



HAL
open science

Intelligent Tutoring Systems - 11th International Conference, ITS 2012, Chania, Crete, Greece, June 14-18, 2012. Proceedings

Stefano A. Cerri, William J. Clancey, Giorgos Papadourakis, Kitty Panourgia

► **To cite this version:**

Stefano A. Cerri, William J. Clancey, Giorgos Papadourakis, Kitty Panourgia (Dir.). Intelligent Tutoring Systems - 11th International Conference, ITS 2012, Chania, Crete, Greece, June 14-18, 2012. Proceedings. Springer Verlag, 7315, 2012, Lecture Notes in Computer Science, 978-3-642-30949-6 (Print) 978-3-642-30950-2 (Online). <lirmm-00799116>

HAL Id: lirmm-00799116

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00799116v1>

Submitted on 11 Mar 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Stefano A. Cerri William J. Clancey
Giorgos Papadourakis Kitty Panourgia (Eds.)

Intelligent Tutoring Systems

11th International Conference, ITS 2012
Chania, Crete, Greece, June 14-18, 2012
Proceedings

Volume Editors

Stefano A. Cerri
LIRMM: University of Montpellier and CNRS
161 rue Ada, 34095 Montpellier, France
E-mail: cerri@lirmm.fr

William J. Clancey
NASA and Florida Institute for Human and Machine Cognition
Human Centered Computing - Intelligent Systems Division
Moffett Field, CA 94035, USA
E-mail: william.j.clancey@nasa.gov

Giorgos Papadourakis
Technological Educational Institute of Crete
School of Applied Technology
Department of Applied Informatics and Multimedia
Stavromenos, P.O. Box 1939
71004 Heraklion, Crete, Greece
E-mail: papadour@cs.teicrete.gr

Kitty Panourgia
Neoanalysis Ltd.
Marni 56, 10437 Athens, Greece
E-mail: kpanourgia@neoanalysis.eu

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-30949-6 e-ISBN 978-3-642-30950-2
DOI 10.1007/978-3-642-30950-2
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012939156

CR Subject Classification (1998): I.2.6, J.4, H.1.2, H.5.1, J.5, K.4.2

LNCS Sublibrary: SL 2 – Programming and Software Engineering

© Springer-Verlag Berlin Heidelberg 2012

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The 11th International Conference on Intelligent Tutoring Systems, ITS 2012, was organized in Chania, Crete, Greece, during June 14–18, 2012. The Call for Papers is printed here to relate the conference’s motivation and theme:

The Intelligent Tutoring Systems (ITS) 2012 conference is part of an on-going biannual series of top-flight international conferences (the ITS conference was launched in 1988) on technologies—systems—that enable, support or enhance human learning. This occurs by means of tutoring—in the case of formal learning—and by exposing learners to rich interactive experiences—in the case of learning as a side effect (informal learning). The “intelligence” of these systems stems from the model-based artificial intelligence technologies often exploited to adapt to the learners (e.g., semantic technologies, user modeling) and also from how today’s technologies (e.g., the Web and service-oriented computing methods) facilitate new emergent collective behaviors. These new practices may outperform previously conceivable learning or tutoring scenarios because they modify significantly the power, speed, and focus of participants’ interactions independently from space and time constraints. The highly interdisciplinary ITS conferences bring together researchers in computer science, informatics, and artificial intelligence on the one side (the “hard” sciences); cognitive science, educational psychology, and linguistics on the other (the “soft” sciences).

The specific theme of the ITS 2012 conference is co-adaptation between technologies and human learning. There are nowadays two real challenges to be faced by ITS. The main technical challenge is due to the unprecedented speed of innovation that we notice in Information and Communication Technologies (ICT), in particular, the Web. Any technology seems to be volatile, of interest for only a short time span. The educational challenge is a consequence of the technical one. Current educational uses of technologies have to consider the impact of ICT innovation on human practices. In particular, new technologies may modify substantially the classical human learning cycle, which since the nineteenth century was mainly centered on formal teaching institutions such as the schools. Educational games are an example of how instructional practice adapts to innovation; another is the measurable role of emotions in learning.

Therefore, our focus for ITS 2012 will be not just on the use of technologies but also the co-adaptation effects. Rapidly evolving technologies entail significant new opportunities and scenarios for learning, thus support the need for analyzing the intersection between new learning practices and innovative technologies to advance both methods and theory for

human learning. This approach especially enables “learning by constructing,” in much the same way as the Web Science movement adds to the classical Web technologies. A new design priority has emerged: reasoned analysis of human communities in different interaction contexts before deploying or applying a new infrastructure or application.

On the one hand this scientific analysis will guide us to avoid well-known pitfalls, on the other it will teach us lessons not only about how to exploit the potential learning effects of current advanced technologies—the applicative approach—but also how to envision, elicit, estimate, evaluate the potential promising effects of new technologies and settings to be conceptualized, specified and developed within human learning scenarios—the experimental approach. We expect this experimental approach to produce long-term scientific progress both in the hard and in the soft sciences, consolidating at the same time important socio-economic benefits from the new infrastructures and the new applications for human learning.

As a result of the Call for Papers, we received more than 200 different contributions evaluated by chairs of four different tracks: the Scientific Paper Track (Chairs: Stefano A. Cerri and William J. Clancey), the Young Researcher Track (Chairs: Roger Azevedo and Chad Lane), the Workshop and Tutorial Track (Chairs: Jean Marc Labat and Rose Luckin). One Panel: “The Next Generation: Pedagogical and Technological Innovations for 2030” were organized by the Panel Chairs: Beverly Woolf and Toshio Okamoto.

For the scientific paper track, we provide a summary of the statistics at the end of the preface. In addition, 14 out of 15 Young Researcher Track papers were accepted, five workshops and two tutorials. There have been four outstanding invited speakers whose contributions have been included in the electronic version of the proceedings.

The scientific papers were evaluated with the help of a popular conference management tool, EasyChair, which was an excellent example of co-adaptation: we were impressed by the space of potential variations in the business process definition and management that is available thanks to the online tool. We believe that the “configuration” choices may have a significant impact on the positive quality of the resulting program.

We chose to assign three “junior” reviewers and one “senior” reviewer to each paper in order to delegate as much as possible to a team of four reviewers the difficult selection task. With the help of EasyChair, we carefully checked the fit of the paper’s topics with the reviewer’s selected topics of expertise and avoided conflict of interests due to proximity, historical, or professional relations.

The process was triple blind: reviewers did not know the authors’ names, authors did not know the reviewers’ names, and reviewers did not know the other reviewers’ names. We guided the evaluation process by means of an evaluation form suggesting to accept about 15% of long papers, 15–30% of short papers and 30% of posters. The reviewer’s evaluations naturally respected our suggestions: out of 177 papers, we accepted 134, consisting of 28 long (16%), 50 short

(28%) and 56 posters (32%). In our view, the quality of long papers is excellent, short papers are good papers, and posters present promising work that deserves attention and discussion.

The decision taken by the senior reviewer was respected in almost all cases, with a very limited number of exceptions that always involved raising the rank of the paper. Our conviction is that the reviewers were very critical, but also extremely constructive, which was confirmed by most of the exchanges with the authors after notification of the decision. The authors of the rejected papers also benefited from a careful review process, with feedback that we hope will help them to improve the quality of the presentations.

We can state without any doubt that ITS 2012 was a very selective, high-quality conference, probably the most selective in the domain.

On the one hand, we wished to guarantee a high acceptance rate and therefore participation at the conference. On the other, we wished to reduce the number of parallel tracks and enable papers accepted as short or long to be attended by most of the participants in order to enhance the historical interdisciplinary nature of the conference and the opportunity for a mutual learning experience. We also wished to increase the number of printed pages in the proceedings for each paper. The result has been to allow ten pages for long papers, six for short ones, and two for posters. The Young Researcher Track's 14 papers are also included in the proceedings (three pages).

The classification by topic in the book reflects viewpoints that are necessarily subjective. What appears as a major phenomenon is that the domain of ITS is becoming increasingly intertwined: theory and experiments, analysis and synthesis, planning and diagnosis, representation and understanding, production and consumption, models and applications. It has not been easy to sort the papers according to topics. In the sequencing of papers in the book, we have tried as much as possible to reflect the sequence of papers in the conference sessions.

We thank first of all the authors, then the members of the Program Committee and the external reviewers, the Steering Committee and in particular Claude Frasson and Beverly Woolf, both present, supportive and positive all the time, the local Organizing Committee, finally each and all the other organizers that are listed on the following pages. Such an event would not have been possible without their commitment, professional effort and patience.

April 2012

Stefano A. Cerri
William J. Clancey
Giorgios Papadourakis
Kitty Panourgia

STATISTICS

By Topic

Topic	Submissions	Accepted	Acceptance Rate	PC Members
Evaluation, privacy, security and trust in e-learning processes	4	2	0.50	5
Ubiquitous and mobile learning environments	5	4	0.80	33
Ontological modeling, Semantic web technologies and standards for learning	7	4	0.57	30
Non-conventional interactions between artificial intelligence and human learning	8	6	0.75	25
Recommender systems for learning	9	4	0.44	31
Informal learning environments, learning as a side effect of interactions	12	10	0.83	32
Multi-agent and service-oriented architectures for learning and tutoring environments	12	8	0.67	22
Instructional design principles or design patterns for educational environments	21	14	0.67	23
Authoring tools and development methodologies for advanced learning technologies	21	12	0.57	34
Discourse during learning interactions	22	20	0.91	17
Co-adaptation between technologies and human learning	22	13	0.59	26
Virtual pedagogical agents or learning companions	23	17	0.74	37
Collaborative and group learning, communities of practice and social networks	23	18	0.78	49
Simulation-based learning, intelligent (serious) games	33	29	0.88	48
Modeling of motivation, metacognition, and affect aspects of learning	33	23	0.70	38
Empirical studies of learning with technologies, understanding human learning on the Web	35	27	0.77	42

Topic	Submissions	Accepted	Acceptance Rate	PC Members
Domain-specific learning domains, e.g., language, mathematics, reading, science, medicine, military, and industry	36	27	0.75	23
Educational exploitation of data mining and machine learning techniques	38	30	0.79	30
Adaptive support for learning, models of learners, diagnosis and feedback	61	44	0.72	64
Intelligent tutoring	79	62	0.78	66

By Country

Country	Authors	Submitted papers
Algeria	2	1.00
Australia	10	4.00
Austria	-	-
Brazil	21	8.02
Canada	40	17.54
Costa Rica	1	0.11
Czech Republic	2	1.00
Denmark	1	1.50
Egypt	1	0.33
Finland	1	1.00
France	31	11.53
Germany	12	3.87
Greece	13	4.83
Hong Kong	2	0.20
India	7	4.33
Ireland	-	-
Italy	-	-
Japan	24	9.75

Country	Authors	Submitted papers
Korea, Republic of	-	-
Latvia	1	0.33
Mexico	2	0.67
The Netherlands	9	3.00
New Zealand	8	4.17
Philippines	6	1.06
Portugal	4	1.00
Romania	4	2.67
Saudi Arabia	2	1.33
Singapore	-	-
Slovakia	-	-
Slovenia	6	1.33
Spain	23	6.25
Switzerland	5	0.71
Taiwan	8	2.00
Tunisia	2	1.33
United Kingdom	15	7.08
United States	178	75.04

Committees

Conference Committee

Conference Chair

George M. Papadourakis Technological Educational Institute of Crete,
Greece

General Chair

Maria Grigoriadou University of Athens, Greece

Program Chairs

Stefano A. Cerri (Chair) LIRMM: University of Montpellier and CNRS,
France

William J. Clancey
(Co-chair) NASA and Florida Institute for Human and
Machine Cognition, USA

Organization Chair

Kitty Panourgia Neoanalysis, Greece

Workshops and Tutorials Chairs

Jean Marc Labat Pierre and Marie Curie University, France
Rose Luckin Institute of Education, UK

Panels Chairs

Beverly Woolf University of Massachussetts, USA
Toshio Okamoto University of Electro-Communications, Japan

Young Researcher Track Chairs

Roger Azevedo McGill University, Canada
Chad Lane University of Southern California, USA

Program Committee

Program Chairs

Stefano A. Cerri (Chair)	LIRMM: University of Montpellier and CNRS, France
William J. Clancey (Co-chair)	NASA and Florida Institute for Human and Machine Cognition, USA

Senior Program Committee

Esma Aimeur	University of Montreal, Canada
Vincent Alevén	Carnegie Mellon University, USA
Ivon Arroyo	University of Massachusetts, USA
Kevin Ashley	University of Pittsburgh, USA
Ryan Baker	Worcester Polytechnic Institute, USA
Joseph Beck	Worcester Polytechnic Institute, USA
Gautam Biswas	Vanderbilt University, USA
Jacqueline Bourdeau	Tele-université, Montreal, Quebec, Canada
Bert Bredeweg	University of Amsterdam, The Netherlands
Paul Brna	University of Edinburgh, UK
Peter Brusilovsky	University of Pittsburgh, USA
Chan Tak-Wai	National Central University, Taiwan
Cristina Conati	University of British Columbia, Canada
Ricardo Conejo	University of Malaga, Spain
Albert Corbett	Carnegie Mellon University, USA
Elisabeth Delozanne	University Pierre et Marie Curie, France
Vania Dimitrova	University of Leeds, UK
Benedict Du Boulay	University of Sussex, UK
Isabel Fernandez-Castro	University of Basque Country, Spain
Claude Frasson	University of Montreal, Canada
Guy Gouarderes	University of Pau, France
Art Graesser	University of Memphis, USA
Peter Hastings	DePaul University, USA
Neil Heffernan	Worcester Polytechnic Institute, USA
W. Lewis Johnson	Alelo Inc., USA
Kenneth Koedinger	Carnegie Mellon University, USA
Jean-Marc Labat	Universite Pierre et Marie Curie, France
Susanne Lajoie	McGill University, Canada
H. Chad Lane	University of Southern California, USA
James Lester	North Carolina State University, USA
Diane Litman	University of Pittsburgh, USA
Chee-Kit Looi	National Institute of Education, Singapore
Rosemary Luckin	University of Sussex, UK

Gordon McCalla	University of Saskatchewan, Canada
Tanja Mitrovic	University of Canterbury, New Zealand
Riichiro Mizoguchi	Osaka University, Japan
Jack Mostow	Carnegie Mellon University, USA
Roger Nkambou	University of Quebec at Montreal, Canada
Stellan Ohlsson	University of Illinois at Chicago, USA
Toshio Okamoto	University of Electro-Communications, Japan
Ana Paiva	INESC-ID and Instituto Superior Tecnico, Technical University of Lisbon, Portugal
Niels Pinkwart	Clausthal University of Technology, Germany
Carolyn Rose	Carnegie Mellon University, USA
Kurt Van Lehn	Arizona State University, USA
Julita Vassileva	University of Saskatchewan, Canada
Rosa Vicari	The Federal University of Rio Grande do Sul, Brazil
Maria Virvou	University of Piraeus, Greece
Vincent Wade	Trinity College Dublin, Ireland
Gerhard Weber	University of Education Freiburg, Germany
Beverly Woolf	University of Massachusetts, USA
Kalina Yacef	University of Sydney, Australia

Program Committee

Mohammed Abdelrazek	King Abdulaziz University, Saudi Arabia
Luigia Aiello	University of Rome, Italy
Colin Alison	St. Andrews University, UK
Ana Arruarte	University of the Basque Country, Spain
Roger Azevedo	McGill University, Canada
Tiffany Barnes	University of North Carolina at Charlotte, USA
Beatriz Barros	University of Malaga, Spain
Maria Bielikova	Slovak University of Technology in Bratislava, Slovakia
Ig Bittencourt	Federal University of Alagoas, Brazil
Emmanuel G. Blanchard	Aalborg University at Copenhagen, Denmark
Steve Blessing	University of Tampa, USA
Joost Breuker	University of Amsterdam, The Netherlands
Nicola Capuano	University of Salerno, Italy
Patricia Charlton	London Knowledge Lab, UK
Zhi-Hong Chen	National Central University, Taiwan
Chih-Yueh Chou	Yuan Ze University, Taiwan
Evandro Costa	Federal University of Alagoas, Brazil
Scotty Craig	University of Memphis, USA
Alexandra Cristea	University of Warwick, UK

Sydney D'Mello	University of Memphis, USA
Hugh Davis	University of Southampton, UK
Michel Desmarais	Polytechnique Montreal, Canada
Cyrille Desmoulins	University of Grenoble, France
Darina Dicheva	Winston-Salem State University, USA
Pierre Dillenbourg	Ecole Polytechnique Federale de Lausanne, Switzerland
Peter Dolog	Aalborg University, Denmark
Pascal Dugenie	IRD: Institut de Recherche pour le Développement, France
Robert Farrell	IBM Research, USA
Vasco Furtado	University of Fortaleza, Brazil
Franca Garzotto	Politecnico di Milano, Italy
Abdelkader Gouaich	LIRMM: University of Montpellier and CNRS, France
Yusuke Hayashi	Osaka University, Japan
Tsukasa Hirashima	Hiroshima University, Japan
Seiji Isotani	University Sao Paulo, Brazil
Patricia Jaques	Universidade do Vale do Rio dos Sinos (UNISINOS), Brazil
Clement Jonquet	University of Montpellier - LIRMM, France
Pamela Jordan	University of Pittsburgh, USA
Vana Kamtsiou	Brunel University, London, UK
Akihiro Kashiara	University of Electro-Communications, Japan
Kathy Kikis-Papadakis	FORTH, Crete, Greece
Yong Se Kim	Sungkyunkwan University, Republic of Korea
Philippe Lemoisson	CIRAD - TETIS, Montpellier, France
Stefanie Lindstaedt	Graz University of Technology and Know-Center, Austria
Chao-Lin Liu	National Chengchi University, Taiwan
Vincenzo Loia	University of Salerno, Italy
Manolis Mavrikis	London Knowledge Laboratory, UK
Riccardo Mazza	University of Lugano/University of Applied Sciences of Southern Switzerland, Switzerland
Germana Menezes	
Da Nobrega	Universidade de Brasilia (UnB), Brazil
Alessandro Micarelli	University of Roma, Italy
Kazuhisa Miwa	Nagoya University, Japan
Paul Mulholland	Knowledge Media Institute, The Open University, UK
Chas Murray	Carnegie Learning, Inc., USA
Wolfgang Nejdl	L3S and University of Hannover, Germany
Jean-Pierre Pecuchet	INSA Rouen, France
Alexandra Poulouvassilis	University of London, UK

Andrew Ravenscroft	University of East London, UK
Genaro Rebolledo-Mendez	University of Veracruz, Mexico
Ma. Mercedes T. Rodrigo	Ateneo de Manila University, Philippines
Ido Roll	University of British Columbia, Canada
Paulo Salles	University of Brasilia, Brazil
Jacobijn Sandberg	University of Amsterdam, The Netherlands
Sudeshna Sarkar	Indian Institute of Technology Kharagpur, India
Hassina Seridi	Badji Mokhtar Annaba University, Algeria
Mike Sharples	The Open University, UK
Peter B. Sloep	The Open University, The Netherlands
John Stamper	Carnegie Mellon University, USA
Akira Takeuchi	Kyushu Institute of Technology, Japan
Josie Taylor	Institute of Educational Technology, Open University, UK
Thanassis Tiropanis	University of Southampton, UK
Stefan Trausan Matu	Bucarest Polytechnic, Romania
Andre Tricot	University of Toulouse, France
Wouter Van Joolingen	University of Twente, The Netherlands
Su White	University of Southampton, UK
Diego Zapata-Rivera	Educational Testing Service, USA
Ramon Zatarain-Cabada	Technological Institute of Culiacan, Mexico

Organization Committee

Chair

Kitty Panourgia Neoanalysis, Greece

Members

Dimosthenis Akoumianakis	TEI of Crete, Greece
Yannis Kaliakatsos	TEI of Crete, Greece
Emmanuel S. Karapidakis	TEI of Crete, Greece
Athanasios Malamos	TEI of Crete, Greece
Harris Papoutsakis	TEI of Crete, Greece
Konstantinos Petridis	TEI of Crete, Greece
George Triantafilides	TEI of Crete, Greece

Treasurer: Neoanalysis

Student Volunteers Chairs

Pierre Chalfoun University of Montreal, Canada

Mediterranean Committee

Chair

Mohammed Abdelrazek King Abdulaziz University, Saudi Arabia

Members

Amar Balla ENSI, Algeria
Stephane Bernard Bazan Université Saint Joseph de Beyrouth, Lebanon
Isabel Fernandez-Castro University of Basque Country, Spain
Khaled Guedira Institut Supérieur de gestion, Tunisia
Gianna Martinengo Didael KTS, Milan, Italy
Kyparisia Papanikolaou School of Pedagogical & Technological Education, Greece

Steering Committee

Chair

Claude Frasson University of Montreal, Canada

Members

Stefano Cerri University of Montpellier, France
Isabel Fernandez-Castro University of the Basque Country, Spain
Gilles Gauthier University of Quebec at Montreal, Canada
Guy Gouardères University of Pau, France
Mitsuru Ikeda Japan Advanced Institute of Science and Technology, Japan
Marc Kaltenbach Bishop's University, Canada
Judith Kay University of Sidney, Australia
Alan Lesgold University of Pittsburgh, USA
James Lester North Carolina State University, USA
Roger Nkambou University of Quebec at Montreal, Canada
Fabio Paragua Federal University of Alagoas, Brazil
Elliot Soloway University of Michigan, USA
Daniel Suthers University of Hawai, USA
Beverly Woolf University of Massachussets, USA

External Reviewers

Adewoyin, Bunmi
Alrifai, Mohammad
Bourreau, Eric
Campbell, Antoine
Chauncey, Amber
Cheney, Kyle
Chiru, Costin
Cialdea, Marta
Delestre, Nicolas
Doran, Katelyn

Elorriaga, Jon A.
Falakmasir, Mohammad Hassan
Floryan, Mark
Foss, Jonathan
Gkotsis, George
Gonzalez-Brenes, Jose
Grieco, Claudia
Gross, Sebastian
Guarino, Giuseppe
Gutierrez Santos, Sergio
Guzmán, Eduardo
Hayashi, Yugo
Henze, Nicola
Herder, Eelco
Hicks, Drew
Johnson, Matthew
Kojima, Kazuaki
Koriche, Fred
Labaj, Martin
Larrañaga, Mikel
Le, Nguyen-Think
Lehman, Blair
Lehmann, Lorrie
Limongelli, Carla
Lomas, Derek
Mangione, Giuseppina
Martin, Maite
Mazzola, Luca
Miranda, Sergio

Morita, Junya
Nickel, Andrea
Orciuoli, Francesco
Pardos, Zach
Pardos, Zachary
Peckham, Terry
Pierri, Anna
Pinheiro, Vladia
Rebedea, Traian
Ruiz, Samara
Sciarrone, Filippo
Scotton, Joshua
Sharipova, Mayya
Shaw, Erin
Steiner, Christina M.
Stepanyan, Karen
Sullins, Jeremiah
Thomas, John
Thomas, Keith
Trausan-Matu, Stefan
Tvarozek, Jozef
Urretavizcaya, Maite
van Lehn, Kurt
Vasconcelos, José Eurico
Wu, Kewen
Yudelson, Michael
Zipitria, Iraide
Šimko, Marián

Workshops

Intelligent Support for Exploratory Environments: Exploring, Collaborating, and Learning Together

Toby Dragon, Sergio Gutierrez Santos, Manolis Mavrikis, and Bruce M. McLaren

Workshop on Self-Regulated Learning in Educational Technologies (SRL@ET): Supporting, Modelling, Evaluating, and Fostering Metacognition with Computer-Based Learning Environments

Amali Weerasinghe, Roger Azevedo, Ido Roll, and Ben Du Boulay

Intelligent Support for Learning in Groups

Jihie Kim and Rohit Kumar

Emotion in Games for Learning

Kostas Karpouzis, Georgios N. Yannakakis, Ana Paiva, and Eva Hudlicka

Web 2.0 Tools, Methodology, and Services for Enhancing Intelligent Tutoring Systems

Mohammed Abdel Razek and Claude Frasson

Tutorials

Important Relationships in Data: Magnitude and Causality as Flags for What to Focus on

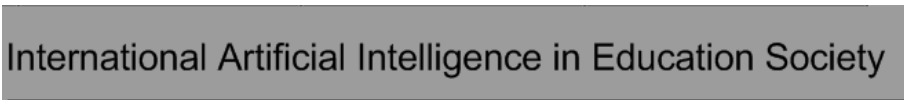
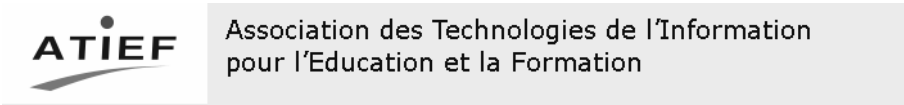
Joseph Beck (WPI)

Parameter Fitting for Learner Models

Tristan Nixon (Carnegie Learning Inc.), Ryan S.J.D. Baker (WPI), Michael Yudelson (CMU), and Zach Pardos (WPI)

Scientific Sponsors

The following scientific associations have granted their scientific support to the conference; their members benefit from a special registration rate.



The conference benefits also from the sponsoring of the following renowned conferences:

- IJCAI : International Joint Conference in Artificial Intelligence
- ECAI : European Conference in Artificial Intelligence
- EDM : Educational Data Mining

Table of Contents

Affect: Emotions

Implicit Strategies for Intelligent Tutoring Systems	1
<i>Imène Jraïdi, Pierre Chalfoun, and Claude Frasson</i>	
Rudeness and Rapport: Insults and Learning Gains in Peer Tutoring . . .	11
<i>Amy Ogan, Samantha Finkelstein, Erin Walker, Ryan Carlson, and Justine Cassell</i>	
On Pedagogical Effects of Learner-Support Agents in Collaborative Interaction	22
<i>Yugo Hayashi</i>	
Exploration of Affect Detection Using Semantic Cues in Virtual Improvisation	33
<i>Li Zhang</i>	
Measuring Learners Co-Occurring Emotional Responses during Their Interaction with a Pedagogical Agent in MetaTutor	40
<i>Jason M. Harley, François Bouchet, and Roger Azevedo</i>	
Visualization of Student Activity Patterns within Intelligent Tutoring Systems	46
<i>David Hilton Shanabrook, Ivon Arroyo, Beverly Park Woolf, and Winslow Burleson</i>	
Toward a Machine Learning Framework for Understanding Affective Tutorial Interaction	52
<i>Joseph F. Grafsgaard, Kristy Elizabeth Boyer, and James C. Lester</i>	
Exploring Relationships between Learners' Affective States, Metacognitive Processes, and Learning Outcomes	59
<i>Amber Chauncey Strain, Roger Azevedo, and Sidney D'Mello</i>	
Mental Workload, Engagement and Emotions: An Exploratory Study for Intelligent Tutoring Systems	65
<i>Maher Chaouachi and Claude Frasson</i>	

Affect: Signals

Real-Time Monitoring of ECG and GSR Signals during Computer-Based Training 72
Keith W. Brawner and Benjamin S. Goldberg

Categorical vs. Dimensional Representations in Multimodal Affect Detection during Learning 78
Md. Sazzad Hussain, Hamed Monkaresi, and Rafael A. Calvo

Cognitive Priming: Assessing the Use of Non-conscious Perception to Enhance Learner’s Reasoning Ability 84
Pierre Chalfoun and Claude Frasson

Games: Motivation and Design

Math Learning Environment with Game-Like Elements: An Incremental Approach for Enhancing Student Engagement and Learning Effectiveness 90
Dovan Rai and Joseph E. Beck

Motivational Factors for Learning by Teaching: The Effect of a Competitive Game Show in a Virtual Peer-Learning Environment 101
Noboru Matsuda, Evelyn Yarzebinski, Victoria Keiser, Rohan Raizada, Gabriel Stylianides, and Kenneth R. Koedinger

An Analysis of Attention to Student-Adaptive Hints in an Educational Game 112
Mary Muir and Cristina Conati

Serious Game and Students’ Learning Motivation: Effect of Context Using Prog&Play 123
Mathieu Muratet, Elisabeth Delozanne, Patrice Torguet, and Fabienne Viallet

Exploring the Effects of Prior Video-Game Experience on Learner’s Motivation during Interactions with HeapMotiv 129
Lotfi Derbali and Claude Frasson

A Design Pattern Library for Mutual Understanding and Cooperation in Serious Game Design 135
Bertrand Marne, John Wisdom, Benjamin Huynh-Kim-Bang, and Jean-Marc Labat

Games: Empirical Studies

Predicting Student Self-regulation Strategies in Game-Based Learning Environments	141
<i>Jennifer Sabourin, Lucy R. Shores, Bradford W. Mott, and James C. Lester</i>	
Toward Automatic Verification of Multiagent Systems for Training Simulations	151
<i>Ning Wang, David V. Pynadath, and Stacy C. Marsella</i>	
Using State Transition Networks to Analyze Multi-party Conversations in a Serious Game	162
<i>Brent Morgan, Fazel Keshtkar, Ying Duan, Pdraig Nash, and Arthur Graesser</i>	
How to Evaluate Competencies in Game-Based Learning Systems Automatically?	168
<i>Pradeepa Thomas, Jean-Marc Labat, Mathieu Muratet, and Amel Yessad</i>	

Content Representation: Empirical Studies

Sense Making Alone Doesn't Do It: Fluency Matters Too! ITS Support for Robust Learning with Multiple Representations	174
<i>Martina A. Rau, Vincent Aleven, Nikol Rummel, and Stacie Rohrbach</i>	
Problem Order Implications for Learning Transfer	185
<i>Nan Li, William W. Cohen, and Kenneth R. Koedinger</i>	
Knowledge Component Suggestion for Untagged Content in an Intelligent Tutoring System	195
<i>Mario Karlovćec, Mariheida Córdoba-Sánchez, and Zachary A. Pardos</i>	

Feedback: Empirical Studies

Automating Next-Step Hints Generation Using ASTUS	201
<i>Luc Paquette, Jean-François Lebeau, Gabriel Beaulieu, and André Mayers</i>	

The Effectiveness of Pedagogical Agents' Prompting and Feedback in Facilitating Co-adapted Learning with MetaTutor	212
<i>Roger Azevedo, Ronald S. Landis, Reza Feyzi-Behnagh, Melissa Duffy, Gregory Trevors, Jason M. Harley, François Bouchet, Jonathan Burlison, Michelle Taub, Nicole Pacampara, Mohamed Yeasin, A.K.M. Mahbubur Rahman, M. Iftekhar Tanveer, and Gahangir Hossain</i>	

Noticing Relevant Feedback Improves Learning in an Intelligent Tutoring System for Peer Tutoring	222
<i>Erin Walker, Nikol Rummel, Sean Walker, and Kenneth R. Koedinger</i>	

ITS in Special Domains

Multi-paradigm Generation of Tutoring Feedback in Robotic Arm Manipulation Training	233
<i>Philippe Fournier-Viger, Roger Nkambou, André Mayers, Engelbert Mephu-Nguifo, and Usef Faghihi</i>	

User-Centered Design of a Teachable Robot	243
<i>Erin Walker and Winslow Burleson</i>	

An Intelligent Tutoring and Interactive Simulation Environment for Physics Learning	250
<i>Lakshman S. Myneni and N. Hari Narayanan</i>	

Guru: A Computer Tutor That Models Expert Human Tutors	256
<i>Andrew M. Olney, Sidney D'Mello, Natalie Person, Whitney Cade, Patrick Hays, Claire Williams, Blair Lehman, and Arthur Graesser</i>	

Developing an Embodied Pedagogical Agent with and for Young People with Autism Spectrum Disorder	262
<i>Beate Grawemeyer, Hilary Johnson, Mark Brosnan, Emma Ashwin, and Laura Benton</i>	

Non Conventional Approaches

WEBSistments: Enabling an Intelligent Tutoring System to Excel at Explaining Rather Than Coaching	268
<i>Yue Gong, Joseph E. Beck, and Neil T. Heffernan</i>	

Automated Approaches for Detecting Integration in Student Essays	274
<i>Simon Hughes, Peter Hastings, Joseph Magliano, Susan Goldman, and Kimberly Lawless</i>	

On the WEIRD Nature of ITS/AIED Conferences: A 10 Year Longitudinal Study Analyzing Potential Cultural Biases	280
<i>Emmanuel G. Blanchard</i>	

Content Representation: Conceptual

Goal-Oriented Conceptualization of Procedural Knowledge	286
<i>Martin Možina, Matej Guid, Aleksander Sadikov, Vida Groznik, and Ivan Bratko</i>	

Context-Dependent Help for Novices Acquiring Conceptual Systems Knowledge in DynaLearn	292
<i>Wouter Beek and Bert Bredeweg</i>	

Towards an Ontology-Based System to Improve Usability in Collaborative Learning Environments	298
<i>Endhe Elias, Dalgoberto Miquilino, Ig Ibert Bittencourt, Thyago Tenório, Rafael Ferreira, Alan Silva, Seiji Isotani, and Patrícia Jaques</i>	

Program Representation for Automatic Hint Generation for a Data-Driven Novice Programming Tutor	304
<i>Wei Jin, Tiffany Barnes, John Stamper, Michael John Eagle, Matthew W. Johnson, and Lorrie Lehmann</i>	

Assessment: Constraints

Exploring Quality of Constraints for Assessment in Problem Solving Environments	310
<i>Jaime Galvez Cordero, Eduardo Guzman De Los Riscos, and Ricardo Conejo Muñoz</i>	

Can Soft Computing Techniques Enhance the Error Diagnosis Accuracy for Intelligent Tutors?	320
<i>Nguyen-Think Le and Niels Pinkwart</i>	

Dialogue: Conceptual

Identification and Classification of the Most Important Moments from Students' Collaborative Discourses	330
<i>Costin-Gabriel Chiru and Stefan Trausan-Matu</i>	

When Less Is More: Focused Pruning of Knowledge Bases to Improve Recognition of Student Conversation	340
<i>Mark Floryan, Toby Dragon, and Beverly Park Woolf</i>	

Coordinating Multi-dimensional Support in Collaborative Conversational Agents	346
<i>David Adamson and Carolyn Penstein Rosé</i>	
Textual Complexity and Discourse Structure in Computer-Supported Collaborative Learning	352
<i>Stefan Trausan-Matu, Mihai Dascalu, and Philippe Dessus</i>	
Dialogue: Questions	
Using Information Extraction to Generate Trigger Questions for Academic Writing Support	358
<i>Ming Liu and Rafael A. Calvo</i>	
Learning to Tutor Like a Tutor: Ranking Questions in Context	368
<i>Lee Becker, Martha Palmer, Sarel van Vuuren, and Wayne Ward</i>	
Learner Modeling	
Analysis of a Simple Model of Problem Solving Times	379
<i>Petr Jarušek and Radek Pelánek</i>	
Modelling and Optimizing the Process of Learning Mathematics	389
<i>Tanja Käser, Alberto Giovanni Busetto, Gian-Marco Baschera, Juliane Kohn, Karin Kucian, Michael von Aster, and Markus Gross</i>	
The Student Skill Model	399
<i>Yutao Wang and Neil T. Heffernan</i>	
Clustered Knowledge Tracing	405
<i>Zachary A. Pardos, Shubhendu Trivedi, Neil T. Heffernan, and Gábor N. Sárközy</i>	
Preferred Features of Open Learner Models for University Students	411
<i>Susan Bull</i>	
Do Your Eyes Give It Away? Using Eye Tracking Data to Understand Students' Attitudes towards Open Student Model Representations	422
<i>Moffat Mathews, Antonija Mitrovic, Bin Lin, Jay Holland, and Neville Churcher</i>	
Fuzzy Logic Representation for Student Modelling: Case Study on Geometry	428
<i>Gagan Goel, Sébastien Lallé, and Vanda Luengo</i>	

Learning Detection

Content Learning Analysis Using the Moment-by-Moment Learning Detector	434
<i>Sujith M. Gowda, Zachary A. Pardos, and Ryan S.J.D. Baker</i>	
Towards Automatically Detecting Whether Student Learning Is Shallow	444
<i>Ryan S.J.D. Baker, Sujith M. Gowda, Albert T. Corbett, and Jaclyn Ocumpaugh</i>	
Item to Skills Mapping: Deriving a Conjunctive Q-matrix from Data ...	454
<i>Michel C. Desmarais, Behzad Beheshti, and Rhouma Naceur</i>	

Interaction Strategies: Games

The Role of Sub-problems: Supporting Problem Solving in Narrative-Centered Learning Environments	464
<i>Lucy R. Shores, Kristin F. Hoffmann, John L. Nietfeld, and James C. Lester</i>	
Exploring Inquiry-Based Problem-Solving Strategies in Game-Based Learning Environments	470
<i>Jennifer Sabourin, Jonathan Rowe, Bradford W. Mott, and James C. Lester</i>	
Real-Time Narrative-Centered Tutorial Planning for Story-Based Learning	476
<i>Seung Y. Lee, Bradford W. Mott, and James C. Lester</i>	

Interaction Strategies: Empirical Studies

An Interactive Teacher's Dashboard for Monitoring Groups in a Multi-tabletop Learning Environment	482
<i>Roberto Martinez Maldonado, Judy Kay, Kalina Yacef, and Beat Schwendimann</i>	
Efficient Cross-Domain Learning of Complex Skills	493
<i>Nan Li, William W. Cohen, and Kenneth R. Koedinger</i>	
Exploring Two Strategies for Teaching Procedures	499
<i>Antoniija Mitrovic, Moffat Mathews, and Jay Holland</i>	
Relating Student Performance to Action Outcomes and Context in a Choice-Rich Learning Environment	505
<i>James R. Segedy, John S. Kinnebrew, and Gautam Biswas</i>	

Using the MetaHistoReasoning Tool Training Module to Facilitate the Acquisition of Domain-Specific Metacognitive Strategies	511
<i>Eric Poitras, Susanne Lajoie, and Yuan-Jin Hong</i>	
An Indicator-Based Approach to Promote the Effectiveness of Teachers' Interventions	517
<i>Aina Lekira, Christophe Després, Pierre Jacoboni, and Dominique Py</i>	
Limiting the Number of Revisions while Providing Error-Flagging Support during Tests	524
<i>Amruth N. Kumar</i>	

Dialogue: Empirical Studies

Towards Academically Productive Talk Supported by Conversational Agents	531
<i>Gregory Dyke, David Adamson, Iris Howley, and Carolyn Penstein Rosé</i>	
Automatic Evaluation of Learner Self-Explanations and Erroneous Responses for Dialogue-Based ITSs	541
<i>Blair Lehman, Caitlin Mills, Sidney D'Mello, and Arthur Graesser</i>	
Group Composition and Intelligent Dialogue Tutors for Impacting Students' Academic Self-efficacy	551
<i>Iris Howley, David Adamson, Gregory Dyke, Elijah Mayfield, Jack Beuth, and Carolyn Penstein Rosé</i>	
How Do They Do It? Investigating Dialogue Moves within Dialogue Modes in Expert Human Tutoring	557
<i>Blair Lehman, Sidney D'Mello, Whitney Cade, and Natalie Person</i>	
Building a Conversational SimStudent	563
<i>Ryan Carlson, Victoria Keiser, Noboru Matsuda, Kenneth R. Koedinger, and Carolyn Penstein Rosé</i>	
Predicting Learner's Project Performance with Dialogue Features in Online Q&A Discussions	570
<i>Jaebong Yoo and Jihie Kim</i>	

Young Researchers Track

Interventions to Regulate Confusion during Learning	576
<i>Blair Lehman, Sidney D'Mello, and Arthur Graesser</i>	
Using Examples in Intelligent Tutoring Systems	579
<i>Amir Shareghi Najar and Antonija Mitrovic</i>	

Semi-supervised Classification of Realtime Physiological Sensor Datastreams for Student Affect Assessment in Intelligent Tutoring	582
<i>Keith W. Brawner, Robert Sottolare, and Avelino Gonzalez</i>	
Detection of Cognitive Strategies in Reading Comprehension Tasks	585
<i>Terry Peckham</i>	
The Effects of Adaptive Sequencing Algorithms on Player Engagement within an Online Game	588
<i>Derek Lomas, John Stamper, Ryan Muller, Kishan Patel, and Kenneth R. Koedinger</i>	
A Canonicalizing Model for Building Programming Tutors	591
<i>Kelly Rivers and Kenneth R. Koedinger</i>	
Developmentally Appropriate Intelligent Spatial Tutoring for Mobile Devices	594
<i>Melissa A. Wiederrecht and Amy C. Ulinski</i>	
Leveraging Game Design to Promote Effective User Behavior of Intelligent Tutoring Systems	597
<i>Matthew W. Johnson, Tomoko Okimoto, and Tiffany Barnes</i>	
Design of a Knowledge Base to Teach Programming	600
<i>Dinesha Weragama and Jim Reye</i>	
Towards an ITS for Improving Social Problem Solving Skills of ADHD Children	603
<i>Atefeh Ahmadi Olounabadi and Antonija Mitrovic</i>	
A Scenario Based Analysis of E-Collaboration Environments	606
<i>Raoudha Chebil, Wided Lejouad Chaari, and Stefano A. Cerri</i>	
Supporting Students in the Analysis of Case Studies for Ill-Defined Domains	609
<i>Mayya Sharipova</i>	
Using Individualized Feedback and Guided Instruction via a Virtual Human Agent in an Introductory Computer Programming Course	612
<i>Lorrie Lehmann, Dale-Marie Wilson, and Tiffany Barnes</i>	
Data-Driven Method for Assessing Skill-Opportunity Recognition in Open Procedural Problem Solving Environments	615
<i>Michael John Eagle and Tiffany Barnes</i>	
Posters	
How Do Learners Regulate Their Emotions?	618
<i>Amber Chauncey Strain, Sidney D'Mello, and Melissa Gross</i>	

A Model-Building Learning Environment with Explanatory Feedback to Erroneous Models	620
<i>Tomoya Horiguchi, Tsukasa Hirashima, and Kenneth D. Forbus</i>	
An Automatic Comparison between Knowledge Diagnostic Techniques	622
<i>Sébastien Lallé, Vanda Luengo, and Nathalie Guin</i>	
The Interaction Behavior of Agents' Emotional Support and Competency on Learner Outcomes and Perceptions	624
<i>Heather K. Holden</i>	
Accuracy of Tracking Student's Natural Language in Operation ARIES!, A Serious Game for Scientific Methods.....	626
<i>Zhiqiang Cai, Carol Forsyth, Mae-Lynn Germany, Arthur Graesser, and Keith Millis</i>	
Designing the Knowledge Base for a PHP Tutor	628
<i>Dinesha Weragama and Jim Reye</i>	
Domain Specific Knowledge Representation for an Intelligent Tutoring System to Teach Algebraic Reasoning	630
<i>Miguel Arevalillo-Herráez, David Arnau, José Antonio González-Calero, and Aladdin Ayesh</i>	
Exploring the Potential of Tabletops for Collaborative Learning	632
<i>Michael Schubert, Sébastien George, and Audrey Serna</i>	
Modeling the Affective States of Students Using SQL-Tutor	634
<i>Thea Faye G. Guia, Ma. Mercedes T. Rodrigo, Michelle Marie C. Dagami, Jessica O. Sugay, Francis Jan P. Macam, and Antonija Mitrovic</i>	
A Cross-Cultural Comparison of Effective Help-Seeking Behavior among Students Using an ITS for Math.....	636
<i>Jose Carlo A. Soriano, Ma. Mercedes T. Rodrigo, Ryan S.J.D. Baker, Amy Ogan, Erin Walker, Maynor Jimenez Castro, Ryan Genato, Samantha Fontaine, and Ricardo Belmontez</i>	
Emotions during Writing on Topics That Align or Misalign with Personal Beliefs	638
<i>Caitlin Mills and Sidney D'Mello</i>	
A Multiagent-Based ITS Using Multiple Viewpoints for Propositional Logic	640
<i>Evandro Costa, Priscylla Silva, Marlos Silva, Emanuele Silva, and Anderson Santos</i>	

Simulation-Based Training of Ill-Defined Social Domains: The Complex Environment Assessment and Tutoring System (CEATS).....	642
<i>Benjamin D. Nye, Gnana K. Bharathy, Barry G. Silverman, and Ceyhun Eksin</i>	
Empirical Investigation on Self Fading as Adaptive Behavior of Hint Seeking	645
<i>Kazuhisa Miwa, Hitoshi Terai, Nana Kanzaki, and Ryuichi Nakaïke</i>	
Scripting Discussions for Elaborative, Critical Interactions	647
<i>Oliver Scheuer, Bruce M. McLaren, Armin Weinberger, and Sabine Niebuhr</i>	
Design Requirements of a Virtual Learning Environment for Resource Sharing	649
<i>Nikos Barbalios, Irene Ioannidou, Panagiotis Tzionas, and Stefanos Paraskeuopoulos</i>	
The Effectiveness of a Pedagogical Agent's Immediate Feedback on Learners' Metacognitive Judgments during Learning with MetaTutor ...	651
<i>Reza Feyzi-Behnagh and Roger Azevedo</i>	
Supporting Students in the Analysis of Case Studies for Professional Ethics Education	653
<i>Mayya Sharipova and Gordon McCalla</i>	
Evaluating the Automatic Extraction of Learning Objects from Electronic Textbooks Using ErauzOnt	655
<i>Mikel Larrañaga, Ángel Conde, Iñaki Calvo, Ana Arruarte, and Jon A. Elorriaga</i>	
A Cognition-Based Game Platform and Its Authoring Environment for Learning Chinese Characters	657
<i>Chao-Lin Liu, Chia-Ying Lee, Wei-Jie Huang, Yu-Lin Tzeng, and Chia-Ru Chou</i>	
Effects of Text and Visual Element Integration Schemes on Online Reading Behaviors of Typical and Struggling Readers	660
<i>Robert P. Dolan and Sonya Powers</i>	
Fadable Scaffolding with Cognitive Tool	662
<i>Akihiro Kashihara and Makoto Ito</i>	
Mediating Intelligence through Observation, Dependency and Agency in Making Construals of Malaria	664
<i>Meurig Beynon and Will Beynon</i>	

Supporting Social Deliberative Skills in Online Classroom Dialogues: Preliminary Results Using Automated Text Analysis	666
<i>Tom Murray, Beverly Park Woolf, Xiaoxi Xu, Stefanie Shipe, Scott Howard, and Leah Wing</i>	
Using Time Pressure to Promote Mathematical Fluency	669
<i>Steve Ritter, Tristan Nixon, Derek Lomas, John Stamper, and Dixie Ching</i>	
Interoperability for ITS: An Ontology of Learning Style Models	671
<i>Judi McCuaig and Robert Gauthier</i>	
Skill Diaries: Can Periodic Self-assessment Improve Students' Learning with an Intelligent Tutoring System?	673
<i>Yanjin Long and Vincent Aleven</i>	
An Optimal Assessment of Natural Language Student Input Using Word-to-Word Similarity Metrics	675
<i>Vasile Rus and Mihai Lintean</i>	
Facilitating Co-adaptation of Technology and Education through the Creation of an Open-Source Repository of Interoperable Code	677
<i>Philip I. Pavlik Jr., Jaclyn Maass, Vasile Rus, and Andrew M. Olney</i>	
A Low-Cost Scalable Solution for Monitoring Affective State of Students in E-learning Environment Using Mouse and Keystroke Data	679
<i>Po-Ming Lee, Wei-Hsuan Tsui, and Tzu-Chien Hsiao</i>	
Impact of an Adaptive Tutorial on Student Learning	681
<i>Fethi A. Inan, Fatih Ari, Raymond Flores, Amani Zaier, and Ismahan Arslan-Ari</i>	
Technology Enhanced Learning Program That Makes Thinking the Outside to Train Meta-cognitive Skill through Knowledge Co-creation Discussion	683
<i>Kazuhisa Seta, Liang Cui, Mitsuru Ikeda, and Noriyuki Matsuda</i>	
Open Student Models to Enhance Blended-Learning	685
<i>Maite Martín, Ainhoa Álvarez, David Reina, Isabel Fernández-Castro, Maite Urretavizcaya, and Susan Bull</i>	
ZooQuest: A Mobile Game-Based Learning Application for Fifth Graders	687
<i>Gerard Veenhof, Jacobijn Sandberg, and Marinus Maris</i>	
Drawing-Based Modeling for Early Science Education	689
<i>Wouter R. van Joolingen, Lars Bollen, Frank Leenaars, and Hannie Gijlers</i>	

An OWL Ontology for IEEE-LOM and OBAA Metadata	691
<i>João Carlos Gluz and Rosa M. Vicari</i>	
Classifying Topics of Video Lecture Contents Using Speech Recognition Technology	694
<i>Jun Park and Jihie Kim</i>	
An Agent-Based Infrastructure for the Support of Learning Objects Life-Cycle	696
<i>João Carlos Gluz, Rosa M. Vicari, and Liliana M. Passerino</i>	
Cluster Based Feedback Provision Strategies in Intelligent Tutoring Systems	699
<i>Sebastian Gross, Xibin Zhu, Barbara Hammer, and Niels Pinkwart</i>	
A Web Comic Strip Creator for Educational Comics with Assessable Learning Objectives	701
<i>Fotis Lazarinis and Elaine Pearson</i>	
A Layered Architecture for Online Lab-Works: Experimentation in the Computer Science Education	703
<i>Mohamed El Amine Bouabid, Philippe Vidal, and Julien Broisin</i>	
A Serious Game for Teaching Conflict Resolution to Children	705
<i>Joana Campos, Henrique Campos, Carlos Martinho, and Ana Paiva</i>	
Towards Social Mobile Blended Learning	707
<i>Amr Abozeid, Mohammed Abdel Razek, and Claude Frasson</i>	
Learning Looping: From Natural Language to Worked Examples	710
<i>Leigh Ann Sudol-DeLyser, Mark Stehlik, and Sharon Carver</i>	
A Basic Model of Metacognition: A Repository to Trigger Reflection . . .	712
<i>Alejandro Peña Ayala, Rafael Dominguez de Leon, and Riichiro Mizoguchi</i>	
Analyzing Affective Constructs: Emotions ‘n Attitudes	714
<i>Ivon Arroyo, David Hilton Shanabrook, Winslow Burleson, and Beverly Park Woolf</i>	
Interactive Virtual Representations, Fractions, and Formative Feedback	716
<i>Maria Mendiburo, Brian Sulcer, Gautam Biswas, and Ted Hasselbring</i>	

An Intelligent System to Support Accurate Transcription of University Lectures	718
<i>Miltiades Papadopoulos and Elaine Pearson</i>	
Multi-context Recommendation in Technology Enhanced Learning	720
<i>Majda Maâtallah and Hassina Seridi-Bouchelaghem</i>	
Author Index	723

Implicit Strategies for Intelligent Tutoring Systems

Imène Jraidi, Pierre Chalfoun, and Claude Frasson

Université de Montréal, Dept. of Computer Science and Operations Research
2920 chemin de la tour, H3T-1J8 QC, Canada
{jraidiim, chalfoun, frasson}@iro.umontreal.ca

Abstract. Nowadays several researches in Intelligent Tutoring Systems are oriented toward developing emotionally sensitive tutors. These tutors use different instructional strategies addressing both learners' cognitive and affective dimensions and rely, for most of them, on explicit strategies and direct interventions. In this paper we propose a new approach to augment these tutors with new implicit strategies relying on indirect interventions. We show the feasibility of our approach through two experimental studies using a subliminal priming technique. We demonstrate that both learners' cognitive and affective states can be conditioned indirectly and show that these strategies produce a positive impact on students' interaction experience and enhance learning.

Keywords: Implicit tutoring strategies, Unconscious processes, Subliminal priming, Affect, Cognition.

1 Introduction

The development of Intelligent Tutoring Systems (ITS) began in the 1970's in an attempt to enhance computer based instruction with artificial intelligence methods to provide a highly individualized teaching and feedback tailored to the needs of the learner. Their aim is to support learning by simulating human tutors' pedagogical skills and domain expertise and produce the same kind of learning and flexibility that might occur between teachers and students [1]. In the recent years, the dynamics of learning has been shifting steadily from purely cognitive aspects of teaching to affect-sensitive tutors. This change can mainly be explained by recent advances in cognitive science, artificial intelligence, and neuroscience showing that the brain mechanisms associated to emotions are not only related to cognitive processes, such as reasoning, but also solicited in perception, problem solving and decision making [2].

Indeed, various research areas including education, psychology, computational linguistics, and artificial intelligence have shown a growing interest in the close links between affect and learning [3-8] as emotions have an impact on attention, motivation, memorization, and information processing [9]. This fact is especially true in the ITS community where several researchers have developed emotionally intelligent tutors able to respond to students on a personal level, identifying their actual emotional states and adapting their teaching accordingly [3, 8, 10-12].

However these tutors still need to be enhanced to track changes in learners' mental states especially when cognitive tasks such as reasoning and decision making occur. Furthermore, most of these tutors rely on explicit¹ strategies and direct interventions when interacting with the learner. These strategies can be in some cases excessive, inappropriate, or intrusive to the dynamics of the learning session. They can also be approximate or target basically, superficial aspects of the interaction.

In this research we propose to enhance these strategies with implicit interventions that can be more subtle and have the ability to target deeper affective and cognitive aspects involved in the learning process. Our work is based on evidence from the neuropsychology that suggests that cognition and affect involve some forms of implicit or unconscious processing that can be implicitly solicited [13]. In this paper we propose an approach to augment these tutors with new implicit strategies relying on indirect interventions in a problem solving environment in order to enhance cognitive abilities such as reasoning and condition affective states such as self-esteem while learning takes place.

2 Previous Work

A variety of explicit strategies have been developed and evaluated within different learning environments providing both cognitive and affective feedback.

Some researches are oriented toward psychological theories to induce positive emotions in students [14], while others mostly use punctual, task-related and less intrusive interventions that can be directly integrated in the dynamics of the learning session. These strategies rely on a variety of task-based support policies to respond to particular students' affective states. This can be in the form of examples or definitions to help the students understand specific concepts. For example in [6], if the tutor realizes that the learner is bored, he engages him in a variety of stimulating tasks, a particular challenge or a game. If frustration is detected, the tutor provides statements or corrects certain information that the learner might have poorly assimilated. In [3], the ITS reacts differently to frustrated learners. It provides an indication or other similar problems to help the learner. In case of boredom, and depending on the situation, the tutor proposes an easier problem to motivate the learner in solving it, or increases the level of difficulty if the problem is too easy. The corrective mechanisms involved here are all direct and explicit and are applied a posteriori, that is once the student makes a mistake or reacts negatively to a situation.

Other approaches integrate more sophisticated companion technologies using life-like characters allowing real-time affective interactions between virtual pedagogical agents and learners [3, 4, 11, 12]. These agents may have a human appearance dialoguing with learners, communicating various messages of encouragement or

¹ We define an explicit strategy as a tutoring intervention that occurs with a person's sensory or conscious perception. By contrast, an implicit strategy is an intervention that cannot be consciously perceived or reported by the learner. This implies that this intervention occurs without a learner's awareness and hence that does not interrupt the dynamics of the learning session.

congratulation; appearing to care about a learner's progress. Some agents can work with the students on the same task, as study partners [8] and exchange on the problem they are solving by either offering advice and encouragement or helping them in coping with negative emotions (such as frustration, boredom, or fatigue). These companions can also adopt an empathetic response [3, 11, 12]. For instance in [12] an agent intervenes looking concerned when users are asked stressing interview questions leading to a drop in stress levels as measured by skin conductance. Although some of these tutors use physiological sensors to monitor students' affect, none uses brain data to monitor mental states and adjusts learning to cerebral changes that can occur in learners' cognitive reasoning processes. In another study [10], it is shown that empathetic responses of pedagogical learning companions improve learners' interest and self-efficacy but not learning outcomes. These agents can even simulate the behavior of the learner by adopting his expressions and actions [3, 4, 11, 12]. For example, Burleson [4] uses an agent that mimics the facial expressions and movements of the learner. It may for example, smile if it sees the learner smiling.

However, the aforementioned strategies only target aspects related to the direct interaction between the learner and the tutor. These strategies do not address the unconscious mechanisms underlying important cognitive and affective processing related to learning. The goal of this research is to augment, not replace, current learning strategies with implicit interventions. We believe that the complimentary nature of these new strategies can endow the current tutors with the ability to investigate, and hopefully enhance, the unconscious processes involved in emotional processing, reasoning and decision making. The following section will explain in more details these proposed strategies as well as present results from conducted studies.

3 Implicit Strategies for Intelligent Tutoring Systems

As discussed in the previous section, current ITS rely on explicit strategies that address only measured variables involved in the direct interaction between the tutor and the learner. In this paper we aim to enhance these strategies with implicit interventions that address more subtle unconscious processes involved in both learners' affect and cognition. The basis of this work relies on previous findings from neuropsychological studies suggesting the possibility of nonconscious perception (or perception without awareness), because of the existence of a boundary threshold between conscious and unconscious perception [15]. A stimulus below this threshold of awareness (called subliminal stimulus) cannot be consciously perceived but can yield emotional and cognitive reactions [15, 16]. This phenomenon is known as subliminal perception: it unconsciously solicits affective and cognitive mechanisms in the human brain [13].

Masked priming is one of the most widely used technique for subliminal perception [15]. It consists in projecting for a very short time, a subliminal stimulus or prime such as a word or a valenced image preceded and/or followed by the projection of a mask for a particular time. This mask is usually in the form of a set of symbols that have nothing to do with the prime in order to elude its conscious detection [15]. In this research, our goal is to enhance existing tutoring strategies with masked priming

techniques. We will present two different priming approaches, namely affective priming and cognitive priming applied in problem solving environments.

3.1 Affective Priming

Affective priming is a masked priming technique that consists in exposing participants to affective stimuli that can only be unconsciously perceived in order to implicitly elicit emotional reactions. These stimuli can be in the form of images or words charged with valenced semantics (e.g. smiling faces or positively connoted words). This technique is based on work in implicit memory [17] and automatic and unconscious processes related to stereotypes and attitudes [18]. This body of work suggests that implicit and unconscious elements are found in several psychological manifestations among which, emotional regulation. This technique has been studied in areas such as social behavior, advertising and stereotypes (see [16] for a review). Furthermore, these studies show that emotions are more likely influenced by these unconsciously perceived stimuli than by consciously perceived stimuli.

Conducted Study. We used an affective priming technique in order to implicitly condition learners’ affect within a logical problem solving environment. The aim of the study was to enhance learners’ implicit self-esteem while they follow a learning session about logics. The goal is to teach learners how to infer a logical rule from a series of data in order to find the missing element of a sequence. The session starts with a tutorial giving instructions and examples to get learners accustomed with the user interface and types of questions, then a series of logical exercises are given.

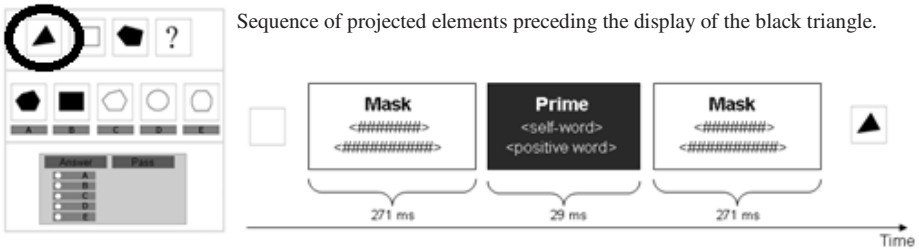


Fig. 1. Affective priming during problem solving

Three modules are taught, each one is concerned with specific forms of data: the first module deals with geometrical shapes, the second module with numbers and the third module focuses on letters (see [19] for more details). Learners are asked to respond as quickly and efficiently as possible to each of the 15 questions of the quiz. They are informed that they can either respond or pass the question and that a correct answer = 4 points, an incorrect answer = -1, and a no-answer = 0. Questionnaire materials (shapes, numbers and letters) are presented sequentially on the screen and subliminal primes are projected just before the materials.

In this study, a particular form of masked priming was used, namely the Evaluative Conditioning (EC) technique [20]. This method consists in subliminally projecting self-referential words (conditioned stimulus) such as “I” or the first name of the learner, paired with positively valenced words (unconditioned stimulus) such as “efficient”, “success” or “smart”. The idea behind EC is that this conditioning implicitly influences the semantic structure of associations (between words) in memory and hence, the automatic affective reactions resulting from this pairing [20]. Fig. 1 gives an overview of how the masked priming took place. Each prime, consisting of a self-referential word and a positive word, is projected for 29 ms and is preceded and followed by a 271 ms mask composed of sharp (#) symbols.

A total of 39 participants with a mean age of 27.31 ± 6.87 years, were recruited for the experiment. Participation was compensated with 10 dollars. They were randomly assigned either to the experimental condition ($N = 20$, 13 males) or to the control condition, without priming ($N = 19$, 11 males). Participants’ self-esteem was assessed with the Initial Preference Task (IPT), a widely used technique for assessing implicit self-esteem, using the Ipsatized double-correction scoring algorithm (see [21] for more details).

Two sensors were used to measure participants’ emotional reactions, namely galvanic skin response, known to be correlated to emotional arousal (low to high) and blood volume pulse from which heart rate, known to be correlated to valence (positive to negative), was extracted [22]. The proportions of emotions characterized with a positive valence and a neutral arousal (Target emotion proportions) were assessed with regards to the baseline values, to measure participants’ positive emotional activations using a two dimensional model of affect [23]. These specific emotions are assumed to provide a maximum of efficiency and productivity in learning [24].

Table 1. Experimental results of the affective priming study

	Experimental condition		Control condition	
	M	SD	M	SD
Self-esteem	1.68	0.94	1.08	0.99
Target emotion proportions	48.15	37.32	40.36	34.02
Final score	33.4	12.36	25.5	9.87
Number of passed answers	0.95	0.83	2.11	1.91

Further variables were recorded to measure learners’ performance, namely, the final score in the test and the number of passed answers. Results are summarized in Table 1. A significant evidence for the conditioning effect on self-esteem was found. Primed participants showed significantly higher self-esteem than the control group, $F(1, 37) = 4.84$, $p < 0.05$ ($M = 1.68$ vs. 1.08). Besides, they showed significantly higher proportions of target emotions, $F(1, 583) = 6.03$, $p < 0.05$, in the logical quiz ($M = 48.15\%$) with regards to non primed participants ($M = 40.36\%$). The scores of the test were also better in the experimental group ($F(1, 37) = 4.37$, $p < 0.05$, $M = 33.4$ vs. 25.5), and unlike the control group, participants were taking more risks by answering more questions (lower number of passed answers, $F(1, 37) = 7.45$, $p < 0.05$, $M = 0.95$ vs. 2.11) even if they were not completely sure of their answers.

3.2 Cognitive Priming

Like affective priming, cognitive priming is a masked priming technique, but with a different objective; the stimulus used (an answer or a hint about a question for example) is aimed toward positively enhancing specific cognitive processes such as reasoning or decision making toward the goal of implicitly enhancing knowledge acquisition. This technique is based on numerous findings in neuroscience providing evidence from the brain wave activity using electro-encephalography (EEG), demonstrating that unconsciously perceived stimuli can reach orthographic, lexical and motor levels of representations (see [25] for more details).

Conducted Study. In this second study, we used a cognitive priming technique in order to implicitly enhance learners’ reasoning abilities within a problem solving environment. The aim of the study was to use cognitive priming to specifically enhance the analogical reasoning abilities of students while learning how to construct an odd magic square of any order with the use of neither a calculator nor one mental arithmetic operation. A magic square of order n is a square containing n^2 distinct integers disposed in a way such as all the n numbers contained in all rows, columns or diagonals sum to the same constant (leftmost part of Fig. 2).

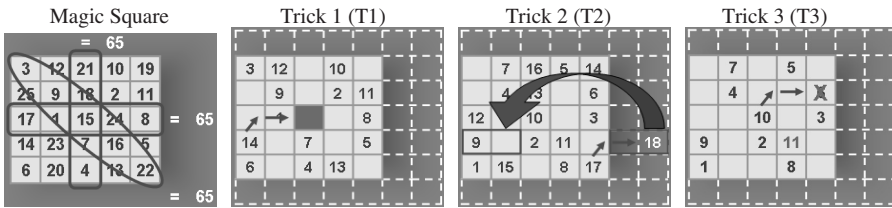


Fig. 2. Magic square and the three tricks taught

To construct such a magic square, three tricks are required (Fig. 2). These tricks are cumulative and thus the difficulty increases with each trick. The solution to the three tricks was not presented. Instead, the learners had to infer their own solutions, correctly figure out the different algorithm used in each trick and answer a series of 13 questions (see [26] for more details). Furthermore, learners were instructed to make the fewest amounts of mistakes possible whilst taking the shortest amount of time. Then, learners reported how they figured out each trick by choosing between the following: “I deduced the trick by intuition, logic, a little of both” (Trick answer type variable). Learners also reported how they answered each question by choosing between the following: “I answered the question by guessing the answer, by intuition or by logical deduction” (Question answer type variable). Learners’ brain activity (using EEG) was recorded to investigate changes in mental activity during reasoning.

In this study, two experimental conditions with different types of primes were considered namely Answer_cues and Miscues. The former condition intended to enable learners to reason faster while deducing the tricks. The primes are in the form of

arrows pointing at the answer to each trick as displayed in Fig.2. The Miscues condition is intended to mislead the learner using primes (arrows) that point to the wrong square on the screen. The idea of the study was to assess the effect of each type of prime on reasoning. In both conditions, primes (Answer_cues and Miscues) were displayed in each trick throughout the study. Each prime is projected for 33 ms and is followed and preceded by a mask of 275 ms consisting of random geometric figures.

A total of 43 participants with a mean age of 27 ± 3.5 years, were recruited for the experiment and were compensated with 10 dollars; they were randomly assigned either to the Answer_cues condition ($N = 14$, 7 males), to the Miscues condition ($N = 14$, 6 males) or to the Control condition, without priming ($N = 15$, 7 males).

We were interested in examining results related to performance (number of mistakes) with regards to the way learning occurred (Trick answer type), the way learners answered questions (Question answer type) and the group (Answer_cues, Miscues, Control). Significant effects were only found for the variables Trick answer type*group with regards to the number of mistakes with the following combinations: Logic*Answer_cues ($p = 0.002$, $\alpha = 0.05$, $\chi^2 = 16.949$), A little of both*Answer_cues ($p = 0.048$, $\alpha = 0.05$, $\chi^2 = 9.117$). Results seem to indicate that only Answer_cues, and not miscues, do significantly influence logical reasoning and decision making when learning a trick logically.

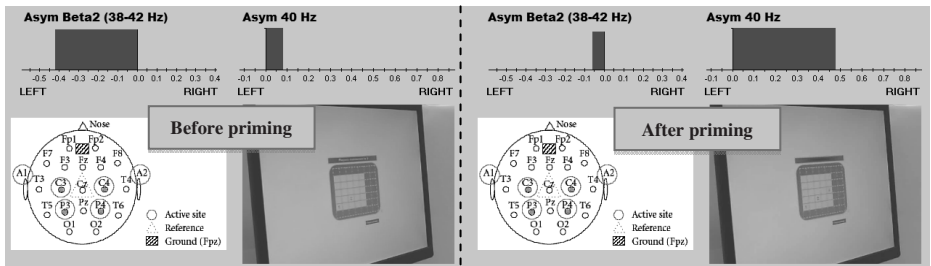


Fig. 3. Recording of cerebral changes following cognitive priming in intuitive reasoning

From the EEG data we were interested in investigating changes in two metrics that have previously been reported as relevant indicators of insightful problem solving (40Hz right asymmetry) [27] and complex arithmetic processing (Beta2 left asymmetry) [28]. We observed that the asymmetry values for the 40Hz ($p = 0.003$, $\alpha = 0.05$) and Beta2 ($p = 0.04$, $\alpha = 0.05$) in the Answer_cues group are significantly different than the Miscues group for the third and most difficult trick. Participants of the Answer_cues group seem to shift their attention from a complex arithmetic process (Beta2 left asymmetry decrease) toward an “insightful” problem solving strategy (40Hz right asymmetry increase), thus involving the right side of the brain, known to be an important actor in insightful problem solving. Fig. 3 depicts one such example recorded during learning where we see a female learner reporting learning the third trick (T3) by intuition. We see the decrease in Beta2 in the left brain and the increase in 40Hz in the right brain after priming, illustrating that a combination of these two metrics could indeed be an interesting indicator of a change in the reasoning strategy from complex arithmetics to an insightful reasoning during problem solving.

4 Discussion

The two presented studies have shown that learners' self-esteem and abilities to reason in a problem solving environment can be augmented through the use of affective and cognitive priming. The two experiments rely on a masked priming technique consisting in projecting stimuli below the threshold of conscious awareness (positively charged words paired with self-referential words and hints about the task).²

The affective and cognitive dimensions of these implicit interventions have been investigated and results have shown that electro-physiological sensors can provide current tutors with relevant information regarding the positive impact of the proposed approach in terms of learners' emotional and mental reactions. In light of those results, we believe that a hybrid ITS, one using both explicit and implicit strategies, can greatly enhance the interaction with a learner, enabling him to optimize his learning experience with both direct and indirect aspects. The proposed approach for integrating our implicit tutor is illustrated below.

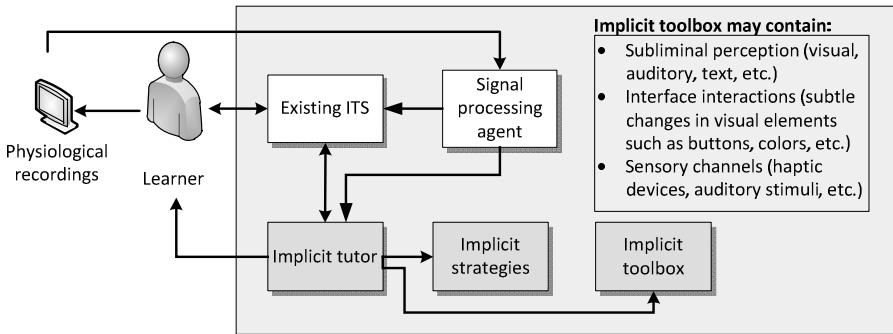


Fig. 4. Proposed approach

The implicit tutor is continuously communicating with the explicit tutor in search of the best strategy to apply in a given situation. To achieve this objective, the implicit tutor can choose one or multiple implicit strategies (cognitive and/or affective priming) as well as a *kind* of stimulus to project. The implicit toolbox is essentially a guide of all existing stimuli that are applicable for a situation. In this paper, we have presented two studies employing different subliminal stimuli (visual and textual). However, we believe that two issues need to be addressed, before this integration could take place in a real learning environment. First, the implicit strategies have to be tested in complex, real-life, lessons where deeper learning may take place and compare results. Second, ethical issues of deploying these strategies should also be explored. It would be interesting for example to reproduce these studies while informing the learners of *what* will they get but not *how*. In other words, explain that the system is built to provide help in a subtle way without revealing the kind of stimuli that will be used from the toolbox and crosscheck results.

² In both experiments, none of the participants has reported seeing the primes during the tasks.

5 Conclusion

This paper discusses a new approach to enhance ITS with implicit tutoring strategies targeting subtle indirect aspects in the interaction between the tutor and the learner. These strategies are based on unconsciously perceived interventions that address the automatic mechanisms associated to learners' affective and cognitive processing inherent to learning using the subliminal perception. We demonstrated our approach through two experimental studies showing two different applications of the masked priming technique, namely affective priming and cognitive priming. We showed that both learners' cognitive and affective states can be conditioned implicitly and that these strategies can produce a positive impact on students' interaction experience and enhance learning. The first study showed that affective priming had a positive impact on learners' outcomes, self-esteem, and emotional reactions. The second study showed that cognitive priming enhanced learners' reasoning abilities and EEG data demonstrated that cognitive abilities such as analogical reasoning can potentially be monitored, assessed and positively influenced under priming conditions.

In our future work, we are interested in developing a tutor that will integrate both implicit and explicit interventions. This tutor will select appropriate strategies according to learners' profile, and real time data from learners' progress and emotional and mental reactions.

Acknowledgments. We acknowledge the National Science and Engineering Research Council (NSERC) and the Tunisian Ministry of Higher Education and Scientific Research for funding this work.

References

1. Seidel, R.J., Park, O.: An Historical Perspective and a Model for Evaluation of Intelligent Tutoring Systems. *Journal of Educational Computing Research* 10(2), 103–128 (1994)
2. Phelps, E.A.: Emotion and cognition: Insights from Studies of the Human Amygdala. *Annual Review Psychology* 57, 27–53 (2006)
3. Arroyo, I., Ferguson, K., Johns, J., Dragon, T., Meheranian, H., Fisher, D., Barto, A., Mahadevan, S., Woolf, B.P.: Repairing Disengagement With Non-Invasive Interventions. In: *Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*. IOS Press (2007)
4. Bursleson, W.: Affective learning companions: strategies for empathetic agents with real-time multimodal affective sensing to foster meta-cognitive approaches to learning, motivation, and perseverance, MIT PhD thesis (2006)
5. Conati, C.: Probabilistic Assessment of User's Emotions in Educational Games. *Applied Artificial Intelligence* 16, 555–575 (2002)
6. D'Mello, S., Graesser, A.: Automatic detection of learner's affect from gross body language. *Applied Artificial Intelligence* 23, 123–150 (2009)
7. Litman, D., Forbes-Riley, K.: Annotating student emotional states in spoken tutoring dialogues. In: *SIGdial Workshop on Discourse and Dialogue*, Boston, USA, pp. 144–153 (2004)
8. Woolf, B., Bursleson, W., Arroyo, I., Dragon, T., Cooper, D., Picard, R.: Affect aware tutors: recognising and responding to student affect. *Int. J. Learn. Technol.* 4(3/4), 129–164 (2009)

9. Pekrun, R.: The Impact of Emotions on Learning and Achievement: Towards a Theory of Cognitive/Motivational Mediators. *Applied Psychology* 41(4), 359–376 (1992)
10. Kim, Y.: Empathetic virtual peers enhanced learner interest and self-efficacy. In: Workshop on Motivation and Affect in Educational Software, in Conjunction with the 12th International Conference on Artificial Intelligence in Education (2005)
11. McQuiggan, S.W., Lee, S.Y., Lester, J.C.: Early Prediction of Student Frustration. In: Pava, A.C.R., Prada, R., Picard, R.W. (eds.) *ACII 2007*. LNCS, vol. 4738, pp. 698–709. Springer, Heidelberg (2007)
12. Prendinger, H., Ishizuka, M.: The empathic companion: A character-based interface that addresses users' affective states. *Applied Artificial Intelligence* 19, 267–285 (2005)
13. Kouider, S., Dehaene, S., Jobert, A., Le Bihan, D.: Cerebral Bases of Subliminal and Supraliminal Priming during Reading. *Cereb. Cortex* 17(9), 2019–2029 (2007)
14. Mayer, J.D., Allen, J., Beaugard, K.: Mood inductions for four specific moods: A procedure employing guided imagery vignettes with music. *Journal of Mental Imagery* 19, 133–150 (1995)
15. Del Cul, A., Baillet, S., Dehaene, S.: Brain dynamics underlying the nonlinear threshold for access to consciousness. *Public Library of Science, Biology* 5(10), 2408–2423 (2007)
16. Hassin, R., Uleman, J., Bargh, J.: *The new unconscious*. Oxford University Press, Oxford (2005)
17. Tulving, E., Schacter, D.: Priming and human memory systems. *Science* 247(4940), 301–306 (1990)
18. Bargh, J.A., Chaiken, S., Govender, R., Pratto, F.: The generality of the automatic attitude activation effect. *Journal of Personality and Social Psychology* 62(6), 893–912 (1992)
19. Jraidi, I., Frasson, C.: Subliminally Enhancing Self-esteem: Impact on Learner Performance and Affective State. In: Alevén, V., Kay, J., Mostow, J. (eds.) *ITS 2010*. LNCS, vol. 6095, pp. 11–20. Springer, Heidelberg (2010)
20. Grumm, M., Nestler, S., Von Collani, G.: Changing explicit and implicit attitudes: The case of self-esteem. *Journal of Experimental Social Psychology* 45(2), 327–335 (2009)
21. LeBel, E.P., Gawronski, B.: How to find what's in a name: Scrutinizing the optimality of five scoring algorithms for the name-letter task. *European Journal of Personality* 23, 85–106 (2009)
22. Lang, P.J.: The emotion probe: Studies of motivation and attention. *Am. Psy.* 50(5), 372–385 (1995)
23. Russell, J.: A circumplex model of affect. *JPSP* 39, 1161–1178 (1980)
24. Kaiser, R.: Prototypical development of an affective component for an e-learning system. Master Thesis, Department of Computer Science, University of Rostock, Germany (2006)
25. Gaillard, R., Naccache, L., Pinel, P., Clémenceau, S., Volle, E., Hasboun, D., Dupont, S., Baulac, M., Dehaene, S., Adam, C., Cohen, L.: Direct Intracranial, fMRI, and Lesion Evidence for the Causal Role of Left Inferotemporal Cortex in Reading. *Neuron* 50, 191–204 (2006)
26. Chalfoun, P., Frasson, C.: Subliminal cues while teaching: HCI technique for enhanced learning. *Advances in Human Computer Interaction: Special Issue on Subliminal Communication in Human-Computer Interaction* (January 2011)
27. Sandkühler, S., Bhattacharya, J.: Deconstructing insight: EEG correlates of Insightful Problem Solving. *PLOS One* 3(1) (2008)
28. Hyungkyu, K., Jangsik, C., Eunjung, L.: EEG Asymmetry Analysis of the Left and Right Brain Activities During Simple versus Complex Arithmetic Learning. *Journal of Neurotherapy* 13 (2009)

Rudeness and Rapport: Insults and Learning Gains in Peer Tutoring

Amy Ogan¹, Samantha Finkelstein¹, Erin Walker²,
Ryan Carlson¹, and Justine Cassell¹

¹ Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA 15213

² School of Computing, CIDSE, Arizona State University, Tempe, AZ 85282

{aao, slfink, rcarlson, justine}@cs.cmu.edu,
erin.a.walker@asu.edu

Abstract. For 20 years, researchers have envisioned artificially intelligent learning companions that evolve with their students as they grow and learn. However, while communication theory suggests that positivity decreases over time in relationships, most tutoring systems designed to build rapport with a student remain adamantly polite, and may therefore inadvertently *distance* the learner from the agent over time. We present an analysis of high school friends interacting in a peer tutoring environment as a step towards designing agents that sustain long-term pedagogical relationships with learners. We find that tutees and tutors use different language behaviors: tutees express more playfulness and face-threat, while tutors attend more to the task. This face-threat by the tutee is associated with increased learning gains for their tutor. Additionally, a small sample of partners who were strangers learned less than friends, and in these dyads increased face-threat was negatively correlated with learning. Our findings support the idea that learning companions should gradually move towards playful face-threat as they build relationships with their students.

Keywords: Rapport, impoliteness, virtual peers, ECA, teachable agent.

1 Introduction

Peer tutoring, a paradigm in which one student tutors another of a similar ability, results in deep learning gains for the tutor [1]. Peer tutoring provides a social motivation for the tutor to attend more in order to effectively explain concepts [2]. In addition, the tutor engages in a series of cognitive steps that improve learning, such as constructing explanations and reflecting on errors [3]. The tutee plays an active role in this process by challenging, contradicting, and questioning the tutor's moves [3] causing the tutor to engage in increased reflection and self-explanation [1].

In the ITS community, an effort has been underway to develop virtual characters that act as a tutee, or *teachable agent*, in order to leverage the benefits of human peer tutoring [4, 5, 6]. However, most teachable agents focus on the cognitive elements of the interaction and, to date, none have been designed based on analyses of the social behaviors that emerge as a part of successful peer tutoring. There is therefore great

opportunity to expand on the social capabilities of teachable agents in order to create rapport in the service of increased learning. Ideally, these social teachable agents – and other kinds of virtual peers - will be able to build long-term relationships with students to support them in their educational goals [as proposed in 7].

Researchers have previously designed polite intelligent tutors based on Goffman’s theory of face, that is, the public self-image that people project [8]. Brown and Levinson describe positive face as the desire for one’s image to be appreciated and negative face as the desire to not be impeded in one’s actions [9]. Existing systems avoid threatening positive and negative face by giving praise, providing reassurance, or hedging requests [10, 11]. However, while politeness serves a function early in a relationship, positivity is claimed to decrease as rapport increases in human-human interactions [12]. Culpeper’s theory of impoliteness [13] describes the role of behaviors such as insults and challenges which are considered face-threatening; they harm the addressee’s positive or negative face, and may cause offense [9]. However, impoliteness has a number of functions in conversation. It may serve to upend power imbalances [13] or even to reinforce solidarity and rapport among people with preexisting relationships [14, 15]. Teens in particular have been shown to use “rude” language to positive social effect [16]. For that reason in this paper we evaluate the strategies and functions these language behaviors effect within particular contexts [17].

Evidence suggests that impoliteness is important in human-agent relationships as well. Our previous work demonstrated that negative remarks (such as teasing and frustration) directed to a virtual tutee in a think-aloud protocol were associated with increased learning gains on the part of the tutor [18]. There have been a few efforts to create intelligent tutoring systems that use rudeness (such as sarcasm) as a rapport-building mechanism [e.g., 19]. These systems were positively received by students, but were not based on analyses of human-human interaction, and learning gains were not assessed. We know that learners apply the same norms of social interaction to learning companions as to human conversational partners [20]; therefore, understanding human-human behavior is critical in the development of a system able to develop a natural social relationship with the learner over time, in the service of learning.

In this work we analyze dialogues between pairs of students participating in a peer tutoring intervention by annotating 54 conversations for language features (e.g., complaining), which we group into conversational strategies (e.g. face threat), and also code for social functionality (positivity and impoliteness). These students were friends and thus are presumed to have pre-existent relationships. We use these data to investigate two research questions: in human-human peer tutoring dialogues, can we link particular surface level language features to social conversational strategies, such as face-threat, and does this linkage differ between tutees and tutors (*RQ1*)? Do these conversational strategies relate to social functions, and does this have an effect on peer tutor learning (*RQ2*)? An exploratory analysis of 6 dyads of strangers (presumed not to have prior relationships) allows us to address a third research question that may provide insight into the design of relationship-building systems that evolve over time: How do the relationship-affecting conversational strategies of strangers relate to learning, and differ from friends (*RQ3*)? Our results yield specific design guidelines for implementing relationship-building behaviors in an interactive tutoring system – specifically, a teachable agent, grounded in our findings from human-human tutoring.

2 Study

To assess the social behaviors of real students in a peer tutoring context, we re-examined data collected for a previous study to evaluate the impact of an intervention that monitored students' collaboration and could provide adaptive support [21]. A peer tutor and tutee interacted over chat while the tutee worked on algebra problems. Participants were 130 8th-10th grade students (49 male) with diverse racial backgrounds from one American high school who had previously received classroom instruction on relevant domain material. Participants were asked to sign up for the study with a friend. Those who were interested but had no partner were matched with another unmatched participant. 54 dyads were friends and 6 dyads were strangers. Participants took a 20-minute pre-test on relevant math concepts, and then spent 20 minutes working alone with the computer to prepare for tutoring. One student in each dyad was randomly assigned the role of tutor, while the other was given the role of tutee. They spent the next 60 minutes engaging in tutoring. Finally, students were given a domain post-test isomorphic to the pretest, and compensated.

3 Data Annotation

We analyzed the tutoring dialogues using a scheme we developed to capture three levels of relationship-building and signaling: specific language behaviors, the conversational strategies they contribute to, and their associated social functions. The distinction between these three levels was drawn from work in pragmatics [8] that allows us to interpret the different social functions of groups of language behaviors used in context (such as insults used to indicate solidarity and therefore build rapport, or politeness used to indicate distance and therefore push away) [17]. Much of our analysis focuses on the friend dyads: 5,408 utterances from 108 participants over 54 sessions. 2,333 of these utterances were produced by the tutee and 3,075 by the tutor.

Thirteen surface-level language behaviors, shown in Table 1, were coded by two independent raters, based on research on impoliteness [13], positivity in tutorial dialogues [22], and computer-mediated communication [23]. Each utterance could receive more than one code. Counts of features were normalized by the total number of utterances spoken by that participant. Based on the Principal Components Analysis presented in section 4.1 below, codes were grouped into factors representing conversational strategies. Each utterance was also annotated for two types of social functions, motivated by the literature on rapport-building and -maintaining. This entailed examining the interlocutor's *response* to a given act; the same utterance may serve a different social function depending on its reception. The positivity code was expanded beyond politeness to encompass other indicators of positivity such as those used by Boyer [22] including empathy, praise, and reassurance, in addition to cooperative talk. ($M_{\text{tutor}}=14\%$; $M_{\text{tutee}}=17\%$; Cohen's $K=.79$). The impoliteness code expresses negativity, combining both cooperative rudeness such as teasing and banter (e.g. "I hate youuuu :D"), and uncooperative rudeness which seems to intend to cause offense (e.g., "your horrible at this.") [15] ($M_{\text{tutor}}=8\%$; $M_{\text{tutee}}=12\%$; Cohen's $K=.72$).

4 Results

4.1 Surface-Level Language Features and Role Differences (*RQ1*)

With the goal of understanding how surface-level language features contributed to social conversational strategies in peer tutoring dialogue, and what strategies were most frequent, we performed a Principal Components Analysis (PCA) with Varimax rotation. This allowed us to move from a focus on individual behaviors to understanding how particular *types* of behaviors are used in meaningful ways in this population. The annotated language features collapsed into four factors, which explained 76% of the total variance. Table 1 shows the mapping between the language features and factors. Based on the pragmatics theory cited, we interpret the four factors as follows:

Playfulness to lighten the mood or mitigate negativity: *Laughter, extra letters, emoticons, off-task behavior, inclusive complaining (about a third person).*

Face-threat remarks directed toward the partner: *Direct insults, condescension/brags, challenges, exclusive complaining (about the other person).*

Attention-getting to draw the partner back on task: *Message enforcers, pet names.*

Emphasis to add emotive features: *Excessive punctuation, capitalization.*

Table 1. Annotation scheme divided into factors, with mean normalized behaviors for tutors and tutees, and Cohen’s kappa between raters. Significantly higher values (comparing tutor to tutee) marked with *($p < .05$), **($p < .01$), ***($p < .001$).

Factor 1: Playfulness		K	Tutor M	Tutor SD	Tutee M	Tutee SD
Laughter (L)	“hahaha,” “lol!”	1	4.3%	.07	6.2%	.08
Extra letters (EL)	“tutor help meeeee”	.94	4.8%	.07	7.2%	.10
Emoticons (E)	“h8u <3,” “did it! :-D”	1	2.3%	.04	3.4%	.05
Inc. complain (I)	“this thing is such butt”	.78	4.8%	.05	8.4%**	.08
Off-task (O)	“I am so hungryyyyy!”	.71	7.3%	.08	9.9%	.11
Factor 2: Face-threat						
Direct insult (DI)	“you’re being so weird.”	1	1.3%	.02	2.3%	.03
Condescension (C)	“... obviously add now. Duh.”	1	0.9%	.02	2.9%***	.04
Challenge (Ch)	tutor: okay bro now subtract tutee: “nao, not right, I add.”	.91	3.0%	.03	5.6%***	.06
Excl. complain (EC)	“you’re making no sense dude”	8	1.2%	.02	3.4%***	.04
Factor 3: Attention						
Enforcer (Ef)	“pay attention – now divide.”	.85	1.7%***	.02	.39%	.01
Vocatives (V)	“homie, that’s not right.”	.9	3.0%	.04	5.0%	.06
Factor 4: Emphasis						
Punctuation (P)	“I don’t even see an rt?!?!?”	.89	6.5%	.09	9.9%	.12
Capitalization (Ca)	“I SAID you ADD THE BD!!”	1	3.1%	.06	5.9%	.11

The PCA allowed us to compute a regression value for each of these four factors for each participant, which represented their utterances in terms of these values (e.g., the total ‘face threat’ value of the conversation). We then investigated how tutee and tutor utterances differed along these factors by running a MANOVA with role as the independent variable, and four dependent variables: playfulness, face threat, attention, and emphasis. We found that tutees used more *playful* ($F(1,107)=8.33$, $p<.01$) and *face-threatening* strategies ($F(1,107)=16.62$, $p<.001$), while tutors used more *attention-getting* strategies ($F(1,107)=9.72$, $p<.01$). *Emphasis* use was equivalent ($p>.1$).

These results indicate that tutees are responsible for introducing playful and face-threatening strategies in the conversation, while tutors instead bring attention back to the task. The following is a representative example from the corpus:

[] **Tutee:** *I need help tuter*
 [] **Tutor:** *What do you do next?*
 [E,DI,Ch,Ca] **Tutee:** *it told me aks why you got it wrong. ANSWER: your stupid XD*
 [Ef,V,P,Ca] **Tutor:** *dude! STOP! Add VT to both sids!*

Typically, tutees’ requests for help involve excessive punctuation or extra letters, both shown to contribute to “playfulness in language” in texting [23]. Tutors respond to these requests with on-task utterances (e.g. “now you need to add gh to both sides.”) If tutees reply with face-threat, tutors use vocatives and message enforcers to bring the conversation back on task. We explore this interplay further in section 4.3.

4.2 Conversational Strategies and Social Functions (RQ1)

With the results of the PCA, we examined the social conversational strategies effected by off-task social language such as complaining and exclamations. We next analyze the social functions of expressions of positivity and impoliteness in particular, and the relationship between conversational strategies and social functions. We investigated tutor and tutee differences using a MANOVA with role as the independent variable, and positivity and impoliteness as the dependent variables. Given the PCA results, it is not surprising to find that tutees were significantly more impolite ($F(1,107)=7.74$, $p<.01$) than tutors, and marginally less positive ($F(1,107)=3.60$, $p=.06$).

We thus analyzed the connection between conversational strategies and social functions separately for tutors and tutees using bivariate correlations. While the tutees primarily expressed positivity with *playfulness* ($r=.276$, $p<.05$), tutors expressed positivity with *emphasis* ($r=.359$, $p<.01$) in addition to *playfulness* ($r=.436$, $p=.001$). That is, tutees primarily achieved conversational positivity through playful non-standard writing, complaining about the task they were doing, or interjecting off-task comments into the dialogue. Tutors used these techniques, but additionally expressed positivity through emphasizing their utterances with excess punctuation and using the caps lock, such as to praise their partner (e.g. “YAY you DID IT!!!”).

Differences were also apparent in impoliteness, which tutees primarily expressed through *face-threatening* features ($r=.5$, $p<.001$) and *attention-getting* features ($r=.306$, $p<.05$). In contrast, tutors used only *attention-getting* features such as message enforcers to indicate impoliteness ($r=.517$, $p<.001$). These correlations are

supported by qualitative analyses of the data, such as the following example where the tutor continues to keep the conversation on task despite the tutees' face threat.

[DI, C, EC] **Tutee:** *your horrible at this*

[Ef] **Tutor:** *thanks... i try. Just restart the problem.*

[EC] **Tutee:** *can you actually say something that i can fully understand*

[] **Tutor:** *add vt then you have to solve for t so subtract bh from both sides*

Despite these apparent differences, a correlation was found in the language use of tutor-tutee pairs, ranging from a weak correlation for *attention-getting* ($r=.244, p<.05$) to very strong correlations, e.g., *playfulness* ($r=.840, p<.001$). This result indicates synchrony or coordination in the dyads, a marker of the kind of rapport that characterizes long-term relationships [12]. So while partners identify their roles (tutor or tutee) through conversational strategies, they also index their rapport by not straying too far from the partner's language patterns.

4.3 Learning Gains, Language Features, and Social Functions (RQ2)

Our second research question investigates how language use relates to learning outcomes. To address the design of teachable agents, we examined the relation between tutees' behaviors and their partner's learning gains. A stepwise regression looking at the four conversational strategies, the social functions, and the interactions among these features ($r^2=.07$, $F(1,107)=1.824$, $p=.1$) found that face threat is a positive predictor of learning gains ($\beta=.375$, $t=2.22$, $p=.03$), while the interaction term of face threat x positivity is a negative predictor of learning gains ($\beta=-.320$, $t=-1.86$, $p=.06$). This means that as face threat increases, tutors learn more, and the learning benefits of face threat can be enhanced by appropriate use of positivity. In essence, face threatening conversational strategies with socially positive functions enhance the learning interaction. On the other hand, high positivity with low face threat from the tutee is actually associated with lower levels of learning. That is, positive social interaction that does not contain the kind of face-threatening behavior that characterizes rapport in this age group may either signal less rapport, or actually reduce the connection between the dyad, and therefore reduce learning gains. In addition, a lack of face-threatening interactions may indicate a lack of the challenging tutor moves by the tutee, that increase the cognitive benefits of the interaction.

In order to explore how these functions were associated with learning, we quantified how tutors reacted to the use of positivity and face threat by the tutee. We used transition matrices to evaluate the conditional probability of a feature occurring in one turn based on the presence of another in the prior turn (collapsing consecutive utterances by the same speaker to form turns). Thus, we calculate the probability that the tutor will use feature *B* given that the tutee used feature *A* in the previous turn. By examining these transition matrices (see Table 2 for values), we can identify common response patterns in the dyads. We found that when the tutee exhibits positivity, the tutor is no more likely to respond with positivity (42%) than with a response that contains no coded social features (46%). Generally, these instances with no codes are task-related, non-emotive statements such as "ok add five". When a tutee exhibits

face-threat, on the other hand, the tutor is more likely (57%) to respond with no social features (indicating that the tutor is likely using task features). Thus, while tutors demonstrate no particular pattern of response to positivity, they are likely to respond to negativity with strategies to keep the conversation on-task. Negative behaviors such as face-threats on the part of the tutor therefore are more likely to elicit effective tutoring behaviors than are positive behaviors such as praise.

Reversing the direction of the conditional probability demonstrated that while tutee behavior with respect to positivity is similar to observed tutor behavior, tutee behavior when the tutor exhibits face threat is very different. Whereas a tutor is not likely to engage her tutee, the tutee is just as likely to fire back with impoliteness (36%) as she is to refrain (33%). The imbalance of power between the two roles within the context of an existent friendship may lead the tutee to try to regain the upper hand through face-threat, while the tutor tries to regain authority through task behavior.

Table 2. Selected entries from transition matrix, for friends. Left-most column shows initiator and language feature exhibited. Transition percentages indicate number of times feature was seen in partner's response, divided by the total number of partner responses to feature. Because features can co-occur in an utterance, response percentages may not always sum to 1.

		Partner response			
		None (%)	Positivity (%)	Impoliteness (%)	Off-topic (%)
Tutee	Positivity	46	42	8	24
	Face threat	57	19	23	15
Tutor	Positivity	38	42	11	23
	Face threat	33	16	36	19

4.4 Relationships: Friends and Strangers (RQ3)

In addition to the fifty-four dyads analyzed above, six dyads participated in the study who either did not sign up with a friend, or whose schedule changed requiring them to be partnered with another unmatched participant. Given literature that suggests that the demonstration of rapport differs between friends and strangers (those who are building rather than maintaining rapport) [13], these dyads provide a contrast to the data from partners who were already friends.

An ANOVA with role and friend as independent variables and partner learning gains as the dependent variable shows that friends had significantly greater learning gains than strangers ($F(1,120)=4.71$, $p=.03$; $M_{\text{stranger}}=-.17$, $SD_{\text{stranger}}=.35$, $M_{\text{friend}}=.02$, $SD_{\text{friend}}=.28$), while role was not significant in this analysis ($p>.1$). Given that strangers tended to learn less from the intervention, we investigated whether the factors related to their learning were equivalent to those of friends. A stepwise regression showed that *face threat* is a strong *negative* predictor of learning gains for strangers (overall model: $r^2=.44$, $F(1,11)=8.516$, $p=.015$; effects of *face threat*: $\beta=-.678$, $t=-2.92$, $p=.015$). In other words, in direct contrast to friends, greater amounts of

face-threatening behaviors by non-friend tutees actually do threaten the relationship, and hence are associated with lower learning gains for the tutor.

The behavior transition matrices demonstrate that, for strangers, in every possible transition, a response containing no coded features was more likely than any other behavior. That is, strangers tend to produce task-related, non-emotive statements in response to all other behaviors. Furthermore, when the stranger tutee exhibits face threat towards the tutor, only 14% of the time will the tutor reply with impoliteness, with other transitions producing similar results. It is also notable that there are only three instances of face threat from the stranger tutors, and in each case the tutee responds with no coded features. Strangers are also much more hesitant to respond to positivity. Compared to friends, we see strangers responding with a much more restricted set of behaviors, suggesting a discomfort with confrontation not seen in friend dyads [24]. When face threat does happen, it is not beneficial for learning.

5 Discussion and Conclusions

Though most intelligent tutoring systems that attempt to build rapport with the learner do so through politeness, actual peer tutors employ a great deal of impolite and face-threatening behavior. In this paper, we have analyzed chat data from a computer-supported peer tutoring intervention to investigate how peer tutors and tutees use surface-level language features to contribute to a set of particularly teen-like communicative strategies. These strategies interact with positive and negative social relationship functions in ways which correlate with learning gains. Importantly, the pre-existing social relationships between the partners also matter, as this chain of effects differs in interesting ways between friends and strangers.

Through a factor analysis that investigated groupings of thirteen language features, we determined ways in which peer tutors and tutees use these various features to accomplish playfulness, face-threat, attention-getting, and emphasis communicative strategies. Understanding how students use the same features to achieve different positive and negative communicative strategies within a dialogue will allow us to develop teachable agents who are able to index the language features of a community to respond appropriately to their partner both socially and cognitively.

An investigation of how tutors and tutees differentially use these communicative strategies showed that tutees tend to be responsible for the bulk of positive and negative social input in a dialogue, while tutors keep the interaction on track by directing the tutee's attention. Yet, tutors and tutees do act in synchrony, with dyads displaying correlated levels of each of the four communicative strategies, and their consequent social functions such as positivity and impoliteness. The synchrony between the partners is an index of their friendship, while the asynchrony in the use of social language – and negativity in particular – may be demonstrating an attempt by the tutee to redress the power differential of the two roles, and by the tutor to maintain the higher status of instructor. This conflict, however, keeps within the frame of friendship as demonstrated by the lack of negative response by the tutor to the tutee's insults.

It is undoubtedly the fact that the friendship supports – or even thrives on – so much apparent negativity that leads to our result that increased face threat on the tutee’s part leads to increased learning on the tutor’s part. Tutees are keeping the tutors on their intellectual toes by challenging their help, demanding explanations, and questioning their methods [3]. What we show in this work is that these playground strategies of playful insults, criticisms, and condescension [16] can serve the same goal as the challenges and contradictions that mark good peer tutoring [3]. It is likely, however, that impoliteness has its limits; excessive criticism or insult may fail at both social and cognitive goals. Accordingly, we find that positivity also plays a critical role in these tutoring interactions, as it enhanced the learning benefits of face threatening acts, while it was not associated with learning on its own. The interactions between these factors are complex, and leave ample room for future work.

Though preliminary analyses indicate that even strangers will use some face-threatening behaviors during tutorial dialogues, among these dyads, the presence of such behaviors leads to reduced learning gains for both the tutor and the tutee. We cannot and should not assume that a teachable agent and its tutor begin as friends. Thus, in the design of such agents, we will want to investigate the effectiveness of a model that begins with very few face-threatening behaviors. Neither should we abandon hope, however, that the agent and his tutor will embark on a relationship over time. We therefore propose that over multiple sessions, the agent begin to drive the learning by becoming increasingly face-threatening through challenges and even insults, while maintaining a synchrony with the tutor’s usage of face-threat in return.

As friendship between partners was not randomly controlled in this data, future work should investigate tutor-tutee rapport between friends and strangers in a more controlled setting. And as children perform many more social moves than we coded, future directions should examine additional behaviors, such as a breakdown of politeness moves typically referenced in intelligent tutoring systems [10]. It is also important to note that our data is rooted within a particular context; specifically, that of teenage American students interacting through a textual interface. While our results may not generalize beyond this context, it is fortunately one ripe for research in educational technology. In any case, if tutoring systems and virtual peers are to play a role as long-term learning companions, they must have the ability to evoke, signal and maintain relationships in ways appropriate to the age group they are built for, and that they must be capable of changing those relational strategies over time.

Acknowledgements. This research was supported by NSF Awards DRL-0910176 and IIS-0968485 and the IES, U.S. Dept. of Ed., through R305A090519 to Carnegie Mellon University. It was also supported by the Pittsburgh Science of Learning Center, funded by NSF Award SBE-0836012. The opinions expressed are those of the authors and do not represent views of the funders.

References

1. Sharpley, A., Irvine, J., Sharpley, C.: An examination of the effectiveness of a cross-age tutoring program in mathematics for elementary school children. *American Educational Research Journal* 20(1), 101–111 (1983)

2. Rohrbeck, C.A., Ginsburg-Block, M.D., Fantuzzo, J.W., Miller, T.R.: Peer-assisted learning interventions with elementary school students: a meta-analytic review. *Journal of Educational Society* 95(2), 240–257 (2003)
3. Webb, N.: Peer interaction and learning in small groups. *International Journal of Educational Research* 13(1), 21–39 (1989)
4. Brophy, S., Biswas, G., Katzberger, T., Bransford, J., Schwartz, D.: Teachable agents: combining insights from learning theory and computer science. *Artificial Intelligence in Education* 50, 21–28 (1999)
5. Matsuda, N., Yarzebinski, E., Keiser, V., Raizada, R., Stylianides, G.J., Cohen, W.W., Koedinger, K.R.: Learning by Teaching SimStudent – An Initial Classroom Baseline Study Comparing with Cognitive Tutor. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS, vol. 6738, pp. 213–221. Springer, Heidelberg (2011)
6. Gulz, A., Silvervarg, A., Sjoden, B.: Design for off-task interaction - Rethinking pedagogy in technology enhanced learning. In: *Proceedings of the 10th IEEE International Conference on Advanced Learning Technologies* (2010)
7. Chan, T., Baskin, A.: Studying with the prince: the computer as a learning companion. In: ITS 1988 Conference, Montreal, Canada, pp. 194–200 (1988)
8. Goffman, E.: *The presentation of self in everyday life*. Doubleday, NY (1959)
9. Brown, P., Levinson, S.: Universals in Language Usage: Politeness phenomena. In: Goody, E.N. (ed.) *Questions and Politeness: Strategies in Social Interaction*. University Press, London (1978)
10. McLaren, B., DeLeeuw, K., Mayer, R.: Polite web-based intelligent tutors: Can they improve learning in classrooms? *Computers and Education* 56(3), 574–584 (2011)
11. Johnson, W.L., Rizzo, P.: Politeness in Tutoring Dialogs: “Run the Factory, That’s What I’d Do”. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) ITS 2004. LNCS, vol. 3220, pp. 67–76. Springer, Heidelberg (2004)
12. Tickle-Degnen, L., Rosenthal, R.: The Nature of Rapport and its Nonverbal Correlates. *Psychological Inquiry*, 285–293 (1990)
13. Culpeper, J.: Towards an anatomy of impoliteness. *Journal of Pragmatics* 25(3), 349–367 (1996)
14. Straehle, C.A.: “Samuel?” “Yes dear?” Teasing and conversational rapport. In: Tannen, D. (ed.) *Framing in Discourse*. Open University Press, New York (1993)
15. Keinpointer, M.: Varieties of rudeness: types and functions of impolite utterances. *Functions of Language*, 251–287 (1997)
16. Ardington, A.: Playfully negotiated activity in girls’ talk. *Journal of Pragmatics* 38(1), 73–95 (2006)
17. Mills, S.: Gender and politeness. *Journal of Politeness Research* 1(2), 263–280 (2005)
18. Ogan, A., Finkelstein, S., Mayfield, E., D’Adamo, C., Matsuda, N., Cassell, J.: “Oh dear Stacy!” Social Interaction, Elaboration, and Learning with Teachable Agents. In: *To appear in Proceedings of CHI 2012* (2012)
19. Graesser, A., McNamara, D.: Self-Regulated Learning in Learning Environments with Pedagogical Agents that Interact in Natural Language. In: *The Measurement of Learners’ Self-Regulated Cognitive and Metacognitive Processes While Using Computer-Based Learning Environments*, pp. 234–244 (2010)
20. Cassell, J.: Towards a Model of Technology and Literacy Development: Story Listening Systems. *Journal of Applied Developmental Psychology* 25(1), 75–105 (2004)
21. Walker, E., Rummel, N., Koedinger, K. R.: Adaptive support for CSCL: Is it feedback relevance or increased accountability that matters? In: *Proceedings of the 10th International Conference on Computer-Supported Collaborative Learning*, pp. 334–342 (2011)

22. Boyer, K.E., Phillips, R., Wallis, M., Vouk, M.A., Lester, J.C.: Balancing Cognitive and Motivational Scaffolding in Tutorial Dialogue. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 239–249. Springer, Heidelberg (2008)
23. Herring, S., Zelenkauskaite, A.: Symbolic Capital in a Virtual Heterosexual Market. *Written Communication* 26, 5–31 (2009)
24. Cassell, J., Gill, A., Tepper, P.: Coordination in Conversation and Rapport. In: *Proceedings of the Workshop on Embodied Natural Language*. Association for Computational Linguistics (2007)

On Pedagogical Effects of Learner-Support Agents in Collaborative Interaction

Yugo Hayashi

College of Information Science and Engineering, Ritsumeikan University
1-1-1, Nojihigashi, Kusatsu, Shiga, 525-8577 Japan
yhayashi@fc.ritsumei.ac.jp

Abstract. The present study was conducted to investigate if and how conversational agent can facilitate explanation activity that is conducive to learning. This was investigated through two experiments where pairs of participants, who were enrolled in a psychology course, engaged in a task of explaining to their partners the meanings of concepts of technical terms taught in the course. During the task, they interacted with a conversational agent, which was programmed to provide back-channel feedbacks and meta cognitive suggestions to encourage and facilitate conversational interaction between the participants. The findings of the experiments suggested that (1) a conversational agent can facilitate a deeper understanding of concept when participants are attentive to its presence, and (2) affective positive feedbacks from conversational agent facilitates explanation and learning performance.

Keywords: collaboration, explanation activities, pedagogical agents, affective learning.

1 Introduction

Advances in communication technologies made it possible to develop a system which aids human interaction and supports cognitive operation. One of such enterprises includes researches to develop embodied conversational agents to support educational system. In the fields of cognitive science and learning science, researchers on collaborative learning have shown that successful understanding or acquisition of new concepts depends greatly on how explanations are provided. In this study the task of explanation is experimentally investigated by using a conversational agent that serves as a teaching assistant. The purpose of the experiment is to find out if the presence of conversational agents facilitates learning and what kind of feedback from the agents is most conducive to successful learning performance.

2 Related Work and Relevant Questions

2.1 Collaborative Problem Solving

In cognitive science, several studies on collaborative problem solving revealed how concepts are understood or learned. For example, researchers have shown that asking

reflective questions for clarification to conversational partners is an effective interactional strategy to gain a deeper understanding of a problem or a concept (e.g. [12, 3, 15, 13]). It has also been demonstrated that the use of strategic utterances such as asking for explanation or providing suggestions can stimulate reflective thinking and meta cognition involved in understanding a concept.

All these studies suggest that how well one can explain is the key to understanding and learning of a concept. Explanation may, however, be successful if people have difficulties in retrieving and associating relevant knowledge required for explanation activity. This has been reported to be the case especially among novice problem solvers [4, 10]. Also, it may not help learn a concept if people cannot communicate well each other as in when, for example, they use technical terms or phrases unknown to others [7].

One of the ways to help collaborative problem solvers is to introduce a third-person or a mentor who can facilitate the task by using prompts such as suggestions and back-channels. In actual pedagogical situation, however, it is often difficult for one teacher to monitor several groups of collaborators and to supervise their interaction during explanation. Recent studies by [8, 1] demonstrated that the use of conversational agents that act as educational companions or tutors can facilitate learning process. Yet, it has not been fully understood if and what kinds of support by such agents would be more helpful for collaborative problem solvers. In this article, the author will further investigate this question through the use of meta-cognitive suggestions, and affective expressions.

2.2 Pedagogical Conversational Agents as Learning Advisers

In the field of human computer interaction, researches have conducted a number of experimental studies which involve the use of pedagogical agents (e.g. [9, 5]). In the next section, the author will explain the factors that are important for pedagogical conversational agents as learning advisers.

The Effects of Monitoring and Presence of others. One of the important considerations in the study involving human performance is the effect of the "external factor" or the social influence from other people around. Studies in social psychology have suggested that work efficiency is improved when a person is being watched by someone, or, that the presence of an audience facilitates the performance of a task. This impact that an audience has on a task-performing participant is called the "audience effect". Another relevant concept on task efficiency, but from a slightly different perspective, is what is called "social facilitation theory". The theory claims that people tend to do better on a task when they are doing it in the presence of other people in a social situation; it implies that person factors can make people more aware of social evaluation. [16], who reviewed social facilitation studies concluded that the presence of others have positive motivational affects. [8] is one of the experimental studies which investigated the effects of a programmed agent. In this experiment, an agent, which played the role of an assistant, was brought in to help a participant who explained a concept. In the experiment, three different environments were set up for

the 'explaining activity'. They were: (1) two participants working with a text-based prompt, (2) two participants working with a visual image of pedagogical agent which produced a text-based prompt, (3) one participant working with a visual image of pedagogical agent which produced a text-based prompt (in this setup, participants did not have a human co-learner and directly interacted with the agent). The result showed that the participants in the last two conditions did better than the first where only textual prompts were presented. It also showed that the participants in the second condition did not engage in the explanation activity as much as those in the third. The first finding of [8] that the participants in the last two conditions, who worked with the agent, performed better may be attributed to the fact that their task of explanation was being watched or monitored by the agent.

These results suggest that participants would do better in the task of explanation if they are more conscious of the presence of the agents or if they are given an explicit direction to pay attention to the agent. This is our first research question investigated in Experiment 1 described below.

The Effects of Affective Feedback. Another point to be taken into consideration in studies of human performance is the "internal factor" or the affective factor, which is just as important as the "external factor" discussed above. They affect people's performance in either negative or positive ways and several studies reported that such factors are especially important in learning activities [1]. For example, [2] revealed that positive moods can increase memory performance. [11] also demonstrated that positive state of mind can improve text comprehension. Moods may affect the performance of human activities both verbally and non-verbally. In a study by [9], which examined how positive and negative comments from conversational agents affect learning performance, a pictorial image of an agent was programmed to project a textual message to the participant; in the positive condition, a visual avatar produced a short comment like "this task looks fun", while in the negative condition, it produced a short comment like "I don't feel like doing this, but we have to do it any-way". The results showed that the conversational agents that provided the participants with comments in a positive mood furnished them with a higher motivation of learning.

The studies discussed above suggest that the performance of explanation would also be enhanced if suggestions are given in positive mood either verbally or through visual feedbacks. This is our second research question investigated in Experiment 2 described below.

Research Goal and Hypothesis. The goal of this study is to experimentally investigate if and in what ways conversational agents can facilitate understanding and learning of concepts. The role of an agent was to assist the paired participants explain concepts to their partners during the collaborative peer-explanation activity. The hypotheses tested in this study were:

1. the presence of a conversational agent during collaborative learning through explanation task facilitates learners' understanding of a concept (Hypothesis 1 or H1)
2. the use of positive expressions provided by a conversational agent facilitates collaborative learners' understanding of concepts. (Hypothesis 2 or H2)

3 Method

3.1 Experimental Task and Procedure

The two experiments were conducted in a room where the computers were all connected by a local area network. In both experiments, the participants were given four technical terms printed on a sheet of paper. They were: 'schema', 'short-term / long-term memory', 'figure-ground reversal', and 'principle of linguistic relativity', which had been introduced in a psychology class. They were asked to describe the concepts of these words. After this pre-test, they logged in the computer and used the program installed in a USB flash drive (see the next section for detail). The pairs of participants were communicated through the chat program and one of the paired participants was instructed to explain to their partner the meanings of the words presented on their computer screen one by one. When two of the four concepts were explained to their partner, they switched the roles and the other partner explained the rest of the two words to his/her partner. All participants received the same prompts of suggestions from the agent on how explanations should be given and how questions should be asked about the concepts. After this intervention, they took the same test in the post-test. The descriptions of the concepts they provided in the post-test were compared with those of the pre-test to analyze if the participants gained a deeper understanding of the concepts after the collaborative activity. The whole process of the experiment took approximately 80 minutes.

3.2 Experimental System

In the experiments, a computer-mediated chat system was set up through computer terminals connected via a local network and the interactions of the participants during the activity were monitored. The system used in the experiments was programmed in Java (see Fig. 1). The system consists of three program modules of Server, Chat Clients, and Agent, all of which are simultaneously activated. The pedagogical agent used in this study is a simple rule-based production system typical of artificial intelligence (The agent system is developed by the author's previous study). It is capable of meaningfully responding to input sentences from users and consists of three main modules: Semantic Analyzer, Generator, and Motion Handler. Textual input of all conversational exchanges produced by paired participants is sent to the semantic analyzer of the conversation agent. The semantic analyzer then scans the text and detects keywords relevant to the concepts if they are being used in the explanation task (e.g. "I think that a *schema* is some kind of *knowledge* that is used based on one's own *experience*." (detected key words are shown in bold italic). Next, the extracted keywords are sent to the working memory in the generator and processed by the rule base, where various types of rule-based statements such as 'if X then Y' are stored to generate prompt messages (if there are several candidates of matching statements for the input keywords, a simple conflict-resolution strategy is utilized). When the matching process is completed, prompt messages are selected and sent back to the working memory in the generator. The messages generated by the rule base are also sent to the

motion handler module to activate an embodied conversation agent, a computer-generated virtual character which can produce human-like behaviors such as blinking and head-shaking (See next sections for details).

Several types of output messages are presented by the agent depending on the content of input text from the participants (see Table 1 below for examples). Only short back channels are sent when there are several related key words in a text (Type1 output); Messages of encouragement are given when the agent detects some keywords related to the target concept (Type 2 output, Type 3 output, Type 4 output).

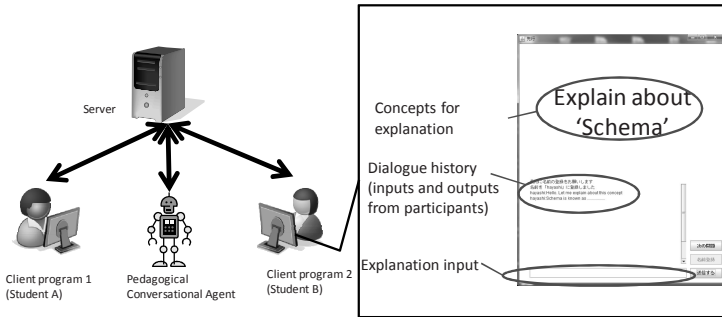


Fig. 1. Experimental environment and screenshot of the chat system

Table 1. Types of output messages from the agent

Type of output messages	Examples
Input messages (Detected key words are in Bold)	"I think that a schema is some kind of knowledge that is used based on one's own experience ."
Type1output: back channels	"That's the way", "Keep going! ", "Um-hum"
Type 2 output: Suggestion	"You used few important keywords. Try to explain from a different perspective."
Type 3 output: Suggestion(positive)	"Wow! You used a few very good keywords. That's great! It is better if you explain it from a different perspective!"
Type 4 output: Suggestion(negative)	"Well, you used few keywords. That is not enough. It is not satisfactory unless you explain it from a different perspective."

3.3 Participants and Conditions

In this study, a total of 173 participants participated in two experiments (114 participants for Experiment 1 and 59 participants for Experiment 2). The participants were all undergraduate students who were taking a psychology course and participated in them as part of the course work. They were randomly assigned to three conditions, which varied with respect to how prompts of suggestions were presented and how

conversational agents were used (see the sections below for details). In conditions of odd numbers, a group by three participants was composed.

Experiment 1. The purpose of Experiment 1 was to test H1: the presence of a conversational agent during explanation task facilitates understanding of concepts. This was investigated through three conditions (See Fig. 2). In the first condition (Group SST, $n = 37$), participants were provided with (just) text-based prompts which provided them with suggestions to facilitate the explanation task. In the second condition (Group SSA, $n = 38$), the participants were provided with text-based prompts through a chat-dialogue setup and also with a picture of a conversational agent shown on the display. Also, the participants were told that the agent will play the role of mentor; this direction was included to make them more conscious of being monitored by the agent. The third condition (Group SSA+, $n = 39$) was the same as the second condition except that the virtual character was an embodied conversational agent which uses its hand gestures while the participants chat on the computer. The figure was manipulated by the 2D-image/avatar-design tool (<http://avatarmaker.abi-station.com/>). The second and third conditions were used in order to find out the effects of pictorial presentation of an agent upon the explanation task. In both of these conditions, a pedagogical agent provided participants with back-channel feed-backs as they chat (see Table 1 for examples of backchannels).

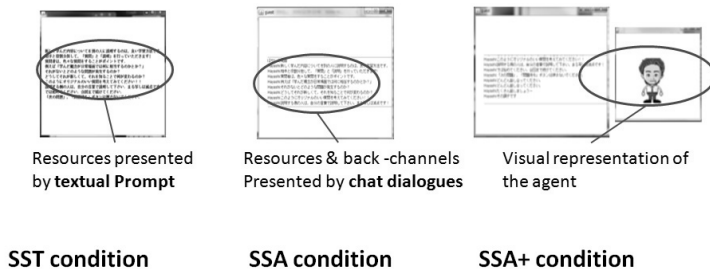


Fig. 2. Experimental conditions for Experiment 1

Experiment 2. Experiment 2 was conducted to test H2: the use of positive comments by conversational agent facilitates explanation activities and as a result, fosters understanding of concepts. To find out how affective factors influence the task of explanation, two types of avatars with more realistic appearance were created using a 3D-image/animation-design tool called Poser 8 (www.e-frontier.com): one is the "positive agent" with friendly facial expression and the other is the "negative agent" with unfriendly facial expression, which were used for the "positive condition" and the "negative condition" of the experiment, respectively. In the positive condition (Group SSA+P, $n = 31$), the participants were given positive suggestions, which were synchronized with the facial expressions of the positive agent. In the negative condition (Group SSA+N, $n = 28$), the participants were given negative suggestions, which were synchronized with the facial expressions of the negative agent (See Fig. 3). The messages were given through chat dialogue and the virtual character moved its hand

gestures while the participants chat on the computer (For examples of suggestion for the conversational agent see Table 1).

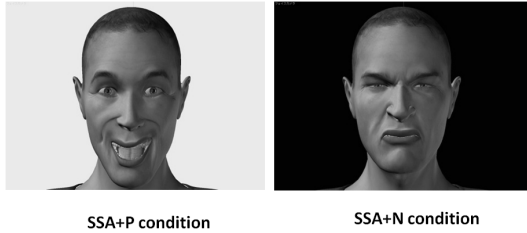


Fig. 3. Positive and negative facial expressions of the agent in Experiment 2

Dependant Variables. After Experiment 1 and Experiment 2, the participants who took the pre-test and post-test were asked to describe the concepts of the same technical words. The results of the pre- and post- tests were then compared to find out how the explanation task with different conditions facilitated their understanding or learning of the concepts. For the comparison, their descriptions were scored in the following way: 1 point for a wrong description or no description, 2 points for a nearly-correct description, 3 points for a fairly-correct description, 4 points for an excellent description, and 5 points for an excellent description with concrete examples. It was judged that the greater the difference in scores between the two tests the higher the degree of the effect of explanation.

4 Results

4.1 Experiment 1

The results of the Experiment 1 showed that the participants' understanding of the concepts (see Fig. 4 left). The vertical axis represents the average scores of the tests for the three groups at the times of pre- and post- tests. A statistical analysis was performed using a 2 x 3 mix factorial ANOVA with the two evaluation test-times (the pre-test vs. the post-test) and the three groups with different task conditions (SST vs. SSA vs. SSA+) as independent factors.

There was significant interaction between the two factors ($F(2,111) = 11.78, p < .01$). First, an analysis of the simple main effect was done on each level of the interface factor. In the SST, SSA, and SSA+ condition, the average scores in post-test was higher than pre-test respectively ($F(1,111) = 21.76, p < .01$; $F(1,111) = 119.59, p < .01$; $F(1,111) = 104.4, p < .01$). Next, an analysis of the simple main effect was done on each level of the period factor. In the pre-test, there no differences between conditions ($F(2,222) = 1.27, p = .28$). Although in the post-test there were differences between conditions ($F(2,222) = 20.27, p < .01$). Further analysis on the post-test was

conducted using the Ryan's method. Results indicate that the average score of SSA+ was higher than SST, and the average score of SSA was higher than SST respectively ($p < .01$; $p < .01$). There were no differences between SSA and SSA+ ($p = .35$). The overall results of Experiment 1 suggests that the collaborative activities facilitated the participants' understanding or learning of the concepts more when the presence of the third party, which gave suggestions for explanations, was made more explicit; in other words, the results show that H1 was supported.

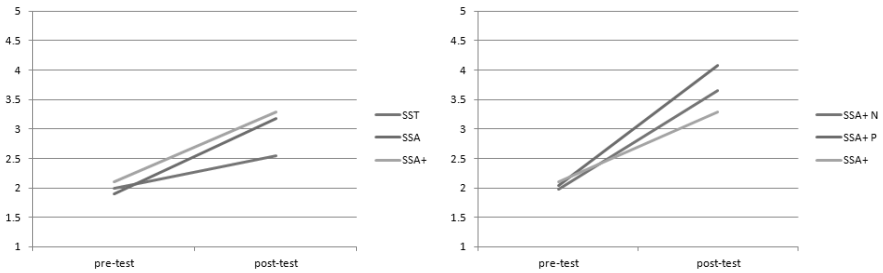


Fig. 4. Results of experiment 1(left) and experiment 2(right)

4.2 Experiment 2

The results of the Experiment 2 showed that the participants' understanding of the concepts (see Fig. 8 right). The vertical axis represents the average scores of the tests for the three groups at the times of pre- and post- tests. A statistical analysis was performed using a 2×3 mix factor ANOVA with the two evaluation test-times (the pre-test vs. the post-test) and the three groups with different affective conditions (SSA+N vs. SSA+P vs. SSA+) as independent factors. For the group with SSA+ condition, the same data used in Experiment 1 was used in Experiment 2.

There was significant interaction between the two factors ($F(2, 95) = 10.90$, $p < .01$). First, an analysis of the simple main effect was done on each level of the interface factor. In the SSA+N, SSA+P, and SSA+ condition, the average scores in post-test was higher than pre-test respectively ($F(1,95) = 172.86$, $p < .01$; $F(1,95) = 254.50$, $p < .01$; $F(1,95) = 87.85$, $p < .01$). Next, an analysis of the simple main effect was done on each level of the period factor. In the pre-test, there no differences between conditions ($F(2,190) = 0.48$, $p = .62$). Although in the post-test there were differences between conditions ($F(2,190) = 18.64$, $p < .01$). Further analysis on the post-test was conducted using the Ryan's method. Results indicate that the average score of SSA+P was higher than SSA+N and the average score of SSA+P was higher than SSA+, and the average score of SSA+N was higher than SSA+ respectively ($p < .01$; $p < .01$; $p < .01$). The overall results of Experiment 2 suggests that the collaborative activities facilitated the participants' understanding or learning of the concepts more when the positive suggestions were; in other words, the results show that H2 was supported.

5 Discussion

5.1 H1: Effects of the Presence of a Conversational Agent

The results of Experiment 1 suggested that the use of a conversational agent which provide relevant suggestions is more effective to facilitate explanation activities that result in a deeper understanding of concepts (i.e., Group SSA+ > Group SST, Group SSA > Group SST). The present experiment also provided some new evidence on the effectiveness of "audience effect", the effect of making people aware of the presence of a mentor, and the use of cognitive suggestions and back-channels, which was not investigated in similar studies in the past (e.g. [8]). One interesting finding in this experiment was that there was no difference between the group which was not provided with a visual representation of the agent (SSA+) and that which was provided with a visual representation (SSA). It may be that a mere mentioning of the instruction to the participants such as "the agent is your mentor and it's watching you", without showing the visual image of the agent, was sufficient enough to derive the "audience affect" [16]. On the contrary, it can also be predicted that the visual representation of the agent in the experiment did not have a discriminating effect upon the degree of attention as much it was expected to. This will be further discussed below.

5.2 H2: Effects of the Affective Expressions of the Conversational Agent

The results of Experiment 2 suggested that the greater the affective input from the conversational agent the more it can facilitate explanation activities which leads to a deeper understanding of concepts (i.e., Group SSA+P > Group SSA+N > Group SSA+). This experiment, examined the effects of affective expressions using both 'verbal message' and 'visual representation, which few others have looked into (e.g. [9]). As noted above, one very interesting finding was that Group SSA+N, to which suggestions and facial expressions of negative kind were given, scored higher than Group SSA+, to which suggestions and facial expressions of neutral kind were given, though not as high as Group SSA+P, to which suggestions and facial expressions of positive kind were given. This may suggest that the participants actually paid more attention and worked harder when they received negative comments than they received neutral comments. Some studies claim that negative comments presented through the media have a strong facilitation effects on memory [14]. The possibility that negative comments had a strong facilitating effect on this condition might be related to such effects. This point will be further investigated elsewhere.

6 Conclusion and Future Work

The present study investigated the effectiveness of the use of a conversational agent in a collaborative activity, where paired participants explained each other the meaning of technical terms taught in a psychology class for a better understanding. Conversational agents were used to encourage and facilitate the students' interaction through both

verbal and visual input. The experimental results suggested that the awareness of the presence of a conversational agent can trigger a deeper understanding of a concept during an explanation and that not only positive input but negative input from the conversational agent facilitate explanation activities and thus enhance learning performances. Pedagogical agent can play several different roles for collaborative learning activities and several studies have looked into the effectiveness of the use of a pedagogical agent with different roles. For example, [1] investigated the effectiveness of the use of a pedagogical agent which plays the roles of an expert teacher, a motivator, and a mentor (both an expert and motivator). However, not much is known yet about what roles it can play effectively. Another issue to be further investigated is the effect of the personality of the agent upon these roles. These and other related topics need to be further studied in future.

References

1. Baylor, A.L., Kim, Y.: Simulating instructional roles through pedagogical agents. *International Journal of Artificial Intelligence in Education* 15(1), 95–115 (2005)
2. Bower, G.H., Forgas, J.P.: Mood and social memory. In: Forgas, J.P. (ed.) *Handbook of Affect and Social Cognition*, pp. 95–120. LEA, NJ (2001)
3. Chi, M.T.H., Bassok, M., Lewis, M.W., Reimann, P., Glaser, R.: Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science* 13, 145–182 (1989)
4. Coleman, E.B.: Using explanatory knowledge during collaborative problem solving in science. *The Journal of Learning Sciences* 7(3&4), 387–427 (1998)
5. Graesser, A., McNamara, D.: Self-regulated learning in learning environments with pedagogical agents that interact in natural language. *Educational Psychologist* 45(4), 234–244 (2010)
6. Gulz, A., Haake, M.: Design of animated pedagogical agents – A look at their look. *International Journal of Human-Computer Studies* 64(4), 322–339 (2006)
7. Hayashi, Y., Miwa, K.: Prior experience and communication media in establishing common ground during collaboration. In: *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, pp. 528–531 (2009)
8. Holmes, J.: Designing agents to support learning by explaining. *Computers & Education* 48(4), 523–547 (2007)
9. Kim, Y., Baylor, A.L., Shen, E.: Pedagogical agents as learning companions: The impact of agent emotion and gender. *Journal of Computer Assisted Learning* 23(3), 220–234 (2007)
10. King, A.: Guiding knowledge construction in the classroom: Effects of teaching children how to question and how to explain. *American Educational Research Journal* 30, 338–368 (1994)
11. Mayer, D.K., Turner, J.C.: Discovering emotion in classroom motivation research. *Educational Psychologist* 37(2), 107–114 (2002)
12. Miyake, N.: Constructive interaction and the interactive process of understanding. *Cognitive Science* 10(2), 151–177 (1986)
13. Okada, T., Simon, H.: Collaborative discovery in a scientific domain. *Cognitive Science* 21(2), 109–146 (1997)

14. Reeves, B., Nass, C.: *Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places*. Cambridge University Press, New York (1996)
15. Salomon, G.: *Distributed cognition: Psychological and educational considerations*. Cambridge University Press, New York (2001)
16. Zajonc, R.B.: Social facilitation. *Science* 149, 271–274 (1965)

Exploration of Affect Detection Using Semantic Cues in Virtual Improvisation

Li Zhang

School of Computing, Engineering and Information Sciences
University of Northumbria, UK
li.zhang@northumbria.ac.uk

Abstract. Affect interpretation from multithreaded online conversations is a challenging task. Understanding context and identifying target audiences are very crucial for the appropriate interpretation of emotions implied in an individual input embedded in such online social interactions. In this paper, we discuss how context is used to interpret affect implied in conversational inputs with weak affect indicators embedded in multithreaded social interactions. Topic theme detection using latent semantic analysis is applied to such inputs to identify their discussion themes and potential target audiences. Relationships between characters are also taken into account for affect analysis. Such semantic interpretation of the dialogue context also shows great potential in the recognition of metaphorical phenomena and the development of a personalized intelligent tutor for drama improvisation.

Keywords: Affect and topic theme detection, and multithreaded interaction.

1 Introduction

It is inspiring and challenging to produce an intelligent agent who is capable of conducting drama performance, interpreting social relationships, context, general mood and emotion, reasonably sensing others' inter-conversion, identifying its role and participating intelligently in open-ended improvisational interaction. Online interaction with such an agent may also enable young people to engage in effective personalized learning. However, it is never an easy task even for human teachers to interpret learners' emotional expressions appropriately. Intelligent agents sometimes will need to incorporate information derived from multiple channels embedded in the interaction context to interpret the learners' emotions. The research conducted by Kappas [1] discussed several different emotions embedded in 'smile' facial expressions during social interaction and the importance of the understanding and employment of the related social context for the accurate interpretation of the implied affect in such expressions. Such cognitive study poses new challenges to computer scientists for intelligent agent development. The research presented in this paper has focused on the production of intelligent agents with the abilities of interpreting dialogue contexts semantically to support affect detection as the initial exploration.

Our research is conducted within a previously developed online multi-user role-play virtual framework, which allows school children aged 14 – 16 to perform drama improvisation. In this platform young people could interact online in a 3D virtual drama stage with others under the guidance of a human director. In one session, up to five virtual characters are controlled on a virtual stage by human users (“actors”), with characters’ (textual) “speeches” typed by the actors operating the characters. The actors are given a loose scenario around which to improvise, but are at liberty to be creative. An intelligent agent with an affect detection component is also involved in improvisation and detects affect from human characters’ each individual input. It was able to detect 15 emotions without taking any contexts into consideration.

Moreover, the previous processing was mainly based on pattern-matching rules that looked for simple grammatical patterns partially involving specific words [2]. It proved to be effective enough to detect affect from inputs containing strong clear emotional indicators such as ‘yes/no’, ‘thanks’ etc. There are also situations that users’ inputs contain very weak affect signals, thus contextual inference is needed to further derive the affect conveyed in such inputs. Moreover, inspection of the transcripts also indicates that the dialogues are often multi-threaded. This refers to the situation that social responses of different discussion themes to previous several speakers are mixed up. Therefore the detection of the most related discussion themes using semantic analysis is very crucial for the accurate interpretation of the emotions implied in those with ambiguous target audiences and weak affect indicators.

2 Related Work

There is much well-known research for the creation of affective virtual characters. Endrass, Rehm and André [3] carried out study on the culture-related differences in the domain of small talk behaviour. Their agents were equipped with the capabilities of generating culture specific dialogues. Recently textual affect sensing has also drawn researchers’ attention. Neviarouskaya et al. [4] provided textual affect sensing to recognize judgments, appreciation and different affective states. Although some linguistic contexts introduced by conjunctions were considered, the detection was still limited to the analysis of individual input. Ptaszynski et al. [5] employed context-sensitive affect detection with the integration of a web-mining technique to detect affect from users’ input and verify the contextual appropriateness of the detected emotions. However, their system targeted interaction only between an AI agent and one human user in non-role-playing situations. Comparing with the above work, our work focuses on the following aspects: (1) real-time affect sensing for basic and complex emotions in inputs with strong affect indicators; (2) the detection of the most related social contexts and target audiences using semantic interpretation for the processing of inputs with weak affect indicators; and (3) context-based affect detection with the consideration of relationships and the target audiences’ emotions.

3 Semantic Interpretation of Interaction Contexts

We noticed that the language used in the collected transcripts is often complex and invariably ungrammatical, and also contains a large number of weak cues to the affect that is being expressed. These cues may be contradictory or may work together to enable a stronger interpretation of the affective state. In order to build a reliable and robust analyser, it is necessary to undertake several diverse forms of analysis and to enable these to work together to build stronger interpretations. Thus in this work, we integrate contextual information to further derive affect embedded in contexts to provide affect detection for those without strong affect indicators.

Since human language is very diverse, terms, concepts and emotional expressions can be described in various ways. Especially if the inputs contain weak affect indicators, other approaches focusing on underlying semantic structures in the expressions should be considered. Thus latent semantic analysis (LSA) [6] is employed to calculate semantic similarities between sentences to derive discussion themes and potential target audiences for those inputs without strong affect signals.

Latent semantic analysis generally identifies relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. In order to compare the *meanings or concepts* behind the words, LSA maps both words and documents into a ‘concept’ space and performs comparison in this space. In detail, LSA assumes that there is some underlying latent semantic structure in the data which is partially obscured by the randomness of the word choice. This random choice of words also introduces noise into the word-concept relationship. LSA aims to find the smallest set of concepts that spans all the documents. It uses a statistical technique, called singular value decomposition, to estimate the hidden concept space and to remove the noise. This concept space associates syntactically different but semantically similar terms and documents. We use these transformed terms and documents in the concept space for retrieval rather than the original terms and documents.

In our work, we employ the semantic vectors package [7] to perform LSA, analyze underlying relationships between documents and calculate their similarities. This package provides APIs for concept space creation. It applies concept mapping algorithms to term-document matrices using Apache Lucene, a high-performance, full-featured text search engine library implemented in Java [7]. We integrate this package with the affect detection component to calculate the semantic similarities between test inputs and training documents. In this paper, we target the transcripts of the school bullying¹ and Crohn’s disease² scenarios for context-based affect analysis.

In order to compare user inputs with documents belonging to different topic categories, we have to collect some sample documents with strong topic themes from the Experience project (www.experienceproject.com). These articles belong to 12

¹ The bully, Mayid, is picking on a new schoolmate, Lisa. Elise and Dave (Lisa’s friends), and Mrs Parton (the school teacher) are trying to stop the bullying.

² Peter has Crohn’s disease and has the option to undergo a life-changing but dangerous surgery. He needs to discuss the pros and cons with friends and family.

discussion categories including Education, Family & Friends, Health & Wellness etc. Since we intend to perform topic theme detection for the transcripts of the bullying and Crohn's disease scenarios, we extracted sample articles close enough to these scenarios including articles of Crohn's disease (five), school bullying (five), family care for children (five), food choice (three), school life including school uniform (10) and school lunch (10). Phrase and sentence level expressions implying 'disagreement' and 'suggestion' are also gathered from the several other articles published on the website. Thus we have training documents with eight themes including 'Crohn's disease', 'bullying', 'family care', 'food choice', 'school lunch', 'school uniform', 'suggestions' and 'disagreement'. Affect detection from metaphors often poses great challenges. In order to detect a few metaphorical phenomena, we include five types of metaphorical examples published on the following website: <http://knowgramming.com>. These include cooking, family, weather, farm and mental metaphors. Individual files are used to store each type of metaphors. All the sample files of these 13 categories are put under one directory for further analysis. The following interaction of the bullying scenario is used to demonstrate how we detect the discussion themes for those inputs with weak affect indicators.

1. Lisa: *can I go to the loo miss* [neutral]
2. Mayid: and *wot is with the outfit, geeky or wot!!* [angry]
3. Mayid: *y u wna cry sum more in da toilet!* [angry]
4. Dave: Oh, dear, please be more brave. Help will be here soon. [*Played by the AI agent*]
5. Elise: *shut ya face just cuz u buy urs at de jumble sale* [angry]
6. Mayid: *hahahaha* [happy]
7. Mayid: and u buy urs at the rag market! [*Target audience: Elise; angry*]
8. Elise: *whatever u piece of dirt.* [angry]
9. Elise: Lisa *how r u?* [neutral]
10. Mayid: *piece of dirt, yeh and im proud!!! u piece of s**** [angry]
11. Mayid: Lisa is fine. Nothing is wrong with her. [*Target audience: Elise and Lisa: angry*]
12. Dave: are these all desperate people? [*Played by the AI agent*]
13. Mayid: *ur da desperate one dave!!!* [angry]
14. Dave: Do I have anything to do with it? [*Played by the AI agent*]
15. Mayid: no u dnt! So *get frikin lost!* [angry]

Affect implied by the inputs with strong affect indicators (illustrated in italics) in the above interaction is detected by the previous processing. Dave was played by the AI agent. The inputs without an affect label followed straightaway are those with weak affect indicators (7th & 11th inputs). Therefore further processing is needed to recover their most related discussion themes and identify their most likely audiences in order to identify implied emotions more accurately. Our general idea for the detection of discussion themes is to use LSA to calculate semantic distances between each test input and all the training files with clear topic themes. Semantic distances between the test input and the 13 topic terms (such as 'disease', 'bullying') are also calculated. The detected topics are derived from the integration of these semantic similarity outputs. We start with the 7th input to demonstrate the theme detection.

First of all, in order to produce a concept space, the corresponding semantic vector APIs are used to create a Lucene index for all the training samples and the test file ('test_corpus1.txt' contains the 7th input). This generated index is also used to create term and document vectors, i.e. the concept space. Various search options could be used to test the generated concept model. In order to find the most effective approach to extract the topic themes, we provide rankings for all the training files and the test input based on their semantic distances to a topic theme as the first step. We achieve this by searching for document vectors closest to the vector for a specific term (e.g. 'bullying'). The 7th input obtains the highest ranking for the topic theme, 'clothes', among all the rankings for the eight non-metaphorical topics. But there are multiple ways to describe a topic theme (e.g. 'disagreement'). It affects the file ranking results more or less if different terms indicating the same themes are used. Thus we need to use other more effective search methods to accompany the above findings.

Another effective approach is to find the semantic similarity between documents. All the training documents contain clear discussion themes indicated by their file names. If the semantic distances between training files and the test file are calculated, then it provides another source of information for topic theme detection. Therefore we use the CompareTerms semantic vector API to find out semantic similarities between all the training corpus and the test document. We provide the top five rankings for semantic similarities between the training documents and the 7th input in Figure 1.

```
Similarity of "disagree1.txt" with "test_corpus1.txt": 0.3753488428354625
Similarity of "bullied3.txt" with "test_corpus1.txt": 0.24211021149610681
Similarity of "food1.txt" with "test_corpus1.txt": 0.22516070145013847
Similarity of "school_uniform.txt" with "test_corpus1.txt": 0.20377660285294324
Similarity of "farm_metaphor.txt" with "test_corpus1.txt": 0.18034050861738923
```

Fig. 1. Part of the output for semantic similarities between training documents and the test file

The outputs indicate that the 7th input is more closely related to 'disagreement (disagree1.txt)' and 'bullying (bullied3.txt)' topics although it is also semantically close to school uniform. In order to identify its target audiences, we have to conduct topic theme detection since the 6th input until we find the input with similar topics. The previous pre-processing identified the 6th input implied 'laughter', unrelated to both of the above themes. Thus we focus on the 5th input from Elise. It is identified to show the same two themes as those for the 7th input. Thus the audience of the 7th input is Elise, who implied 'anger' in the 5th input with strong affect indicators.

The research of Wang et al. [8] also discussed feedback of artificial listeners can be influenced by relationships and personalities. In our application, relationships are thus employed to advise affect detection in social contexts. In this example, since Elise and Mayid have a negative relationship and Elise showed 'anger' in the most recent 'bullying' input, Mayid is most likely to indicate resentful 'anger' in the 7th input also with a 'bullying' theme. Thus the 7th input implies an 'angry' emotion. Similarly the 11th input is detected most closely related to the topics of 'bullying' and 'family care' with Lisa and Elise (9th) as identified audiences. Since Mayid has a negative

relationship with both of them, he is more likely to indicate ‘bullying’. Thus the 11th input implies an ‘angry’ emotion.

Therefore, appraisal rules are generated to reflect the above description and reasoning to derive affect in social contexts for those inputs without strong affect indicators. The rules accept the target audiences’ emotions and relationships between the audiences and the speaker for affect interpretation. Moreover, the semantic-based processing goes beyond pattern matching and evaluation results indicated that it can be well applied to real conversation contexts of bullying and disease.

4 Evaluation and Conclusion

We have taken previously collected transcripts recorded during our user testing to evaluate the efficiency of the updated affect detection component with contextual inference. In order to evaluate the performances of the topic theme detection and the rule based affect detection in social contexts, three transcripts of the Crohn’s disease scenario are used. Two human judges are employed to annotate the topic themes of the extracted 300 inputs from these test transcripts using these 13 topic categories. Cohen’s Kappa was used to measure the inter-annotator agreement between human judges and the result was 0.83. Then the 265 example inputs with agreed theme annotations are used as the gold standard to test the performance of the topic theme detection. A keyword pattern matching baseline system was used to compare the performance with that of the LSA. We have obtained an averaged precision, 0.736, and an averaged recall, 0.733, using the LSA while the baseline system achieved an averaged precision of 0.603 and an averaged recall of 0.583 for the 13 topic detection. The detailed results indicated that discussion themes of ‘bullying’, ‘disease’ and ‘food choices’ were very well detected by our semantic-based analysis. The discussions on ‘family care’ and ‘suggestion’ topics posed most of the challenges. Generally the semantic-based interpretation achieves reasonable and promising results. The human judges have also annotated these 265 inputs with the 15 frequently used emotions. The inter-agreement between human judge A/B is 0.63. While the previous version achieves 0.46 in good cases, the new version achieves 0.56 and 0.58 respectively. Inspection of the annotated test transcripts by the new version of the AI agent indicates that many expressions regarded as ‘neutral’ previously were annotated appropriately as emotional expressions. 50 articles from the Experience website were also used to evaluate the semantic-based topic detection. The processing achieved a 66% accuracy rate in comparatively unfamiliar contexts.

Moreover, in future work, we intend to extend the emotion modeling with the consideration of personality and culture. We are also interested in topic extraction to support affect interpretation, e.g. the suggestion of a topic change indicating potential indifferent to the current discussion theme. It will also ease the interaction and make human characters comfortable if our agent is equipped with culturally related small talk behavior. We believe these are crucial aspects for the development of effective personalized intelligent pedagogical agents.

References

1. Kappas, A.: Smile when you read this, whether you like it or not: Conceptual challenges to affect detection. *IEEE Transactions on Affective Computing* 1(1), 38–41 (2010)
2. Zhang, L.: Exploitation on Contextual Affect Sensing and Dynamic Relationship Interpretation. *ACM Computers in Entertainment* 8(3) (2010)
3. Endrass, B., Rehm, M., André, E.: Planning Small Talk Behavior with Cultural Influences for Multiagent Systems. *Computer Speech and Language* 25(2), 158–174 (2011)
4. Neviarouskaya, A., Prendinger, H., Ishizuka, M.: Recognition of Affect, Judgment, and Appreciation in Text. In: *Proceedings of the 23rd International Conference on Computational Linguistics*, Beijing, China, pp. 806–814 (2010)
5. Ptaszynski, M., Dybala, P., Shi, W., Rzepka, R., Araki, K.: Towards Context Aware Emotional Intelligence in Machines: Computing Contextual Appropriateness of Affective States. In: *Proceeding of IJCAI* (2009)
6. Landauer, T.K., Dumais, S.: Latent semantic analysis. *Scholarpedia* 3(11), 4356 (2008)
7. Widdows, D., Cohen, T.: The Semantic Vectors Package: New Algorithms and Public Tools for Distributional Semantics. In: *IEEE Int. Conference on Semantic Computing* (2010)
8. Wang, Z., Lee, J., Marsella, S.: Towards More Comprehensive Listening Behavior: Beyond the Bobble Head. In: Vilhjálmsón, H.H., Kopp, S., Marsella, S., Thórisson, K.R. (eds.) *IWA 2011*. LNCS, vol. 6895, pp. 216–227. Springer, Heidelberg (2011)

Measuring Learners' Co-Occurring Emotional Responses during Their Interaction with a Pedagogical Agent in MetaTutor

Jason M. Harley, François Bouchet, and Roger Azevedo

McGill University, Dept. of Educational and Counselling Psychology, Montréal, Canada
jason.harley@mail.mcgill.ca

Abstract. This paper extends upon traditional emotional measurement frameworks used by ITSs in which emotions are analyzed as single, discrete psychological experiences by examining co-occurring emotions (COEs) (e.g., Conati) through a novel methodological approach. In this paper we examined the occurrence of students' embodiment of basic single discrete emotions (SDEs) and COEs (in addition to neutral) using an automatic facial expression recognition program, FaceReader 4.0. This analysis focuses on the sub goal setting task of learners' ($N = 50$) interaction with MetaTutor, during which a pedagogical agent assisted students to set three relevant sub goals for their learning session. Results indicated that neutral and sadness were the SDEs experienced most by students and also the most represented emotions in COE pairs. COEs represented nearly a quarter of students' embodied emotions.

Keywords: Emotions, affect, intelligent tutoring systems, pedagogical agents, co-occurring emotions, learning, human-computer interaction, co-adaptation.

1 Co-Occurring Emotions during Learning with ITSs

Effective learning and students' experience of emotions are deeply intertwined in a variety of learning contexts [1-3]. Researchers' shared understanding of this educational tenet and its application to designing computer-based learning environments has had important implications for the development of ITSs, specifically, the development of ITSs that are able to detect, model, and adapt to changes in learners' emotional fluctuations. This paper extends upon this work by measuring learners' experience of co-occurring emotions (COEs). COEs are emotional states that occur simultaneously, where their discrete characteristics (e.g., valence, intensity) are maintained, but they are experienced in tangent with other emotional states (e.g., happiness and surprise). It is crucial that we are able to detect, measure and adapt to students' COEs during their interactions with ITSs because there are meaningful differences between a student's experience of a single discrete emotion (SDE) (e.g., anger) in comparison to the same student's experience of a pair of SDEs (e.g., anger and surprise).

In our review of the literature we found only one ITS system which considered co-occurring emotions [4], as opposed to only considering and measuring emotions as

discrete, non-overlapping states (i.e., SDEs) [1,3,5]. A review of theories of emotions revealed only two references to COEs; neither discussed COEs as a major theoretical component [6-7]. These examples suggest that COEs have both a theoretical and methodological basis for existing and being measured and that their absence in ITS literature and other emotions literature is a shortcoming, also stated by [4].

The purpose of this paper is to examine the occurrence of COEs using a novel trace data methodology, in which learners' emotions are measured with an automatic facial recognition program, FaceReader [8]. In this paper learners' emotions were measured while they interacted with a pedagogical agent (PA) during the sub goal setting task of their interaction with MetaTutor [9]. Our research questions included: (1) what proportion of all emotions that learners' embodied, during the sub goal setting task, are COEs vs. SDEs? and (2) which pairs of COEs are most prominent?

2 Methods

2.1 Participants

50 undergraduate students from two large, public universities in North America participated in this study. Participants (74% female, 68% Caucasian) were randomly assigned to either a control condition or a prompt and feedback condition.

2.2 MetaTutor and Apparatus

MetaTutor is a multi-agent ITS and hypermedia-learning environment which consists of 41 pages of text and static diagrams about the human circulatory system [9]. The sub goal setting task, part of the sub goal setting phase of learners' interaction with MetaTutor, is the focus of our study and ranged between 1m09s and 6m03s ($M = 2m22s$, $SD = 1m10s$). This difference in time is due to participants' varying abilities to set three sub goals for learning as much as they could about the circulatory system at an appropriate level of detail, as well as the PA's scaffolding strategy.

A Microsoft LifeCam™ webcam was used to record participants' faces during their interaction with MetaTutor. The camera was mounted above the monitor and videos were recorded as WMV files, with a frame rate varying from 20 to 60 frames per second. In order to classify the embodiment of learners' emotions, we used Noldus FaceReader™ 4.0, a software program that analyzes participants' facial expressions and provides a classification of their emotional states using: (1) an Active Appearance Model to model their faces and (2) an artificial neural network with seven outputs corresponding to Ekman and Friesen's 6 basic emotions [10] in addition to neutral. Imported face videos were analyzed using FaceReader's pre-calibration and general model settings. FaceReader has been validated through comparison with human coders' ratings of basic emotions and specified acted emotions [11- 12].

2.3 Data Analysis

FaceReader provides a score between 0 and 1, for each frame of each participant's video for each of Ekman's six basic emotions, in addition to neutral. FaceReader also provides information about the dominant emotional state (computed with a proprietary algorithm using the scores of the seven emotional states in the previous frames) and timestamp information regarding the on and offset of the hierarchical rankings of these states. In order to be able to compare the results obtained to FaceReader's default proprietary algorithm, we replicated it as closely as possible in order to evaluate (for every frame) not only the primary emotional state, but also the secondary one (when it existed), using the following steps:

- First, we calculated a list of emotions, whose scores were above a minimal threshold value of 0.01 for more than 0.5s, while not disappearing completely (either because no face could be found in the frame or because their score was below 0.01) for more than 1s. The score associated with each selected emotion was either the one given by FaceReader for that frame, if available (i.e., if a face had been found in the frame), or the previous frame's score for that emotion.
- To order the emotions of the previous list, and to avoid a sequence of quick alternations from one frame to another between two emotions with very close scores, we calculated the primary (resp. secondary) emotional state as the one having the highest (resp. second highest) mean score over the past 0.5s.
- If the score of the secondary emotional state deviated no more than 0.15 from the score of the primary emotional state, we identified the emotional state of the considered frame as being a co-occurring emotional state.

Using this method, for the sample of 50 participants considered, we obtained a 91% level of agreement between the primary emotional state calculated by FaceReader and the one we calculated (97% if we also considered the value of the secondary state). In order to aggregate the data from participants, since each of the 50 videos had been recorded with a different frame rate, we normalized the sum of each emotion or pair of emotions using the frame rate value for the video. We also normalized the sum of each emotion or pair of emotions displayed in Table 1 (hence all participants have the same weight, regardless of the time spent to set sub goals). In total this analysis examined 224,582 judgments of emotional states made by FaceReader across participants.

3 Results

3.1 What Proportion of All Emotions that Learners' Embodied during the Sub Goal Setting Task Are COEs vs. SDEs?

When looking at all the possible embodiments of emotions, both SDEs and all possible pairs of COEs (see Table 1), we see that the discrete state of neutral was the emotional state with the greatest proportion (30.77%), followed by the discrete states of sadness (18.25%), happiness (10.73%) and disgust (9.33%). These four SDEs made up 69.08% of all the possible embodiments of emotions, which increased to approximately 77% of the emotions when the SDEs scared (2.00%), anger (3.22%) and surprise (2.77%) are included. The remaining 23% are different combinations of COEs.

3.2 Which Pairs of COEs Are Most Prominent?

Summing each of the different basic COEs in addition to neutral revealed that 12.45% of emotional states involved the emotion neutral co-occurring with other emotional states, 12.64% involved sadness, 7.19% involved disgust, 5.74% involved happiness, 4.34% involved anger, 2.52% involved surprise, and 1.00% involved scared. These proportions exceed 23% because of the overlapping nature of co-occurring emotions. By looking at column 5 of Table 1, we can see that the co-occurring emotional pairs which learners experienced most often included: neutral and sad (4.77%), sad and disgusted (2.99%), happy and sad (2.40%), and neutral and disgusted (2.39%). These emotional states had a greater proportion of co-occurrence than several of the single, discrete emotional states, including scared and surprised.

Table 1. Proportions of Learners' SDE and COEs during the Sub Goal Setting Task

Emotion		Co-occurrence of emotions (in %)				Number of subjects embodying			
A	B	A&B	B&A	A&B or B&A	Difference A&B vs. B&A	A&B	B&A	A&B or B&A	A&B and B&A
Neutral	-	30.77	-	30.77	-	49	-	49	-
Happy	-	10.73	-	10.73	-	41	-	41	-
Sad	-	18.25	-	18.25	-	48	-	48	-
Angry	-	3.22	-	3.22	-	33	-	33	-
Surprised	-	2.77	-	2.77	-	24	-	24	-
Scared	-	1.99	-	1.99	-	14	-	14	-
Disgusted	-	9.33	-	9.33	-	39	-	39	-
Neutral	Happy	0.89	0.91	1.80	-0.02	32	29	34	27
Neutral	Sad	2.27	2.50	4.77	-0.24	43	46	46	43
Neutral	Angry	1.00	0.64	1.64	0.35	30	24	32	22
Neutral	Surprised	0.93	0.54	1.46	0.39	19	15	21	13
Neutral	Scared	0.21	0.18	0.39	0.03	13	8	14	7
Neutral	Disgusted	1.25	1.13	2.39	0.12	31	26	32	25
Happy	Sad	1.38	1.02	2.40	0.37	29	27	34	22
Happy	Angry	0.10	0.07	0.17	0.04	12	11	14	9
Happy	Surprised	0.11	0.12	0.23	-0.01	9	7	11	5
Happy	Scared	0.09	0.11	0.21	-0.02	8	7	12	3
Happy	Disgusted	0.45	0.48	0.93	-0.03	20	22	24	18
Sad	Angry	1.13	0.72	1.85	0.41	25	19	28	16
Sad	Surprised	0.27	0.12	0.39	0.16	14	11	15	10
Sad	Scared	0.14	0.11	0.25	0.03	10	8	11	7
Sad	Disgusted	1.47	1.51	2.99	-0.04	30	32	35	27
Angry	Surprised	0.02	0.07	0.09	-0.06	4	5	5	4
Angry	Scared	0.02	0.02	0.04	0.00	4	3	4	3
Angry	Disgusted	0.29	0.27	0.56	0.01	14	15	17	12
Surprised	Scared	0.05	0.02	0.07	0.02	5	4	6	3
Surprised	Disgusted	0.13	0.16	0.28	-0.03	10	9	11	8
Scared	Disgusted	0.02	0.02	0.04	-0.01	4	3	4	3

Note: The seven SDEs in lines 3 to 9 of column 1 are ordered arbitrarily. All subsequent emotions in columns 1 and 2 follow the same repeating order as the first seven until all possible pairs of emotions (i.e., COEs) have been exhausted. Columns 3 and 4 represent the proportions for which the emotions in columns 1 and 2 were the dominant emotion when paired together. Column 5 represents the proportions of co-occurring emotions pairs (sum of column 3 and 4).

4 Discussion, Conclusions and Future Directions

Our results provide us with the means to draw several interesting tentative conclusions about an important component of the psychological process of emotions that we know little about. First, that COEs, while not representing a majority of the emotional states experienced, do represent a sizeable portion, which reinforces the need to study and understand them. Second, we see that learners' proportional experience of COEs are similar to their experience of SDEs (i.e., sadness and neutral are common components in the most common pairings). Third, this paper highlights the prominence of learners' experience of neutral and sadness during the sub goal setting task of their learning session with MetaTutor. It is possible that learners experienced sadness in response to their proposed sub goals being rejected by the PA, especially since the great majority of learners failed to set their sub goals independently. In noting the prominence of learners' embodiment of neutral, it is important to remember that it is a commonly over-looked emotional state by researchers who measure emotions [1-3,10]. In this analysis, we operationalized neutral as a psychological state in which participants are not experiencing one of the six basic emotions or a positive or negative valence. The purpose of investigating learners' experiences of a neutral state is to measure their baseline state, which allows one to measure fluctuations in emotions. Neutral has a particularly important role to play in examining learners' emotional responses in ITSs as it is not necessarily realistic to expect the average undergraduate student to be in a positively-valenced emotional state (e.g., happiness, engagement) throughout the session. In these cases, neutral may be a signal that learners are in an emotional state where they are not emotionally distracted and can therefore learn (an important bottom line).

This paper represents our first exploration of a complex, but important addition to the psychological process of emotions and how it applies to MetaTutor and may apply to other ITSs and contexts. Future directions include using multiple channels to measure SDEs and COEs, including self-reports and physiological sensors, in order to cross-validate our findings. This is an important next step because our current method for detecting co-occurring emotions is data-driven and relies only on one channel, which excludes learner-centered emotions (e.g., curiosity and boredom). We are also interested in looking, not only at the alignment of SDEs and COEs with events, but at the fluctuations between various SDEs and COEs. This is an especially important direction because it will help further our understanding regarding the nature of co-occurring emotions as complex psychological processes.

Acknowledgements. The research presented in this paper has been supported by a doctoral fellowship from the Fonds Québécois de recherche - Société et culture (FQRSC) awarded to the first author and funding from the Social Sciences and Humanities Research Council of Canada (413-2011-0170) and the National Science Foundation (DRL 0633918 and IIS 1008282) awarded to the third author. We would like to thank Reza Feyzi-Behnagh, Melissa Duffy, Gregory Trevors, Melissa Stern, Amy Johnson, Amber Chauncey, Candice Burkett, Ashley Fike, Ronald Landis and Jonathan Burlison for their help with data collection and to acknowledge feedback from Robert Bracewell.

References

1. McQuiggan, S.W., Lester, J.C.: Modelling affect expression and recognition in an interactive learning environment. *Inter. Journal of Learning Technology* 4, 216–233 (2009)
2. Pekrun, R.: Emotions as Drivers of Learning and Cognitive Development. In: Calvo, R.A., D'Mello, S.K. (eds.) *New Perspectives on Affect and Learning Technologies*, pp. 23–39. Springer, New York (2011)
3. Woolf, B., Bursleson, W., Arroyo, I., Dragon, T., Cooper, D., Picard, R.: Affect-aware tutors: recognizing and responding to student affect. *International Journal of Learning Technology* 4, 129–164 (2009)
4. Conati, C., Maclaren, H.: Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction* 19, 267–303 (2009)
5. D'Mello, S.K., Craig, S.D., Graesser, A.C.: Multimethod assessment of affective experience and expression during deep learning. *Inter. Jour. of Learn. Tech.* 4, 165–187 (2009)
6. Pekrun, R.: The Control-Value Theory of Achievement Emotions: Assumptions, Corollaries, and Implications for Educational Research and Practice. *Educational Psychology Review* 18, 315–341 (2006)
7. Ekman, P., Friesen, W.V.: Constants across cultures in the face and emotion. *J. Pers. Soc. Psychol.* 17, 124–129 (1971)
8. VicarVision: FaceReader 4.0 [Computer software]. Noldus Information Technology, Wageningen, The Netherlands (2011)
9. Azevedo, R., Behnagh, R., Duffy, M., Harley, J.M., Trevors, G.J.: Metacognition and self-regulated learning in student-centered leaning environments. In: Jonassen, D., Land, S. (eds.) *Theoretical Foundations of Student-Center Learning Environments*, pp. 216–260. Erlbaum, Mahwah (2012)
10. Ekman, P.: An argument for basic emotions. *Cognition & Emotion* 6, 169–200 (1992)
11. van Kuilenburg, H., Wiering, M., den Uyl, M.: A Model Based Method for Automatic Facial Expression Recognition. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) *ECML 2005. LNCS (LNAI)*, vol. 3720, pp. 194–205. Springer, Heidelberg (2005)
12. Terzis, V., Moridis, C.N., Economides, A.A.: Measuring instant emotions during a self-assessment test: the use of FaceReader. In: *Proceedings of the 7th Inter. Conf. on Methods and Techniques in Behavioral Research*, pp. 18:1–18:4. ACM, New York (2010)

Visualization of Student Activity Patterns within Intelligent Tutoring Systems

David Hilton Shanabrook¹, Ivon Arroyo¹,
Beverly Park Woolf¹, and Winslow Burleson²

¹ Department of Computer Science, University of Massachusetts Amherst

² School of Computer Science and Informatics, Arizona State University

Abstract. Novel and simplified methods for determining low-level states of student behavior and predicting affective states enable tutors to better respond to students. The Many Eyes Word Tree graphics is used to understand and analyze sequential patterns of student states, categorizing raw quantitative indicators into a limited number of discrete states. Used in combination with sensor predictors, we demonstrate that a combination of features, automatic pattern discovery and feature selection algorithms can predict and trace higher-level states (emotion) and inform more effective real-time tutor interventions.

Keywords: user modeling, pattern discovery, student emotion, engagement.

1 Introduction

Tutoring systems have demonstrated effective learning over large amounts of students in classrooms in public schools [1][8][2], and some studies have shown evidence that the adaptive nature of tutoring systems is responsible for higher learning rates [3]. However, even the most effective tutoring system will fail if the students behavior is not receptive to the material being presented. Although individualized learning provided by tutoring systems has been beneficial overall, its effectiveness might be increased if maladaptive student behaviors could be identified and modeled.

Recent research has utilized dynamic assessment of a students performance to enhance the effectiveness of their tutor sessions [3]. Many research groups use physiological sensors and tutor metrics to predict emotions [5][6]. While often predictive, sensors are hard to deploy in real-life situations; they often require non-standard hardware and modeling is contingent on labeling. Labeling refers to the correlation of physiological metrics, to emotional states, (e.g., by self-reporting, observation, etc.) which introduces error. Using tutor log data alone, (e.g., incorrect attempts, etc.) can avoid these issues.

This current work refines previous work in inferring and predicting student behavioral state based on tutor data. The process is a variation and an extension of time-based motif discovery [4] in student behaviors, now used for the prediction of emotional states. Prior research has used data mining to discover

patterns in the problem states that defined student behavior [5] [6]. One of those studies began the process of categorizing raw problem metrics regarding time on task, accuracy and help received into more meaningful categories, or states [1] [9]. Motif discovery was used to find engagement patterns in windows of 10 student-problem interactions. These patterns could then be used to define new student behavior states. Limitations of this work were the difficulty of defining meaning to the patterns; redundancies among the states and the lack of clear meaning of some of the binning categories. Attempts to view the data visually were also difficult due to the large number of states.

Guided by the findings reported from the literature, discussed above, our method examines student interaction with the tutor during problem solving. However, rather than looking at short-term behaviors over the lapse of one problem and relating it with higher level latent states or outcomes (e.g., emotions, mastery), we examine frequent behavioral patterns over several problems, and their predictive power of higher level affective states. The next sections describe this methodology.

2 The Tutor and the Student Data

The data comes from students working with Wayang, an adaptive tutoring system that helps students learn to solve standardized-test questions, in particular state-based exams taken at the end of high school in the USA. This multimedia tutoring system teaches students how to solve geometry, statistics and algebra problems. To answer problems students choose a solution from a list of multiple-choice options. Students are provided immediate feedback when they click on an answer. Students may click on a help button for hints, and hints are displayed in a progression from general suggestions to bottom-out solution.

An empirical evaluation was conducted involving 295 high school students from classes in public high schools, Spring 2009. Students used Wayang for a week during one-hour periods instead of their regular math class. Students progressed through various topics such as expressions with variables, perimeter, triangles, equations. Every 5 minutes, and at the end of a math problem, students were asked how they were feeling, which they reported in a scale of 1 (low) to 5 (high). As the students worked the tutor logged problem metrics such as `timeToFirstAttempt`. During some testing, hardware sensors (mental state camera, skinconductance bracelet, pressure sensitive mouse, and posture sensitive chair) collected realtime physiological student data.

3 Methodology

During the **Data Pre-processing Stage**, the continuous problem metrics are binned into discrete states. Our original approach to binning was to simply convert the continuous metrics into discrete (3-5 bins) states, in a logical manner when appropriate. For instance, `timeToFirstAttempt`, a positive skewed metric, was binned into less than 4 seconds, insufficient time to read the problem [7],

Table 1. Student States

<i>State</i>	<i>Description</i>	<i>Possible Intervention</i>
NOTR	Not reading problem before first attempt	Decrease problem difficulty, invoke help, read problem aloud.
SOF	Solved first attempt without help	Increase problem difficulty.
BOTT	Getting answer from help	Decrease difficulty, gaming intervention
GIVEUP	Stopped before answering	Decrease difficulty, gaming intervention
ATT	Solved after 1 or 2 attempts without help	On task behavior, show full problem solution after correct answer is entered.
SHINT	Solved with help	On task behavior.
GUESS	Guessing	Help intervention.

and two other bins, low and high. The other raw metrics were similarly binned resulting four state descriptors per problem each with three to five bins.

Visualization lead to the realization the four descriptors contained redundant information and a simpler approach would yield as accurate problem state. With timeToFirstAttempt, the binned less than 4 seconds was named the not read (NOTR) state. This state was prioritized over all other metrics, if the student is not reading the problem before attempting to answer other metrics were not relevant.

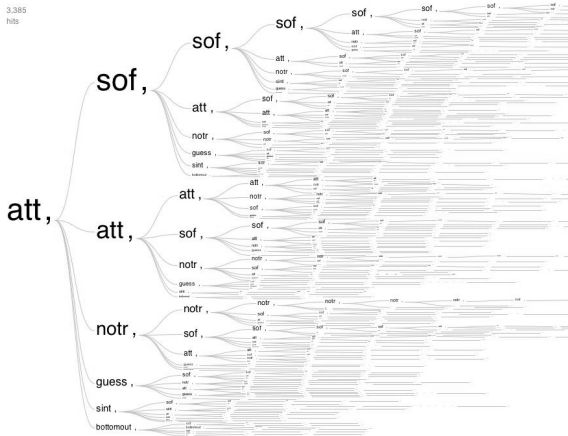


Fig. 1. Problem state patterns

This greatly simplified our problem state; the original seven binned categories create 135 possible combination states. With this prioritization we now have only six states: NOTR (not reading problem), SOF (solved on first attempt), BOTT (bottom out), GIVEUP, ATT (valid attempts) or SHINT (solved with hints). SOF categorizes all problems that are solved on first attempt without invoking

help, indicating problem level should be increased. BOTT implies gaming with intervention of disallowing bottom out hint. GIVEUP indicates user quit problem and problem difficulty would be decreased. ATT indicates a student working on-task at an appropriate level; support provided to ensure continued success. SHINT similar to ATT and invoking help but no support needed. GUESS indicates the student needs either help or lower problem difficulty. These problem states were sorted by student and time, resulting in a 23,325 problem state string sequence representing 295 students across multiple sessions and schools.

During the **Pattern-Analysis Stage**, a descriptive graphics tool, IBMs Many Eyes Word Tree algorithm is used to quickly gain an understanding of patterns [10]. A word tree is typically used as a method for graphically summarizing text, for example, gaining insight into a famous speech by viewing the word sequences and their frequency. Applying the algorithm to our problem string allowed us to quickly discover the most frequent patterns of behavior. Figure 1 shows the total 1280 ATT (attempted and solved) events. Most frequently ATT was followed by a SOF event (see top tree). The second level of the tree shows that the sequence ATT ATT the highest frequent event changes to the ATT event, i.e. the shift in behavior occurs after two ATT states (see second tree and top branch). This indicates the ATT state is more often a solitary event, where the ATT ATT pattern will continue in the ATT state. Thus, from the analysis the most frequent 3 problem state patterns (e.g., NOTR-NOTR-NOTR) are determined (see third tree and second branch).

The last stage is the **Feature Selection and Model Building Stage**, in which we identify the benefit of these state-classifications and patterns over raw descriptors and sensor data in prediction. We used the 7 states (S) (e.g., ATT=true, NOTR=false) and 14 most frequent 2 problem state patterns (3S). So each student-problem interaction row has associated with it: a) variables for raw descriptors of the interaction with that problem (e.g. hints seen = 2, time spent = 2 minutes, incorrect attempts = 0); b) a state-based classification of the interaction (S); c) 14 binary variables for the presence or absence of the most common patterns (3S) during the last 3 problems seen. We evaluated the contribution of adding or removing these S states triplet-motifs (3S) in the prediction of emotion at time t , where the motifs describe tutor activities at time $t-1$, $t-2$ and $t-3$.

4 Results

Stepwise regression was used to construct a linear model with significant predictors, and overall model fit (Table 2). The results suggest the addition of states and their patterns improves prediction. Similar results for frustration suggest that, when sensors are not present, adding states and state-patterns contributes to a better prediction of frustration. While incorrect attempts over the last problem keeps being important, as well as hints seen and the presence of the female character, a variety of other states over the last problem (SOF, GIVEUP, SHINT) are important predictors, and SOF_SOF_ATT in particular.

Table 2. R Values for the prediction of CONFIDENCE / FRUSTRATION

	<i>RW</i>	<i>S</i>	<i>RW + S</i>	<i>S + 3S</i>	<i>RW+S+3S</i>
None	0.39/ 0.39	0.32/ 0.34	0.41/ 0.42	0.34/ 0.37	0.42/ 0.43
Camera	0.40/ 0.46	0.37/ 0.41	0.40/ 0.48	0.37/ 0.41	0.40/ 0.48
Seat	0.39/ 0.47	0.31/ 0.43	0.39/ 0.50	0.34/ 0.45	0.41/ 0.51
Mouse	0.41/ 0.41	0.35/ 0.33	0.42/ 0.41	0.38/ 0.33	0.44/ 0.41
Wrist	0.55/ 0.42	0.41/ 0.37	0.55/ 0.46	0.41/ 0.43	0.55/ 0.48

Cross-validation revealed small gains in accuracy for the state-based models, 1%-5%, and 3%-10%, compared to the baseline models, last problem raw features. We analyzed the relationships of the problem states and emotions, see Figure 2. Reading upper left panel, confident readings, the "positive" states, attempts, solved on first and solved with help, all showed positive confident, while the "negative" states, guessing, bottom out, not reading, quit generally negative confidence. The opposite is true in the frustrated panel, lower left. The Pearson Chi-Square test for independence shows statistical significance and a CramerV of 0.116.

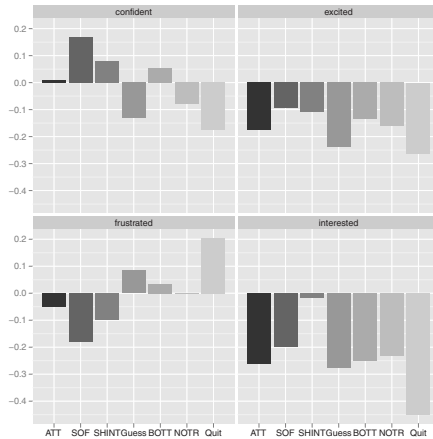


Fig. 2. State/Emotion Relationship

5 Discussion and Future Work

We described a data-driven approach toward automatic prediction of students emotional states without sensors and while students are still actively engaged in their learning. We created models from students ongoing behavior. A cross-validation revealed small gains in accuracy for the more sophisticated state-based models and better predictions of the remaining unpredicted cases, compared to the baseline models. An important opportunity exists for tutoring systems to optimize not only learning, but also long-term attitudes related to students' emotions while using software. By modifying the context of the tutoring system including students perceived emotion around mathematics, a tutor can now optimize and improve a students mathematics attitudes.

A variety of changes can be made that might improve the predictive power of models. For instance, we might choose the two most frequent triplet patterns starting with a specific state. It is possible that rare patterns work better at

predicting some emotions, particularly infrequent ones. Last, it is unclear if we need to look at the last 3 states, or only last 2 states.

After highly accurate states have been found, future work consists of refining emotion models to predict desirable and undesirable learning states and attitudes. The outcome of the current study will be used to respond with interventions; responding based on different levels of assessment of engagement and emotions combined.

References

- [1] Arroyo, I., Beal, C.R., Murray, T., Walles, R., Park Woolf, B.: Web-Based Intelligent Multimedia Tutoring for High Stakes Achievement Tests. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) ITS 2004. LNCS, vol. 3220, pp. 468–477. Springer, Heidelberg (2004)
- [2] Arroyo, I., Mehranian, H., Woolf, B.: Effort-based Tutoring: An Empirical Approach to Intelligent Tutoring. In: Proceedings of the 3rd International Conference on Educational Data Mining, Pittsburgh, PA (2010b)
- [3] Baker, R.S., Corbett, A.T., Koedinger, K.R., Roll, I.: Detecting When Students Game the System, Across Tutor Subjects and Classroom Cohorts. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 220–224. Springer, Heidelberg (2005)
- [4] Chui, B., Keogh, E., Lonardi, S.: Probabilistic discovery of time series motifs. In: Proceedings of Knowledge Discovery in Data, pp. 493–498 (2003)
- [5] Cooper, D., Arroyo, I., Woolf, B.P.: Actionable Affective Processing for Automatic Tutor Interventions. In: Calvo, R.A., D’Mello, S. (eds.) New Perspectives on Affect and Learning Technologies. Springer, New York (in press)
- [6] D’Mello, S.K., Graesser, A.C.: Automatic Detection of Learner’s Affect from Gross Body Language. *Applied Artificial Intelligence* 23(2), 123–150 (2009)
- [7] Johns, J., Woolf, B.P.: A Dynamic Mixture Model to Detect Student Motivation and Proficiency. In: Proceedings of the National Conference on Artificial Intelligence, p. 163 (2006)
- [8] Koedinger, K.R., Anderson, J.R., Hadley, W.H., Mark, M.A.: Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education* 8(1), 30–43 (1997)
- [9] Lin, J., Keogh, E., Lonardi, S., Patel, P.: Finding motifs in time series. In: Proceedings of the 2nd Workshop on Temporal Data Mining, pp. 53–68 (2002)
- [10] ManyEyes, <http://www-958.ibm.com/software/data/cognos/manyeyes/> (retrieved March 11, 2012)
- [11] Picard, R.W., Papert, S., Bender, W., Blumberg, B., Breazeal, C., Cavallo, D., Machover, T., Resnick, M., Roy, D., Strohecker, C.: Affective Learning—A Manifesto. *BT Technical Journal* 2(4), 253–269 (2004)
- [12] Shanabrook, D., Cooper, D., Woolf, B.: Identifying High-Level Student Behavior Using Sequence-based Motif Discovery. In: Proceedings of EDM, vol. 200 (2010)

Toward a Machine Learning Framework for Understanding Affective Tutorial Interaction

Joseph F. Grafsgaard, Kristy Elizabeth Boyer, and James C. Lester

Department of Computer Science, North Carolina State University
Raleigh, North Carolina, USA

{jfggrafsg, keboyer, lester}@ncsu.edu

Abstract. Affect and cognition intertwine throughout human experience. Research into this interplay during learning has identified relevant cognitive-affective states, but recognizing them poses significant challenges. Among multiple promising approaches for affect recognition, analyzing facial expression may be particularly informative. Descriptive computational models of facial expression and affect, such as those enabled by machine learning, aid our understanding of tutorial interactions. Hidden Markov modeling, in particular, is useful for encoding patterns in sequential data. This paper presents a descriptive hidden Markov model built upon facial expression data and tutorial dialogue within a task-oriented human-human tutoring corpus. The model reveals five frequently occurring patterns of affective tutorial interaction across text-based tutorial dialogue sessions. The results show that hidden Markov modeling holds potential for the semi-automated understanding of affective interaction, which may contribute to the development of affect-informed intelligent tutoring systems.

Keywords: Affect, hidden Markov models, tutorial dialogue.

1 Introduction

Research in recent years has highlighted the interplay of cognition and affect in tutorial interaction. This interplay has implications for the design of intelligent tutoring systems (ITSs) that seek to attain or exceed the effectiveness of expert human tutors. To meet this goal, recent results demonstrated that understanding both the cognitive and affective nature of tutorial interaction may be necessary [1]. Affective phenomena during interactions with ITSs have been examined through a wide array of modalities including self-reports, observation, system logs, dialogue, facial expression, posture, and physiological measures [1]. Prior investigations of facial expression in tutoring identified links between particular facial movements and cognitive-affective states relevant to learning [2].

This paper details the construction and analysis of a descriptive HMM built from task-oriented textual tutorial dialogue annotated with dialogue acts and facial expression annotated from video. Facial movement combinations were annotated in a novel, three-phase protocol to provide rich affective representation within tutorial

dialogue. Analysis of the learned HMM structure revealed five prevalent and persistent patterns of affective tutorial interaction represented by recurring sequences of hidden states. These results show the potential of HMMs for semi-automated understanding of affective tutorial interaction, which may inform integration of affect into future ITSs.

2 Related Work

Few studies have utilized hidden Markov models (HMMs) to model affect within the context of learning. In a recent study based on interactions with AutoTutor [3], HMMs learned transitions primarily consistent with the theory of cognitive disequilibrium. In an earlier study with Wayang Outpost [4], a math ITS for standardized test preparation, an HMM that modeled motivation improved predictive accuracy in a dynamic mixture model for correctness of student responses. Both approaches added constraints on top of those inherent within HMM assumptions. A recent study of human-human tutoring that modeled student brow lowering (an indicator of confusion) using HMMs provided both a predictive model and an analysis of confusion within the tutorial interaction [5]. The work presented here builds on these prior findings by leveraging sixteen facial movements (including brow lowering) in a purely descriptive model built without additional constraints, resulting in a richer representation of affect.

3 Dialogue Corpus and Facial Expression Annotation

A corpus of human-human tutorial dialogue was collected during a tutorial dialogue study [6]. Students solved an introductory computer programming problem and engaged in computer-mediated textual dialogue with a human tutor. The corpus consists of 48 dialogues annotated with dialogue acts, shown in Table 1. Student facial video was collected for post-analysis. (Note that the videos were not shown to tutors.) Seven of the highest quality facial videos were selected for the extent to which the student's entire face was visible during the recording, and for near-even split across genders and tutors. These videos were annotated with facial expressions for the present analysis (selected examples are shown in Figure 1). Tutoring sessions ranged in duration from thirty minutes to over an hour.

The seven selected facial videos were manually annotated using the Facial Action Coding System (FACS), which enumerates the possible movements of the face through a set of facial action units (AUs) [7]. Two certified FACS coders viewed entire videos, encoding facial events of one or more AUs with a start and end frame. Some FACS AUs were excluded due to excessive burden in manual FACS coding (e.g., mouth opening, blinking) or anticipated rarity (e.g., lip pucker, lip funneler). Sixteen were selected for coding: AUs 1, 2, 4-7, 9, 10, 12, 14-17, 20, 23, 24, and 31.

In the first phase of the condensed FACS protocol, the two certified FACS coders independently annotated occurrences of AUs. The coders met in a second phase to produce a combined set of facial event instances without discussing specific AUs, during which event instances were merged or eliminated. By the end of the second phase, the coders agreed completely upon the start and end time of facial events (without discussing specific AUs). In the third phase, one of the coders reviewed where the facial events occurred and decided on precisely which AUs occurred. Finally, the second coder annotated 9.3% of the facial events independently, establishing an agreement average of Cohen's $\kappa=0.67$, comparable with similar studies [2].

Table 1. Dialogue act tags and frequency across the seven sessions (S = student, T = tutor)

Act	Description	S	T
ASSESSING QUESTION	Task-specific query or feedback request	16	29
EXTRA DOMAIN	Unrelated to task	20	26
GROUNDING	Acknowledgement, thanks, greetings, etc.	26	16
LUKEWARM FEEDBACK	Partly positive/negative task feedback	2	12
LUKEWARM CONTENT FDBK	Partly positive/negative elaborated feedback	1	9
NEGATIVE FEEDBACK	Negative task feedback	5	5
NEGATIVE CONTENT FDBK	Negative elaborated feedback	1	34
POSITIVE FEEDBACK	Positive task feedback	10	76
POSITIVE CONTENT FDBK	Positive elaborated feedback	2	5
QUESTION	Conceptual or other query	13	9
STATEMENT	Declaration of factual information	18	143



Fig. 1. Examples of facial action units: AUs 1+2 or “surprise” (left), 14+17 or “doubt” (center), and 4+12 or “confusion and frustration” (right). Arrows indicate facial movements.

This event-based annotation protocol incorporates AU combinations, which denote multiple facial movements occurring at the same time. While related research has indicated some facial expression and emotion correlations [2,7], affect-facial expression mapping is a difficult problem that requires considering the surrounding context. Affective interpretations discussed here are based on the simplified tutorial context offered by computer-mediated tutorial interaction.

4 Hidden Markov Modeling and Discussion

A hidden Markov model (HMM) is defined by an *initial probability distribution* across hidden states, *transition probabilities* between hidden states, and *emission probabilities* for each hidden state and observation symbol pair [8]. HMMs learn a probabilistic structure that preserves patterns within the modeled phenomena, such as the interplay between facial expression and dialogue in affective tutorial interaction.

The facial expression and dialogue data described in Section 3 were merged into sequences of observations needed to build the HMM. Each observation consisted of a facial expression (denoted as facial action units (AUs) [7]), dialogue act or both. The Baum-Welch algorithm with log-likelihood measure was used for model training. Ten random initializations were performed to reduce convergence to local maxima. A hyperparameter optimization outer loop produced candidate HMMs across a range from three to twenty-two hidden states. Average log likelihood was computed across candidate HMMs for each number of hidden states. The models with best average log-likelihood had ten hidden states, and the best-fit model had the highest log-likelihood among these.

With the model in hand, the Viterbi algorithm was applied to map the most probable hidden state to each observation. Exhaustive search to length five across each session's hidden state sequences revealed five frequently recurring sequences (or "patterns") of affective tutorial interaction, shown in Figure 2. Each pattern occurred at a relative frequency greater than 0.05 across multiple sessions. Seven (of ten) hidden states comprised the patterns.

In order to examine the persistence of the five frequently-occurring patterns of affective tutorial interaction, average sequence lengths were calculated for each session (shown in Figure 2). There are subtle differences between relative frequency as a measure of prevalence and average sequence length as a measure of persistence. When the measures agreed (as was often the case), they showed prevalence and persistence of specific patterns of affective tutorial interaction within a particular session. When the measures differed, a persistent pattern recurred in long, but rare, sub-sequences or a prevalent pattern recurred in short sub-sequences.

The average sequence lengths shown in Figure 2 indicate notable differences in affective tutorial interaction within sessions. Thus, it may be possible to group sessions that have similar quantitative profiles. For instance, sessions 6 and 7 both have persistent sequences of PATTERN 2 and PATTERN 4, indicative of persistent student confusion with tutor statements and conversational dialogue during those sessions. Likewise, PATTERN 1 models tutor lecturing and instruction with occasional student participation and student affective states, PATTERN 3 is dominated by student facial displays (mostly surprise and frustration), and PATTERN 5 is largely composed of doubt, surprise, and stress with occasional tutor feedback and statements. In this way, quantitative application of HMMs provides insight into profiles of affective tutorial interaction across tutoring sessions.

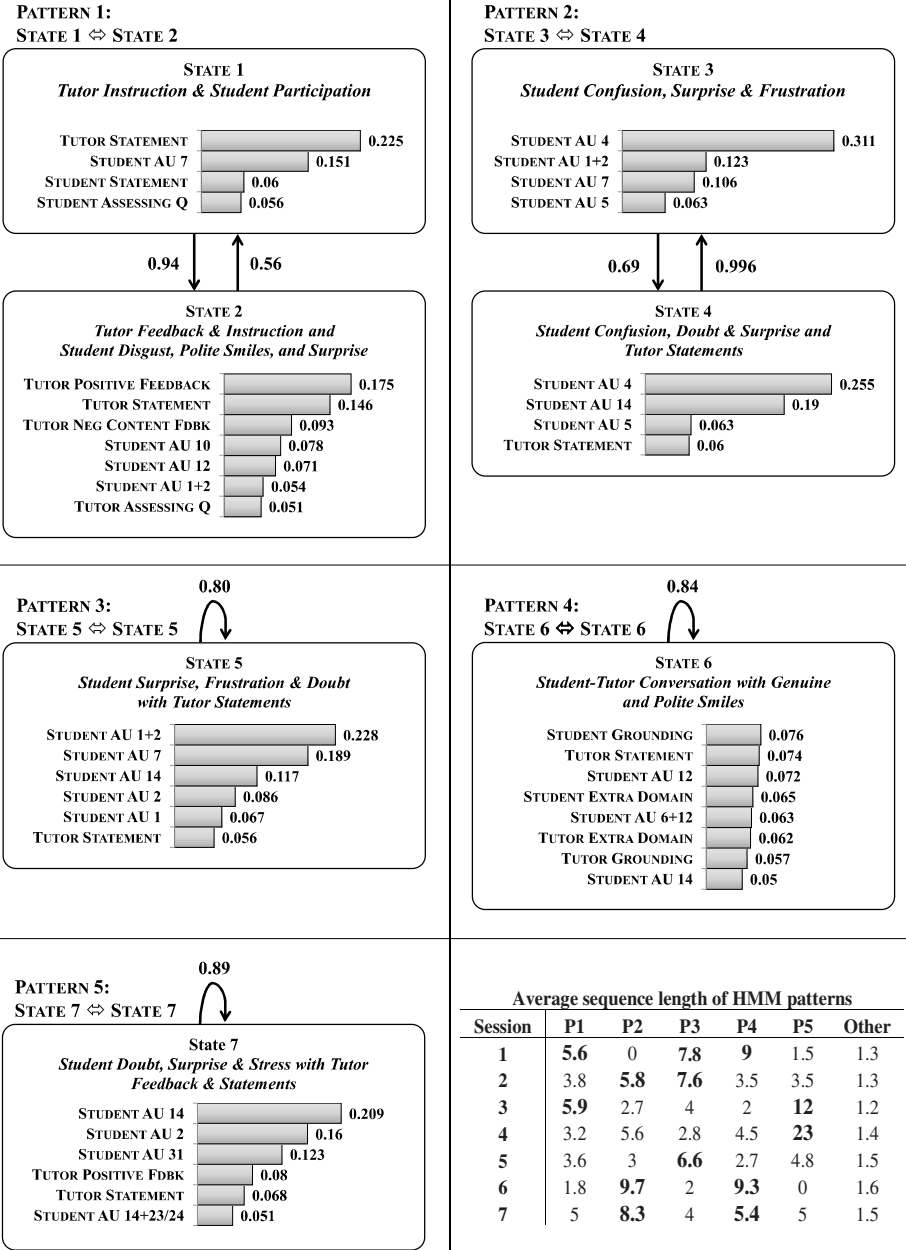


Fig. 2. Five patterns (i.e. frequently recurring sequences of hidden states) of affective tutorial interaction discovered from the best-fit hidden Markov model. Transition probabilities ≥ 0.5 are displayed. Emissions probabilities ≥ 0.05 are shown.

5 Conclusion and Future Work

The descriptive HMM learned from facial expression and task-oriented tutorial dialogue revealed five frequently-occurring patterns of affective tutorial interaction. Each pattern modeled distinct and interpretable segments of the tutoring sessions. A closer inspection of hidden state sequences as they occurred within sessions showed notable differences between sessions.

While this approach toward semi-automated understanding of affective tutorial interaction was successful, there are two primary limitations that highlight important directions for future work. First, manual FACS coding requires substantial manual labor, although this may become irrelevant when sufficient reliability is achieved in automated facial expression recognition. Second, the small sample size was a limiting factor, but using this approach across more tutoring sessions may identify statistical relationships involving discovered patterns of affective tutorial interaction. The quantitative distinctions in prevalence and persistence of discovered patterns of affective tutorial interaction may highlight individual or group-wise differences, leading to correlational analyses of HMM patterns and tutorial outcomes, such as self-efficacy and learning gains.

Further studies investigating the application of machine learning techniques are merited to advance the state of semi-automated affect understanding. Leveraging novel, semi-automated techniques may enable us to better understand affect during learning and contribute to efforts to integrate affect in ITSs.

Acknowledgements. This work is supported in part by the North Carolina State University Department of Computer Science along with the National Science Foundation through Grant DRL-1007962 and the STARS Alliance Grant CNS-1042468. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the participants, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

References

1. D'Mello, S.K., Calvo, R.A.: Significant Accomplishments, New Challenges, and New Perspectives. In: Calvo, R.A., D'Mello, S.K. (eds.) *New Perspectives on Affect and Learning Technologies*, pp. 255–271. Springer, New York (2011)
2. D'Mello, S.K., Lehman, B., Person, N.: Monitoring Affect States During Effortful Problem Solving Activities. *International Journal of Artificial Intelligence in Education* 20 (2010)
3. D'Mello, S.K., Graesser, A.: Modeling Cognitive-Affective Dynamics with Hidden Markov Models. In: *Proceedings of the 32nd Annual Cognitive Science Society*, pp. 2721–2726 (2010)
4. Johns, J., Woolf, B.: A Dynamic Mixture Model to Detect Student Motivation and Proficiency. In: *Proceedings of the 21st National Conference on Artificial Intelligence*, pp. 163–168 (2006)

5. Grafsgaard, J.F., Boyer, K.E., Lester, J.C.: Predicting Facial Indicators of Confusion with Hidden Markov Models. In: D’Mello, S., Graesser, A., Schuller, B., Martin, J.-C. (eds.) *ACII 2011, Part I. LNCS*, vol. 6974, pp. 97–106. Springer, Heidelberg (2011)
6. Boyer, K.E., Phillips, R., Ingram, A., Ha, E.Y., Wallis, M., Vouk, M., Lester, J.: Characterizing the Effectiveness of Tutorial Dialogue with Hidden Markov Models. In: Alevan, V., Kay, J., Mostow, J. (eds.) *ITS 2010. LNCS*, vol. 6094, pp. 55–64. Springer, Heidelberg (2010)
7. Ekman, P., Friesen, W.V., Hager, J.C.: *Facial Action Coding System. A Human Face*. Salt Lake City, USA (2002)
8. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE* 77, 257–286 (1989)

Exploring Relationships between Learners' Affective States, Metacognitive Processes, and Learning Outcomes

Amber Chauncey Strain¹, Roger Azevedo², and Sidney D'Mello³

¹University of Memphis, Memphis, TN 38152, USA

²McGill University, Montreal, Quebec H3A 1Y2, Canada

³University of Notre Dame, Notre Dame, IN, 46556, USA

dchuncey@memphis.edu, roger.azevedo@mcgill.ca, sdmello@nd.edu

Abstract. We used a false biofeedback methodology to investigate interactions among learners' affective states, metacognitive processes, and learning outcomes during multimedia learning. False-biofeedback is a method to induce physiological arousal (and resultant emotions) by presenting learners with audio stimuli of false heartbeats that are either accelerated, baseline, or control (no heartbeat). A path analysis indicated that the most complex relationships among affective states, metacognitive processes, and learning outcomes occurred when learners were presented with accelerated biofeedback. We discuss the implications of our findings for the development of ITSs that are sensitive to the complex relationship among these key processes.

Keywords: emotion, self-regulated learning, metacognition.

1 Introduction

Middle school and high school can be challenging for many young learners. This is in part because they are required to learn about conceptually-rich domains such as physics, ecology, chemistry, and biology. These challenging domains have the potential to elicit a host of negative emotions that may interfere with learners' ability to effectively regulate their learning. While many conceptual models of self-regulated learning (SRL) focus on learners' use of cognitive and metacognitive strategies to regulate their learning [1,2] the majority of these models do not adequately consider the role of emotion in self-regulation during multimedia learning. In order to examine these relationships in a controlled setting, we used a false-biofeedback methodology [3] to induce physiological arousal (and resultant emotions) by presenting learners with audio stimuli of false heartbeats (accelerated and baseline). In some trials we presented learners with no stimulus; these served as control trials. Our purpose for using this methodology, rather than examining emotions as they naturally arose, is that emotions that arise spontaneously during learning are often highly transient, which makes them difficult to study. Therefore, our goal was to use a precise, experimentally controlled method for inducing affect in order to better uncover relationships among affect, metacognition, and learning. In this paper, we use a path analysis approach to uncover

the links among affect, metacognition, and performance across the three false biofeedback conditions. The broader goal is apply knowledge gained about these complex relationships towards the design of more effective intelligent tutoring systems.

2 Method

2.1 Participants

Fifty undergraduate students from a southern public college in the U.S. participated in this experiment. The participants' mean age was 23.3 years ($SD = 7.13$), and there were 34 females (68%) in the sample. There were 54% Caucasians, 44% African Americans, and 2% Latino. All participants received \$20 for participating in the experiment.

2.2 Stimuli and Software

A self-paced multimedia learning environment that comprised 24 slides about the human circulatory system was presented via a computer interface. The interface was configured to deliver content, present comprehension questions, record responses to these questions, obtain self-reports on participants' metacognitive judgments, and monitor response times.

A Reebok Fit Watch 10s strapless heart rate monitor was worn on participants' non-dominant wrist. This heart rate monitor is typically used to detect and display the wearer's current heart rate. However, because previously-recorded baseline and accelerated heart rates were presented to participants (rather than their own heart rate), this function was not used for this experiment.

The two auditory stimuli (baseline and accelerated heart rates) were presented binaurally through headphones. These stimuli began playing when participants opened a content slide and played continuously until participants navigated away from the slide. During baseline trials, participants heard a recording of a resting human heart beat (approximately 70 BPM), and during arousal trials, they heard a recording of a human heart beat at an accelerated rate (approximately 100 BPM). During control trials, no auditory stimulus was presented. We used a within-subjects design, and randomly presented eight slides per biofeedback condition (accelerated, baseline, control). The presentation of these stimuli was counterbalanced across participants.

2.3 Materials and Procedure

The materials for this experiment were a consent form, a demographic questionnaire, and the Affect Grid. The Affect Grid [4] is a single item affect measurement instrument consisting of a 9×9 (valence \times arousal) grid; these are the primary dimensions that underlie affective experience.

The learning session proceeded over 24 trials, with each trial consisting of multiple steps. First, participants viewed either a text based question inference question related

to the content. After reading the question, participants were asked to indicate how easily they could learn the material by making an ease of learning (EOL) judgment.

Next, participants had as much time as necessary to read the content slide. Upon opening the content slide, the learning environment presented either accelerated, baseline, or no biofeedback through participants' headphones. When participants navigated to the next slide, they were prompted to indicate how well they understood what they had just read by making a judgment of learning (JOL). Following the JOL prompt, the text based or inference question was presented again and participants were prompted to answer the question by selecting from one of four multiple choice foils. Next, participants were prompted to indicate how accurate they thought their answer was by making a retrospective confidence judgment (RCJ). For the final step in each trial, participants were prompted to self-report their current level of valence and arousal on the Affect Grid. The completion of the Affect Grid marked the end of one trial. This multi-step process occurred for all 24 trials within the self-paced learning session.

3 Predictions, Results, and Discussion

3.1 Predicted Links between Affect, Metacognition, and Performance

We developed a model (see Fig. 1A) that is grounded in theories of affect [5,6] and a leading model of SRL that emphasizes cognitive processes and metacognitive monitoring and control [2]. The link from arousal to valence (Link 1), indicates that learners' level of arousal influences the kinds of positively or negatively valenced emotions they experience.

An extensive body of research indicates that arousal is predictive of performance outcomes [7]. There is presumably an optimal level of physiological arousal which enhances performance (for example, the kind of arousal that leads to engagement or interest but not intense anxiety). Although it is unclear exactly what the optimal level of arousal is, our model predicts a significant relationship between the intensity of learners' arousal and their overall learning performance (Link 2).

When arousal is moderate, valence is expected to be predictive of learning by affecting learners' metacognitive processes (Link 3). For example, perhaps negative emotions like frustration or confusion lead to *decreased* confidence in learning when learners attribute those negative emotions to their inability to understand the material. Thus, our model proposes a link between valence and judgments of learning.

The remaining links in our proposed model stem directly from theoretical and empirical research on the complex processes of self-regulated learning [2,8,9]. First, we predict a significant relationship between learners' EOLs and JOLs. Specifically, because learners should use previous metacognitive judgments (EOLs) to inform future metacognitive judgments (JOLs), we predict that learners who perceive a topic to be particularly difficult to learn will also report less understanding of that topic, and vice versa (Link 4). Learners' JOLs are typically predictive of overall learning performance [1,9], so our model includes a link between learners' JOLs and learning performance (Link 5). Lastly, we predict that performance will predict learners' RCJs.

This prediction is based on the assumption that self-regulation is a constant and active process in which learners assess their understanding and performance and making the necessary adjustments through the use of control processes. As such, we predict that learners will accurately assess the correctness of their responses to questions about the material and will use that assessment when they make their retrospective confidence judgments (Link 6).

The model was tested with a series of multiple regression analyses aimed at uncovering links between arousal, valence, EOLs, JOLs, RCJs, and performance. Six separate models were constructed for the accelerated, baseline, or no biofeedback conditions across text based and inference questions. However, interesting patterns only emerged when participants had to answer challenging inference questions, so the text-based models will not be discussed here. In the following section we will report only models and coefficients that were significant ($p < .05$) or marginally significant ($p < .10$).

3.2 Results

Control trials, in which participants received no false biofeedback, were the most similar to typical learning episodes since there was no experimental manipulation of emotion. As predicted, we found that EOLs predicted JOLs ($\beta = 0.77$), which in turn predicted performance ($\beta = 0.42$) (see Fig. 1B). However, we failed to find a significant link between performance and RCJs, demonstrating that participants were poor judges of their own performance when they received no biofeedback.

We found no significant links between participants’ affective processes (valence and arousal) and metacognitive processes and performance in the control condition. Overall, the resulting model for control trials suggests that participants did not experience affective states that were salient enough to impact metacognitive processes and performance.

We found a similar pattern in the **baseline** model, but with one important difference. Once again, we found that EOLs significantly predicted JOLs ($\beta = 0.81$), which in turn predicted performance ($\beta = 0.31$) (see Fig. 1C). We also found that during baseline trials participants’ performance was predictive of their RCJs ($\beta = 0.45$). This is interesting, as it demonstrates that presenting baseline biofeedback increased participants’ metacognitive awareness of their own learning. However, in contrast to the predicted model, we failed to find significant links among valence, arousal, JOLs, and performance.

The **accelerated** model was most closely aligned with our predicted model. As with the baseline model, there were significant links between EOLs and JOLs ($\beta = 0.71$), JOLs and performance ($\beta = 0.66$), and performance and RCJs ($\beta = 0.65$) (see Fig. 1D). Most importantly, the accelerated model yielded a significant positive link between valence and JOLs ($\beta = 0.13$), demonstrating that participants who experienced more positively valenced emotions while receiving accelerated biofeedback made more accurate judgments of their understanding of the material. Interestingly, however, we failed to find a significant link between arousal and valence or between arousal and any metacognitive or cognitive processes.

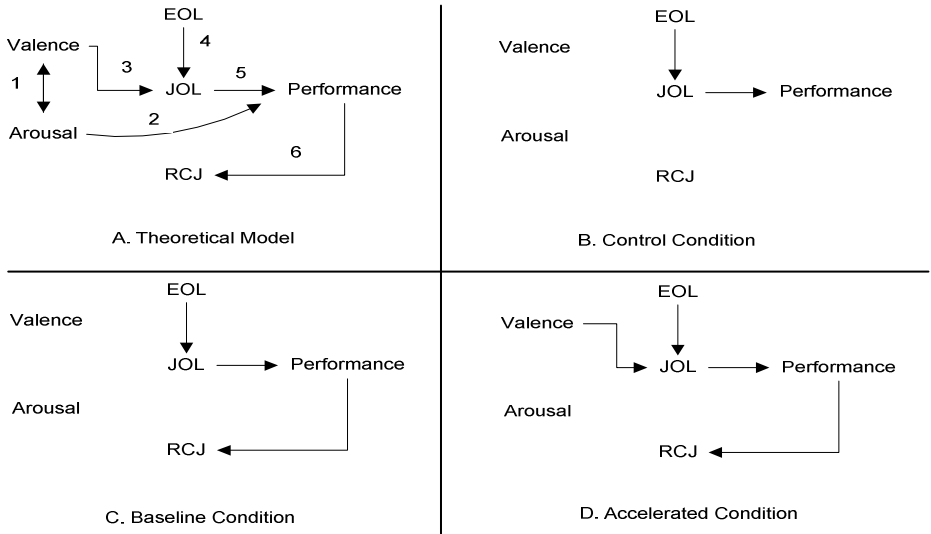


Fig. 1. Theoretical model of links between affect, metacognition, and performance

4 Discussion

In this experiment, we proposed and validated a model which integrated affect, metacognition, and performance during learning. We found that there are distinct models of the relationship among these processes that emerge across different levels of arousal induced by false biofeedback. These results emphasize the need for ITSs to be sensitive to the complex relationship among affect, metacognition, and learning. For example, ITSs that use pedagogical agents to scaffold learners' understanding of complex science topics might benefit from the use of physiological and bodily measures which can detect shifts in learners' emotional and motivational states in real-time. If a learner shifts to a negative emotional state (i.e., stress, boredom), a system which is sensitive to these shifts could help learners transition out of these emotional states by modeling, prompting, and scaffolding appropriate self-regulatory processes.

In conclusion, there is a need for more empirically-driven research directed toward understanding of the role of emotion, metacognition, and performance during multimedia learning. As theoretical, conceptual, and educational implications and methodological techniques are improved, the elusive role of emotion may be disambiguated, leading researchers to more fully understand the consequences of emotion on learning, and to develop ITSs that effectively coordinate learners' cognitive and emotional states.

Acknowledgments. This research was supported by the National Science Foundation (NSF) (DRL 0633918; IIS 0841835; DRL 1008282) awarded to the second author and (HCC 0834847, DRL 1108845) awarded to the third author. Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

References

1. Dunosky, J., Metcalfe, J.: *Metacognition: A textbook for cognitive, educational, life span and applied psychology*. Sage, Newbury Park (2009)
2. Winne, P., Hadwin, A.: The weave of motivation and self-regulated learning. In: Schunk, D., Zimmerman, B. (eds.) *Motivation and Self-Regulated Learning: Theory, Research, and Applications*, Taylor & Francis, NY (2008)
3. Valins, S.: Cognitive effects of false heart rate biofeedback. *Journal of Personality and Social Psychology* 4, 400–408 (1966)
4. Russell, J.A., Weiss, A., Mendelsohn, G.A.: Affect grid: A single-item scale of pleasure and arousal. *Journal of Personality and Social Psychology* 57, 493–502 (1989)
5. Clore, G.L., Ortony, A.: Appraisal theories: How cognition shapes affect into emotion. In: Lewis, M., Haviland-Jones, J.M., Barrett, L.F. (eds.) *Handbook of Emotions*, 3rd edn., pp. 628–644. Guilford Press, New York (2010)
6. Pekrun, R.: The control-value theory of achievement emotions: Assumptions, corollaries, and implications for educational research and practice. *Educational Psychology Review* 18, 315–341 (2006)
7. Zeidner, M.: Test anxiety in educational contexts: Concepts, findings, and future directions. In: Schutz, P., Pekrun, R. (eds.) *Emotions in Education*, pp. 165–184. Academic Press, San Diego (2007)
8. Zimmerman, B.: Investigating self-regulation and motivation: Historical back-ground, methodological developments, and future prospects. *American Educational Research Journal* 45, 166–183 (2008)
9. Leonesio, R.J., Nelson, T.O.: Do different measures of metamemory tap the same underlying aspects of memory? *Journal of Experimental Psychology: Learning, Memory, and Cognition* 16, 464–470 (1990)

Mental Workload, Engagement and Emotions: An Exploratory Study for Intelligent Tutoring Systems

Maher Chaouachi and Claude Frasson

HERON Lab, Computer Science Department
University of Montreal,
2920 chemin de la tour, H3T 1N8, Canada
{chaouacm, frasson}@iro.umontreal.ca

Abstract. Modeling learners' emotional states is a promising tool for enhancing learning outcomes and tutoring abilities. In this paper, we present a new perspective of learner emotional modeling according to two fundamental dimensions, namely mental workload and engagement. We hypothesize that analyzing results from learners' workload and engagement evolution can help Intelligent Tutoring Systems diagnose learners' emotional states and understand the learning process. We demonstrate by an experiment involving 17 participants that learners' mental workload and engagement are closely related to specific emotions with regard to different learning phases.

Keywords: Mental workload, Mental engagement, Emotion modeling, ITS.

1 Introduction

Endowing Intelligent Tutoring Systems (ITS) with abilities to wisely assess and monitor learners' affective and cognitive state has been an important research thrust over few past decades [1-3]. Several ITS with physio-cognitive models aiming to provide intelligent assistance, efficient adaptation and more realistic social communication were developed in the scope of reaching optimal interaction conditions, improving adaptability, enhancing learners' overall performance, skill acquisition and productivity [5-7].

In parallel, a growing body of research in the field of artificial intelligence, human computer interaction, cognition and neuroscience presented various models tracking shifts on users' alertness; engagement and workload and have been successfully used in closed-loop systems or simulation environment [2, 3, 8]. By assessing users' internal state, these systems were able to adapt to users' information processing capacity and then to respond accurately to their needs. The major part of these systems was based on two fundamental mental metrics, namely, mental workload and mental engagement.

Despite disagreement about its nature and definition, mental workload can be seen in terms of human information processing. It reflects the amount of the mental effort and energy invested as in a particular task. Mental engagement is more related to the

level of mental vigilance and alertness. It gives also a wide indication about the level of attention and motivation.

The integration of affective models in ITS added an empathic and social dimension into tutors' behaviors [6, 10]. However there is still a lack of methods helping tutors to analyze more deeply the emotional state of the learner and to diagnose and understand accurately the cognitive origin of an emotion. ITS are still mainly based on learners' performance in analyzing learning process and learners' skill acquisition [4-7]. Providing ITS with adapted tools and models to relate the affective reaction of a learner to his internal mental state can provide a new perspective for tutoring.

In this paper we present an exploratory study of emotions, mental engagement and workload within an educational environment. In particular, we performed an experiment to analyze the behavior of the computed mental metrics with regards to learners' emotional states.

2 Previous Work

Developing EEG indexes for workload assessment is an important field especially in laboratory contexts. A variety of linear and non-linear classification and regression methods were used to determine mental workload in different kinds of cognitive tasks such as memorization, language processing, visual, or auditory tasks. These methods used mainly EEG Power Spectral Density (PSD) bands combined with machine learning techniques [1, 8, 9]. In our previous work, we developed an EEG workload index based on Gaussian Process Regression (GPR) using data gathered from strict laboratory conditions [11]. The index showed to precisely reflect users' workload variation in several cognitive tasks.

Pope and colleagues [2] at NASA developed an EEG-engagement index based on brainwave band power and applied it in a closed-loop system to modulate task allocation. Performance in a vigilance task improved when this index was used as a criterion for switching between manual and automated piloting. Performance improvement was reported using this engagement index for task allocation mode (manual or automated). In this paper, we propose to explore the behavior of these mental metrics in a learning environment with regards to a self-reported emotion. The major contribution of this study is to present the different trends in engagement and workload with regards to emotional state within an educational context.

3 Methodology

Our experimental setup consists of a 6-channel EEG headset sensor and two video feeds. All recorded sessions were replayed and analyzed to accurately synchronize data using necessary time markers. 17 participants were recruited for this research. All participants were briefed about the experimental process and objectives and signed a consent form and were equipped with the EEG-cap. The experimental process consisted on a 10-minutes baseline followed three successive learning activities in trigonometry:

Pretest. This task involved 10 (yes/no/no-response) questions that covered some basic aspects of trigonometry (for instance: “is the tangent of an angle equal to the ratio of the length of the opposite over the length of the adjacent?”). In this part, participants have had to answer to the questions without any interruption, help or time limit.

Learning Session. In this task, participants were instructed to use a learning environment covering the theme of trigonometry and specially designed for the experiment. Two lessons were developed explaining several fundamental trigonometric properties and relationships. The environment provides basic definitions as well as their mathematical demonstrations. Schemas and examples are also given for each presented concept. Several concepts on trigonometry were recalled and the links between the different concepts was clearly explained

Problem Solving. Problems presented during this task are based on participants’ ability to apply, generalize and reason about the concepts seen during the learning session. No further prerequisites were required to successfully resolve the problem except lessons’ concepts. However a good level of implication and concentration is needed to solve the problems. A total of 6 problems with a gradually increasing difficulty level were selected and presented in the same order for all participants. Each problem is a multiple-choice question illustrated by a geometrical figure. A fixed time limit is imposed for each problem varying according to its difficulty level. The problem-solving environment provided also a limited number of hints for each problem.

3.1 Subjective Reporting of Emotional State

After completing each task level (pretest, learning session, or after each problem), participants were asked to evaluate their emotion during the last task execution. Two axes were presented for the learner. The first axis corresponds to the emotional valence and the second axis links the emotional activation. Four main emotional states were then derived: **Q1** (positive valence, high activation), **Q2** (positive valence, low activation), **Q3** (negative valence, high activation) and **Q4** (negative valence, low activation). In order to help learner situate their emotional state examples of emotions which might arise in each state were given (e.g. interest, joy for Q1, confidence, relax for Q2, confusion, frustration for Q3 and boredom, disengagement for Q4).

3.2 EEG Recording

EEG signals were received from sites P3, C3, Pz and Fz as defined by the International 10-20 Electrode Placement System (Jasper 1958). Each site was referenced to Cz and grounded at Fpz. Two more active sites were used namely A1 and A2. Impedance was maintained below 5 Kilo Ohms and the recorded sampling rate was at 256 Hz. A 60-Hz notch filter was applied during the data acquisition to remove EEG noise. In addition, an artifact rejection heuristic was applied to the recorded data using a threshold on the signal power with regards to the baseline. For each participant, EEG data recorded from each channel were transformed into a power spectral density using a

Fast-Fourier Transform (FFT) applied to each 1-second epoch with a 50 % overlapping window multiplied by the Hamming function to reduce spectral leakage.

3.3 Computing the Engagement Index

As previously mentioned, a developed an engagement index was used in closed-loop system to assist pilot with regards to their mental engagement [2]. This index uses three EEG bands: **Theta** (4–8 Hz), **Alpha** (8–13 Hz) and **Beta** (13–22 Hz). The *ratio* used was: $\text{Beta} / (\text{Alpha} + \text{Theta})$. This *ratio* was also found as being the most effective when validated and compared to many other indices [8]. In our study, we used this index within a 40s window sliding technique to smooth the behavior of this index.

3.4 Computing the Workload Index

In order to compute the mental workload an EEG metric, we used our workload assessment model based on kernel Gaussian Process Regression and principal component analysis (PCA) algorithm. Detailed description of this model is presented in [11].

4 Results and Discussions

Our first investigation was to evaluate the progression of the workload and engagement level across the learning tasks. A repeated measures ANOVA revealed that there were significant changes in the EEG_Workload between the learning activities $F(3.23, 51.61) = 2.76, p < 0.05$). Degrees of freedom were corrected using the Greenhouse-Geisser estimates of sphericity ($\epsilon = 0.46$). Post hoc results showed that the EEG_Workload measures significantly increased during the learning session when compared with the pretest ($p < 0.05$). This increase can be explained by the effort produced by learners in understanding concepts and acquiring skills in the learning phase compared to the pretest session where learners responded to questions that did not require particular mental effort. In fact, during pretest no pressure was put on learners who had simply to situate their knowledge in trigonometry. Repeated measures ANOVA revealed no significant change in the engagement index across overall activities $F(3.66, 58.70) = 0.690, p = \text{ns}$, Greenhouse-Geisser correction $\epsilon = 0.52$). However, an expected significant decrease in learners' engagement was registered between the beginning of the lesson and the end of the problem solving phase ($p < 0.05$). In fact, switching from the pretest phase to the learning session (or from a problem to another) can have no effect on the level of engagement, vigilance or alertness of a learner. However, the engagement tended to decrease in general and this effect tended to be more pronounced at the end of the experiment. This trend is expected as the alertness of the learners can be reduced by the fatigue. The opposite trend was registered in learners' workload behavior. Mental workload significantly increased from the beginning to the end of the learning interaction. This tendency is also expected as the problem solving task is more mentally demanding than pretest or learning session (see Fig.1).

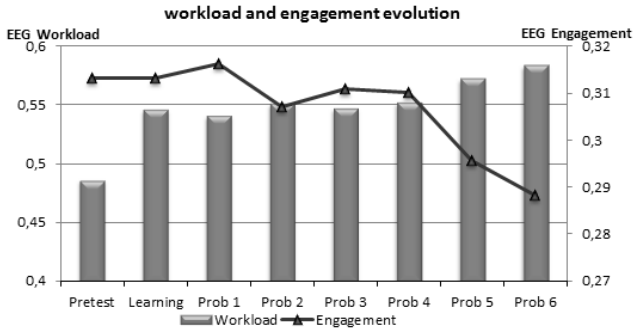


Fig. 1. Mean learners' workload and engagement for each activity

Our next concern was to evaluate how learners reported their emotions with regards to the mental workload and engagement metrics. A one-way ANOVA showed that there is a significant effect of the emotional state reported by the learners on the engagement, $F(3, 132)=3.32, p < 0.05$ and workload $F(3, 132)=4.52, p < 0.05$ for all participants. Specifically, the analysis of this result revealed that mean engagement index values were significantly higher when learner's emotional state was in Q1 (Positive valence and high activation: $M = 0.568, SD = 0.29$) compared to the other quadrants (see figure 2). Giving this result, we can state that positive emotions arising in Q1 (such as interest) seem to lead to the highest level of user engagement. Emotional state Q1 presented also the lowest value of the workload index: $M = 0.51, SD = 0.09$. However, the highest workload value was registered in the Q3 emotional state (Negative valence and high activation: $M = 0.68, SD = 0.13$). This suggests that negative emotions with high activation (e.g. confusion) can signal a high level of mental workload.

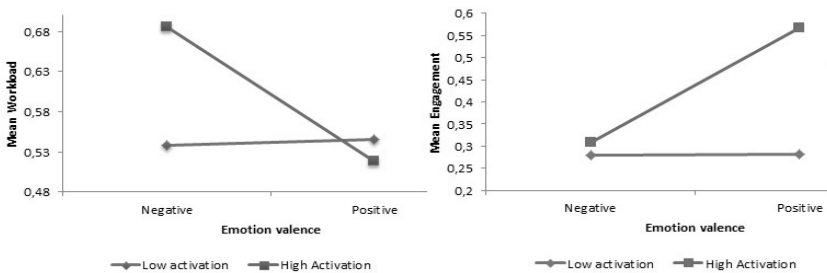


Fig. 2. Workload and engagement interaction with emotional valence and activation

A closer look to these results leads us to notice that higher level of alertness and attention elicit a high emotional activation however these emotions tended to be positive when the mental demand is low and negative in the opposite case. For example if a learner is highly involved in resolving a difficult problem and involves a high

mental focus, emotions such as confusion or frustration could emerge during the activity. Lowest engagement value was recorded for Q4 (Negative valence and low activation: $M = 0.27$, $SD = 0.19$). One-way ANOVA also confirmed this significant impact of emotional state on workload and engagement during the learning phase and the problem solving phase when these activities are taken separately. Two factorial ANOVA were also performed to test the main effects and the interaction effect of the emotional valence (positive, negative) and emotional activation (high, low) on workload and engagement. A significant main effect was obtained for emotional valence on workload $F(1,99)=4,390$ $p<0,05$ and for engagement $F(1,99)=6.237$ $p<0.05$. The interaction of the valence and the activation was also significant on both workload and engagement.

5 Conclusion

In this paper we presented an empirical study of workload and engagement metrics extracted from EEG signals with regards to reported emotional state. Workload and engagement indexes were analyzed with different learning activities of trigonometry. Results showed that there is significantly impact of these metrics on the reported emotion. High level of workload indicated the elicitation of negative emotions whereas engagement level was mainly associated with positive emotions. Future works involve developing a tutor which reacts in real time with the mental metrics. Moreover different strategies for handling and managing mental effort will be explored.

Acknowledgements. we acknowledge the NSERC of Canada and the Tunisian government for funding this work.

References

1. Berka, C., Levendowski, D.J., Cvetinovic, M.M., et al.: Real-Time Analysis of EEG Indexes of Alertness, Cognition, and Memory Acquired With a Wireless EEG Headset. *International Journal of Human-Computer Interaction* 17, 151–170 (2004)
2. Pope, A.T., Bogart, E.H., Bartolome, D.S.: Biocybernetic system evaluates indices of operator engagement in automated task. *Biological Psychology* 40, 187–195 (1995)
3. Prinzel, L.J., Freeman, F.G., Scerbo, M.W.: A Closed-Loop System for Examining Psychophysiological Measures for Adaptive Task Allocation. *International Journal of Aviation Psychology* 10, 393–410 (2000)
4. Baker, R.S., Corbett, A.T., Koedinger, K.R.: Detecting Student Misuse of Intelligent Tutoring Systems. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) *ITS 2004*. LNCS, vol. 3220, pp. 531–540. Springer, Heidelberg (2004)
5. Arroyo, I., Cooper, D.G., Bursleson, W., et al.: Emotion Sensors Go To School. In: *Proceeding of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pp. 17–25. IOS Press (2009)
6. D’Mello, S., Craig, S., Witherspoon, A., et al.: Automatic detection of learner’s affect from conversational cues. *User Modeling and User-Adapted Interaction* 18, 45–80 (2008)

7. Forbes-Riley, K., Rotaru, M., Litman, D.J.: The relative impact of student affect on performance models in a spoken dialogue tutoring system. *User Modeling and User-Adapted Interaction* 18, 11–43 (2008)
8. Wilson, G.F.: An analysis of mental workload in pilots during flight using multiple psychophysiological measures. *Int. J. Aviat. Psychol.* 12, 3–18 (2004)
9. Stevens, R.H., Galloway, T., Berka, C.: EEG-Related Changes in Cognitive Workload, Engagement and Distraction as Students Acquire Problem Solving Skills. In: Conati, C., McCoy, K., Paliouras, G. (eds.) *UM 2007. LNCS (LNAI)*, vol. 4511, pp. 187–196. Springer, Heidelberg (2007)
10. Woolf, B., Bursleson, W., Arroyo, I., Dragon, T., Cooper, D., Picard, R.: Affect-aware tutors: recognising and responding to student affect. *Int. J. Learning Technology* 4(3/4), 129–163 (2009)
11. Chaouachi, M., Jraidi, I., Frasson, C.: Modeling Mental Workload Using EEG Features for Intelligent Systems. *User Modeling and User-Adapted Interaction*, 50–61 (2011)

Real-Time Monitoring of ECG and GSR Signals during Computer-Based Training

Keith W. Brawner and Benjamin S. Goldberg

United States Army Research Laboratory
Human Research and Engineering Directorate
Learning in Intelligent Tutoring Environments (LITE) Laboratory
Simulation and Training Technology Center, Orlando, FL 32826
{keith.w.brawner, benjamin.s.goldberg}@us.army.mil

Abstract. The potential of Intelligent Tutoring Systems (ITSs) to influence learning may be greatly enhanced by the tutor's ability to accurately assess the student's state in real-time and then use this state as a basis to provide timely feedback or alter the instructional content. In order to maximize the ITS' potential to influence learning, the physiological state of students needs to be captured and assessed. Electrocardiogram (ECG) and Galvanic Skin Response (GSR) data has been shown to be correlated to physiological state data, but the development of real-time processing and analysis of this data in an educational context has been limited. This article describes an experiment where nineteen participants interacted with the Cultural Meeting Trainer (CMT), a web-based cultural negotiation trainer. Metrics of mean, standard deviation, and signal energy were collected from the GSR datastream while instantaneous and average heart rate were collected from the ECG datastream using a windowing technique around important interactions. Our analysis assesses these measures across three interaction scenarios. The findings of this experiment influence the appropriateness of instructional intervention, and drive the development of real-time assessment for education.

Keywords: Intelligent Tutoring, Affective Computing, Physiological Sensing, Scenario-Based Training, Instructional Intervention.

1 Introduction

Technology-driven instruction has led to a culture of learning that extends beyond the confines of the conventional classroom. With continual advancements in computing resources and artificial intelligence, computer-based instruction has evolved into a means for providing tailored and personalized educational experiences. This is achieved through the application of Intelligent Tutoring Systems (ITS) that monitor student interactions in real-time and adapt learning events to the individual. ITSs, in certain domains such as mathematics, have been shown to be more effective than traditional classroom instruction [1]. This capability is propagated through web-based systems that produce a one sigma difference, on average, in performance and reduce the need for training support personnel by 70%, and operating costs by 92% [2].

However, expert human tutoring has shown to produce two standard deviations of improvement [3]. Tutors sense and make decisions based upon observations relating to affective states, and are then used by tutors to direct flow and difficulty [4]. This promotes efficiency and thoroughness in decision making and problem solving [5]. While humans sense affect naturally, ITSs must assess the user via sensors. An affect-sensitive ITS monitors the emotional state of the user in order to provide intervention, if appropriate. Sensor technology advancements offer a unique opportunity with this approach, as student interactions and physiological variables can be monitored. This allows for an ITS to respond to an individual student's affective needs, which can improve learning outcomes [6].

There is a strong link between affect, cognition, and learning [7]. Electrocardiogram (ECG) and Galvonic Skin Response (GSR) signals, specifically, have been shown to be significant factors in emotional aspects. Several researchers are beginning to believe the claims that GSR [8] and ECG [17] data are appropriate for response to ITS interactions, which are the sources of measured data in this paper.

If a student is monitored in real-time and assessed to be in a state which is not conducive to learning, there is still the issue of what type of instructional interaction to apply for correction. Two possible methods of instructional intervention that can be implemented within scenario-based training are to reduce specificity of task or provide an unexpected response. It is expected that the response to these types of variations is observable within the ECG and GSR metrics.

2 Methodology

Each participant interacted with the CMT, a web-based system prototype for training bilateral negotiations. The game characteristics are representative of Middle Eastern culture, with scenario interactions presented through static dialogue. Each of the participants experienced 5-6 minute conversations with three individuals, in randomly assigned order. A baseline measurement and break period of 120 seconds was included between each of these interactions. Interactions with the three characters corresponded to information gathering assignments at a hospital following an insurgency attack. The first of these tasks was Well-Defined with No Interruption (WDNI) and involved maintaining small talk with an in-house physician. The second task, which was Ill-Defined with No Interruption (IDNI), was a conversation with the lead physician to gain information without making firm commitments. The third task, which was Ill-Defined with an Interruption (IDI), was intended to gain US support and identify hospital needs with the hospital administrator. The character interrupts discussion by speaking out of turn when an answer is attempted.

The methodology of this paper is heavily based upon the previously reported pilot study [9]. However, there are two large deviations: the type of data collected and the population group of interest. The first difference between experiments is that this study focuses on the ECG and GSR datasets. The second is that this study focused on a population of interest: current cadets of the United States Military Academy (USMA) at West Point.

Thirty-five cadets volunteered as subjects for this study. Following informed consent and collection of demographics, each participant was fitted with ECG and GSR sensors from the Biopac system. Due to noise in data collection and erroneous tagging of gameplay events, only nineteen sets of usable data were identified. These errors in data quality are that of the collection apparatus and the controlling software program, and are not believed to be systemic to GSR data collection methods. Of the nineteen cadets collected, 15 were males (age $\mu = 19.8$, $\sigma = 1.15$) and 4 were females (age $\mu = 19.25$, $\sigma = 0.96$). Participants reported intermediate (58%) or basic (42%) skill with computer games, with none claiming mastery.

The physiological data was collected using a BIOPAC MP 150 system at a 500 Hz sampling rate. This rate allows for the capture of individual heart beats, and meets requirements for GSR analysis. Each participants' signal is preprocessed for areas of interest. These data points, in order, are taken before, after, and halfway between each system interaction. These samples are sixteen seconds (8000 samples) in duration. Sixteen seconds is sufficient time to extract an instantaneous Heart Rate Variability point, and to perform a power analysis in the GSR signal.

The ECG signal has had the following features extracted: the heart rate between the closest two heartbeats to the event, and the averaged heart rate over the interval. The GSR signal, which responds slower to change, has had the following features extracted: the mean, standard deviation, and energy within the interval. All feature extraction in this paper has been performed with the idea of a real-time adaptive ITS in mind, and represents signals that can be communicated to real-time algorithms to determine whether an intervention is required. This is intended to be used in the Generalized Intelligent Framework for Tutoring (GIFT) [10] system, which uses both real-time sensor and performance data to drive personalized instructional intervention.



Fig. 1. Areas of analysis interest

ECG Signal. The ECG signal is processed for real-time QRS detection in accordance with original work on the subject [11]. The signal passes through a slightly improved second-order band pass filter. It then has the derivative taken, is squared, moving window integrated (MWI), and thresholded for heartbeat detection, shown below:

- Filter Response: $\frac{s * w_0}{s^2 + s * \frac{w_0}{Q} + w_0^2}$ (with a center frequency of 5 and a Q value of 4)
- $\frac{d}{dx} = y(nT) = \frac{1}{8} * T[-x(nT - 2T) - 2x(nT - T) + 2x(nT + T) + x(nT + 2T)]$
- Squaring: $y(nT) = [x(nT)]^2$
- MWI: $y(nT) = \left(\frac{1}{N}\right) * [x(nT - (n - 1)T) + x(nT - (N - 2)T) + \dots + x(nT)]$
(N is 30 samples, or a 3.6 millisecond delay for this work)

GSR Signal. The fundamental GSR data item of interest within the window is the change in response to stimulus. As such, the features that have been extracted over the

window are the mean, standard deviation, and signal energy [13]. This is completed by using the steps of smoothing ($y[n] = \frac{1}{\tau}x(n) + \frac{\tau-1}{\tau}y(n-1)$), normalization ($s(t) = \frac{s(t)-\mu_s(t)}{\sigma_s(t)}$), and second difference energy ($\sqrt{\int_t \frac{d^2}{dt^2}(s(t))}$).

3 Results

The post-processed set of ECG and GSR data was used for statistical analysis. Both within-subject and between-subject tests were run looking for statistically reliable differences in the calculated metrics across treatments. The variability in scenario manipulations is hypothesized to produce varying levels of arousal, which should be represented in the collected data. It is important to note that self-reported measures of engagement, via the Independent Television Commission-Sense of Presence Inventory instrument [13], and mood, via the Self-Assessment Manikin [14], were collected following the completion of each scenario, but there was minimal variance in responses between treatments. This analysis focuses on the recorded physiological data.

Analysis showed ECG data to display minimal variance over time and across scenarios, including the IDNI scenario. This can be seen when looking at the correlations between ECG metrics (Instantaneous Heartbeat Rate: [IDI vs. IDNI $r = 0.945$, $p < .0001$; IDI vs. WDNI $r = 0.871$, $p < .0001$; and IDNI vs. WDNI $r = 0.771$, $p < .0001$] and Average Heartbeat Rate [IDI vs. IDNI $r = 0.943$, $p < .0001$; IDI vs. WDNI $r = 0.904$, $p < .0001$; and IDNI vs. WDNI $r = 0.846$, $p < .0001$]). Due to this factor, the results highlight the GSR data.

A non-directional t-Test ($\alpha = .05$) was used to compare the average for all three GSR outputs to identify scenarios that produced significant differences in GSR metrics. Interestingly, results show reliable differences in all metrics when comparing the ill-defined treatments against the well-defined. When evaluating IDI against WDNI, significant differences were found for the average of the windowed-mean (IDI [M = 2.272, SD = 1.08] and WDNI [M = 2.555, SD = 1.23], $t(18) = -2.643$, $p < .025$), the average standard deviation (IDI [M = .027, SD = .019] and WDNI [M = .041, SD = .035], $t(18) = -2.323$, $p < .05$), and the average signal energy (IDI [M = 387787.3, SD = 373776.2] and WDNI [M = 261590.1, SD = 268921.3], $t(18) = 2.414$, $p < .05$). Similarly, the test looking at IDNI compared with WDNI had analogous results, with the exception of the windowed-mean, which reported a p-value just above the .05 threshold. For the two remaining measures, the average standard deviation (IDNI [M = .0234, SD = .016] WDNI [M = .0408, SD = .035], $t(18) = -2.472$, $p < .025$), and the average signal energy (IDNI [M = 373610.4, SD = 315170.5] WDNI [M = 261590.1, SD = 268921.3], $t(18) = 2.965$, $p < .01$) all show statistically reliable differences.

To examine detectable differences within individual subjects, a repeated-measure analysis of variance was conducted, which allows for the observance of data variability created by individual differences. As seen in the between-subject analysis, all three GSR metrics are reporting to be reliably different. The result shows the scenario to have a main effect on the windowed-mean, $F(3, 15) = 4.184$, $p < .05$, the average standard deviation, $F(3, 15) = 4.787$, $p < .025$, and the average signal energy, $F(3, 15) =$

3.643, $p < .05$. Upon further analysis, a pairwise comparison was used to identify the scenarios to have the largest effect on collected GSR data. Of all the compared treatments, only two pairs were reported as being significantly different. Results show individuals in the WDNI condition output had significantly higher GSR scores in the windowed-mean when compared to the IDI treatment, with a mean difference between the two scenarios of 0.283, $p = .05$. As well, participants in the WDNI condition output had significantly lower signal energy scores when compared to the IDNI treatment, with a mean difference between the two scenarios ($p = .025$).

4 Discussion and Future Work

The experiments described above are intended to examine several effects. The first of these is that ECG and GSR measurements will be able to discern a difference between well- and ill-defined scenarios. The second is that, between the ill-defined scenarios, the interjection of an interruption will have an effect of the participants' further responses.

The combination of self-paced instruction, web-based interaction, static character pictures, and text feedback failed to vary heart rate, and lowered survey response across all scenarios, but represents typical web-based instruction response. There were no reportable differences in dependent variables between the IDI and IDNI scenarios. This is an indication that the instructional event of interrupting the user had no effect on their arousal levels. It is an interesting conclusion that within the context of this training environment, this intervention is shown to be an inappropriate instructional strategy to increase engagement. The authors continue to believe that interruption is still a valid strategy among more engaging applications.

Significant differences in the windowed measurements of mean, standard deviation, and signal power were found between the well- and ill-defined scenarios. GSR is a measurement of anxiety, arousal, boredom, frustration, or stress [15]. Interaction scenarios without clear goals, such as in the ill-defined interaction context, are likely to produce lower levels of arousal. This is supported by work examining the relation between performance and stress through compensatory control of one's attention and effort [16]. This effect is observed without regard to self-reported immersion and heart rate response. While it is noted that USMA cadets may have less of a response to being interrupted, there is a clear difference when not given a specific mission.

Future work to assess real-time changes in trainee affect is motivated by the ability of the GSR signal to detect significant differences among experiences. This is encouraging when combined with the wide availability of low-cost GSR sensors. There is additional research being conducted to investigate alternate low-cost sensors, with promising results, and the data stream feature extractions created as part of this work are intended for use within GIFT [10].

Acknowledgements. The authors would like to thank two organizations for this work. We would like to thank SoarTech for the design of the data collection testbed and scenarios. We would also like to thank Dr. Michael Matthews of the Behavior Sciences and Leadership Department at the United States Military Academy at West Point.

References

1. Verdú, E., Regueras, L.M., Verdú, M.J., De Castro, J.P., Pérez, M.A.: Is Adaptive Learning Effective? A Review of the Research. In: Proceedings of the 7th WSEAS International Conference on Applied Computer & Applied Computational Science (ACACOS 2008), pp. 710–715. WSEAS Press, Stevens Point (2008)
2. Woolf, B.: Reasoning about Teaching and Learning. In: 10th Conference of Spanish Association for Artificial Intelligence (CAEPIA 2003) and the 5th Conference on Technology Transfer (TTIA 2003), San Sebastian, Spain (November 2003)
3. Bloom, B.S.: The 2 sigma problem. The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher* 13(6), 4–16 (1984)
4. Lehman, B., Matthews, M., D’Mello, S., Person, N.: What Are You Feeling? Investigating Student Affective States During Expert Human Tutoring Sessions. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 50–59. Springer, Heidelberg (2008)
5. Isen, A.M.: Positive Affect and Decision Making. In: Lewis, M., Haviland, J.M. (eds.) *Handbook of Emotions*, pp. 261–277. Guilford Press, New York (1993)
6. Woolf, B., Bursleson, W., Arroyo, I., Dragon, T., Cooper, D., Picard, R.: Affect-Aware Tutors: Recognizing and Responding to Student Affect. *International Journal of Learning Technology* 4, 129–164 (2009)
7. D’Mello, S.K., Craig, S.D., Gholson, B., Franklin, S., Picard, R.W., Graesser, A.C.: Integrating Affect Sensors in an Intelligent Tutoring System. In: *Affective Interactions: The Computer in the Affective Loop Workshop at 2005 International Conference on Intelligent User Interfaces*, pp. 7–13. AMC Press, New York (2005)
8. Handri, S., Yajima, K., Nomura, S., Ogawa, N., Kurosawa, Y., Fukumura, Y.: Evaluation of Student’s Physiological Response Towards E-Learning Courses Material by Using GSR Sensor. In: 2010 IEEE/ACIS 9th International Conference on Computer and Information Science (ICIS), pp. 805–810. IEEE, Los Alamitos (2010)
9. Goldberg, B.S., Sottolare, R.A., Brawner, K.W., Holden, H.K.: Predicting Learner Engagement during Well-Defined and Ill-Defined Computer-Based Intercultural Interactions. In: D’Mello, S., Graesser, A., Schuller, B., Martin, J.-C. (eds.) *ACII 2011, Part I*. LNCS, vol. 6974, pp. 538–547. Springer, Heidelberg (2011)
10. Sottolare, R., Holden, H., Brawner, K., Goldberg, B.: Challenges and Emerging Concepts in the Development of Adaptive, Computer-based Tutoring Systems for Team Training. In: *IITSEC Emerging Concepts Group Proceedings*, Orlando, Florida (2011)
11. Pan, J., Tompkins, W.J.: A real-time QRS detection algorithm. *IEEE Trans. Biomed. Eng.* BME 32(3), 230–236 (1985)
12. Grundlehner, B., Brown, L., Penders, J., Gyselinckx, B.: The design and analysis of real-time, continuous arousal monitor. In: *Proceedings of the 6th International Workshop on Wearable and Implantable Body Sensor Networks* (2009)
13. Lessiter, J., Freeman, J., Keogh, E., Davidoff, J.: A Cross-Media Presence Questionnaire: The ITC-Sense of Presence Inventory. *Presence* 10(3), 282–297 (2001)
14. Bradley, M.M., Lang, P.J.: Measuring emotion: The Self-Assessment Manikin and the Semantic Differential. *Journal of Behavior Therapy and Experimental Psychiatry* 25(1), 49–59 (1994)
15. Kapoor, A., Bursleson, W., Picard, R.W.: Automatic prediction of frustration. *International Journal of Human-Computer Studies* 65, 724–736 (2007)
16. Hockey, G.R.J.: A state control theory of adaptation to stress and individual differences in stress management. In: Hockey, G.R.J., Gaillard, A.W.K., Coles, M.G.H. (eds.) *Energetics and Human Information Processing*. Martinus Nijhoff, Dordrecht (1986)

Categorical vs. Dimensional Representations in Multimodal Affect Detection during Learning

Md. Sazzad Hussain^{1,2}, Hamed Monkaresi², and Rafael A. Calvo²

¹ National ICT Australia (NICTA), Australian Technology Park, Eveleigh 1430, Australia

² School of Electrical and Information Engineering, University of Sydney, Australia
{Sazzad.Hussain, Hamed.Monkaresi, Rafael.Calvo}@sydney.edu.au

Abstract. Learners experience a variety of emotions during learning sessions with Intelligent Tutoring Systems (ITS). The research community is building systems that are aware of these experiences, generally represented as a category or as a point in a low-dimensional space. State-of-the-art systems detect these affective states from multimodal data, in naturalistic scenarios. This paper provides evidence of how the choice of representation affects the quality of the detection system. We present a user-independent model for detecting learners' affective states from video and physiological signals using both the categorical and dimensional representations. Machine learning techniques are used for selecting the best subset of features and classifying the various degrees of emotions for both representations. We provide evidence that dimensional representation, particularly using valence, produces higher accuracy.

Keywords: Affect, multimodality, machine learning, learning interaction.

1 Introduction

It has been widely acknowledged that affective states (e.g. emotions) underpin learning, supporting both what we learn and how we go about doing it. Learners experience a host of learning-centered emotions such as confusion, boredom, engagement/flow, curiosity, interest, surprise, delight, anxiety, and frustration. These affective states are highly relevant and influential to both the processes and products of learning; many of these states are frequently experienced during tutorial sessions with both Intelligent Tutoring Systems (ITS) as well as human tutors [1-4].

Tutoring systems that are affect-sensitive aim to detect and react to learner emotions not only to improve learning but also to increase task interest and motivation [5]. It is believed that endowing ITSs with a degree of emotional intelligence will improve the computer tutor's understanding of the learner. One of the challenges is improving their affect detection accuracy. It is generally agreed that accuracy can be improved by multimodal information fusion combining signals such as facial expressions, gestures, voice and a variety of physiological ones. Using these signals, researchers enabled ITS with the ability to detect learners' affective states using categorical (e.g. confusion, frustration, etc.) [1, 3, 6] and dimensional (valence,

arousal) [7] representations. In general, the choice of emotion representation influences how these tutoring systems adapt and respond to affective states. This paper investigates both categorical and dimensional representations for automatically detecting learner affective states during interactions with AutoTutor, an ITS with conversational dialogues [8]. Features are extracted from facial video (webcam) and physiological signals to build a user-independent model. A number of studies have attempted to recognize learners' affect from facial expressions and speech [9-11]. Studies using physiological signals, especially in educational contexts, are relatively rare with some exceptions [7, 12]. This study uses features related to changes in skin color of face and head position from video. As for physiology, features related to heart activity, skin response, respiration, facial muscle activity are considered.

Several emotion theories focus on *categorical* representations, which consider discrete emotions (e.g. fear, anger, etc). Another alternative is the *dimensional* representation, where a person's affective states are represented as a point in a multi-dimensional space (e.g. valence, arousal, dominance). Aghaei Pour et al. [12] investigated that ITS feedback (positive, neutral, and negative) and learner affective states were statistically related. A more recent study by Hussain et al. [7], provided an empirical mapping of a set of discrete learning-centered affective states into a valence/arousal space. In this paper, we have used the mapping representation given in [7] to group the categorical affects in the negative (surprise, frustration, confusion, boredom) and positive (delight, curiosity, flow) co-ordinates and label them *negative* and *positive* respectively. Then we use supervised machine learning to classify *negative*, *neutral*, and *positive* from the video and physiological features. Similarly, for the dimensional representation, we also classify valence (*negative*, *neutral*, and *positive*) and compare the results with categorical representation.

The following sections explain the experiment, the computational model, and the results obtained.

2 Experiment and Data Collection

The dataset from [7] is used for the study in this paper. Learners were 20¹ healthy participants (8 males and 12 females) aged from 18 to 30 years. They were equipped with physiological sensors that monitored electrocardiogram (ECG), facial electromyogram (EMG), respiration, and galvanic skin response (GSR). The physiological signals were acquired using a BIOPAC MP150 system. Video was recorded using an ordinary webcam (Logitech Webcam Pro 9000).

During the experiment, subjects completed a 20-minute tutorial session with AutoTutor on topics in computer literacy. During this interaction, videos of the participant's face and of the computer screen were recorded. Participants made affect judgments immediately after the learning session at 10 seconds fixed intervals over the course of viewing their face and screen videos. They were asked to provide two types of judgments: (a) categorical judgments, which included learning-centered

¹ The dataset from 16 learners were used due to physiological sensor and calibration failures in four learners.

affective states, and (b) dimensional judgments consisting of valence/arousal ratings using the 3x3 affective grid.

3 Computational Framework in Matlab

Using the mapping presented in [7], the categorical labels are relabeled as *negative* and *positive* respectively (*neutral* unchanged). Dimensional labels for 1-3 degrees of valence from self-reports were relabeled as *negative*, *neutral*, and *positive* respectively. The computational framework (figure 1) for feature extraction, feature selection and classification is implemented in Matlab with the support of in-house and third party codes/toolboxes.

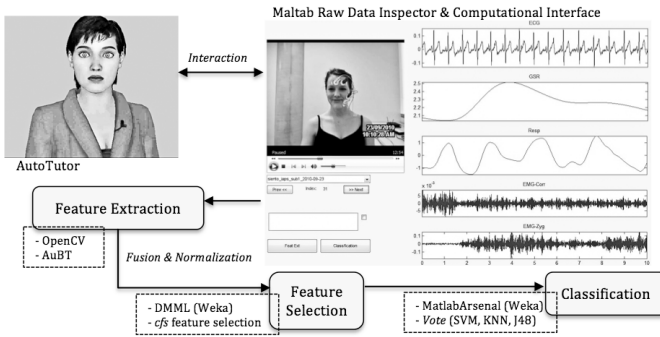


Fig. 1. Computational framework in Matlab for data visualization, feature extraction, feature selection and classification

Feature vectors were calculated using 10 seconds time window corresponding to the duration of each annotation and relabeled into *negative*, *neutral*, and *positive*. Two types of image-based features were explored: geometric and chromatic features. Five geometrical data (*x* and *y* coordinates, *width*, *height* and *area*) were derived which determined the position of the head in each frame. In addition, each frame was separated into *red*, *green* and *blue* colors in different conditions, due to movement or changing illumination sources. A total of 115 features were extracted from the video (59 from geometric and 56 from chromatic). Statistical features were extracted from the different physiological channels using the Augsburg Biosignal toolbox (AuBT) in Matlab. Some features were common for all signals (e.g. *mean*, *median*, and *standard deviation*, *range*, *ratio*, *minimum*, and *maximum*) whereas other features were related to the characteristics of the signals (e.g. *heart rate variability*, *respiration pulse*, *frequency*). A total of 214 features were extracted from the five physiological channel signals (84 from ECG, 42 from EMG, 21 from GSR, and 67 for respiration). All physiological features were considered as a single modality and both head movement and skin color features were considered as video modality. The fusion model contained all features of these two modalities. Data from individual participants were first standardized and then combined achieving a total of 2038 instances.

To reduce the dimensionality of the large number of features a Correlation based

Feature Selection (CFS) method was used for choosing the best subset of features. The CFS technique evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them [29].

We evaluated three classifiers: k-nearest neighbor (KNN), linear support vector machine (SVM), and decision trees. A Vote classifier combined classifiers with the *average probability* rule. The training and testing was performed separately with 10-fold cross validation. Self-reports produce imbalanced class distribution and for this study a high discrepant class distribution was observed for the user-independent model. The high discrepant class distribution influences classification evaluation, therefore, we applied a down sample technique, *spreadsubsample* in Weka, which produces a random subsample with a balanced class distribution. Due to down sampling, 32% data was lost from the categorical representation and 51% data was lost from the dimensional representation.

4 Results and Discussions

Classification results are presented for detecting *negative*, *neutral*, and *positive* for the categorical and dimensional (valence) representations. Firstly, we discuss the features chosen by the feature selection algorithm. For the categorical representation, chromatic and geometric features had almost similar contribution. For valence, chromatic was more dominant compared to geometric. As for physiology, ECG features were the most important for both emotion representations, especially valence. Respiration features were also noticeably important for valence. Similar trend for feature selection was found for their fusion model. GSR was not very useful. The selected features were used to obtain the classification results presented in figure 2.

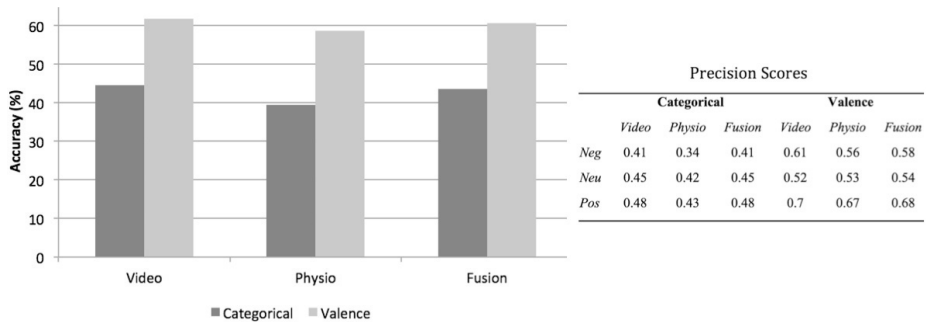


Fig. 2. Classification accuracy for detecting *negative*, *neutral* and *positive* from video, physiology, and fusion for categorical and valence representations²

² Similar trend in overall classification accuracy was observed for the same dataset without applying down sampling techniques (total 2038 instances for both emotion models).

The baseline accuracy was 33%, obtained using the ZeroR classifier. It is clear from figure 2 that the detection accuracy for the valence representation is higher than the categorical one for the two modalities (video, physiology) and their fusion. This is an indicator that the dimensional representation, in this case valence (dimensional) is very suitable for modeling learner affective states compared to the categorical representation. If we investigate the individual modalities, it is observed that the video channel (accuracy of 45% for categorical and 62% for valence) is the best modality for detecting learner affective states using both emotion models. The physiological channel performs slightly lower than the video. The fusion fails to improve the accuracy over the video channel for both representations. Physiological signals have been reported in previous studies to be more useful for detecting arousal [13]. Even though we have not included arousal from the dimensional representation for this study, we were interested to see if the fusion of the two modalities can show any improvement. We briefly present the findings for detecting three degrees (low, medium, high) of arousal. The arousal dataset was highly skewed with most of the labels appearing to be *medium arousal* and less appearing to be *high arousal*. The dataset is down sampled for this analysis, losing 73% of the data. Despite low number of instances, the classification results for the arousal dimension show good accuracy. The fusion of video and physiological features in the arousal model exhibit slightly higher accuracy (64.63%) compared to video (61.48%) and physiology (60%). This could indicate that physiological features are more useful with other modalities for arousal models in learning interactions.

5 Conclusion

We have explored multimodal features for detecting *negative*, *neutral*, and *positive* affective states using the categorical and dimensional representations during learning sessions with ITS. Machine learning techniques have been applied for selecting the best subset of features and classification. The analysis shows that learners' affective states are best detected using the dimensional representation. More importantly, this is evidence that the choice of emotion model plays important role in affect detection during learning interactions. There might be underlying reasons for such results, for example, that dimensional representations might be more natural for emotion modeling, maybe because they prevent linguistic incongruence. This is part of a longstanding debate to which this paper contributes additional evidence.

The video channel achieved the highest detection performance for both representations. However, multimodal features (e.g. physiology) still need to be considered especially for arousal models. The accuracy for detecting *negative*, *neutral*, and *positive* for both emotion models in this study is above random but not extremely high because of the user-independent model. Improved detection accuracy could be achieved in a user-dependent model with other modalities. However, the importance of choosing the suitable emotion model is evident from this paper and should be considered for building better ITS systems.

Acknowledgements. Sazzad Hussain was supported by Endeavour Award and National ICT Australia (NICTA)³.

References

1. D’Mello, S., Craig, S., Witherspoon, A., Mcdaniel, B., Graesser, A.: Automatic detection of learner’s affect from conversational cues. *User Modeling and User-Adapted Interaction* 18, 45–80 (2008)
2. Csikszentmihalyi, M.: *Flow: The psychology of optimal experience*. Harper and Row, New York (1990)
3. Graesser, A., McDaniel, B., Chipman, P., Witherspoon, A., D’Mello, S., Gholson, B.: Detection of emotions during learning with AutoTutor. In: *Proceedings of the 28th Annual Meetings of the Cognitive Science Society*, pp. 285–290 (2006)
4. Craig, S., Graesser, A., Sullins, J., Gholson, B.: Affect and learning: an exploratory look into the role of affect in learning with AutoTutor. *Learning, Media and Technology* 29, 241–250 (2004)
5. Calvo, R.A., D’Mello, S.: *New Perspectives on Affect and Learning Technologies. Explorations in the Learning Sciences, Instructional Systems and Performance Technologies*, vol. 3. Springer, New York (2011)
6. Klein, J., Moon, Y., Picard, R.: This computer responds to user frustration: Theory, design, and results. *Interacting with Computers* 14, 119–140 (2002)
7. Hussain, M.S., AlZoubi, O., Calvo, R.A., D’Mello, S.K.: Affect Detection from Multichannel Physiology during Learning Sessions with AutoTutor. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011. LNCS(LNAI)*, vol. 6738, pp. 131–138. Springer, Heidelberg (2011)
8. Graesser, A.C., Chipman, P., Haynes, B.C., Olney, A.: AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education* 48, 612–618 (2005)
9. Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., Taylor, J.: Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine* 18, 32–80 (2001)
10. Polzin, T.: *Detecting Verbal and Non-verbal cues in the communication of emotion*. Unpublished Doctoral Dissertation, School of Computer Science, Carnegie Mellon University (2000)
11. Yacoob, Y., Davis, L.: Recognizing human facial expressions from long image sequences using optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18, 636–642 (1996)
12. Aghaei Pour, P., Hussain, M.S., AlZoubi, O., D’Mello, S., Calvo, R.A.: The Impact of System Feedback on Learners’ Affective and Physiological States. In: Alevin, V., Kay, J., Mostow, J. (eds.) *ITS 2010. LNCS*, vol. 6094, pp. 264–273. Springer, Heidelberg (2010)
13. Calvo, R.A., D’Mello, S.: *Affect Detection: An Interdisciplinary Review of Models, Methods, and their Applications*. *IEEE Transactions on Affective Computing* 1, 18–37 (2010)

³ NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

Cognitive Priming: Assessing the Use of Non-conscious Perception to Enhance Learner's Reasoning Ability

Pierre Chalfoun and Claude Frasson

Université de Montréal, Dept. of Computer Science and Operations Research
2920 chemin de la tour, H3T-1J8 QC, Canada
{chalfoun, frasson}@iro.umontreal.ca

Abstract. Current Intelligent Tutoring Systems (ITS) employ explicit and direct learning strategies when interacting with learners. Although these ITS use cognitive and logical models to analyze the conscious cognitive processes behind reasoning, we believe that in specific situations during knowledge acquisition, such as reasoning, unconscious cognitive processes are heavily solicited in the brain. In this paper, we will propose a complimentary and novel learning strategy to current ITS aimed at enhancing reasoning in a problem solving environment. This approach, called Cognitive Priming, is based on neural correlates of non-conscious perception. We will present two studies that have positively conditioned learners and enhanced different dimensions of their reasoning skills by employing a technique based on the science of subliminal perception. We will also present relevant cerebral data recorded throughout the studies and discuss the importance of such findings for the community.

Keywords: Cognitive priming, reasoning, problem solving, EEG, ITS.

1 Introduction

For more than twenty five years now, the aim of intelligent Tutoring Systems (ITS) has been to properly adapt learning sessions and material to the learner. Moreover, the availability, ease of use, and affordability of physiological devices have endowed current tutors with the ability to assess student's cognitive and affective states during learning [1]. Amongst the many important cognitive processes that occur during learning, properly assessing the reasoning ability of learners when acquiring knowledge is of paramount importance. Current ITS use explicit and direct learning strategies when interacting with learners and only assess conscious cognitive processes that occur during learning. However, it is now widely accepted that the unconscious mind does play a role in cognitive activity and in learning. Indeed, we believe, based on several experiments and well recorded phenomenon in recent neuroscience literature, that learning is a complex interplay between conscious and unconscious mechanisms in the brain and *exploring* and *assessing* these mechanisms is not only possible, but of great interest [2]. In general, this research is interested in exploring the domain of unconscious cognition and assess, using physiological sensors, the relevant cognitive mechanisms involved during reasoning in a problem solving environment. This paper,

more specifically, will explore and assess the possibility of enhancing learner's reasoning ability in a problem solving environment by employing a technique based on neural correlates of non-conscious perception called "cognitive priming". The idea is to project answers to a problem slightly outside the learner's conscious awareness while active thinking is taking place thus increasing the reasoning process of learners. Contrary to popular belief, a large body of work in neuroscience has put forward strong evidence that learning simple to complex information can be done without perception or complete awareness at the task at hand [3, 4].

2 Cognitive Priming

Before going further, we need to clearly establish the terminology that will be used in this paper. *Unconscious cognition* refers to the wide range of possible effects that unconscious mechanisms in the brain can have on cognitive processes such as learning and complex decision making. *Non-conscious perception* is the sub-branch of unconscious cognition that deals with all sensory-related stimuli that are processed unconsciously (e.g. images or sounds). *Subliminal perception* is a technique that transmits information without overloading the active cognitive channel by projecting a stimulus, called a *prime*, under the human conscious visual threshold. *Masked priming* is one of the most widely used technique for subliminal perception [5]. It consists in projecting for a very short time (20 to 40 ms) a stimulus (such as a word or an image) preceded and/or followed by the projection of a mask (random figures or dashes) for a few hundred milliseconds. *Cognitive priming* is a special case of subliminal perception where the stimulus used (answer to a question for example) is aimed toward enhancing cognitive processes such as reasoning or decision making towards the goal of better knowledge acquisition. The essence of our work is inspired by a framework in the neuroscience of non-conscious perception and more specifically on two landmark papers in *Science* and *Nature* where robust subliminal priming methodologies showed that that genuinely invisible primes could influence processing at a semantic level [6, 7]. In light of all these findings, we have carefully designed subliminal primes in the form of images containing cognitively helpful information to the learner for a problem solving environment. We will now review some relevant work in areas close to our research interests before presenting our experiments.

3 Related Work

To the best of our knowledge, we found no similar work in the ITS/AIED community that uses and assesses cognitive priming to attempt to enhance reasoning in a problem solving environment. The most relevant work however regarding our research has been done by Lowery and colleagues who demonstrated that subliminal primes can increase performance on midterm exams compared to neutral primes and that subliminal priming may have long-term effects on real-world behavior [8]. In HCI, one of the early works regarding subliminal cues for task-supported operation was the text editor program of Wallace [9] where Wallace and colleagues found that the frequency

at which subjects demanded help was much lower when the required information was presented in subliminal matter. The Memory Glasses by DeVaul and colleagues [3] used wearable glasses that projects subliminal cues as a strategy for just-in time memory support. The objective was to investigate the effect of various subliminal cues (correct and misleading) on retention in a word-face learning paradigm and compare recall performance. Another use of priming for memory support can be found in the thesis of Shutte [10] where the author assessed the effects of brief subliminal primes on memory retention during an interference task. Although the results of these priming seemed very encouraging, the author cautions HCI designers that misusing subliminal priming that can lead to critical disruptions of ongoing tasks. After briefly reviewing the relevant work, we will now present the studies conducted where cognitive priming was used in two distinct experimental setups.

4 Empirical Studies Conducted

The learning task set in both of these experiments is to teach the construction of an odd magic square of any order with the use of 3 cumulative visual tricks (T1 to T3) requiring neither a calculator nor complex mental arithmetic operations. For a magic square of 5x5, the three tricks show how to properly fill the boxes in the square with numbers from 1 to 25. Tricks are cumulative and a learner must use T1 to complete T2 and T2 to complete T3. Thus, T3 is more difficult to understand than T1 because it requires learners to have understood T1 and T2 respectively. (see [11] for details).

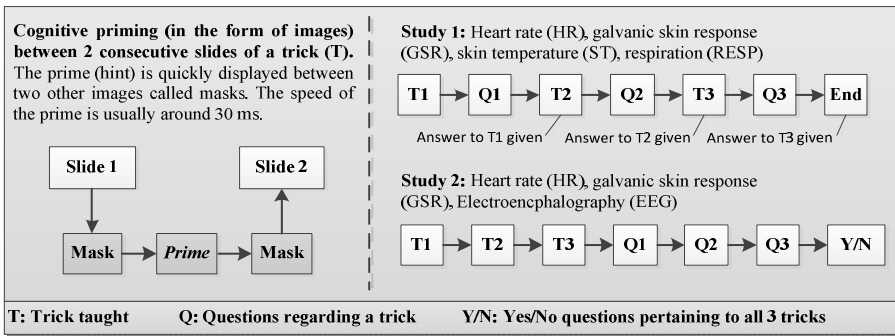


Fig. 1. Overall design for study 1 and 2

The main objective of the studies was to learn the tricks whilst making the fewest possible mistakes. Learning performance metrics were our main criterions for evaluating learners' reasoning performance. The subliminal stimulus, thresholds, prime and masks were carefully chosen following the neural bases of subliminal priming [5] as well as accepted brain methodologies [12].

Study 1: using positive hints as cognitive primes. In this study, learners are shown multiple examples of each trick without explaining how the trick works. We present

for each trick various Power Point slides of before and after states of the magic square for various numbers. Instead of giving away the answer, we ask the subjects to *deduce* the rule by themselves. The control group will try to deduce each trick without subliminal stimuli and conversely the condition group will be conditioned by a subliminal tutor that will prime learners. The primes are in the form of arrows pointing to the proper location on the square where the next number should appear. We will then compare performances, trick completion time and question completion time. Learning takes place in a 3D game-like environment called MOCAS. The interactions between the avatar's learner and the pedagogical agents are done via mouse clicks. The learners are instructed to continue once they are convinced they have discovered the inner working of each trick and cannot go back. They are asked a series of questions by virtual avatars related to the last trick learned. After answering all questions related to each learned trick, the answer to the current trick is revealed before learning the next trick. Physiological signals of the learners were also monitored in real-time and saved for off-line analysis. The signals were heart rate, galvanic skin response, respiration and skin temperature. A total of 30 participants were recruited for the experiment; they were assigned either to the experimental condition (N=15; PositivePrime) or to the control condition (N=15; Control).

Results. Our hypothesis in this study was that cognitive primes were to have positive effects on reasoning, and consequently on performance and mistakes made. The results obtained show that PositivePrime's overall performance was statistically different ($p=.023$, $\alpha=0.05$) than the control group and 2.7 times more efficient on average (44% less mistakes overall with the presence of the subliminal module). Furthermore, subliminal priming at specific intervals seem to significantly reduce the time spend on each question. Indeed, time spent on each question by primed learners is reduced by an overall factor of 1.3 (Single factor ANOVA $p = .023$, $\alpha = 0.05$). It is important to note that NO subliminal priming is done during the questions. All the priming is done during the tricks taught. The answer to the questions is not projected subliminally when the question is asked. We believe these encouraging results can be explained by the fact that the subliminal primes are goal-relevant to the cognitive task at hand and might have acted as a "catalyst" for quickly converging to a solution as observed by previous studies [13]. However, we wanted to verify the validity of this priming strategy by conducting a second study where we introduced primes (called miscues) that were designed to throw off learners in order to compare results with positive primes used here.

Study 2: using positive and misleading hints as cognitive primes. In this study, we are teaching the same lesson (learning how to construct a magic square in 3 simple tricks) but within a 2D system that looks very similar to an online exam session. Although learners still had to infer their own solutions and correctly figure out the algorithm used in each trick, the solution to each trick was never presented (see fig. 1). Thus, each learner had to *induce* the rules and construct a mental model of the overall solution. Furthermore, learners reported how they figured each trick by choosing between the following: I deduced the trick by intuition, logic, a little of both (variable Trick answer type). A fixed time limit of 45 seconds for the questions was imposed.

Failing to give an answer within the allowed time was considered a mistake. Learners also reported how they answered each question by choosing between the following: I answered the question by guessing the answer, by intuition or by logical deduction (variable Question answer type). After giving their answer, a green check or a red cross appears for 2 seconds indicating to the learner if they made a correct or wrong choice respectively. The main intent for these changes is to associate relevant brain states with *how* learners reasoned and resolved problems. Physiological signals of the learners were also monitored in real-time and saved for off-line analysis. The signals were EEG (brainwaves), heart rate and galvanic skin response. A total of 43 participants were recruited for the experiment; they were assigned either to the answer group (N=14; Answer_cues), to the misleading group (N=14; Miscue) or to the control group (N=15; Control). Each learner was compensated with 10\$.

Results. Our hypothesis in this study was that cognitive positive primes only, and not miscues, were to have positive effects on reasoning, and consequently on performance and mistakes made. We examined results related to performance (number of mistakes) with regards to the way learning occurred (Trick answer type), the way learners answered questions (Question answer type) and the group (Answer_cues, Miscue, Control). Significant effects from a four way cross-tabulation analysis were only found for the variables Trick answer type* group with regards to the number of mistakes with the following combinations: Logic*Answer_cues ($p=.002$, $\alpha = 0.05$, chi-square = 16.949), A little of both*Answer_cues ($p=.048$, $\alpha = 0.05$, chi-square = 9.117). Results seem to indicate that only Answer_cues, and not miscues, do significantly influence logical reasoning and decision making when learning a trick logically. From the EEG data we were interested in investigating changes in two metrics that have previously been reported as relevant indicators of insightful problem solving (40Hz right asymmetry) [14] and complex arithmetic processing (Beta2 left asymmetry) [15]. We observed that the asymmetry values for the 40Hz ($p = .003$, $\alpha = 0.05$) and Beta2 ($p = .04$, $\alpha = 0.05$) in the Answer_cues group are significantly different than the Miscue group for the third and most difficult trick. The Answer_cues group seems to shift their attention from a complex arithmetic process (Beta2 left asymmetry decrease) toward an “insightful” problem solving strategy (40Hz right asymmetry increase), thus involving the right side of the brain, known to be an important actor in insightful problem solving. The combination of these two metrics could indeed be an interesting indicator of a change in the reasoning strategy from complex arithmetics to insightful reasoning during problem-solving.

5 Conclusion

We have presented in this paper a novel approach to enhance reasoning and learning in a problem solving environment with cognitive priming. This technique aims at enhancing unconscious processes involved in learning, namely reasoning, by projecting information under the visual threshold of learners without neither overloading the current cognitive channel nor disturbing the learning session. Furthermore, cerebral recording have shown that it might be possible to assess not only classical reasoning

but also intuitive reasoning. Expected benefits from this technique are two-fold. First, the reasoning ability of learners is strengthened by the added information outside of conscious awareness. Second, the supplementary cognitive data does not hamper or interrupt active cognitive processes. Many interesting challenges remain however for future work such as ethical aspects and usage of this technique in a more complex scenario where *deep learning* might occur. We are currently working on all these issues and hope to present relevant findings to the community in the near future.

References

1. Nkambou, R., Bourdeau, J., Mizoguchi, R. (eds.): *Advances in Intelligent Tutoring Systems*. SCI, vol. 308, p. 2010. Springer, Heidelberg (2010)
2. Hassin, R.R., Uleman, J.S., Bargh, J.A.: *The new unconsciousness*, p. 575. Oxford University Press, New York (2005)
3. DeVaul, R.W., Pentland, A., Corey, V.R.: *The Memory Glasses: Subliminal vs. Overt Memory Support with Imperfect Information*. In: *IEEE International Symposium on Wearable Computers 2003*, pp. 146–153. IEEE Computer Society, New York (2003)
4. Nunez, J.P., Vicente, F.D.: *Unconscious learning. Conditioning to subliminal visual stimuli*. *The Spanish Journal of Psychology* 7(1), 15 (2004)
5. Del Cul, A., Baillet, S., Dehaene, S.: *Brain Dynamics Underlying the Nonlinear Threshold for Access to Consciousness*. *PLoS Biology* 5(10), 2408–2423 (2007)
6. Dehaene, S., Naccache, L., Le Clec'h, G., Koechlin, E., Mueller, M., Dehaene-Lambertz, G., van de Moortele, P.F., Le Bihan, D.: *Imaging unconscious semantic priming*. *Nature* 395, 597–600 (1998)
7. Greenwald, A.D., Draine, S.C., Abrams, R.L.: *Three cognitive markers of unconscious semantic activation*. *Science* 273, 1699–1702 (1996)
8. Lowery, B.S., Eisenberger, N.I., Hardin, C.D., Sinclair, S.: *Long-term effect of subliminal priming on academic performance*. *Basic and Applied Social Psychology* 29(2), 151–157 (2007)
9. Wallace, F.L., Flaherty, J.M., Knezek, G.A.: *The Effect of Subliminal HELP Presentations on Learning a Text Editor*. *Information Processing and Management* 27(2/3), 7 (1991)
10. Schutte, P.C.: *Assessing the Effects of Momentary Priming on Memory Retention During an Interference Task*. In: *Computer Science 2005*, p. 103. Virginia Commonwealth University, Virginia (2005)
11. Chalfoun, P., Frasson, C.: *Subliminal cues while teaching: HCI technique for enhanced learning*. *Advances in Human Computer Interaction: Special Issue on Subliminal Communication in Human-Computer Interaction* (2011) (1)
12. Pessiglione, M., Petrovic, P., Daunizeau, J., Palminteri, S., Dolan, R.J., Frith, C.D.: *Subliminal Instrumental Conditioning Demonstrated in the Human Brain*. *Neuron* 59, 561–567 (2008)
13. Strahan, E.J., Spencer, S.J., Zanna, M.P.: *Subliminal priming and persuasion: Striking while the iron is hot*. *Journal of Experimental Social Psychology* 6(38), 13 (2002)
14. Sandkühler, S., Bhattacharya, J.: *Deconstructing insight: EEG correlates of Insightful Problem Solving*. *PLOS One* 3(1) (2008)
15. Hyungkyu, K., Jangsik, C., Eunjung, L.: *EEG Asymmetry Analysis of the Left and Right Brain Activities During Simple versus Complex Arithmetic Learning*. *Journal of Neurotherapy* 13 (2009)

Math Learning Environment with Game-Like Elements: An Incremental Approach for Enhancing Student Engagement and Learning Effectiveness

Dovan Rai and Joseph E. Beck

Computer Science Department, Worcester Polytechnic Institute, USA
{dovan, josephbeck}@wpi.edu

Abstract. Educational games intend to make learning more enjoyable, but carry a potential cost of compromising learning efficiency by consuming both instructional time and student cognitive resources. Therefore, instead of creating an educational game, we create a learning environment with *game-like elements*, the aspects of games that are engaging, but that hopefully do not negatively impact the learning effectiveness of the system. We present an approach of incrementally making a tutor more game-like, and present an evaluation to estimate the effect of game-like elements in terms of their benefits such as enhancing engagement and learning as well as their costs such as distraction and working memory overload. We developed four different versions of a math tutor with different degrees of game-likeness, such as adding narrative and visual feedback. The four systems were pedagogically equivalent consisting of 27 main tutor problems with the same hint and bug messages and mini tutorial lessons. Based on a study with 252 students, we found that students reported more satisfaction with a more “game-like” tutor. Students also took an 11-item pretest and posttest and the students with the most game-like tutor have significant learning gain but there is no reliable difference between the different versions of the tutor.

Keywords: game-like elements, educational games, intelligent tutors, intelligent games, engagement, cognitive overload.

1 Introduction

Intelligent tutors, which are primarily concerned with cognitive aspects of learning, use adaptive, individualized tutoring to students and have shown evidence to improve learning significantly [1]. On the other hand, education researchers have also been interested in computer games due to their immense popularity and affordance of new kinds of interactions. Games can not only enhance the affective aspects of learning, but can also hold the potential to improve cognitive outcomes of learning as well. But despite this intuitive appeal of educational games, there is not enough empirical evidence on effectiveness of educational games [2,3]. There is a relative scarcity of evidence directly comparing the educational effectiveness of educational games vs.

computer tutors; however, comparisons have found an advantage for traditional tutoring approaches over educational games [4,5]. However, tutors, have had difficulties in maintaining students' interest for long periods of time, which limit their use to generate long-term learning [5].

Given these complementary benefits, there has been considerable effort to combine these two fields. However, fulfilling this vision is a challenging design goal and difficult to instantiate. Therefore, we are taking a conservative, incremental research path. Instead of completely integrating educational content into a game framework, we are analyzing and inspecting game-like elements, elements within the game that are engaging, in terms of their pedagogical impact and then integrate the beneficial ones into the tutor. In this paper, we are trying to create a theoretical and experimental framework for assessing these game-like elements.

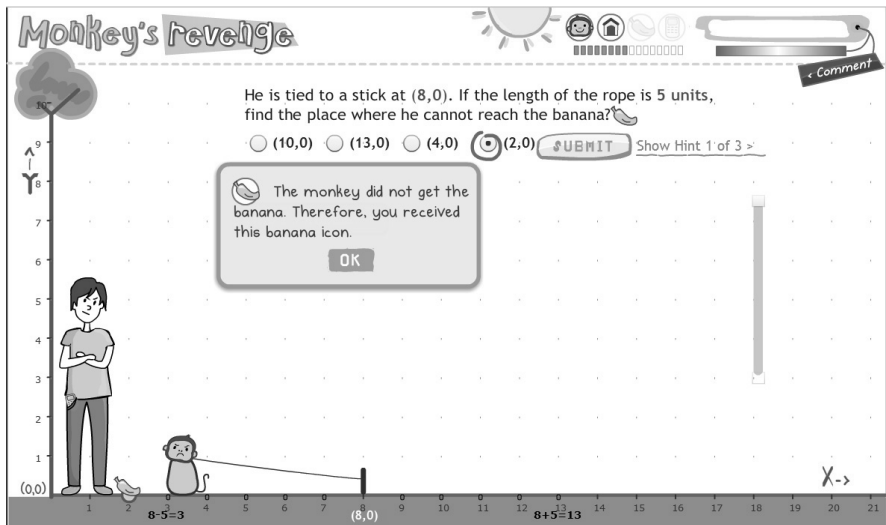


Fig. 1. Screenshot of Monkey's Revenge

Monkey's Revenge (see Figure 1) is a coordinate geometry math-learning environment with game-like elements. The system is a series of 8th grade coordinate geometry problems wrapped in a visual narrative. Students have to help story characters solve the problems in order to move the story forward. Similar to classic computer tutors, students receive hints and bug messages when they encounter difficulties. In the story, a boy, Mike is thrown out of class for playing a game on his cell phone. The day is going to be a strange one as his world is now mapped into coordinates. As a warm-up problem, students have to find out Mike's height in coordinate units based on the (x,y) coordinates of his head. Mike finds a monkey and, being lonely, Mike wants to befriend him and the student helps Mike by giving a name to the monkey. Later Mike builds a house for the monkey, but the monkey is not eager to become domesticated and destroys the house, steals Mike's phone and runs away. The boy tries to get back his phone by throwing balls at the monkey. To move the story

forward, the students have to solve coordinate problems like calculating distance between the boy and the monkey, the slope of the roof and walls of the house, finding points where the monkey tied to a rope cannot reach bananas and finally figure out slopes, intercepts and equation of the line of the path of the ball. The math content gets more advanced as the student progresses within the story.

2 Theoretical Framework

Using games in education has been a topic of great interest and controversy among education researchers generating a growing number of ardent proponents [7,8,22] as well as many unconvinced skeptics [3,9,19]. When we add game-like elements to a tutor, we expect to have a more engaging environment; however, we still do not know how learning changes in the process. Games can improve learning by enhancing affect and motivation and through cognitive support and pedagogical affordances. On the other hand, games can also add various constraints for learning, which are discussed in the following paragraphs.

Practical constraint: Time overload

Game elements consume time that could have been used for instruction. Game environments can be complex and require students to spend time to learn them first.

Intrinsic constraint: Working memory overload

Although details and novelty in a game environment and complexity of the game rules can add excitement and entertainment value in games, they can also overwhelm learners in the case of learning games due to additional working memory load [13] of the learning content. Since non-educational games have a sole purpose of entertaining, they can afford to play with novelty, details and complexity to maximize fun. However, learning games have to deliver learning content, and thus have to restrain on the amount of additional details and complexity.

Goal constraint: Aligning cognitive and affective outcomes

While tutoring systems are primarily concerned with cognitive outcomes and computer games are about maximizing fun, educational games have the objective of enhancing both cognitive and affective outcomes. These two goals are not necessarily in opposition. In fact, they can reinforce each other. But these two outcomes are not always aligned and sometimes affective and cognitive strategies may be in conflict with each other [17]. As mentioned in the previous section, the elements, which enhance excitement and fun, can overwhelm and overload learners. Similarly, the tutorial practices may seem pedantic and diminish students' sense of choice and control and reduce fun [4].

Design constraint: Integration of learning content and game attributes

It is more likely that games will be instructionally effective if the specific characteristics of the game (e.g., setting, player roles and activities, rules, etc.) overlap with

specific instructional objectives. This overlap must be consciously structured on the basis of a thorough analysis of the reasons for the instruction and the instructional objectives to be met [9]. When integration of content and game attributes is unintuitive, it can make learning hard and, when the integration is superficial, it may only add extrinsic motivation hindering intrinsic motivation.

Game-Like Elements

We are not trying to generate formal definitions of games or game elements, but rather we are looking into understanding the properties of game-like elements, which we define as the engaging and interactive aspects of games. Specifically, we are looking into game-like elements such as narrative, immediate visual feedback, visual representation, collecting, sensory stimuli, etc. Even though the game-like elements are defined based on their engaging nature, these elements can have significant pedagogical impact in both positive and negative ways. Our goal is to assess these elements in terms of their pedagogical efficacy and to select and integrate those ones that can be beneficial pedagogically, or at least not hurt the learning.

As we incrementally add game-like elements into a tutor, we may expect to generally have increased fun. But given the complicated relation of games with learning as discussed in the previous section, we do not know how learning changes during the process. We have plotted three plausible tradeoff curves of making tutor more game-like in Figure 2.

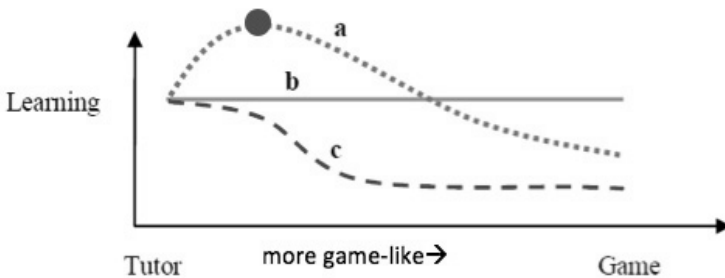


Fig. 2. Three possible tradeoff curves for making tutors more like games

- a. Some game-like elements can be pedagogically beneficial. For example, narrative can enhance learning by adding meaningful and interesting context to the learning content. But, there can be a tradeoff that reduces the benefit after some point. Once the narrative gets too elaborate and complex, it may make learning process complicated and confusing instead.
- b. Some game-like elements may be orthogonal to learning content and may not interfere with, or directly benefit, learning.
- c. Some game-like elements can hurt learning. For example: unguided exploration and pedagogically meaningful choices can leave students confused and possibly making suboptimal decisions.

We want to find the optimal point where the addition of game-like elements maximizes learning. This graph is a simplified representation of the possibilities as it is conceivable that game-like elements could synergize and enhance the effects, or interfere with each other and reduce their individual effects.

Game-Like Elements in Monkey's Revenge

We carefully picked the game-like elements that we thought to be relevant and cognitively supportive to our content.

Embedding domain in a context

Authentic activities: Research on authentic learning has suggested that learning is more efficient and effective when it is embedded in realistic and relevant contexts [12]. Fortunately, our domain of interest, coordinate geometry, has many concrete applications. We tried to incorporate those concrete activities, such as calculating slope of the roof of a house.

Narrative: entertains and engages learners and gives a meaningful context for solving problems. Furthermore, if we use a coherent story, the initial story context can be reused for multiple problems, thus saving effort and cognitive load required reading context for each new word problem, particularly when compared to traditional word problems where the problems tend to have disjoint context.

Visual affordances

Visual problem representation: Graphics not only add appeal but they can help develop mental models, thus reducing the burden on working memory [14]. We used very simple and minimalist visual representation so as not to interfere with the coordinate graph itself. As the problems get harder, they tend to be more abstract and it becomes harder and unintuitive to have concrete representations.

Immediate visual feedback: We have used immediate visual feedback for student responses to serve both engagement and learning objectives. Immediate visual feedback makes the interface more interactive, giving users sense of control and reinforcement. When the feedback is appealing and interesting, it adds to sensory stimuli. In addition to providing positive reinforcement on correct responses, visual feedback on incorrect responses provides students with information about the magnitude of the error and how it relates to the correct solution [7].

Other game-like elements

Collection: Students can collect badges after each level as they master a sub-skill.

Building: Students have to solve different problems to build a house. Using various sub-skills to create a single structure, students can see how different mathematical concepts can be integrated within a single entity.

Personalization: Students can name the monkey. Though this seems a small addition on the designer's part, but students were very excited about this feature.

3 Methodology

Our approach is to assess each individual game-like element's effects on learning and engagement through controlled experiments. But due to the limitation of the number of students we were able to get for the study, we could not test all combinations of game-like elements. Therefore, we focused on the two elements we thought would have the most impact: narrative and immediate visual feedback. We created four different versions of Monkey's Revenge with different combinations of game-like elements. All versions had same 27 math problems in the same sequence. Students also get the same hints and bug messages and two mini tutorial lessons, and the pedagogical help was identical across conditions. By making all the tutors pedagogically equivalent and changing one individual game-like element at a time, we are just looking at the affective and pedagogical impact of the particular individual game-like element.

Table 1. Four experimental tutor versions with different degree of game-likeness

Tutor Version	Game like elements		
	Immediate visual feedback	Narrative	Other game-like elements
A: Monkey's revenge	Yes	Yes	Yes
B: Monkey's Revenge without visual feedback	No	Yes	Yes
C: Monkey's Revenge without narrative	Yes	No	Yes
D: Basic tutor	No	No	No

Participants

A total of 252 middle school (12-14 year olds) students from four urban schools of the United States participated in this study. This intervention was designed as a homework assignment. Unfortunately, most teachers chose to use it as a within-class activity, and only 45 students worked on it as homework. Students were randomly assigned to the four groups, where the randomization was within each class.

Data Collection

Within the tutor, we collected survey questions, logged performance data and also administered pretest and posttest. We created two parallel forms of our test, A and B, and randomly assigned each student one form as the pretest and the other form as the posttest. In this way the pre- and the post-test were equally difficult for each

condition on average, and we can compute learning gains fairly. We could not use the same form of the test for pre- and post-test, since prior experience showed students would not take the posttest seriously since they recalled seeing the questions.

4 Results

Since the exercise took 80 minutes to complete on average, which is longer than a regular class period, only 118 students were able to complete the exercise.

Cognitive and Time Overload

The mean correct responses among the experimental groups are almost the same, with condition A doing a little better than groups C and D, even though it had a lower pretest score. So, we are assuming that pictures and story might not have added difficulty, at least for solving the problems that students had prior knowledge on. Students in tutor version A spent 5 minutes more on narrative which is a very small fraction of the total time spent on the exercise.

Table 2. Students' performance across experimental conditions (means and 95% CI)

Tutor version	Pretest percent correct	Problems correct in the tutor (max=27)	Minutes spent on narrative and instruction
A (N=35)	68%	20.3±1.1	10
B (N=26)	64%	19.8±2	13
C (N=27)	70%	18.6±1.2	9
D (N=31)	74%	18.5±1.5	5

Liking and Satisfaction

We asked the students survey questions with a 5 point Likert scale from “strongly disagree”(1) to “strongly agree”(5).

“I liked this tutor.”; “This tutor is fun.”; “This tutor helped me learn.”; “This is better than the computer math programs I have used before.”

From Table 3, we found a gradient across increasing levels of game-likeness where liking the tutor increases as the tutor becomes more game-like. We had received this same trend in our previous study [21]. Narrative seemed to be more effective as a game-like element than immediate feedback. However, statistically, the three groups with game-like elements are similar to each other and different from “Basic tutor”. Based on students' rating of the tutor and game-like elements, we can conclude that adding game-like elements increased students' liking and satisfaction with the tutor relative to the basic tutor ($p < 0.01$). A different study [20] with online casual games had found that music and sound effect had no effect on game play time duration while addition of visual animation made them play longer.

Table 3. Students' survey response across experimental conditions (means and 95% CI)

Tutor version	Like tutor	Had fun	Tutor helped	Better than other programs
A (N=34)	4.0±0.3	4.1±0.4	3.9±0.3	3.9±0.3
B (N=25)	3.9±0.4	3.9±0.4	3.6±0.4	3.7±0.4
C (N=27)	3.6±0.5	3.3±0.5	3.2±0.5	3.8±0.5
D (N=28)	3.0±0.5	3.0±0.5	3.1±0.5	3.4±0.5

We also collected open feedback from the students to get a qualitative assessment of the tutor. The following is a sample of students' open comment feedbacks:

"I think that overall this is a very interesting and useful tool for kids who are learning coordinate geometry. it is more interesting than online math tools i have used in the past. however, the hints should be more to the point instead of restating information given in the question. :)"

"I think that the storyline added a bit of fun to normal boring Math. I liked this program and i hope to see it again in the future. :)"

"I like the fact that it was a walk though process with icons to tell us what to do. The story was a little distracting but it made solving the problems fun."

"Get rid of the stupid animation we are algebra students not 4th graders.... u need to focus on the math and not the stupid animation"

Learning Gain

We created two sets of 11-item questionnaire (3 multiple choice and 8 open response). The two sets were balanced in terms of problem difficulty.

Table 4. Students' gain across experimental conditions (means and 95% CI)

Tutor version	Pretest percent correct	Learning gain
A (N=31)	66%	10% ± 9%
B (N=17)	69%	5% ± 7%
C (N=23)	70%	7% ± 8%
D (N=23)	74%	3% ± 7%

We were not able to find any conclusive results or patterns in students' learning gains. The large standard error suggests students were not taking the test seriously, that the test was not long enough to estimate student learning, or some combination of both. One conclusion is that only tutor version A had learning gains reliably different than 0, and none of the tutor versions reliably differed from each other.

We also observed individual test questions and the learning gains from the open response questions.

Table 5. Students' gain across pretest items (means)

Extra tutoring	Yes		No				Yes	No
	3	4	6	7	8	9	10	11
Pretest percent correct	47%	51%	76%	71%	70%	88%	41%	64%
Learning gain	13%	18%	5%	10%	-1%	6%	14%	5%

We had also incorporated two mini lessons within the tutor. If students made certain number of wrong attempts or asked for certain number of hints, the tutor assumes that the student has difficulty in the skill and then takes her through a tutorial lesson where she goes through different screens solving smaller problem steps at a time. This is very similar to scaffolding used in intelligent tutors. Those two lessons correspond to the pre and posttest items 3,4 and 10. Coincidentally, these same problems have highest gain, which suggests that the extra tutoring was effective. But the same items also have lowest pre test score (we had intentionally designed the mini lessons for harder problems), which also leaves a possibility that it is more of regression towards the mean or simply having more room for students to demonstrate growth.

5 Future Work and Conclusions

We created four pedagogically equivalent versions of a math learning environment with varying degree of game-likeness. We found that students' showed more liking of the tutor version with game-like elements. Narrative was possibly more effective as a game-like element than immediate feedback. Students with the most game-like tutor had significant learning gain but we were not able to find any differences among the four versions of the tutor. However, students gained significantly in the problems where they received extra tutoring which suggests that adding more tutorial features in a game-like environment leads to higher learning.

As a future work, we would like to replicate this study with a longer intervention spanning over multiple days. Since the tutor versions are pedagogically equivalent in terms of hints and tutorials, it might be hard to find difference in learning gain if the time is fixed. If we make the time unfixed as in a homework setting, we expect a scenario where students with the more game-like version of the tutor find it more engaging and work longer and finally have a higher learning gain.

We have created an iterative experimental framework of assessing each individual game-like element in terms of its affective and pedagogical impact so that we can find the optimal point of learning. Based on this study, we conclude that game-like features such as narrative and immediate visual feedback make students more receptive of the tutor and adding extra tutoring increases learning gain.

Acknowledgements. We would like to thank ASSISTment team in Worcester Polytechnic Institute for helping with setting up the experiment. The first author has been supported by International Fulbright Science and Technology Phd Scholarship.

References

1. Koedinger, K.R., Corbett, A.: Cognitive tutors: Technology bringing learning science to the classroom. In: *The Cambridge Handbook of the Learning Sciences*, pp. 61–78. Cambridge University Press (2006)
2. O’Neil, H., Wainess, R., Baker, E.: Classification of learning outcomes: Evidence from the computer games literature. *The Curriculum Journal* 16(4), 455–474 (2005)
3. Sitzmann, T.: *A Meta-Analytic Examination of the Instructional Effectiveness of Computer-Based Simulation Games*. Personnel Psychology (2011)
4. Easterday, M.W., Alevan, V., Scheines, R., Carver, S.M.: Using Tutors to Improve Educational Games. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011. LNCS(LNAI)*, vol. 6738, pp. 63–71. Springer, Heidelberg (2011)
5. Jackson, G.T., McNamara, D.S.: Motivational impacts of a game-based intelligent tutoring system. In: Murray, R.C., McCarthy, P.M. (eds.) *Proceedings of the 24th International Florida Artificial Intelligence Research Society (FLAIRS) Conference*, pp. 519–524. AAAI Press, Menlo Park (2011)
6. Alevan, V., Myers, E., Easterday, M., Ogan, A.: Toward a framework for the analysis and design of educational games. In: *Third IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning*, pp. 69–76 (2010)
7. Malone, T.W., Lepper, M.R.: Making learning fun: A taxonomy of intrinsic motivations for learning. *Aptitude, Learning and Instruction* 3, 223–253 (1987)
8. Gee, J.P.: *What Video Games Have to Teach Us About Learning and Literacy*. Palgrave/Macmillan, New York (2003)
9. Hays, R.T.: *The effectiveness of instructional games: A literature review and discussion*. Naval Air Warfare Center Training Systems Division, Orlando (2005)
10. Wilson, K.A., Bedwell, W.L., Lazzara, E.H., Salas, E., Burke, S.C., Estock, J.L., Orvis, K.L., Conkey, C.: Relationships between game attributes and learning outcomes: Review and research proposals. *Simulation & Gaming* 40(2), 217–266 (2008)
11. Garris, R., Ahlers, R., Driskell, J.E.: Games, Motivation and learning: A research and practice model. *Simulation & Gaming* 33(4), 441–467 (2002)
12. Shaffer, D.W., Resnick, M.: "Thick" Authenticity: New Media and Authentic Learning. *Journal of Interactive Learning Research* 10(2), 195–215 (1999)
13. Sweller, J.: Cognitive load theory, learning difficulty and instructional design. *Learning and Instruction* 4, 295–312 (1994)
14. Hegarty, M., Mayer, R.E., Monk, C.A.: Comprehension of Arithmetic Word Problems: A Comparison of Successful and Unsuccessful Problem Solvers. *Journal of Educational Psychology* 87, 18–32 (1995)
15. Glymour, C., Scheines, R.: Causal modeling with the TETRAD program. *Synthese*, 37–64 (2004)
16. Wittgenstein, L.: *Philosophical Investigations*. Prentice Hall (1953)
17. Boyer, K.E., Phillips, R., Wallis, M., Vouk, M., Lester, J.: Balancing Cognitive and Motivational Scaffolding in Tutorial Dialogue. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) *ITS 2008. LNCS*, vol. 5091, pp. 239–249. Springer, Heidelberg (2008)
18. Mayer, R.: *Multimedia Learning*, 2nd edn. Cambridge University Press, NY (2009)

19. Clark, R.E.: Games for Instruction? Presentation at the American Educational Research Association, New Orleans, LA (2011)
20. Andersen, E., Liu, Y., Snider, R., Roy, S., Zoran, P.: Placing a Value on Aesthetics in On-line Casual Games. In: ACM CHI Conference on Human Factors in Computing Systems (2011)
21. Rai, D., Beck, J.E.: Causal Modeling of User Data from a Math Learning Environment with Game-Like Elements. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS, vol. 6738, pp. 528–530. Springer, Heidelberg (2011)

Motivational Factors for Learning by Teaching: The Effect of a Competitive Game Show in a Virtual peer-Learning Environment

Noboru Matsuda¹, Evelyn Yarzebinski¹, Victoria Keiser¹, Rohan Raizada¹,
Gabriel Stylianides², and Kenneth R. Koedinger¹

¹ School of Computer Science, Carnegie Mellon University
{mazda, eey2, keiser, rohanr, krk}@cs.cmu.edu

² Department of Education, University of Oxford
gabriel.stylianides@education.ox.ac.uk

Abstract. To study the impact of extrinsic motivational intervention, a competitive Game Show was integrated into an on-line learning environment where students learn algebra equation solving by teaching a synthetic peer learner, called SimStudent. In the Game Show, a pair of SimStudents competed with each other by solving challenging problems to achieve higher ratings. To evaluate the effectiveness of the Game Show in the context of learning by teaching, we conducted a classroom study with 141 students in 7th to 9th grade. The results showed that to facilitate students' learning, the Game Show setting must be carefully designed so that (1) the Game Show goal and learning goal are aligned, and (2) it fosters a symbiotic scenario in which both winners and losers of the game show learn.

Keywords: Learning by teaching, Teachable Agent, Motivation and Engagement, SimStudent, Machine learning.

1 Introduction

The goal of the current paper is to explore the impact of extrinsic motivational intervention, in the form of a competitive Game Show, for tutor learning. Empirical studies show that performance-contingent rewards (i.e., competitors earn rewards based on their performance) facilitate intrinsic motivation more than competitively-contingent rewards (i.e., only the winner earns a reward) [1, 2]. In the current paper, we test the effect of a Game Show feature realized in an on-line learning environment in which students interactively tutor a synthetic peer, called SimStudent [3-5]. In the Game Show, students observe their SimStudents competing with each other by solving challenging problems. The goal of the Game Show for (human) students is to tutor their SimStudents well enough to earn the highest rating. Since this is an instance of performance-contingent rewards (instead of only a winner receiving a reward), we hypothesized that the presence of the Game Show would positively affect the students' intrinsic motivation, which would further facilitate tutor learning.

The present study is a part of an on-going effort to understand the cognitive and social factors that govern the effect of tutor learning [3-5]. It is well known that students learn by teaching others [6-8]. However, it has been only recently that researchers started to explore the underlying cognitive and social principles of tutor learning. This intellectual evolution is largely due to the recent developments of educational technology for a synthetic peer learner, aka teachable agent [9]. Such a technology enables researchers to collect detailed *process data* (cf. the *outcome data* typically measured by an achievement test) showing interactions between the students and the teachable agent. Collecting process data from human students teaching each other is very challenging and actually rarely done for field studies [6].

Betty's Brain is one of the pioneering projects using process data to probe tutor learning [10, 11]. The Betty's Brain system also has a game show feature [10, 12]. In the game show, students earn points by wagering on how well their Betty's Brain agents answer problems in the game show. Although the game show has been observed to play a central role in Betty's learning environment, the presence of the game show has not been controlled to study its effect for tutor learning.

In the current paper, we will explore the following research questions: (1) Does the proposed Game Show facilitate tutor learning? And, if so, (2) how does participation in the Game Show affect tutor learning?

2 APLUS with SimStudent and Competitive Game Show

2.1 Overview of SimStudent

SimStudent is a machine-learning agent that learns procedural skills inductively from examples [3-5]. In the context of learning by teaching, SimStudent attempts to solve a problem one step at a time by making a suggestion for the step (by applying a skill learned). SimStudent then asks about the correctness of the suggestion. If the suggestion receives negative feedback, SimStudent may suggest an alternative action on the step. When SimStudent has no other suggestions, it asks the human student to demonstrate the step as a hint. SimStudent generalizes examples (both from feedback and hints) using domain specific background knowledge and generates hypotheses in the form of production rules that best explain the examples.

Generating production rules inductively from examples is a complicated task. SimStudent uses a hybrid learning algorithm that involves (1) inductive logic programming to learn *when* to apply a production rule, (2) a version space to learn upon *what* to focus attention, and (3) an iterative-deepening depth-first search to learn *how* to change the problem state.

2.2 Overview of APLUS

In order to use it as a teachable agent for peer tutoring, SimStudent is embedded into an online, game-like learning environment called APLUS (Artificial Peer Learning environment Using SimStudent). There is a Tutoring Interface in APLUS that allows the student and SimStudent to collaboratively solve problems. To pose a new problem for

SimStudent, the student enters an equation into the first row of the Tutoring Interface. As SimStudent makes suggestions for each step, they are placed into the Tutoring Interface and the student can use the [Yes/No] button to provide feedback. When SimStudent requires a hint, the student demonstrates the next correct step in the interface.

In APLUS, SimStudent occasionally prompts students to explain their tutoring actions by asking “why” questions [5]. Such questions include (1) the reason to select a particular problem to solve (e.g., “Why should I do this problem?”), (2) the reason for an incorrect suggestion (e.g., “Why am I wrong?”), and (3) the reason for student’s demonstration (e.g., “Why did you do such?”). The student responds to SimStudent’s question either by using drop down menus or free text input.

APLUS provides several resources to assist students in peer learning. Because the student is also learning how to solve equations, he/she may get stuck. There are Worked-out Examples shown in the Tutoring Interface for students to review. There is a Unit Overview with a description of the process of solving equations. A Problem Bank is available with suggested problems that the student may use for tutoring. It also provides a quiz, which students can use to measure SimStudent’s progress. A summary of the quiz results then appears in a separate window, showing the correctness of the solution steps suggested by SimStudent. See [3-5] for more details.

2.3 Competitive Game Show

In the Game Show, a pair of SimStudents competes by solving challenging problems. Fig. 1 shows a sample screenshot of the Game Show window. The same Game Show window is displayed on each student’s screen. Two SimStudent avatars (e.g., Stacy and Amy in Fig. 1) are displayed in the middle of the screen. There is also a Game Show host displayed on the left. In one Game Show competition, there are five problems to solve. The Game Show host provides the first problem that is randomly selected from about 40 different patterns of problems with randomly generated constants and coefficients. The two competing (human) students then take turns and each provide two problems. When a problem is provided, students can see their own SimStudent working, filling in the problem-solving interface (same as the Tutoring Interface) on their own Game Show window.

When all five problems are solved, the Game Show host brings up a review screen on which the students can review the solutions that their SimStudents made for each problem. The correctness of each step is indicated.

Before entering the Game Show, students select their opponents on the match-up screen as shown in Fig. 2. There is a list of students waiting to be matched-up. They can also chat with each other. When a student selects a potential opponent, the opponent’s SimStudent avatar will be displayed along with the profile of the opponent showing a history of game results. The expected rating after a win or loss is also shown. The student can challenge any of the students on the waiting list, or he/she can simply wait for someone to issue a challenge. When receiving a challenge, the student will be shown the expected rating after the game. The challengee can either accept or reject the request for a challenge.

The students were told that their goals for Game Show is to teach SimStudent well and have it attain as high of a *rating* as possible. All SimStudents start at a rating of

25. Ratings are calculated based on the relative rating of the winner and the loser. The calculation is similar to the ELO chess rating system [13]. The winner's rating increases, and the loser's rating decreases. The amount of gain is proportional to the difference in the ratings between the two contestants – the bigger the difference, the more they gain or lose. It must be noted that *even when one wins against a lower rated opponent, the winner's rating still increases—even just a small amount.*

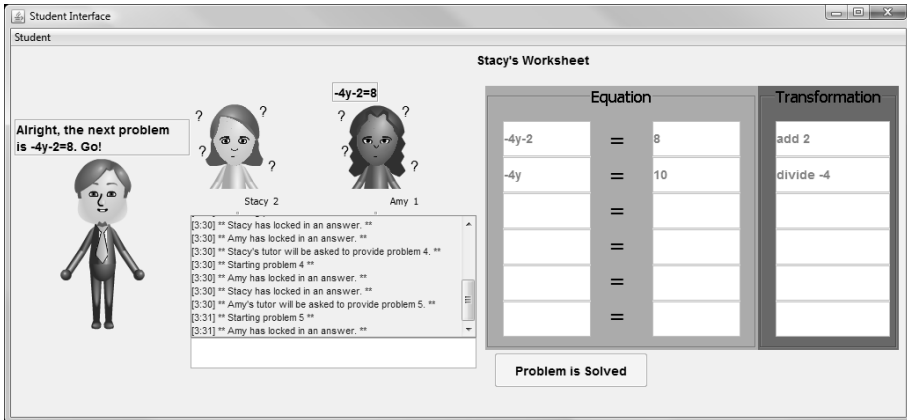


Fig. 1. The Game Show window. A pair of SimStudents competes by solving problems entered by the student tutors.

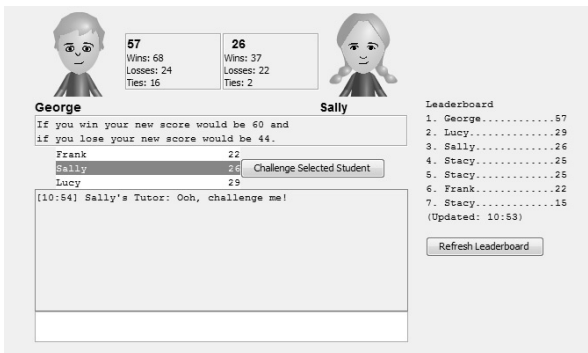


Fig. 2. The match-up screen. The student can select an opponent from the list. A profile for the selected student is then displayed with his/her rating and the history of game results.

3 Evaluation Study

3.1 Participants

One high school in Pittsburgh, PA, participated in the study under the supervision of Pittsburgh Science of Learning Center (www.learnlab.org). There were seven classes

with a total of 141 students participating, about 20 students per class. The study was conducted as a class-level randomized control trial among six algebra I classes, and a within-class randomized control trial for one class. Of the students that participated in the study, 88 completed the pretest, participated in all three study-days, and took the posttest. Of those 88 students, there were 69 students who also took the delayed-test. These 69 students were included in our analysis in Section 4.1, and the 88 students are included in the rest of the analyses.

3.2 Methods

The intervention lasted for three 42-minute class periods. The students' task was to teach SimStudent how to solve equations with variables on both sides. Students in the Game Show condition switched between APLUS and Game Show as often as they liked, and were actually told to do so to enhance their SimStudent's knowledge for better Game Show performance. The students in the baseline condition did not have a Game Show, and were told that their goal was to have SimStudent pass the quiz.

During these three days, the system automatically logged all of the students' activities, including problems tutored, feedback provided, steps performed, examples reviewed, hints requested by SimStudent, quiz attempts, game shows initiated, game show wins and losses, game show rating, and game show opponents challenged, etc.

There was a pre- and post-test the days immediately before and after the intervention periods. There was also a delayed test two weeks after the post-test.

3.3 Measures

We measured students' learning gain through a two part on-line test: a Procedural Skill Test (PST) and a Conceptual Knowledge Test (CKT). Three isomorphic versions of the test were randomly used for the pre, post, and delayed tests.

The PST has three sections: (a) ten equation solving items, (b) twelve items to determine if a given operation is a logical next step for a given equation, and (c) five items to identify the incorrect step in a given incorrect solution. The CKT has two sections divided into: (a) 38 true/false items about basic algebra vocabulary, and (b) ten true/false items to determine if two given algebra expressions are equivalent.

Following the post-test, students had the option to take a questionnaire comprised of (a) 16 items on a 7-point Likert-scale that measured different types of motivation and (b) one free response item. There are four constructs on the questionnaire with reliabilities of 0.79 (mastery), 0.77 (performance), 0.55 (strategy), and 0.49 (affect). Because the affect construct reliability is so low, we shall exclude it from the analysis.

To quantify students' engagement during tutoring, several variables in the process data that might have arguably reflected the degree of students' commitment and care about their SimStudents' learning were used (see Section 4.2). One such example is the quality of self-explanation students provided. There were 2008 student responses for SimStudent's occasional "why" questions. Three human coders categorized these responses into "deep" and "shallow" responses.

4 Results

4.1 Test Scores

Fig. 3 shows the average test scores for each condition on pre-, post-, and delayed-tests for PST (a) and CKT (b). The mixed-design analysis revealed that there was a main effect of test (pre vs. post vs. delayed) for the PST scores; $F(2,66) = 8.81, p < 0.001$. Comparing to the mean of the pre-test (0.38, $SD=0.20$), the mean of the post-test, 0.45 ($SD=0.20$; $t(68) = -3.61, p < 0.01$), and the delayed-test, 0.46 ($SD=0.23$; $t(68)=-4.31, p < 0.00$), were significantly higher. The difference between the post-test and delayed-test scores was not significant. There was a trend of a main effect of the condition, the Game Show condition (GS) vs. the Baseline control condition (BL); $F(1,67) = 3.24, p = 0.08$. No interaction among the test and condition existed.

There was a condition difference for the PST pre-test score; $M_{GS} = 0.45$ ($SD=0.19$) vs. $M_{BL} = 0.34$ ($SD=0.18$); $t(86) = -2.94, p < 0.01$. However, an ANCOVA did not confirm the main effect of the condition for the PST post-test scores; adjusted mean, factoring in pretest scores, was $M_{GS} = 0.50$ vs. $M_{BL} = 0.42$.

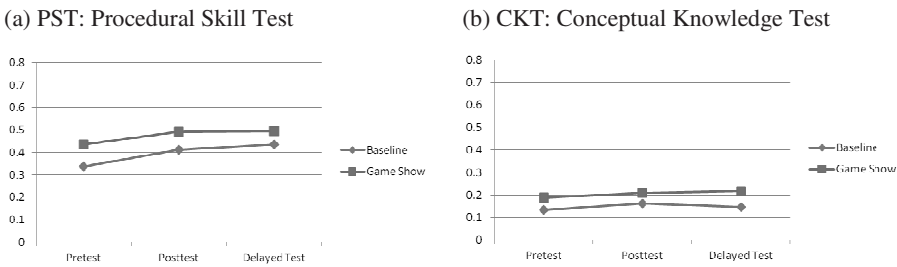


Fig. 3. Test scores for the PST (a) and the CKT (b)

In sum, the students' scores on the PST increased from pre- to post-test, showing the effect of APLUS and SimStudent. However, there was no observed significant effect of the Game Show for tutor learning related to the baseline.

4.2 Tutoring Engagement and Motivation

To understand why there was a lack of effect of Game Show for tutor learning, we probed the process data to see if the students were motivated and/or engaged in tutoring. Most remarkably, it turned out that the Game Show students responded to SimStudent's questions significantly better than the Baseline students; the mean probability of disregarding SimStudent's questions was $GS=0.07$ ($SD=0.12$) vs. $BL=0.17$ ($SD=0.25$); $t(63)=2.27, p < 0.05$. Also, the Game Show students showed a significantly higher probability of entering a "deep" response to SimStudent's questions than the Baseline students; the mean probability of entering a "deep" response was $GS=0.24$ ($SD=0.18$) vs. $BL=0.16$ ($SD=0.15$); $t(85)=-2.34, p < 0.05$. There was no difference in

the amount of words used per response. *These results affirmatively suggest that the Game Show students were more “engaged” in tutoring than the Baseline students.*

The students in the Game Show condition tutored on significantly fewer problems than the baseline condition; mean number of tutored problems for GS=11.1 (SD=5.0) vs. BL=18.6 (SD=7.57); $t(75)=5.45, p<0.00$. However, there was no condition difference on the average duration of tutoring per problem. The students in the baseline condition tutored more problems, but the game show students spent just as long on each problem when they were doing it, showing that *the Game Show did not hurt engagement by encouraging them to hurry back to the game show. The Game Show students achieved the same level on the PST post-test by tutoring fewer problems.*

There was no condition difference for the student’s mean response for the questionnaire constructs. There are a few statistically significant correlations between “engagement” factors (process data) and “motivation” factors (questionnaire), but their correlation coefficients were relatively small. There was no significant correlation between the learning gain measures (both on PST and CKT) and “motivation” factors.

4.3 Game Show Participation

Table 1 shows descriptive statistics for the students’ participation in the Game Show. Many students tended to challenge lower rated opponents more than higher rated opponents. The average rating difference for a given challenge that initiated a contest (challenging opponent’s delta) is -2.4, meaning that students challenged opponents who had a rating 2.4 below them. Likewise, students tended to accept a challenge from lower rated opponents – the average accepting opponent’s delta is -2.0.

Once entered into the Game Show, students rarely went back to tutor SimStudent, regardless of a win or loss. Instead, they tended to find lower rated opponents for an easy win. This increased their chance of winning, which is a reasonable strategy for the current Game Show scenario – the goal of the Game Show was to gain the highest final rating. However, this was not an ideal strategy for tutor learning – students did not find it necessary to tutor their SimStudents after they won the game.

Table 1. Descriptive statistics of the Game Show participation. The numbers show the average followed by standard deviation in the parentheses.

Final rating	26.8 (16.0)	Time in Game Show (min.)	45.3 (26.4)
Ratio of challenge/accept	4.5 (3.6)	Probability of win	0.46 (0.34)
Prob. tutoring after win	0.11 (0.18)	Prob. tutoring after loss	0.24 (0.35)
Challenging opponent’s delta	-2.4 (13.8)	Accepting opponent’s delta	-2.0 (11.5)

The above findings imply that the students were more focused on the *performance* goal (i.e., to increase rating) as opposed to the *learning* goal (to better learn the subject matter). To understand how students selected their opponents, we grouped students based on their preference in selecting opponents. Fig. 4 shows a scatter plot for the average difference of ratings when students challenged (X-axis) and when they

accepted others' challenge requests (Y-axis). On both axes, the difference is relative to the student's rating (i.e., student's rating minus the opponent's rating). There was a strong correlation between these two variables; $r(40)=0.57$, $p<0.00$. Those who tended to challenge higher rated students also tended to accept challenges from higher rated students. The same is true for the opposite direction.

In Fig. 4, the top right quadrant shows students who, on average, challenged higher rated opponents and accepted challenge requests from higher rated opponents. This group of students could be labeled as *Risky Challengers* (RC), because they preferred to win against strong competitors (probably) to make a big rating leap on a win. On the other hand, the bottom left quadrant shows students who, on average, challenged lower rated opponents and accepted challenge requests from lower rated opponents. This group of students could be labeled as *Strategic Winners* (SW), because they were more focused on winning the game for a small but steady rating accumulation. There were 12 (29.3%) RC students and 15 (36.6%) SW students among 41 GS students.

We hypothesized that the RC students learned more than the SW students, because the RC students needed to tutor their SimStudents better than the SW students to win the game against the higher rated opponents. To our surprise, there was no statistically significant difference in tutor learning (measured as the normalized gain on the PST) between SW and RC; 0.08 (SD = 0.41) vs. 0.03 (SD = 0.17), $t(25)=0.38$, $p=0.71$. The hypothesis about the RC students' better learning is not supported.

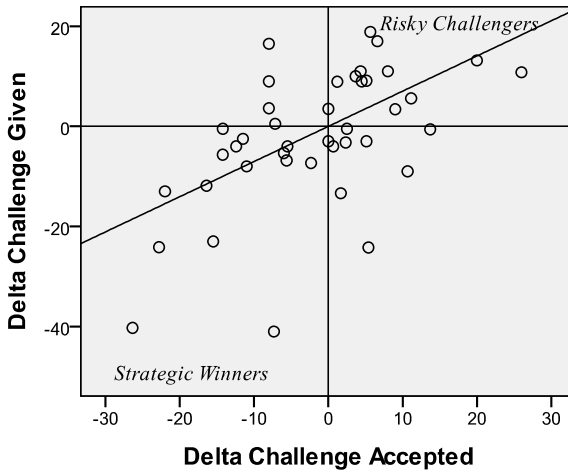


Fig. 4. Scatter plot showing students' average rating difference when they challenge (X-axis) and when they accept their friends' challenge requests (Y-axis). On both axes, the difference is computed by subtracting the opponent's rating from the student's rating. The regression line coefficient: $y=0.70*x$. $r^2=0.33$, $r(40)=0.57$, $p<0.00$.

When analyzing who competed with whom, there was a clear pattern of a predatory group dynamics. About 70% of chance, the competition was between a RC student

and a SW student. In other words, *the SW students won the game at the cost of the RC students' loss.*

The above observations imply that *the current design of Game Show, especially its goal (to achieve the highest rating among the contestants) and the way students select their opponents, is rather harmful for tutor learning.*

5 Discussion

Targeting lower rated opponents is a reasonable strategy to achieve the goal of the Game Show, but is not an ideal strategy for tutor learning. This motivates us to redesign the Game Show. A striking fact is that there was no difference in learning gain between Risky Challengers and Strategic Winners (Fig. 4). The Risky Challengers, who should have been motivated to make their SimStudent a stronger competitor by tutoring better, did not actually tutor better (or more appropriately) than the Strategic Winners. This implies that simply changing the pairing schema to prevent students from challenging lower rated opponents is not sufficient to increase the impact of the Game Show for tutor learning. *There must be a fine alignment between the goal of the Game Show and tutor learning.* In this regard, our data actually support Miller et al's claim [14] that such a *knowledge-dependency*, i.e., the relation between the pedagogically targeted concepts and the knowledge required to interact successfully with the game environment, is key for successful learning. In the game show used in the Betty's Brain system, the game goal is directly correlated with the learning goal.

To align the goals of the Game Show and learning, the schema to match up competitors should be changed so that it guarantees that the winners' SimStudents have high competency of solving the target equations (i.e., equations with variables on both sides, in the current study). One such idea is to discourage students from challenging lower rated opponents (by, for example, putting a restriction on the lower bound for the rating of an opponent to challenge), or to restrict the range of available opponents so that the difference in ratings is within a desired zone.

Although the goal of the Game Show is to earn the highest rating, the individual competitions formed a natural winner-loser distinction between the Strategic Winners and the Risky Challengers. Since there must be winners and losers, this predatory structure is an essential characteristic of the competitive game-show type of feature, which is known to be harmful [2]. Therefore, *letting students form definite winners and definite losers must be avoided to prevent only a small group of students learning at the cost of other students.*

To realize such a "symbiotic" Game Show setting that provides equal learning opportunities for all students regardless of the result of the competition, we must encourage the losers to tutor SimStudent more. One simple idea is to force students to tutor SimStudent after a loss. Embedding virtual game-show contestants with various levels of competency and setting the goal of Game Show to beat the virtual contestants would also resolve the situation.

6 Conclusion

The results from a classroom *in vivo* experiment revealed the misalignment of the goals between the motivational Game Show feature and the learning task. The students' focus was on performance in the Game Show was achieved without actually committing to learning (i.e., tutoring SimStudent better and more appropriately). The study data also suggested that a symbiotic Game Show setting is required for an ideal learning environment that would foster learning for both winners and losers.

We also discussed a few suggestions for future improvements based on the study data and discussions. We will continue our efforts by pursuing an evidence-based iterative-design and engineering process to explore the theory of learning by teaching using the SimStudent technology.

Acknowledgements. The research reported here was supported by National Science Foundation Award No. DRL-0910176 and the Institute of Education Sciences, U.S. Department of Education, through Grant R305A090519 to Carnegie Mellon University. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education. This work is also supported in part by the Pittsburgh Science of Learning Center, which is funded by the National Science Foundation Award No. SBE-0836012.

References

1. Vansteenkiste, M., Deci, E.L.: Competitively contingent rewards and intrinsic motivation: Can losers remain motivated? *Motivation and Emotion* 27(4), 273–299 (2003)
2. Reeve, J., Deci, E.L.: Elements of the competitive situation that affect intrinsic motivation. *Personality and Social Psychology Bulletin* 22(1), 24–33 (1996)
3. Matsuda, N., Keiser, V., Raizada, R., Tu, A., Stylianides, G., Cohen, W.W., Koedinger, K.R.: Learning by Teaching SimStudent: Technical Accomplishments and an Initial Use with Students. In: Alevan, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6094, pp. 317–326. Springer, Heidelberg (2010)
4. Matsuda, N., Yarzebinski, E., Keiser, V., Raizada, R., Stylianides, G.J., Cohen, W.W., Koedinger, K.R.: Learning by Teaching SimStudent – An Initial Classroom Baseline Study Comparing with Cognitive Tutor. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A., et al. (eds.) AIED 2011. LNCS, vol. 6738, pp. 213–221. Springer, Heidelberg (2011)
5. Matsuda, N., et al.: Studying the Effect of Tutor Learning using a Teachable Agent that asks the Student Tutor for Explanations. In: Proceedings of the International Conference on Digital Game and Intelligent Toy Enhanced Learning, DIGITEL 2012 (to appear, 2012)
6. Roscoe, R.D., Chi, M.T.H.: Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions. *Review of Educational Research* 77(4), 534–574 (2007)
7. Cohen, E.G.: Restructuring the classroom: Conditions for productive small groups. *Review of Educational Research* 64(1), 1–35 (1994)
8. Cohen, P.A., Kulik, J.A., Kulik, C.L.C.: Education outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal* 19(2), 237–248 (1982)

9. Brophy, S., et al.: Teachable agents: Combining insights from learning theory and computer science. In: Lajoie, S.P., Vivet, M. (eds.) *Proceedings of the International Conference on Artificial Intelligence in Education*, pp. 21–28. IOS Press, Amsterdam (1999)
10. Schwartz, D.L., et al.: Interactive metacognition: monitoring and regulating a teachable agent. In: Hacker, D.J., Dunlosky, J., Graesser, A.C. (eds.) *Handbook of Metacognition in Education*, pp. 340–358. Routledge, New York (2009)
11. Biswas, G., et al.: Measuring Self-Regulated Learning Skills through Social Interactions in a teachable Agent Environment. *Research and Practice in Technology Enhanced Learning*, 123–152 (2010)
12. Chase, C., et al.: Teachable Agents and the Protégé Effect: Increasing the Effort Towards Learning. *Journal of Science Education and Technology* 18(4), 334–352 (2009)
13. Elo, A.: *The Rating of Chessplayers, Past and Present*. Arco (1978)
14. Miller, C.S., Lehman, J.F., Koedinger, K.R.: Goals and learning in microworlds. *Cognitive Science* 23(3), 305–336 (1999)

An Analysis of Attention to Student – Adaptive Hints in an Educational Game

Mary Muir and Cristina Conati

Department of Computer Science, University of British Columbia
{marymuir, conati}@cs.ubc.ca

Abstract. We present a user study to investigate which factors affect student attention to user-adaptive hints during interaction with an educational computer game. The game is Prime Climb, an educational game designed to provide individualized support for learning number factorization skills in the form of hints based on a model of student learning. We use eye-tracking data to capture user attention patterns on the game adaptive-hints, and present results on how these patterns are impacted by factors related to existing user knowledge, hint timing, and attitude toward getting help in general. We plan to leverage these results in the future for making hint delivery adaptive to these factors.

Keywords: Adaptive help, educational games, pedagogical agents, eye-tracking.

1 Introduction

The ability of providing interventions that are adaptive to each student's specific needs is one of the distinguishing features of intelligent tutoring systems (ITS). One of the most widespread forms of adaptive interventions is to provide hints designed to gradually help students through specific educational activities when they have difficulties proceeding on their own [14]. Despite the wide adoption of adaptive hints, there is an increasing body of research showing their possible limitations, going from students *gaming the system*, i.e., using the hints to get quick answers from the ITS [see 7 for an overview], to *help avoidance*, i.e. students not using hints altogether [e.g., 8, 9]. In this paper, we are interested in investigating the latter issue. More specifically, we seek to gain a better understanding of which factors may affect a student's tendency to attend to adaptive hints that the student has not explicitly elicited.

This research has three main contributions to the ITS field. First, while previous work on help avoidance focused on capturing and responding to a student's tendency to avoid requesting hints [e.g., 8, 9], here we investigate how students react when the hints are provided unsolicited. A second contribution is that we look at attention to adaptive hints during interaction with an educational computer game (edu-game), whereas most previous work on student usage (or misuse) of hints has been in the context of more structured problem solving activities. Providing adaptive hints to support learning during game play is especially challenging because it requires a

trade-off between fostering learning and maintaining engagement. We see the results we present in this paper as valuable information that can be leveraged to achieve this tradeoff. The third contribution of our work is that we use eye-tracking data to study user attention patterns to the adaptive-hints, an approach not previously investigated in hint-related research. In [13], we presented a preliminary qualitative analysis of eye-tracking data for two students playing Prime Climb, and edu-game for number factorization. In this paper, we extend that work by presenting a more extensive quantitative analysis based on data from 12 students.

After discussing related work, we describe Prime Climb in further detail. Next, we illustrate the user study we conducted for data collection. Finally, we discuss results related to how user attention patterns are impacted by factors related to user existing knowledge, hint timing, and attitude toward getting help in general. We also present preliminary results on how attention to hints affects subsequent game playing.

2 Related Work

Edu-games are seen as one of the most promising new forms of computer-based education; however, while there is ample evidence that they are highly engaging, there is less support for their educational effectiveness [e.g., 1, 2, 16]. User-adaptive edu-games are receiving increasing attention [e.g., 3, 4, 5, 15] as a way to improve edu-games effectiveness. However, most of the existing work has not been formally evaluated in terms of how adaptive edu-game components affect edu-game effectiveness. There has also been rising interest in using eye-tracking to gain insights on the cognitive and affective processes underlying a user's performance with an interactive system [e.g., 6, 10, 11, 12]. In this paper, we extend the use of gaze information to understand if/how users attend to an educational game's adaptive interventions. Adaptive incremental hints are commonly used in ITS, but their effectiveness is in question because of extensive evidence that students can misuse them. Two main categories of help misuse have been investigated so far in the context of ITS for problem solving. The first is *gaming the system*, i.e., repeatedly asking for help or entering wrong answers on purpose to get to bottom-out hints that explicitly tell a student how to perform a problem solving step and move on [7]. The second is *help avoidance*, i.e., not asking for help when needed [8]. Several models have been developed to detect in real-time, instances of gaming behavior and intervene to reduce this behavior [see 7 for an overview]. Aleven et al., [8] present a model that detects both gaming the system as well as help avoidance. In [9], this model is used to generate hints designed to improve students' help seeking behavior in addition to hints that help with the target problem solving activity. Not much work, however, has been done on understanding if/how students process adaptive hints that they have not elicited. In [9], the authors suggest that students often ignore these hints. A similar hypothesis was brought forward in [3], based on preliminary results on student attention to hints in Prime Climb, the game targeted in this paper. Those results were based on hint display time (duration of time a hint stays open on the screen) as a rough indication of attention. In [13], however, initial results based on the analysis of gaze data from two Prime Climb

players suggested that students sometimes pay attention to hints. The results we present here confirm this finding and extend it by presenting an analysis of factors that impact attention.

3 The Prime Climb Game

In Prime Climb, students practice number factorization by pairing up to climb a series of mountains. Each mountain is divided into numbered hexagons (see Figure 1), and players must move to numbers that do not share common factors with their partner's number, otherwise they fall. To help students, Prime Climb includes the Magnifying Glass, a tool that allows players to view the factorization for any number on the mountain in the device at the top-right corner of the interface (see Figure 1). Prime Climb also provides individualized textual hints, both on demand and unsolicited. Unsolicited hints are provided in response to student moves and are designed to foster student learning during game playing by (i) helping students when they make wrong moves due to lack of factorization knowledge; (ii) eliciting reasoning in terms on number factorization when students make correct moves due to lucky guesses or playing based on game heuristics. Prime Climb relies on a probabilistic student model to decide when incorrect moves are due to a lack of factorization knowledge vs. distraction errors, and when good moves reflect knowledge vs. lucky guesses. The student model assesses the student's factorization skills for each number involved in game playing, based on the student's game actions [3]. Prime Climb gives hints at incremental levels of detail, if the student model predicts that the student doesn't know how to factorize one of the numbers involved in the performed move. The hint sequence includes a *tool* hint that encourages the student to use the magnifying glass to see relevant factorizations. If the student needs further help, Prime Climb gives *definition* hints designed to re-teach "what is a factor" via explanations and generic examples (e.g., see Figure 1). There are two different factorization definitions: "*Factors are numbers that divide evenly into the number*" and "*Factors are numbers that multiply to give the number*". The game alternates which definition to give first, and presents the second the next time it needs to provide a definition hint. The examples that accompany the definitions change for every hint, and are designed to help illustrate the given definitions while still leaving it to the student to find the factorization of the numbers relevant to the performed move. Finally, Prime Climb provides a *bottom-out* hint giving the factorization of the two numbers involved in the move (e.g., "*You fell because 84 and 99 share 3 as a common factor. 84 can be factorized as...*"). Students can access the next available hint by clicking on a button at the bottom of the current hint (See Figure 1). Otherwise, hints are given in progression as the student model calls for a new hint. A hint is displayed until the student selects to access the next hint or to resume playing (by clicking a second button available at the bottom of the hint). It should be noted that the Prime Climb *bottom-out* hints focus on making the student understand her previous move in terms of factorization knowledge; they never provide explicit information on how to move next. Thus, the Prime Climb hints are less conducive to a student gaming the system than bottom-out hints giving more

explicit help [e.g. 7]. As a matter of fact, previous studies with Prime Climb show that students rarely ask for hints. Most of the hints the students see are unsolicited.

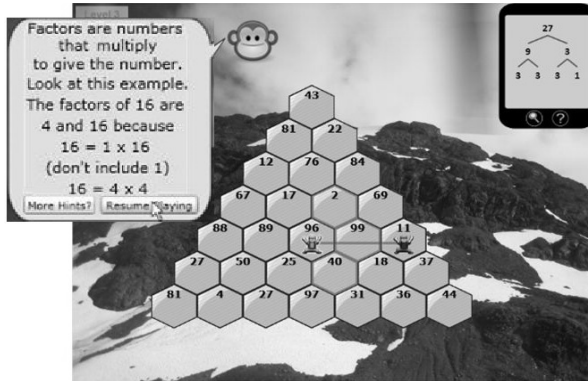


Fig. 1. The Prime Climb Interface

4 User Study on Attention to Hints

The study we ran to investigate students' attention to Prime Climb's adaptive hints relied on a Tobii T120 eye-tracker, a non-invasive desktop-based eye-tracker embedded in a 17" display that collects binocular eye-tracking data.

Twelve students (6 female) from grades 5 and 6 (six students for each grade) participated in the experiment. Participants first took a pre-test testing their ability to identify the factors of individual numbers and common factors between two numbers (16 numbers were tested overall). They then underwent a calibration phase with the Tobii eye-tracker. Next, they each played Prime Climb with an experimenter as a partner. The game was run on a Pentium 4, 3.2 GHz machine with 2GB of RAM, with the Tobii acting as the main display screen. Finally, participants took a post-test equivalent to the pre-test and completed a questionnaire on their game experience.

To analyze the attention behaviors of our study participants with respect to the received adaptive hints, we define an area of interest (*Hint AOI*) that covers the text of the hint message. We use two complementary eye-gaze metrics as measures of user attention to hints. The first is *total fixation time*, i.e., total time a student's gaze rested on the *Hint AOI* of each displayed hint. Total fixation time gives a measure of overall attention to hints, but does not provide detailed information on how a hint was actually processed (e.g., it cannot differentiate between a player who stares blankly at a hint vs. one who carefully reads each word). Furthermore, it is not ideal to compare attention to the different types of hints in Prime Climb because they have different lengths on average (15 words for *tool* hints; 17 words for *bottom-out* hints; 36 words for *definition* hints). Thus, our second chosen metric is the ratio of fixations per word (*fixations/word*), a measure that is independent of hint length and gives a sense of how carefully a student scans a hint's text.

5 Factors Affecting Attention to Hints: Results

The study game sessions lasted 33 minutes on average ($SD = 15$). There was no improvement from pre to post-test performance, with participants scoring an average of 74% ($SD = 31\%$) in the pre-test, an average of 72% ($SD = 31\%$) on post-test and an average percentage learning gain of -0.02 ($SD = 0.06$). Consistent with previous Prime Climb studies, students rarely asked for help. One student asked for four hints, two students asked for hints twice, and two other students requested one hint. Prime Climb, however, generated unsolicited hints frequently: an average of 51 hints per player, ($SD = 23$), with an average frequency of 37 seconds ($SD = 44$). Thus, lack of system interventions can be ruled-out as a reason for lack of learning. If anything, it is possible that the hints happened too frequently, interfering with game playing and leading students to ignore them.

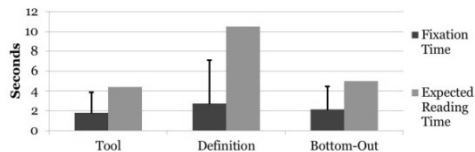


Fig. 2. Average Fixation Time for Prime Climb Hint Types

In order to investigate this idea further, we first compared average fixation time on each hint type with the *expected reading time* (calculated using the 3.4 words/second rate from [17]), which is the time it would take an average-speed reader to read the hint. Figure 2 shows that average fixation time is much shorter than *expected reading time* but the high standard deviation in all three measures shows a trend of selective attention. In the rest of this section, we investigate which factors influenced a student’s decision to attend a hint or not. One obvious factor is whether the hints generated were justified, i.e., whether the probabilistic student model that drives hint generation is accurate in assessing a student’s number factorization knowledge. Unfortunately we can only answer this question for the numbers tested in the post-test, which are about 10% of all the numbers covered in Prime Climb. The model *sensitivity* on post-test numbers (i.e., the proportion of actual positives which are correctly identified as such) is 89%, indicating that the model generally did not underestimate when a student knew a post-test number and thus it likely triggered justified hints on them. It should be noted, however, that for post-test numbers the student model is initialized with prior probabilities derived from test data from previous studies. For all the other numbers in Prime Climb, the model starts with generic prior probabilities of 0.5. Thus, the model’s assessment of how student factorization knowledge on these numbers evolved during game play was likely to be less accurate than for post-test numbers, and may have generated unjustified hints.

Bearing this in mind, we looked at the following additional factors that may influence student attention to hints in our dataset. *Move Correctness* indicates whether the hint was generated in response to a correct or to an incorrect move. *Time of Hint* sets

each hint to be in either the first or second half of a student’s interaction with the game, defined by the median split over playing time. *Hint Type* reflects the three categories of Prime Climb hints: *Definition*, *Tool*, and *Bottom-out*. *Attitude* reflects student’s general attitude towards receiving help when unable to proceed on a task, based on student answers to a related post-questionnaire item, rated using a Likert-scale from 1 to 5. We divided these responses into three categories: *Want help*, *Neutral*, and *Wanted no help*, based on whether the given rating was greater than, equal to, or less than 3 respectively. *Pre-test score* represents the student percentage score in the pre-test as an indication to student pre-existing factorization knowledge.

5.1 Factors That Affect Attention to Hints Measured by Total Fixation Time

We start our analysis looking at total fixation time on a displayed hint as a measure of attention. We ran a $2(\text{Time of Hint}) \times 3(\text{Hint Type}) \times 2(\text{Move Correctness}) \times 3(\text{Attitude})$ general linear model with *pre-test score* as a co-variant, and total fixation time as the dependent measure. We found the following interaction effects¹:

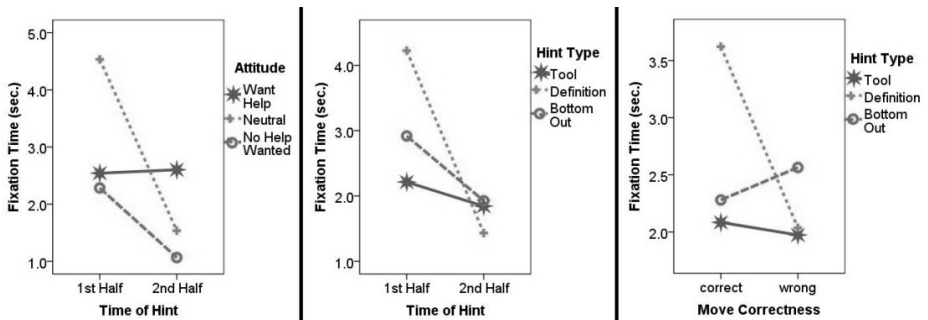


Fig. 3. Interaction effects between: (Left) *Time of Hint* and *Attitude*; (Middle) *Time of Hint* and *Hint Type*. (Right) *Move Correctness* and *Hint Type*.

- *Attitude* and *Time of Hint*. $F(2,447) = 5.566$, $p=0.004$, $\eta^2 = 0.024$ (see Figure 2, Left). Fixation time for those with a neutral help attitude dropped from being the highest among the three groups in the first half of the game to being very low in the second half. For students who do not want help, fixation time is the lowest of the three groups in the first half of the game, and drops to even lower during the second half. Fixation time for those who wanted help did not change.
- *Time of Hint* and *Hint Type*, $F(2,447) = 5.963$, $p=0.003$, $\eta^2=0.026$. (see Figure 2, Middle). Fixation time drops for all hint types between the first and second half of the game. The drop, however, is statistically significant only for *definition* hints, suggesting that these hints became repetitive and were perceived as redundant despite the inclusion of varying examples that illustrate the definitions.

¹ We also found main effects for both *Time of Hint* and *Attitude*, but we don’t discuss them in detail because they are further qualified by the detected interactions.

- *Hint Type* and *Move Correctness*, $F(2,447) = 3.435$, $p=0.033$, $\eta^2=0.015$. (see Figure 2, Right). Players had significantly higher fixation time on *definition* hints caused by correct moves than on those caused by incorrect moves². There were no statistically significant differences between fixation times on correct vs. incorrect moves for the other two hint types. We find the result on *definition* hints somewhat surprising, because we would have expected hints following correct moves to be perceived as redundant and thus attended less than hints following incorrect moves. It is possible, however, that the very fact that hints after correct moves were unexpected attracted the student attention.

5.2 Factors That Affect Attention to Hints Measured by Fixations/Word

To gain a better sense of how students looked at hints when they were displayed, we ran a general linear model with the same independent measures described above (*Time of Hint*, *Hint Type*, *Move Correctness*, *Attitude*, and *pre-test scores*) with fixations/word as the dependent measure. We found three main effects

- *Attitude*, $F(2,447) = 6.722$, $p=0.001$, $\eta^2=0.029$, Students who wanted no help had the lowest fixations/word (Avg. 0.25, $SD = 0.30$), significantly lower than the other two groups. The difference between the help (Avg. 0.36, $SD = 0.38$) and neutral group (Avg. 0.31, $SD = 0.28$) is not significant, but the trend is in the direction of the help group having higher fixation/word than the neutral group.
- *Pre-test score*, $F(1,447) = 6.614$, $p=0.01$, $\eta^2=0.015$. Students with the lowest (below 65%) and highest (above 94%) scores had fewer fixations/word than students with intermediate scores. For high knowledge students, this effect is likely due to the hints not being justified. We can only speculate that, for low knowledge students, the effect may be due to a general lack of interest in learning from the game.
- *Hint Type*, $F(2,447) = 31.683$, $p<0.001$, $\eta^2=0.124$. *Definition* hints (Avg. 0.17, $SD = 0.22$) had a statistically significantly lower fixation/word than either *Tool* (Avg. 0.35, $SD = 0.38$) or *Bottom-out* hints (Avg. 0.34, $SD = 0.32$), possibly due to the fact that students tended to skip the actual definition part of the hints, which does not change, in order to get to the factorization examples at the bottom.

We also found two interaction effects, both involving *Move Correctness* (see Figure 4). The first interaction is with *Hint Type*, $F(2,447) = 11.141$, $p<0.001$, $\eta^2=0.013$. Fixations/word on *Bottom-out* hints drops significantly between those given after a correct move (Avg. 0.48, $SD = 0.27$) and those given after an incorrect move (Avg. 0.19, $SD = 0.22$). This result confirms the positive effect that *Move Correctness* seems to have on attention to hints found in the previous section for *definition* hints. Here, the effect possibly indicates that students are scanning *Bottom-out* hints for correct moves carefully in order to understand why they are receiving this detailed level of hint when they are moving well. The second interaction is with *Time of Hint*,

² There is also a significant difference between fixation time on *definition* hints after correct moves and the other two type of hints after correct moves, but this difference is likely an effect of definition hints being longer, as we discussed in section 4.

$F(1,447)=3.922$, $p=0.048$, $\eta^2=0.009$ and shows that fixations/word drops significantly between hints for correct moves given in the first and the second half of the game, suggesting that the aforementioned surprise effect of hints for correct moves fades as the game progresses.

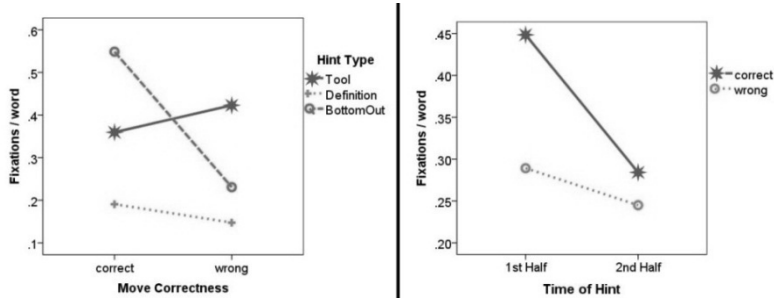


Fig. 4. Interaction effect between: (Left) *Time of Hint* and *Move Correctness*; (Right) *Move Correctness* and *Hint Type*

5.3 Factors That Affect Attention: Discussion

All of the factors that we explored (*Time of Hint*, *Hint Type*, *Attitude*, *Move Correctness* and *Pre-test Scores*) affected to some extent attention to the Prime Climb hint, and the results of our analysis can be leveraged to improve attention to these hints. We found, for instance, that attention to hints decreases as the game proceeds, and the drop is highest for *definition* hints, suggesting that these hints are too repetitive and should be either varied or removed. If a student has an existing attitude toward help, this attitude generates consistent patterns of attention to hints throughout the game (low attention for those who do not want help, higher attention for those who do). This result suggests that general student attitude toward receiving help should be taken into account when generating adaptive hints, and strategies should be investigated to make hints appealing for those students who do not like receiving help. Similarly, strategies should be devised to make students with low knowledge (as assessed by the student model) look at the hints, since our results indicate that these students tend not to pay attention, although they are the ones who likely need hints the most. We also found that students with a neutral attitude toward help had much less consistent attention behavior than the students who wanted help and the students who did not. The neutral students showed quite high attention to hints in the first half of the interaction, but dropped almost to the lowest in the second half, confirming that the Prime Climb hints should be improved to remain informative and engaging as the game proceeds. In the next section, we show initial evidence that improving attention to hints as discussed here is a worthwhile endeavor because it can improve student interaction with the game.

6 Effect of Attention to Hints on Game Playing

In this section, we look at whether attention to hints impact students' performance with Prime Climb. In particular, we focus on the effect of attention to hints on correctness of the subsequent player's move. As our dependent variable, *Move Correctness After Hint*, is categorical (e.g., the move is either correct or incorrect), we use logistic regression to determine if *Fixation Time*, *Fixations per word* and *Hint Type* are significant predictors of *Move Correctness After Hints*³.

Table 1. Logistic regression results for *Move Correctness After Hint*

	B (SE)	p	95% CI for Odds Ratio		
			Lower	Odds Ratio	Upper
Fixations/word	0.98 (0.44)	0.03	1.12	2.68	6.39

Table 1 shows the results of running logistic regression on these data, indicating that *Fixations per word* is the only significant predictor of *Move Correctness After Hints*. The odds ratio greater than 1 indicates that, as fixations/word increases, the odds of correct moves also increases. This suggests that when the players read the hints more carefully, their next move is more likely to be correct. The results of the logistic regression also indicate that the type of hint student pay attention to does not impact move correctness. This finding is consistent with the fact that, in Prime Climb, *bottom-out* hints do not provide direct information on what to do next; they only explain how to evaluate the player's *previous* move in terms of number factorization, and this information cannot be directly transferred to the *next* move. Still, it appears that some form of transfer does happen when students pay attention to the hints, helping them make fewer errors on subsequent moves. This finding suggests that further investigation on how to increase student attention to hints is a worthwhile endeavor, because it can improve student performance with the game, and possibly help trigger student learning.

7 Conclusions and Future Work

In this paper, we presented a user study to investigate which factors affect student attention to user-adaptive hints during interaction with an educational computer game. This work contributes to existing research on student use and misuse of adaptive hints in ITS by looking at how students react to hints when they are provided unsolicited by the system, as opposed to explicitly requested by the student or obtained via gaming strategies. There are two additional aspects that are innovative in this work. The first is that we focus on adaptive hints provided by an edu-game, i.e., in a context in which

³ The data points in our dataset are not independent, since they consist of sets of moves generated by the same students. Lack of independence can increase the risk of making a type 1 error due to overdispersion (i.e., ratio of the chi-square statistic to its degrees of freedom is greater than 1), but this is not an issue in our data set. ($\chi^2 = 6.41$ df = 8).

it is especially challenging to provide didactic support because it can interfere with game playing. The second is that we use eye-tracking data to analyze student attention. We found that attention to hints is affected by a variety of factors related to user existing knowledge, hint timing/context and attitude toward getting help in general. The next step in this research will be to leverage these findings to improve the design and delivery of the Prime Climb hints. We also plan to extend the Prime Climb student model to use eye-tracking data in real-time for assessing if and how a student is attending to hints, and intervene to increase attention when necessary.

References

1. de Castell, S., Jenson, J.: Digital Games for Education: When Meanings Play. *Intermedialities* 9, 45–54 (2007)
2. Van Eck, R.: Building Artificially Intelligent Learning Games. In: Gibson, D., Aldrich, C., Prensky, M. (eds.) *Games and Simulations in Online Learning: Research and Development Frameworks*, pp. 271–307. Information Science Pub. (2007)
3. Conati, C., Manske, M.: Evaluating Adaptive Feedback in an Educational Computer Game. In: Ruttkay, Z., Kipp, M., Nijholt, A., Vilhjálmsson, H.H. (eds.) *IVA 2009. LNCS*, vol. 5773, pp. 146–158. Springer, Heidelberg (2009)
4. Peirce, N., Conlan, O., Wade, V.: Adaptive Educational Games: Providing Non-invasive Personalised Learning Experiences. In: *Second IEEE International Conference on Digital Games and Intelligent Toys Based Education (DIGITEL 2008)*, Banff, Canada, pp. 28–35 (2008)
5. Johnson, W.L.: Serious use for a serious game on language learning. In: *Proc. of the 13th Int. Conf. on Artificial Intelligence in Education*, Los Angeles, USA (2007)
6. Conati, C., Merten, C.: Eye-tracking for user modeling in exploratory learning environments: An empirical evaluation. *Knowl.-Based Syst.* 20(6), 557–574 (2007)
7. Baker, R., Corbett, A., Roll, I., Koedinger, K.: Developing a generalizable detector of when students game the system. *User Model. User-Adapt. Interact.* 18(3) (2008)
8. Alevan, V., McLaren, B.M., Roll, I., Koedinger, K.R.: Toward Tutoring Help Seeking. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) *ITS 2004. LNCS*, vol. 3220, pp. 227–239. Springer, Heidelberg (2004)
9. Roll, I., Alevan, V., McLaren, B.M., Ryu, E., Baker, R.S.J.d., Koedinger, K.R.: The Help Tutor: Does Metacognitive Feedback Improve Students' Help-Seeking Actions, Skills and Learning? In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006. LNCS*, vol. 4053, pp. 360–369. Springer, Heidelberg (2006)
10. Bee, N., Wagner, J., André, E., Charles, F., Pizzi, D., Cavazza, M.: Interacting with a Gaze-Aware Virtual Character. In: *Workshop on Eye Gaze in Intelligent Human Machine Interaction, IUI 2010* (2010)
11. Prasov, Z., Chai, J.: What's in a gaze? the role of eye-gaze in reference resolution in multimodal conversational interfaces. In: *IUI 2008* (2008)
12. Muldner, K., Christopherson, R., Atkinson, R., Bursleson, W.: Investigating the Utility of Eye-Tracking Information on Affect and Reasoning for User Modeling. In: Houben, G.-J., McCalla, G., Pianesi, F., Zancanaro, M. (eds.) *UMAP 2009. LNCS*, vol. 5535, pp. 138–149. Springer, Heidelberg (2009)

13. Muir, M., Conati, C.: Understanding Student Attention to Adaptive Hints with Eye-Tracking. In: Ardissono, L., Kuflik, T. (eds.) UMAP 2011 Workshops. LNCS, vol. 7138, pp. 148–160. Springer, Heidelberg (2012)
14. Woolf, B.P.: Building intelligent interactive tutors: Student-Centered strategies for revolutionizing elearning. Morgan Kaufmann (2008)
15. Easterday, M., Alevan, V., Scheines, R., Carver, S.: Using Tutors to Improve Educational Games. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS, vol. 6738, pp. 63–71. Springer, Heidelberg (2011)
16. Linehan, C., Kirman, B., Lawson, S., Chan, G.: Practical, Appropriate, Empirically-Validated Guidelines for Designing Educational Games. In: CHI 2011, pp. 1979–1988 (2011)
17. Just, M., Carpenter, P.: The Psychology of Reading and Language Comprehension, Boston (1986)

Serious Game and Students' Learning Motivation: Effect of Context Using Prog&Play

Mathieu Muratet¹, Elisabeth Delozanne¹,
Patrice Torguet^{2,4}, and Fabienne Viallet^{3,4}

¹ LIP6, Université Pierre et Marie Curie, 4 place Jussieu,
75005 Paris, France

{mathieu.muratet,elisabeth.delozanne}@lip6.fr
² IRIT

patrice.torguet@irit.fr

³ UMR EFTS

fabienne.viallet@univ-tlse3.fr

⁴ Université Paul Sabatier, 118 route de Narbone, 31400 Toulouse, France

Abstract. This paper deals with an analysis of a large-scale use of Prog&Play¹, a game-based learning environment specially designed to teach the basics of programming to first year university students. The study relies mainly on a motivation survey completed by 182 students among 258 who used the serious game for 4 to 20 hours in seven different university settings. Our findings show that the students' interest for Prog&Play is not only related to the intrinsic game quality, it is also related to the teaching context and mainly to the course schedule and the way teachers organize sessions to benefit from the technology.

Keywords: Serious games, programming, algorithms, motivation.

1 Introduction

Many studies report the growing disinterest of students in developed countries for science in general and for computer science in particular [1,12]. To face an urgent need to improve the level of understanding of computer science as an academic and professional field, many countries are implementing curricula to teach computational thinking² [2,14]. At the same time, an important effort is underway to define pedagogical approaches that will make thinking in terms of computer science more accessible and attractive to all students. These approaches include international competitions between schools³ or between countries⁴. Other studies show that video games are a successful way to increase student motivation by making learning fun. For example, they support problem-based learning and

¹ Prog&Play is an open source serious game freely downloadable at
http://www.irit.fr/ProgAndPlay/index_en.php

² Programming Skills Development, <http://pskills.ced.tuc.gr/>

³ Bebras, <http://www.bebas.org/en/welcome>

⁴ International Olympiad in Informatics, <http://www.ioinformatics.org/>

experiential learning, and they provide immediate feedback, enabling students to self-assess their actions or strategies [11]. The work presented here is a contribution to that field of research. Our basic assumptions are (i) that video games are exciting for students, and (ii) that they can also provide a good context to embed the teaching of computer programming.

Our project, called Prog&Play, aims at increasing students' motivation for learning the basics of programming by writing programs to manipulate the units of a real-time strategy game (RTS). If students implement efficient strategies, they will improve their chance to defeat their enemies and to win missions. In a previous paper, we detailed the design, implementation and evaluation of Prog&Play [10]. In this paper, we investigate how students' motivation is related to the teaching context. First, we discuss background and related work. Then, we present the different experiments we conducted to test Prog&Play with 258 undergraduate students and 20 teachers in different university settings. Finally, we analyse the results to outline guidelines for a successful use of Prog&Play and suggest further avenues of research.

2 Background and Related Work

A popular use of a game-based learning approach to teach programming is asking students to implement their own video game. Chen and Cheng [3] use C++ to enable students to build a small-to-medium scale interactive computer game in one semester. Tools like Scratch [9] or Alice2 [7] are used to make first programming experiences more engaging.

Another approach consists in using programming games where the player has to write computer programs or scripts in order to control the actions of game units. In Colobot⁵, users colonise planets using robots that they program in a specific object-oriented language similar to C++. Other projects do not use a storytelling approach but rely on competition to increase motivation. Robocode [6] is a Java programming game, where the goal is to program a robot tank to fight against other tanks programmed by other players. Other such games are Gun-Tactyx⁶ using the SMALL language or Robot Battle⁷ using a specific script language.

In the Prog&Play project, to ensure contextual learning, we use a storytelling approach where students have to carry out missions as in Colobots, but it is also possible to organize competition between students' programs. Moreover, to adapt to different teaching contexts, Prog&Play provides a large choice of programming languages to command game units: Ada, C/C++, Compalgo, Java, OCaml and Scratch. Prog&Play relies on three basic principles: (i) learners program the game units with simple programs involving functions from a teacher customizable library; (ii) learners see the results of their programs in the game context where they influence the game results; and (iii) learners' engagement

⁵ Colobot, <http://www.cobot.com/colobot/index-e.php>

⁶ Gun-Tactyx, <http://apocalyx.sourceforge.net/guntactyx/>

⁷ Robot Battle, <http://www.robotbattle.com/>

is based on storytelling or competition. Our storytelling approach embeds the pedagogical objectives in different missions to be carried out. While our competitive approach motivates students to improve their programs in order to beat other players.

3 Evaluation

Our goals in designing and implementing Prog&Play were to produce benefits in terms of students' motivation and curricular-specific learning outcomes. As Prog&Play was not used as a standalone learning environment, but was used in different actual university settings, it was difficult to detect the learning outcomes due to Prog&Play or to the teachers' specific pedagogical strategy. To evaluate Prog&Play, (i) we used an iterative and collaborative design and evaluation method involving teachers in order to understand how they implement Prog&Play in the different introductory programming courses they were responsible for, and (ii) we delivered a post questionnaire to students. Our research question was: Is there a relationship between students motivation and the teaching context in which Prog&Play was used and which context is more beneficial?

3.1 Usage Settings and Participants

We studied usage of Prog&Play in seven different settings (noted S1 to S7) involving 258 students and 20 teachers. Teachers organized the pace, schedule and evaluation of students work with respect to their institutional constraints. No member of the Prog&Play design team was involved as a teacher in S4, S6 and S7. In S4 and S5, Prog&Play practice sessions were mandatory and integrated within the regular course, while in the other settings, it was used in addition to the regular course. In S6 and S7, both teachers especially designed courses called "Learning with Information Technology" and "Learning differently" to investigate new pedagogical approaches with Prog&Play in two different universities.

In every setting, Prog&Play was already installed on computers and a teacher was in the room presenting the teaching concepts, the environment, the library and providing help when asked by students. Only in the 6th setting, after 5 sessions with a teacher, students had to complete the game at home with the teacher's or peers' e-mail support to install the game or to debug their programs.

3.2 Materials

To collect information on students' motivation, we designed a questionnaire using the hierarchy of players' needs proposed by Siang and Rao [15] and Greitzer *et al.* [5]. These authors adapted Maslow's original hierarchy of needs to define seven criteria to be fulfilled to motivate players in a game: *rules need* (need 1); *safety need* (need 2); *belongingness need* (need 3); *esteem need* (need 4); *need to know and understand* (need 5); *aesthetic need* (need 6); and *self actualization need* (need 7). Following these authors, our assumption was that the degree of satisfaction within this hierarchy of needs was a significant indicator of motivation.

3.3 Results and Analysis

We considered only questionnaires that were fully completed by students (S1: 13/15; S2: 23/35; S3: 16/16; S4: 29/60; S5: 91/99; S6: 10/18; S7: 0/15; Total: 182/258). We compared (Table 1) students' satisfaction rates in each setting by means of Likert items on the seven need levels. Only a quarter of the students were satisfied in S4 and S5, where Prog&Play practice sessions were mandatory in the regular course schedule. In S1, S2 and S3 where Prog&Play was used in addition to the regular course schedule (as a workshop or practical exercises for students with low grades), the rate of satisfied students was 4 students out of 10. And in S6 where Prog&Play is used as a project assignment, the rate rose to 6 out of 10.

These results suggest that Prog&Play is better implemented within projects, workshops or supplementary practical sessions. We conjecture that Prog&Play is not a game that *teaches* computer programming basics, but it provides a micro-world [13] where students can explore the effects of their different programming constructs and learn from the feedback given by the micro-world. Students use taught programming concepts in an appealing context (RTS) whereas, in regular teaching, they are required to use them in a mathematical context (and they are evaluated using them in such an abstract context).

Table 1. Usage of Prog&Play in seven different settings and global satisfaction

	N	Language, Teaching context and Time spend on game	SR*
S1	15	Compalgo, Workshop apart from regular teaching, 5 * 1h30	4.6/10
S2	35	C, Practice for failing students in addition to regular teaching, 3 * 1h30	3.8/10
S3	16	Java, Workshop apart from regular teaching, 3 * 1h30	4.1/10
S4	60	C, Compulsory practice sessions for every student, 5 * 1h30	2.7/10
S5	99	OCaml, Compulsory practice sessions for every student, 2 * 2h	2.6/10
S6	18	C, Workshop part of a regular IT course, 6 * 2h + homework	6.3/10
S7	15	C, Workshop, regular teaching designed for failing students, 5 * 2h	unreported

* Satisfaction Rate

In addition, we hypothesize that the schedule is an important motivation factor. In regular teaching (S2, S3, S5), teachers split the game scenario into different sessions to fit the pace of programming concepts being introduced, whereas the gameplay would require a more continuous gameflow [4] built on the progression of the missions. Moreover, teachers urged students to finish on time by giving them a solution, while in a normal game session, players often enjoy finding solutions on their own. The course agenda is easier to adapt when the game is used as an add-on to the existing teaching materials (S6, S7).

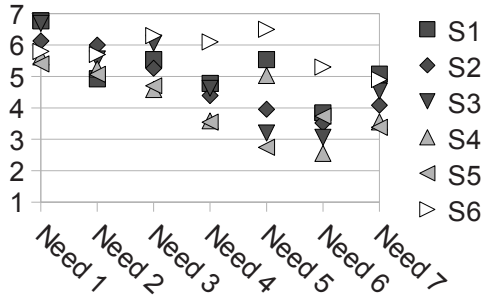


Fig. 1. Mean satisfaction for each player's need in each setting

Figure 1 studied the mean satisfaction for each need level of Siang and Rao's hierarchy in each setting. It shows a greater dispersion of answers on need 5 (Need to know and understand). Need 5 is defined as the necessity for the player to discover new game elements in order to reuse them in future parts of the game. In Prog&Play, this need is satisfied through discovering new units, with their own features, as well as new ways to command units (through programming constructs). This reinforces our hypothesis that the discovery part is important for motivation and learning and requires a sufficient exploration time for players. Satisfaction for need 5 seems therefore highly tied to the time allocated to the game in the teaching agenda.

4 Conclusion

In this paper, we have briefly described Prog&Play, a game-based learning environment, and presented data collected when it was introduced in different university settings. The questionnaires collected from students suggest a clear influence of the teaching setting on students' motivation: a workshop or a project based course in addition to a traditional introductory course, is clearly more beneficial than just plugging Prog&Play sessions within a traditional course. Furthermore, we identified that giving enough time to students to discover the game world and rules is a key feature to improve game understanding and therefore to increase their motivation.

Data collected suggest that, using a serious game only as an illustration tool inside regular teaching doesn't seem to be very beneficial to motivation. In S6 and S7 where there were less time constraints and where the game flow was continuous, students enjoyed advantages inherited from video games: they carried out actions within the game and observed their effects on the game to improve their knowledge of programming constructs. The opportunity for students to carry out useless, redundant or incorrect actions within a serious game providing feedback [16] is fundamental to catch players attention and to allows them to understand programming concepts deeply. A student in setting 6 described very well the motivation induced by exploring the game: *"The solution of the seventh mission took a long time to be achieved. Lots of ideas were considered*

and left unused. In the end, hundreds of code lines were written. I saw my army destroyed many many times. But, each attempt brought me closer to victory and kept me in suspense. Due to this suspense I completed this mission”.

Acknowledgments. We thank Rebecca Freund, Thomas Joufflineau and John Wisdom for helping with English, and teachers from Universities and IUT of Toulouse and Paris who used Prog&Play in their course.

References

1. ACM, IEEE-CS: Computer Science Curriculum 2008: An Interim Revision of CS 2001. ACM Press and IEEE Computer Society Press, New York (2008)
2. Archambault, P.: Un enseignement de la discipline informatique en Terminale scientifique. In: DIDAPRO 4 - Dida & STIC, pp. 205–212 (2011)
3. Chen, W.-K., Cheng, Y.C.: Teaching Object-Oriented Programming Laboratory With Computer Game Programming. IEEE Transactions on Education 50(3), 197–203 (2007)
4. Csikszentmihalyi, M.: Flow - The Psychology of optimal Experience. Harper Perennial (1991)
5. Greitzer, F.L., Kuchar, O.A., Huston, K.: Cognitive science implications for enhancing training effectiveness in a serious gaming context. J. Educ. Resour. Comput. 7(3), art. 2 (2007)
6. Hartness, K.: Robocode: using games to teach artificial intelligence. J. of Comput. Sciences in Colleges 19(4), 287–291 (2004)
7. Kelleher, C., Cosgrove, D., Culyba, D., Forlines, C., Pratt, J., Pausch, R.: Alice2: Programming without Syntax Errors. In: 15th Annual Symposium on the User Interface Software and Technology (2002)
8. Leutenegger, S., Edgington, J.: A games First Approach to Teaching Introductory Programming. In: 38th SIGCSE Technical Symposium on Computer Science Education, vol. 39(1), pp. 115–118 (2007)
9. Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., Resnick, M.: Scratch: A Sneak Preview. In: 2nd International Conference on Creating Connecting, and Collaborating through Computing, pp. 104–109 (2004)
10. Muratet, M., Torguet, P., Viallet, F., Jessel, J.-P.: Experimental feedback on Prog&Play: a serious game for programming practice. Computer Graphics Forum, Eurographics 30(1), 61–73 (2011)
11. Oblinger, D.: The Next Generation of Educational Engagement. J. of Interactive Media in Education (2004)
12. Papastergiou, M.: Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation. J. Comput. Educ. 52(1), 1–12 (2009)
13. Papert, S.: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, New York (1980)
14. Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O’Grady-Cuniff, D., Boucher Owens, B., Stephenson, C., Verno, A.: CSTA K-12 Computer Science Standards. CSTA Standards Task Force (2011) (revised 2011)
15. Siang, A.C., Rao, R.K.: Theories of learning: a computer game perspective. In: Multimedia Software Engineering, pp. 239–245 (2003)
16. Thomas, P., Yessad, A., Labat, J.-M.: Petri nets and ontologies: tools for the “learning player” assessment in serious games. In: Advanced Learning Technologies, pp. 415–419 (2011)

Exploring the Effects of Prior Video-Game Experience on Learner’s Motivation during Interactions with HeapMotiv

Lotfi Derbali and Claude Frasson

Département d’informatique et de recherche opérationnelle
Université de Montréal, C.P. 6128, Succ. Centre-ville Montréal, Québec, Canada H3C 3J7
{derbalil, frasson}@iro.umontreal.ca

Abstract. This study explores the effects of prior video-game experience on learner’s motivation in a serious game environment. 20 participants were invited to play our serious game, called HeapMotiv, intended to educate players about the heap data structure. HeapMotiv is comprised of three missions (*Heap-Tetris*, *Heap-Shoot* and *Heap-Sort*). We used Keller’s ARCS theoretical model of motivation and physiological sensors (heart rate, skin conductance and electroencephalogram) to record learners’ reactions during interactions with different missions. Results from non-parametric tests supported the hypothesis that physiological patterns and their evolution are objective tools to directly and reliably assess effects of prior video-game experience on learner’s motivation.

Keywords: Motivation, serious games, prior experience, physiological sensors, electroencephalogram (EEG).

1 Introduction

Even though motivation within learners tends to vary across subject areas, educators consider motivation to be desirable and to result in better learning outcomes. In Intelligent Tutoring Systems (ITS) context, learners’ interactions with systems and especially Serious Games (SG) have always been considered to be intrinsically motivating. However, learners’ negative emotions or amotivational states, such as boredom or disengagement, have been known to appear following a certain period of these interactions and possibly elicit motivational problems or even cause the learners to start “gaming” the system. Having tools to assess learner’s motivation during interactions with ITS is important to reduce, and eventually repair, motivational problems. Indeed, tutors can adapt their strategies and interventions and respond intelligently to learners’ needs, objectives and interests.

Furthermore, efforts to overcome learners’ motivational problems have mainly been focused on tutor’s strategies or instructional design aspects of the systems. For example, Hurley [1] developed interventional strategies to increase the learner’s self-efficacy and motivation in an online learning environment. She extracted and then validated rules for interventional strategy selection from expert teachers by using an

approach based on Bandura's Social Cognitive Theory and by observing the resulting learners' behaviour and progress. Goo and colleagues [2] showed that tactile feedback, sudden view point change, unique appearance and behaviour, and sound stimuli played an important factor in increasing students' attention in virtual reality experience. Arroyo and colleagues [3] evaluated the impact of a set of non-invasive interventions in an attempt to repair students' disengagement while solving geometry problems in a tutoring system. They claimed that showing students' performance after each problem re-engages students, enhances their learning, and improves their attitude towards learning as well as towards the tutoring software.

Some researchers have also found out strategies that teachers use in order to facilitate students' motivation toward tasks and goals of learning process. Teachers usually report that the proficiency in tasks may vary considerably depending upon the learners' familiarity and prior experience with themes, concepts, genre, characters, etc. Brandwein [4] clarified that teachers provide familiar tasks for students to construct understanding by connecting what they know with the essentials they are trying to learn. Wiley [5] defined interest as the state a student is in when s/he desires to know more about a subject and claimed that a student can be more interested in something s/he already knows about. He assumed that learners basic grounding in the subject and prior experience catalyze the construction of new, more coherent knowledge. He proposed to gain the interest by using concrete, real-life examples which will be familiar to the students, or when that is difficult, by using allegories or metaphors. Other studies have nevertheless shown that the creation of unfamiliar situations and events and paradoxical or conflicting experiences for the student facilitates attention and engagement (e.g., [6]). The learner's readiness to persevere when faced with unfamiliar and challenging learning situations opens up opportunities for success and achievement. Understanding the effects of prior experience on learner's motivation is of particular significance for our research work. In this paper, we precisely aim to assess the effects of prior video-game experience on learner's motivation during interactions with our SG called HeapMotiv. Our experimental study combines psychometric instruments with physiological recordings, namely heart rate (HR), skin conductance (SC) and electroencephalogram (EEG). We ask the following research question: What are relevant physiological patterns during learner's interactions with different missions of HeapMotiv and how are they correlated with learner's motivation and prior video-game experience?

2 Prior Video-Game Experience and Learner's Motivation

We developed a serious game, called HeapMotiv, which intends to educate players about the binary heap data structure. HeapMotiv is a 3D-labyrinth that has many routes with only one path that leads to the final destination. Along the paths of this labyrinth, several information signs were placed to help learners while finding destination. A learner has to play a mission before obtaining a sign direction. In its current version, HeapMotiv is comprised of three 2D-missions (*Heap-Tetris*, *Heap-Shoot* and

Heap-Sort), each intended to educate players about some basic concepts of binary heap, general purpose properties and application to sort elements of an array.

In the present study, we explore prior video-game experience in terms of the match between each mission of HeapMotiv and the learner’s previous experience with video games. From this viewpoint, a mission that involves objects and rules from a well-known game is considered as familiar and will be attributed to “with prior experience” class, whereas that involves objects and rules not consistent with previous learner experience is classed “without prior experience”. The common-sense assumption was made in the present study in order to divide different missions into with and without prior experience classes.

Table 1. Classification of missions of HeapMotiv

Mission	Description	Class
<i>Heap-Tetris</i>	It is based on traditional Tetris game where a learner has to move nodes during their falling using the arrows to fill a binary tree without violating the heap property.	“With prior experience” (Everybody is previously familiar with Tetris which is one of the greatest games of the entire time.)
<i>Heap-Shoot</i>	It is based on shooter games. A learner has to spot violations of shape and heap properties and has then to fix these violations by shooting misplaced nodes.	“With prior experience” (Most commonly, the purpose of a shooter game is to shoot opponents and proceed through missions without the player character dying. A common resource found in many shooter games is ammunition.)
<i>Heap-Sort</i>	It is a comparison-based sorting algorithm to create a sorted array. It begins by building a binary heap out of the data set, and then removing the largest item and placing it at the end of the partially sorted array.	“Without prior experience” (Although sorting algorithms were widespread used in different applications, their relative unfamiliarity impeded their acceptance for non computer science students. Heap-sort is then an unfamiliar algorithm compared to selection and insertion sorting algorithms.)

Furthermore, the ARCS model of motivation [6] has been chosen to theoretically assess learner’s motivation. Keller used existing research on motivational psychology to identify four categories of motivation: *Attention*, *Relevance*, *Confidence* and *Satisfaction*. We have also used objective measures that are not directly dependent on a learner’s perception. In our empirical approach, we relied on two non-invasive physiological sensors: HR and SC. These sensors are typically used to study human affective states. However, we decided to add another interesting and important sensor: EEG. Indeed, brainwave patterns have long been known to give valuable insight into

the human cognitive process and mental state. More precisely, our EEG analysis relies on the “attention ratio” or *Theta/Low-Beta* which is widely used in neurobehavioral studies [7]. According to [7], low-level attention is characterized by “a deviant pattern of baseline cortical activity, specifically increased slow-wave activity, primarily in the theta band, and decreased fast-wave activity, primarily in the beta band, often coupled”. It is also common knowledge within the neuro-scientific community that investigations of cerebral activity limited to one area of the brain may offer misleading information regarding complex states such as attention and motivation. We have therefore investigated different cerebral areas to study simultaneous brainwave changes.

3 Experiment

Twenty volunteers (10 female) were invited to play our serious game HeapMotiv in return of a fixed compensation (mean age was 23.7 ± 6.8 years). They had no prior knowledge of heap data structure. Almost all participants said they were either very or fairly familiar with Tetris (100%) or Shooter (78%) games while only 7% of them have been familiar with some kind of sort algorithms. This is consistent with our assumption in the classification of different missions in section 2.

Following the signature of a written informed consent form, each participant was placed in front of the computer monitor to play the game. SC and HR sensors were attached to the fingers of participants’ non-dominant hands, leaving the other free for the experimental task. EEG was recorded by using a cap with a linked-mastoid reference. The sensors were placed on four selected areas (Fz, F3, C3 and Pz) according to the international 10-20 system. The motivational measurement instrument called Instructional Materials Motivation Survey IMMS [6] was used following each mission to assess learners’ motivation. Due to time constraints and in order to achieve minimum disruption to learners, we used a short IMMS form which contained 16 out of the 32 items after receiving the advice and approval from John Keller. 10 pre-test and 10 post-test quizzes about general knowledge of binary tree and knowledge presented in HeapMotiv were also administered to compare learners’ performance. All participants have played each mission three times and have completed the game. EEG was sampled at a rate of 256 Hz. A power spectral density was computed to divide the EEG raw signal into the two following frequencies: *Theta* (4-8 Hz) and *Low-Beta* (12-20 Hz) in order to compute the attention ratio (*Theta/Low-Beta*) as described above. We also computed an index representing players’ physiological evolution throughout the mission with regards to each signal signification. This index, called Percent of Time (*PoT*), represents the amount of time, in percent, that learners’ signal amplitude is lower (or higher) than a specific threshold. The threshold considered for each signal is the group’s mean signal. The *PoT* index is a key metric enabling us to sum-up learners’ entire signal evolution for a mission.

4 Experimental Results

First, we report general results regarding learners' learning and motivation during interactions with different missions. We administered pre-tests and post-tests questionnaires pertaining to the knowledge taught in *HeapMotiv* and compared results. The Wilcoxon signed ranks test showed a significant positive change in learner's performance in terms of knowledge acquisition ($Z=5.03$, $p<.001$). Furthermore, significant differences for the general motivational scores as well as some categories of the ARCS model were also observed between two mission classes. Results of Wilcoxon signed ranks tests showed significant differences of reported *Attention/Confidence/Motivation* scores between *Heap-Tetris* and *Heap-Sort* missions (*Attention*: $Z=-2.59$, $p<.01$; *Confidence*: $Z=-2.53$, $p<.01$; *Motivation*: $Z=-1.93$, $p<.05$). Similar result have been found between *Heap-Shoot* and *Heap-Sort* missions, *except* for the *Attention* category (*Confidence*: $Z=-2.36$, $p<.05$; *Motivation*: $Z=-1.89$, $p<.05$). However, no significant differences for *Heap-Tetris* and *Heap-Shoot* missions, *except* for the *Attention* category ($Z=-2.02$, $p<.05$).

Second, we report results of correlation run on physiological data recorded during learners' interactions with different missions of *HeapMotiv*. Analysis of "with prior experience" class (*Heap-Tetris* and *Heap-Shoot* missions) showed that significant relationships between the *Attention* category and *PoT-F3* and *PoT-C3* indexes, as well as the general motivation and *PoT-C3* index (*Attention/PoT-F3*: spearman's $\rho=.49$, $n=40$, $p<.001$; *Attention/PoT-C3*: spearman's $\rho=.44$, $n=40$, $p<.01$; *Motivation/PoT-C3*: spearman's $\rho=.32$, $n=40$, $p<.05$). For "without prior experience" class (*Heap-Sort* mission), significant correlations have been found between motivation and *PoT-SC* index, as well as *Attention* and *PoT-F3* and *PoT-C3* indexes (*Motivation/PoT-SC*: spearman's $\rho=.51$, $n=20$, $p<.001$; *Attention/PoT-F3*: spearman's $\rho=.44$, $n=20$, $p<.01$; *Attention/PoT-C3*: spearman's $\rho=.36$, $n=20$, $p<.01$).

These results positively answer our main research question (What are relevant physiological patterns during learner's interactions with different missions of *HeapMotiv* and how are they correlated with learner's motivation and prior video-game experience?). Learners were more interested in something they already know about and consequently had high motivation during *Heap-Tetris* and *Heap-Shoot* missions. The *Heap-Sort* mission which belongs to "without prior experience" class led to a lack of learners' attention and confidence. One explanation may be that difficulties, doubt, and initial failure have been known to appear during learning a new skill or confronting unfamiliar challenges. Furthermore, effects of prior video-game experience on learners' attention and motivation can reliably be monitored and related to changes in the *PoT* of skin conductance and EEG F3 and C3 areas. However, even the significant differences between *Confidence* scores that learners were reported for two mission classes, non-significant correlation results between the *Confidence* category and physiological data have been found. This suggests that the potential of only using physiological analysis in our comparative study is limited because, up until now, we cannot totally rely on physiological assessment in the identification of the effects of prior video-game experience on general learner's motivation (or ARCS categories). One reason may be the limitation of the attention ratio (*Theta/Low-Beta*) which seems to be inappropriate to identify EEG patterns other than those correlated with the *Attention* category.

5 Conclusion and Future Work

In this paper, we have assessed the effects of prior video-game experience on learner's motivation. Results have shown that a mission that involves objects and rules from a well-known game and most closely matches previous experience seems to elicit specific physiological trends in learners, especially observable in the attention ratios. Our results seem to show the relevance and importance of adding the EEG in our empirical study. The present work is capable of extension in several directions. Regarding the physiological analysis, it is preferable to explore alternative EEG frequency ratios based on additional brainwaves such as *Alpha* (8-12 Hz) and *High-Beta* (20-32 Hz) in order to highlight other patterns correlated with learner's motivation. We also plan to address a complementary study to understand distinctive physiological changes associated with varying difficulty levels of different missions.

Acknowledgments. We thank the Tunisian government and the Natural Sciences and Engineering Research Council of Canada (NSERC) for their support. We also thank Tim Autin who has been working on HeapMotiv development.

References

1. Hurley, T.: Intervention Strategies to Increase Self-efficacy and Self-regulation in Adaptive On-Line Learning. In: Wade, V.P., Ashman, H., Smyth, B. (eds.) AH 2006. LNCS, vol. 4018, pp. 440–444. Springer, Heidelberg (2006)
2. Goo, J.J., Park, K.S., Lee, M., Park, J., Hahn, M., Ahn, H., Picard, R.W.: Effects of Guided and Unguided Style Learning on User Attention in a Virtual Environment. In: Pan, Z., Aylett, R.S., Diener, H., Jin, X., Göbel, S., Li, L. (eds.) Edutainment 2006. LNCS, vol. 3942, pp. 1208–1222. Springer, Heidelberg (2006)
3. Arroyo, I., Ferguson, K., Johns, J., Dragon, T., Mehranian, H., Fisher, D., Barto, A., Mahadevan, S., Woolf, B.: Repairing disengagement with non invasive interventions. In: The 13th International Conference on Artificial Intelligence in Education, pp. 195–202. IOS Press, Los Angeles (2007)
4. Brandwein, P.: Rethinking HowWe Do School—and for Whom. In: Tomlinson, C.A. (ed.) The Differentiated Classroom: Responding to the Needs of All Learners. Association for Supervision & Curriculum Development (1999)
5. Wiley, D.: Getting students interested: An integrated approach to Keller's ARCS model of motivational design. Instructional Design Project (2000)
6. Keller, J.M.: Development and use of the ARCS model of motivational design. *Journal of Instructional Development* 10, 2–10 (1987)
7. Lansbergen, M.M., Arns, M., van Dongen-Boomsma, M., Spronk, D., Buitelaar, J.K.: The increase in theta/beta ratio on resting-state EEG in boys with attention-deficit/hyperactivity disorder is mediated by slow alpha peak frequency. *Progress in Neuro-Psychopharmacology and Biological Psychiatry* 35, 47–52 (2011)

A Design Pattern Library for Mutual Understanding and Cooperation in Serious Game Design

Bertrand Marne¹, John Wisdom², Benjamin Huynh-Kim-Bang¹,
and Jean-Marc Labat¹

¹ LIP6, University Pierre et Marie Curie, 4 place Jussieu 75270 Paris, France
{Bertrand.Marne, Benjamin.Huynh-Kim-Bang,
Jean-Marc.Labat}@lip6.fr

² L'UTES, University Pierre et Marie Curie, 4 place Jussieu 75270 Paris, France
John.Wisdom@upmc.fr

Abstract. With serious games (SG) design it is difficult to offset fun and learning, especially when commercial partners, with different goals and methods, are involved. To produce an effective combination of fun and learning, we present our Design Pattern Library to address this issue. This library is aimed to help teachers fully take part in serious game design and to encourage mutual understanding between the different stakeholders enhancing cooperation.

Keywords: serious games, methodology, design patterns, game design, instructional design, cooperation, pedagogy.

1 Introduction

Serious game design usually comes down to how to help teachers understand the needs and methods of game-designers¹; and vice versa; furthermore, how to facilitate mutual understanding between these stakeholders and others involved in the design process.

To address these problems, one goal of our research team is to provide some design tools to facilitate collaboration, cooperation, and mutual understanding between the teachers, the designers and other stakeholders, not yet involved in the process of game creation.

We chose to build a Design Pattern Library integrating our conceptual framework based on six facets [1, 2] to allow everybody concerned to speak the same language, to be on the same conceptual wave length, and to allow some insight into the design process. We shall first discuss the previous work on Design Patterns and present our library. Next, we shall present our fieldwork applying the library to it.

¹ We can broadly group the stakeholders into two categories, the pedagogical experts and the game experts (by pedagogical experts or teachers we mean knowledge engineers, teachers, educators, and domain specialists. By game experts we mean game designers, level designers, game producers, sound and graphics designer, and so on).

2 Previous Work and Methodology

The state of the art on how to design serious games does not contain many references to Design Patterns. However, whether related work on ITS and video games, or guides to good practices or repositories of rules and principles, we find in the literature many elements that have formed the basis for our work. We studied DPs in education and e-learning, e.g. in Intelligent Tutoring Systems [3] or analyzing usage in learning systems [4]. But they do not take into account the game-playing dimension needed to design an SG.

The work specifically oriented towards serious games or at least video games seemed best suited to facilitate SG design. We therefore sought some aspects that might encompass the concept of Design Patterns as defined by Alexander [5] and described by Meszaros [6].

One of the first aspects is their organization. The list of eleven DPs for Educational Games, constructed from interviews with students (gamers) by Plass and Homer [7] lacks overall coherence. It seemed to us both difficult to use in fieldwork and to add to. The collaborative DP library, developed on the web by Barwood and Falstein [8], is another example. More than 400 patterns are (tag) referenced. But this very number would require much organization to facilitate the search for patterns and especially their use as a reference system for the various experts.

Gee [9] (a list of principles organized according to design problems), Aldrich [10] (a sophisticated encyclopædic DP library), and Schell [11] (questions for game designers organized according to workflow) provide an interesting structural framework for both their DP libraries or design methods. But we mostly retain the work of Kiili [12], and Björk & Holopainen [13] which is closest to Alexander's [5]. Indeed, their library has an overall coherence that is both simple to understand and functional. Their DPs refer to one another to create a Pattern Language. Moreover, Björk & Holopainen [13] insist that their Design Patterns are not intended to define what is good or to give guidelines, but to catalogue known references to build a vocabulary to enable participants to discuss design.

The latter DPs were similar to what we were looking for in DPs for serious games. Unfortunately, unlike those of Kiili [12], Plass and Homer [7], and even in a way Aldrich [10] and Schell [11], the work of Björk and Holopainen [13] is not at all oriented towards the serious (pedagogical) aspect of games. Their DPs can indeed be used for designing the fun aspect of an SG, but very few of them can contribute to combining both fun and education as we would like in an SG.

We have nevertheless retained some of them. Those retained as such are followed by the words "(GD)" in the list of our Design Patterns (Section 3.2). Other patterns were adapted, such as "*Serious Boss*", an adaptation of "*Boss Monster (GD)*".

On the other hand, the work of Kiili [12] focuses on the design of serious games and therefore on their serious dimension. But unlike Björk and Holopainen [13] (200 DPs from interviews with 7 game designers), and also Gee [9] (who examined many successful games involved in learning), and Schell [11] (who provides 100 "lenses" for analyzing the design of serious games from his experience as a game designer and producer of many games), etc., Kiili [12] built his library from his ex-

perience designing only one single game (*AnimalClass*). In fact, the library is still rather poor (8 DPs classified in 6 categories). However, their relevance is great and we decided to adopt some of them in our library. They are followed by “(K)” (See section 3.2). We compiled all our collected data covering different design experience, knowledge, and methods using the Design Patterns provided by Meszaros [6].

Focused on our goal of helping experts to collaborate on serious game design, we used an empirical method to build our Design Pattern Library. To discover new patterns, we also examined different sources and studied their content. For instance, we have made an in-depth analysis of six Serious Games mixing fun and education (*StarBank*, *Blossom Flowers*, *Hairz’ Island*, *Ludville* produced by KTM-Advance and *Donjons & Radon* produced by Ad-Invaders²).

3 Our DP Collaborative Library

The library consists of 42 Design Patterns classified within our conceptual framework: *The Six Facets of Serious Game Design* [1, 2]³. We shall present the entire library in a list where it is organized with the Facets (Section 3.2). But first, we shall present one example to illustrate how DPs can best be used: “*Reified Knowledge*” (Facet #4: *Problems and Progression*). Please note that Design Patterns are typically written in *italics*.

3.1 Pattern: *Reified Knowledge*⁴

Context: The particular game the team is designing involves a variety of competence and knowledge issues.

Problem: How can one help users become more aware of their acquired knowledge?

Forces: Several problems arise. How can we make the player aware of the progress he has made for each skill or activity without taking him out of the Flow? How can we use this type of information to enhance his/her motivation and enjoyment of the game?

Solution: Represent items of knowledge or competencies (skills) with virtual objects to be collected. If the player has acquired the requisite skill or piece of knowledge, he/she will be given an object symbolizing this or that knowledge acquisition.

For instance, in *America’s Army 3*, medals can be won when special deeds are accomplished. For example, a user wins a “*distinguished auto-rifleman*” medal when he/she has won 50 games as a rifleman in combat. Medals, however, do not further player progress in the game; and are more a way of reifying the playing style by rendering it concrete. The user can see his/her acquisitions either in knowledge or skills

² <http://www.ktm-advance.com> and <http://www.ad-invaders.com>

³ <http://seriousgames.lip6.fr/site/spip.php?page=facets>

⁴ <http://seriousgames.lip6.fr/site/?Reified-Knowledge>

embodied in medals awarded. Every medal is placed in a showcase, and thus is exhibited as a means of recapitulating what has been acquired.

Example: In *Ludiville* (a KTM-Advance game for a bank), knowledge about home loans is reified by beautiful trading cards (as in a game called *Magic the Gathering*). Once having learnt a new piece of knowledge, players obtain the related card, which they can use later in the game to meet new challenges

Related Patterns: *Object Collection*: also used to motivate players who like to collect things.

3.2 The Content of Our Design Pattern Library

- **[Facet #1] Pedagogical Objectives:** *Categorizing Skills, Price Gameplay vs. Educational Goals*
- **[Facet #2] Domain simulation:** *Simulate Specific Cases, Build a Model for Misconceptions, Elements that Cannot be Simulated, An Early Simulator, Do not Simulate Everything*
- **[Facet #3] Interactions with the simulation:** *Museum, Social Pedagogical Interaction, Serious Boss, Protege Effect (K), Advanced Indicators, Validate External Competencies, Questions – Answers, New Perspectives, Pedagogical Gameplay, Microworld Interaction, Time for Play /Time for Thought, Quick Feedbacks, Teachable Agent (K), In Situ Interaction, Pavlovian Interaction, Debriefing*
- **[Facet #4] Problems and Progression:** *Measurement achievements, Surprise, Smooth Learning Curve (GD), Fun Reward, Game Mastery, Freedom of Pace, Reified Knowledge*
- **[Facet #5] Decorum:** *Object Collection, Local Competition, Loquacious People, Graduation Ceremony, Fun Context, Wonderful World, Narrative Structures (GD), Serious Varied Gameplay, Informative Loading Screens, Hollywoodian Introduction, Comical World*
- **[Facet #6] Conditions of use:** *Two Learners Side by Side*

Our Collaborative Design Pattern Library for serious games is also available with full details on the web⁵.

4 Fieldwork and Discussion

We had the opportunity to test the Design Patterns on several occasions. First they were presented to 20 students of a video game school (ENJMIN⁶): future game designers, programmers, and project managers. We explained the concept of Design Pattern and each DP was shown to them. Next the students could ask questions to clarify the meaning. Finally, they answered a questionnaire on each of these Design Patterns.

⁵ <http://seriousgames.lip6.fr/DesignPatterns>

⁶ ENJMIN “*Ecole Nationale du Jeu et des Medias Interactifs Numériques*” is a video game school at Angoulême, France.

Secondly, Design Patterns have been tested by two teachers who wish to make serious games. One is a university English teacher to help French students apply for a graduate course in the USA. The other is a Junior High biology and geology teacher designing a game about the body's immune system. We will present the results and conclusions of these tests.

The ENJMIN students are video game experts, and so did not learn much about making video games from our Design Patterns. Indeed, at that time, our library contained mostly DPs describing game design. However, they were not yet versed in the area of serious games, and were indeed interested in having new Design Patterns based on the educational aspects of game design.

On the other hand, the two teachers were interested in those Design Patterns that were originally meant for game designers.

The first project, called *Graduate Admission*, was our first attempt to design a game with the help of the DP Library. We started by exploring the game design possibilities using the DP *Game-Based Learning Blend*. Several other DPs were used while the design process, such as: *Narrative structure (GD)*, *Time for Play /Time for Thought*, *Debriefing*, *Reified Knowledge*, etc. The DPs allowed the teacher to structure his project and to go deeper into the cultural and especially game design issues involved. For instance, without these tools, he would probably not have thought about the use of symbolic objects as metaphors for knowledge acquisition.

The second game project design focused on finding a suitable game type for teaching the immune system. The teacher chose to begin by exploring the DP Library for inspiration. The DP *Time for Play /Time for Thought* seemed very interesting both because it was adapted to the challenges posed by the teaching of immunology: the difficulty for the student to be able to keep in mind the matching mechanisms between body defenses and microbes while they are applying these in their activities (exercises). Moreover, he found the meta-cognitive aspect of this Design Pattern very stimulating. It finally allowed him to choose the right type of game play: Tower Defense. This type of game allows players first to prepare their strategies, then check, in an action phase, if the strategy is valid; and finally, they can move on to a reflective phase where they can adjust or modify their initial strategy and so on.

For both those teachers, the Design Pattern Library allowed them to find gameplay solutions for pedagogical problems.

As a conclusion, these first two opportunities to apply DPs to SG design has shown that they could indeed give one group of experts (the educational team) a language that would help them understand the aims, means, and methods of another group (game designers). Vice versa, we need to complete the DP Library with patterns focused more on pedagogy to allow the video game specialists to understand the skills and competences of the teachers in greater depth.

5 Conclusion and Future Avenues of Research

The Design Pattern Library fits well into our Six Facets Conceptual Framework and should in the long run enhance the game design process especially for those project

members who are not specialized in video games or pedagogical ones. However, it appears necessary to improve and to complete this library by focusing more on the latter field as the number of DPs here needs to be increased and completed in greater depth. To achieve this aim, we have created a collaborative web site⁷ where future members of our community can make suggestions and propose novel DPs of their own. Moreover they can vote and comment on Design Patterns, or translate them into another language.

References

1. Marne, B., Huynh-Kim-Bang, B., Labat, J.-M.: Articuler motivation et apprentissage grâce aux facettes du jeu sérieux. In: Actes de la Conférence EIAH 2011, pp. 69–80. Université de Mons, Mons (2011)
2. Capdevila Ibáñez, B., Marne, B., Labat, J.M.: Conceptual and Technical Frameworks for Serious Games. In: Proceedings of the 5th European Conference on Games Based Learning, pp. 81–87. Academic Publishing Limited, Reading (2011)
3. Devedzic, V., Harrer, A.: Software Patterns in ITS Architectures. *Int. J. Artif. Intell.* Ed. 15, 63–94 (2005)
4. Delozanne, E., Le Calvez, F., Merceron, A., Labat, J.: A Structured Set of Design Patterns for Learner’s Assessment. *Journal of Interactive Learning Research* 18, 309–333 (2007)
5. Alexander, C., Ishikawa, S., Silverstein, M.: A pattern language. Oxford University Press, US (1977)
6. Meszaros, G., Doble, J.: A pattern language for pattern writing. In: *Pattern Languages of Program Design-3*, pp. 529–574. Addison-Wesley Longman Publishing Co., Inc., Boston (1997)
7. Plass, J.L., Homer, B.D.: Educational Game Design Pattern Candidates. *Journal of Research in Science Teaching* 44, 133–153 (2009)
8. Barwood, H., Falstein, N.: The 400 Project, http://www.theinspiracy.com/400_project.html
9. Gee, J.P.: Good video games + good learning: collected essays on video games, learning, and literacy. Peter Lang, New York (2007)
10. Aldrich, C.: The complete guide to simulations and serious games: how the most valuable content will be created in the age beyond Gutenberg to Google. John Wiley and Sons, San Francisco (2009)
11. Schell, J.: *The Art of Game Design: A book of lenses*. Morgan Kaufmann (2008)
12. Kiili, K.: Foundation for problem-based gaming. *British Journal of Educational Technology* 38, 394–404 (2007)
13. Björk, S., Holopainen, J.: *Patterns in game design*. Cengage Learning (2005)

⁷ <http://seriousgames.lip6.fr/DesignPatterns>

Predicting Student Self-regulation Strategies in Game-Based Learning Environments

Jennifer Sabourin, Lucy R. Shores, Bradford W. Mott, and James C. Lester

North Carolina State University, Raleigh, North Carolina, USA
{jlrobiso, lrshores, bwmott, lester}@ncsu.edu

Abstract. Self-regulated learning behaviors such as goal setting and monitoring have been found to be key to students' success in a broad range of online learning environments. Consequently, understanding students' self-regulated learning behavior has been the subject of increasing interest in the intelligent tutoring systems community. Unfortunately, monitoring these behaviors in real-time has proven challenging. This paper presents an initial investigation of self-regulated learning in a game-based learning environment. Evidence of goal setting and monitoring behaviors is examined through students' text-based responses to update their 'status' in an in-game social network. Students are then classified into SRL-use categories that can later be predicted using machine learning techniques. This paper describes the methodology used to classify students and discusses initial analyses demonstrating the different learning and gameplay behaviors across students in different SRL-use categories. Finally, machine learning models capable of predicting these categories early into the student's interaction are presented. These models can be leveraged in future systems to provide adaptive scaffolding of self-regulation behaviors.

Keywords: Self-regulated learning, machine learning, early prediction.

1 Introduction

Understanding and facilitating students' self-regulated learning behaviors has been the subject of increasing attention in recent years. This line of investigation is fueled by evidence suggesting the strong role that self-regulatory behaviors play in a student's overall academic success [1]. Self-regulated learning (SRL) can be described as "the process by which students activate and sustain cognitions, behaviors, and affects that are systematically directed toward the attainment of goals" [2]. Unfortunately, students can demonstrate a wide range of fluency in their SRL behaviors [3] with some students lagging behind their peers in their ability to appropriately set and monitor learning goals.

For this reason, the ability to identify and support students' SRL strategies has been the focus of much work in the intelligent tutoring systems community [4,5,6]. Such work has focused primarily on examining SRL in highly structured problem-solving and learning environments. However, understanding and scaffolding students' SRL behaviors is especially important in open-ended learning environments where

goals may be less clear and students do not necessarily have a clear indicator of their progress. In order to be successful in this type of learning environment, students must actively identify and select their own goals and evaluate their progress accordingly. Unfortunately, students do not consistently demonstrate sufficient self-regulatory behaviors during interactions with these environments, which may reduce the educational potential of these systems [7,8]. Consequently, further investigation of the role of SRL in open-ended learning environments is crucial for understanding how these environments can be used as effective learning tools.

This work describes a preliminary investigation of self-regulatory behaviors of students in a game-based science mystery, CRYSTAL ISLAND. During interactions with the CRYSTAL ISLAND environment, students were prompted to report on their mood and status in a way that is similar to many social networking tools available today. Though students were not explicitly asked about their goals or progress, many students included this information in their short, typed status statements. This data is used to classify students into low, medium, and high self-regulated learning behavior classes. Based on these classifications we investigate differences in student learning and in-game behaviors in order to identify the role of SRL in CRYSTAL ISLAND. Machine learning models are then trained that are capable of accurately predicting students' SRL-use categories early into their interaction with the environment, offering the possibility for timely intervention. The implications of these results and areas of future work are then discussed.

2 Related Work

Self-regulated learning (SRL) is a term used to describe the behaviors of students who actively control their learning goals and outcomes [9]. Among other things, SRL involves students actively setting goals and making conscious choices to measure and evaluate their progress towards them. Self-regulated learners deliberately reflect on their knowledge and learning strategies and make adjustments based on past success and failure. While it seems all students apply self-regulatory behaviors during learning, the degree of competency is unfortunately broad, even among students of the same age [3]. Additionally, there is evidence that individuals who are better able to regulate their learning in intentional and reflective ways are more likely to achieve academic success [1]. To mediate these differences, intervention research focused on process goals and feedback has been conducted in traditional classrooms and has yielded positive results [9,10,11].

Beyond the traditional classroom, identifying and scaffolding SRL strategies has been a focus of much work in the intelligent tutoring systems community as well. For example, in MetaTutor, a hypermedia environment for learning biology, think-aloud protocols have been used to examine which strategies students use, while analysis of students' navigation through the hypermedia environment helps to identify profiles of self-regulated learners [6]. Similarly, researchers have identified patterns of behavior in the Betty's Brain system that are indicative of low and high levels of self-regulation [5]. Prompting students to use SRL strategies when these patterns of

behavior occur has shown promise in improving student learning. Conati *et al.* have examined the benefits of prompting students to self-explain when learning physics content in a computer-based learning environment [4].

While previous work has focused primarily on examining SRL in highly structured problem-solving and learning environments, there has also been work on identifying SRL behaviors in open-ended exploratory environments. For example, work by Shores *et al.* has examined early prediction of students' cognitive tool use in order to inform possible interventions and scaffolding [12]. Understanding and scaffolding student's SRL behaviors is especially important in open-ended learning environments where goals may be less clear and students do not necessarily have a clear indicator of their progress [13]. In order to be successful in this type of learning environment, students must actively identify and select their own goals and evaluate their progress accordingly. While the nature of the learning task may have implicit overarching goals such as 'completing the task' or 'learning a lot,' it is important for students to set more specific, concrete and measurable goals [14].

This work focuses on examining SRL within the context of narrative-centered learning. *Narrative-centered learning environments* are a class of serious games that tightly couple educational content and problem solving with interactive story scenarios. By contextualizing learning within narrative settings, narrative-centered learning environments tap into students' innate facilities for crafting and understanding stories [15]. Narrative-centered learning environments have been developed that teach negotiation skills [16] and foreign languages [17] through conversational interactions with virtual characters. Scientific inquiry has been realized in interactive mysteries where students play the roles of detectives [18,19]. While these environments are capable of providing rich, engaging experiences [18], they should not overload students by providing too many possible paths for learning [7]. Appropriate goal-setting is necessary to succeed in these learning environments, making the ability to recognize and support students' SRL strategies especially critical.

3 Method

An investigation of students' SRL behaviors was conducted with CRYSTAL ISLAND, a game-based learning environment being developed for the domain of microbiology that follows the standard course of study for eighth grade science in North Carolina. CRYSTAL ISLAND features a science mystery set on a recently discovered volcanic island. Students play the role of the protagonist, Alex, who is attempting to discover the identity and source of an unidentified disease plaguing a newly established research station. The story opens by introducing the student to the island and the members of the research team for which her father serves as the lead scientist. As members of the research team fall ill, it is her task to discover the cause and the specific source of the outbreak. Typical game play involves navigating the island, manipulating objects, taking notes, viewing posters, operating lab equipment, and talking with non-player characters to gather clues about the disease's source. To progress through the mystery, a student must explore the world and interact with other characters while forming questions, generating hypotheses, collecting data, and testing hypotheses.

Table 1. SRL Tagging Scheme

SRL Category	Description	Examples
<i>Specific reflection</i>	Student evaluates progress towards a specific goal or area of knowledge	“I am trying to find the food or drink that caused these people to get sick.” “Well...the influenza is looking more and more right. I think I'll try testing for mutagens or pathogens – [I] ruled out carcinogens”
<i>General reflection</i>	Student evaluates progress or knowledge but without referencing a particular goal	“I think I'm getting it” “I don't know what to do”
<i>Non-reflective</i>	Student describes what they are doing or lists a fact without providing an evaluation	“testing food” “in the lab”
<i>Unrelated</i>	Any statement which did not fall into the above three categories is considered unrelated, including non-word or unidentifiable statements	“having fun” “arghhh!”

A study with 296 eighth grade students was conducted. Participants interacted with CRYSTAL ISLAND in their school classroom, although the study was not directly integrated into their regular classroom activities. Pre-study materials were completed during the week prior to interacting with CRYSTAL ISLAND. The pre-study materials included a demographic survey, researcher-generated CRYSTAL ISLAND curriculum test, and several personality questionnaires including *personality* [20] and *goal orientation* [21]. Students were allowed approximately 55 minutes to attempt to solve the mystery. Immediately after solving the mystery, or after 55 minutes of interaction, students moved to a different room in order to complete several post-study questionnaires including the curriculum post-test.

Students' affect data were collected during the learning interactions through self-report prompts. Students were prompted every seven minutes to self-report their current mood and status through an in-game smartphone device. Students selected one emotion from a set of seven options, which consisted of the following: *anxious*, *bored*, *confused*, *curious*, *excited*, *focused*, and *frustrated*. After selecting an emotion, students were instructed to briefly type a few words about their current status in the game, similarly to how they might update their status in an online social network. These status reports were later tagged for SRL evidence use using the following four ranked classifications: (1) *specific reflection*, (2) *general reflection*, (3) *non-reflective statement*, or (4) *unrelated* (Table 1). This ranking was motivated by the observation that setting and reflecting upon goals is a hallmark of self-regulatory behavior and that specific goals are more beneficial than those that are more general [14]. Students were then given an overall SRL score based on the average score of their statements. An even tertiary split was then used to assign the students to a Low, Medium, and High SRL category.

4 Results

Data was collected from 296 eighth grade students from a rural North Carolina middle school. After removing instances with incomplete data or logging errors, there were 260 students remaining. Among the remaining students, there were 129 male and 131 female participants varying in age and ethnicity. A total of 1836 statements were collected, resulting in an average of 7.2 statements per student. All statements were tagged by one member of the research team with a second member of the research team tagging a randomly selected subset (10%) of the statements to assess the validity of the protocol. Inter-rater reliability was measured at $\kappa = 0.77$, which is an acceptable level of agreement. General reflective statements were the most common (37.2%), followed by unrelated (35.6%), specific reflections (18.3%) and finally non-reflective statements (9.0%).

4.1 Analyzing Self-Regulation Behaviors

The first objective of this investigation was to explore differences in student learning based on self-regulatory tendencies. Student learning, as measured by normalized learning gains from the pre-test to post-test, was compared for the three SRL groups. An ANOVA indicated a difference in learning gains between the groups ($F_{(2, 257)} = 4.6, p < 0.01$). Tukey post-hoc comparisons indicated that both High and Medium SRL students experienced significantly better learning gains than Low SRL students at the $\alpha = 0.05$ level. Analyses also indicated that there were significant differences on pre-test scores between groups ($F_{(2, 257)} = 5.07, p < 0.01$) suggesting that students with high SRL tendencies may be better students or perhaps their increased prior knowledge helped them to identify and evaluate their goals more efficiently. Figure 1 shows the pre- and post- test scores across groups, highlighting both the differences in pre-knowledge and learning during interaction with CRYSTAL ISLAND.

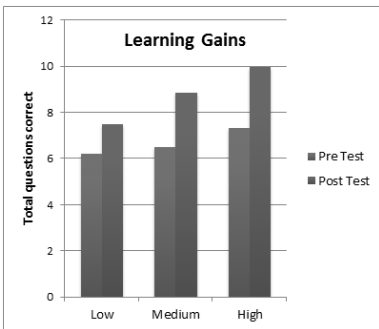


Fig. 1. Learning gains by SRL group

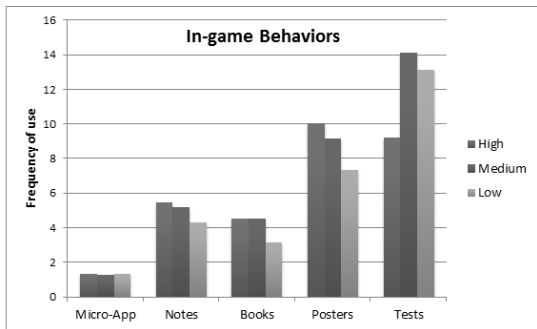


Fig. 2. In-game behaviors by SRL group

The next set of analyses was conducted to investigate differences in student behavior based on their SRL tendencies. A chi-squared analysis indicated that the percentage of students who solved the mystery did not differ significantly based on SRL group ($\chi^2(2, N=260) = 4.72, p = 0.094$). Additionally, an ANOVA indicated there was no significant difference in the number of goals completed during the interaction.

While a significant difference in students' abilities to solve the mystery was not found, there were differences in the in-game resources that students used. Resources expected to be most beneficial to learning and self-regulation included a *microbiology app* on the students' in-game smartphone which provides a wealth of microbiology information, *books* and *posters* that are scattered around the island with additional information, a notebook where students can record their own *notes*, and finally a *testing* machine where students formulate hypotheses and run the relevant tests. ANOVAs for student use of each of these features indicated a significant difference in student use of *posters* ($F_{(2, 257)} = 5.28, p < 0.01$), and *tests* ($F_{(2, 257)} = 5.59, p < 0.01$). While the differences in the use of other devices were not significant, interesting trends emerged (Figure 2). High SRL students appear to make more use of the curricular resources in the game such as books and posters and also take more notes than the lower SRL students. Interestingly, High SRL students run significantly fewer tests than Medium or Low SRL students (as indicated by Tukey post-hoc comparisons). Abundant use of the testing device is often indicative of students gaming the system or failing to form good hypotheses in advance. This finding suggests that High SRL students may be more carefully selecting which tests to run and are perhaps obtaining positive test results earlier than Medium and Low SRL students.

4.2 Predicting Self-Regulation Behaviors

These results highlight several important factors relating to self-regulation. First, the post-interaction method of classifying students into Low, Medium, and High SRL categories appears to yield meaningful groupings of students. Second, these classifications have significant implications for student learning. Students in the High SRL group have a higher level of initial knowledge than Low SRL students and through interactions with CRYSTAL ISLAND, increase this gap in knowledge. This highlights the importance of identifying the Low SRL students so they can receive supplementary guidance to help bridge this gap. Finally, the results indicate that High SRL students utilize the environment's curricular features differently and likely more effectively than Low SRL students. This finding suggests that scaffolding to direct Low SRL students towards more effective use of these resources could be an appropriate mechanism for bridging the learning gap.

However, in order to make use of these findings, Low SRL students must be identified early into the interaction so they can be provided with the necessary scaffolding. The current procedure for identifying these students is performed manually after the interaction has been completed, which does not allow for early interventions. It is also desirable to only provide additional scaffolding to the Low SRL students since the other students appear to be effectively using the environment already and may potentially be harmed by additional interventions. For these reasons, the next goal of this research was to train machine-learning models to predict students' SRL-use categories early into their interaction with CRYSTAL ISLAND.

Table 2. Predictive models and evaluation metrics (for predictive accuracy, * and ** indicate a significant improvement over the prior prediction at $p < .05$ and $.01$, respectively)

<i>Model</i>	<i>Predictive Accuracy</i>				<i>Low-SRL Recall</i>			
	Initial	Report ₁	Report ₂	Report ₃	Initial	Report ₁	Report ₂	Report ₃
Naïve Bayes	44.2	43.5	46.1*	50.5*	0.47	0.28	0.54	0.52
Neural Network	42.3	43.8	46.5*	45.5	0.44	0.45	0.49	0.52
Log. Reg.	42.7	51.2**	47.7	54.5**	0.45	0.65	0.66	0.73
SVM	43.5	46.9*	45.7	51.4**	0.51	0.55	0.56	0.62
Decision Tree	42.7	46.2*	48.1*	57.2**	0.45	0.55	0.71	0.71

In order to predict students' SRL-use categories, a total of 49 features were used to train machine-learning models. Of these, 26 features represented personal data collected prior to the student's interaction with CRYSTAL ISLAND. This included demographic information, pre-test score, and scores on the personality, goal orientation, and emotion regulation questionnaires. The remaining 23 features represented a summary of students' interactions in the environments. This included information on how students used each of the curricular resources, how many in-game goals they had completed, as well as evidence of off-task behavior. Additionally, data from the students' self-reports were included, such as the most recent emotion report and the character count of their "status."

In order to examine early prediction of the students' SRL-use categories, these features were calculated at four different points in time resulting in four distinct datasets. The first of these (**Initial**) represented information available at the beginning of the student's interaction and consequently only contained the 26 personal attributes. Each of the remaining three datasets (**Report_{1,3}**) contained data representing the student's progress at each of the first three emotion self-report instances. These datasets contained the same 26 personal attributes, but the values of the remaining 23 in-game attributes differentially reflected the student's progress up until that point. The first self-report occurred approximately 4 minutes into game play with the second and third reports occurring at 11 minutes and 18 minutes, respectively. The third report occurs after approximately one-third of the total time allotted for interaction has been completed, so it is still fairly early into the interaction time.

Each of these datasets was used to train a set of machine learning classifiers including: Naïve Bayes, Decision Tree, Support Vector Machine, Logistic Regression, and Neural Network. These models were trained and evaluated using 10-fold cross-validation with the WEKA machine learning toolkit [22]. The predictive accuracies of these models are shown in Table 2. All of the learned models were able to offer a predictive accuracy statistically significantly better than a most-frequent class baseline (at $p < 0.01$). Due to the fact that the classes were identified using an even tertiary split, the most frequent class (Medium) model has a predictive accuracy of 33.5%. Additionally, most models demonstrated gains in predictive accuracy further into the interaction.

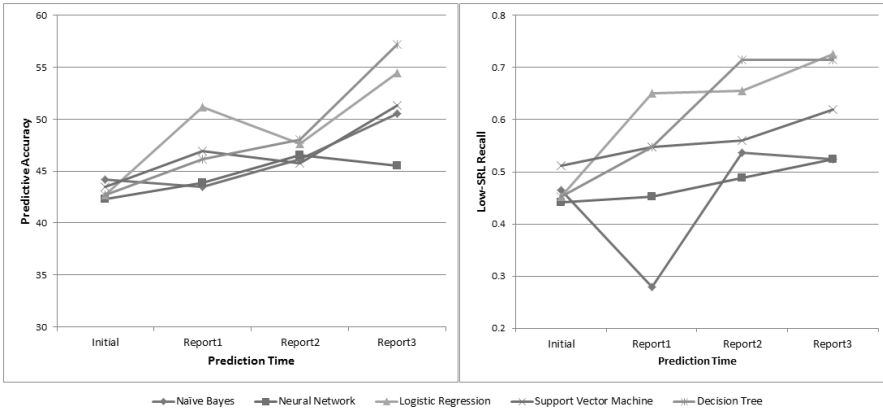


Fig. 3. Predictive accuracy and Low-SRL recall improvements across time

Of the models attempting to predict SRL class before any interaction with the environment, the model with the best performance is the Naïve Bayes model (44.3%). However, there are no significant differences in predictive accuracy between any of the models trained on this dataset. Alternatively, of the models trained with the most data, the Decision Tree model achieves the highest predictive accuracy (57.2%), and is statistically significantly better than the other models trained on this dataset ($p < 0.05$). In general, it appears that the two models with the best overall performance are the Decision Tree and Logistic Regression models.

In addition to predictive accuracy, we are also particularly interested in the models’ abilities to distinguish Low SRL students as these students would be the targets of additional support. For this reason, we compared the models’ levels of recall for the Low SRL class (Figure 3). These results again demonstrate a steady growth in the ability to correctly recognize Low SRL students. Additionally, the Decision Tree and Logistic Regression models again distinguish themselves in their ability to outperform the remaining models. These results indicate that using either model, or perhaps a combination of both models, will offer promise in being able to identify and support Low SRL students early into their interaction with CRYSTAL ISLAND.

5 Discussion

This work presents an initial analysis of students’ natural self-regulated learning activities in the narrative-centered learning environment, CRYSTAL ISLAND. Results indicate that undirected prompts have the potential to show students’ use of goal setting and monitoring. Additionally, the findings suggest that self-regulated learners tend to make better use of in-game curricular resources and may be more deliberate in their actions. Though highly self-regulated learners were not more likely to solve the mystery, they did demonstrate significantly higher learning gains as a result of their interaction. These results point to the importance of being able to identify students with tendencies towards low self-regulation in order to provide appropriate

scaffolding. The machine learning models discussed in this paper show significant promise in being able to predict a student's SRL abilities early into their interaction with CRYSTAL ISLAND.

These findings point to several natural directions for future work. The most prominent of these is developing intervention mechanisms for aiding student self-regulation. Specifically, the results of this work point to the ways that in-game curricular resources can be used effectively. Low SRL students could receive additional support in their use of these resources. Alternatively, it may be that these students suffer in their abilities to recognize and set appropriate goals. This goal-setting behavior could be made more explicit using the game-based nature of the environment.

Understanding how to effectively incorporate these strategies into narrative-centered learning environments is an important area for future investigation. Drawing on ongoing empirical investigations of learning, problem solving, and engagement can support the exploration of a broad range of potential techniques for further enhancing student SRL skills. In particular, investigating individualized instruction strategies and designing SRL features for narrative environments that account for individual differences is an important next step in this line of investigation.

Acknowledgments. The authors wish to thank members of the IntelliMedia Group for their assistance, Omer Sturlovich and Pavel Turzo for use of their 3D model libraries, and Valve Software for access to the Source™ engine and SDK. This research was supported by the National Science Foundation under Grants REC-0632450, DRL-0822200, and IIS-0812291. This material is based upon work supported under a National Science Foundation Graduate Research Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Zimmerman, B.J.: Self-regulated learning and academic achievement: An overview. *Educational Psychologist* 25, 3–17 (1990)
2. Schunk, D.H.: Attributions as Motivators of Self-Regulated Learning. In: Schunk, D.H., Zimmerman, B.J. (eds.) *Motivation and Self-Regulated Learning: Theory, Research, and Applications*, pp. 245–266 (2008)
3. Ellis, D., Zimmerman, B.J.: Enhancing self-monitoring during self-regulated learning of speech. In: Hartman, H.J. (ed.), pp. 205–228. Kluwer, Dordrecht (2001)
4. Conati, C., VanLehn, K.: Towards Computer-Based Support of Meta-Cognitive Skills: a Computational Framework to Coach Self-Explanation. *International Journal of Artificial Intelligence in Education* 11, 398–415 (2000)
5. Biswas, G., Jeong, H., Roscoe, R., Sulcer, B.: Promoting Motivation and Self-Regulated Learning Skills through Social Interactions in Agent-Based Learning Environments. In: 2009 AAAI Fall Symposium on Cognitive and Metacognitive Educational Systems (2009)
6. Azevedo, R., Johnson, A., Chauncey, A., Burkett, C.: Self-Regulated Learning with Meta-Tutor: Advancing the Science of Learning with MetaCognitive Tools. In: Khine, M., Saleh, I. (eds.) *New Science of Learning: Cognition, Computers and Collaboration in Education*, pp. 225–248. Springer, New York (2010)

7. Kirschner, P.A., Sweller, J., Clark, R.E.: Why Minimal Guidance during instruction does not work: An analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. *Educational Psychologist* 41, 75–86 (2006)
8. Alfieri, L., Brooks, P., Aldrich, N., Tenenbaum, H.: Does Discovery-Based Instruction Enhance Learning. *Journal of Education Psychology* 103(1), 1–18 (2011)
9. Schunk, D.H., Zimmerman, B.J.: Self-regulation and learning. In: Reynolds, W.M., Miller, G.E. (eds.), vol. 7, pp. 59–78. Wiley & Sons, New York (2003)
10. Schunk, D.H., Swartz, C.W.: Goals and progress feedback: Effects on self-efficacy and writing achievement. *Contemporary Educational Psychology* 18(3), 337–354 (1993)
11. Schunk, D.H., Swartz, C.W.: Writing strategy instruction with gifted students: Effects of goals and feedback on self-efficacy and skills. *Roeper Review* 15(4), 225–230 (1993)
12. Shores, L.R., Rowe, J.P., Lester, J.C.: Early Prediction of Cognitive Tool Use in Narrative-Centered Learning Environments. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS, vol. 6738, pp. 320–327. Springer, Heidelberg (2011)
13. Land, S.: Cognitive requirements for learning with open-ended learning environments. *Educational Technology Research and Development* 48(3), 61–78 (2000)
14. Zimmerman, B.: Goal Setting: A Key Proactive Source of Academic Self-Regulation. In: Schunk, D.H., Zimmerman, B.J. (eds.) *Motivation and Self-Regulated Learning: Theory, Research, and Applications*, pp. 267–286. Routledge, New York (2008)
15. Bruner, J.S.: *Acts of Meaning*. Harvard University Press, Cambridge (1990)
16. Kim, J., et al.: BiLAT: A game-based environment for practicing negotiation in a cultural context. *International Journal of Artificial Intelligence in Education* 19(3), 289–308 (2009)
17. Johnson, W.L.: Serious Use of a Serious Game for Language Learning. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence in Education*, pp. 67–74 (2007)
18. Rowe, J.P., Shores, L.R., Mott, B.W., Lester, J.C.: Integrating Learning, Problem Solving, and Engagement in Narrative-Centered Learning Environments. *International Journal of Artificial Intelligence in Education* (2011)
19. Ketelhut, D.J.: The impact of student self-efficacy on scientific inquiry skills: An exploratory investigation in ‘River City’, a multi-user virtual environment. *Journal of Science Education and Technology* 16(1), 99–111 (2007)
20. McCrae, R., Costa, P.: *Personality in Adulthood: A Five-Factor Theory Perspective*, 2nd edn. Guilford Press, New York (1993)
21. Elliot, A., McGregor, H.A.: A 2 x 2 achievement goal framework. *Journal of Personality and Social Psychology* 80, 501–519 (2001)
22. Hall, M., Frank, E., Holmes, G., Pfahring, B., Reutmann, P., Witten, I.: The WEKA Data Mining Software: An Update. *SIGKDD Explorations* 11(1) (2009)

Toward Automatic Verification of Multiagent Systems for Training Simulations

Ning Wang¹, David V. Pynadath², and Stacy C. Marsella²

¹ Curious Lab LLC, Westchester, CA USA
ningwang@curiouslab.com

² USC Institute for Creative Technologies, Playa Vista, CA USA
{pynadath,marsella}@ict.usc.edu

Abstract. Advances in multiagent systems have led to their successful application in experiential training simulations, where students learn by interacting with agents who represent people, groups, structures, etc. These multiagent simulations must model the training scenario so that the students' success is correlated with the degree to which they follow the intended pedagogy. As these simulations increase in size and richness, it becomes harder to guarantee that the agents accurately encode the pedagogy. Testing with human subjects provides the most accurate feedback, but it can explore only a limited subspace of simulation paths. In this paper, we present a mechanism for using human data to verify the degree to which the simulation encodes the intended pedagogy. Starting with an analysis of data from a deployed multiagent training simulation, we then present an automated mechanism for using the human data to generate a distribution appropriate for sampling simulation paths. By generalizing from a small set of human data, the automated approach can systematically explore a much larger space of possible training paths and verify the degree to which a multiagent training simulation adheres to its intended pedagogy.

Keywords: multiagent training simulation, serious games.

1 Introduction

Virtual worlds inhabited by autonomous agents are increasingly being used for experiential training and education (e.g., [1,5,8,12,14]). These virtual worlds provide an engaging environment in which students develop skills that can transfer to real-world tasks. To faithfully capture unpredictable real-world settings, simulations are populated by synthetic agents that ideally exhibit the same kind of complex behaviors that humans would exhibit [8,14]. The creation of these environments raises considerable challenges. Foremost, a student's experience in the environment must be consistent with pedagogical goals and doctrine. Notably, success and failure in the environment must be aligned with the skills and knowledge that the system is designed to teach.

From an instructional perspective, the use of complex multiagent virtual environments raises several concerns. The central question is what is the student learning—is it consistent with training doctrine and will it lead to improved student's performance?

Negative training can arise in training environments due to discrepancies between simulation and the real world, as well as discrepancies between simulation and pedagogical goals. With inaccurate models, undesirable strategies may instead appear effective, leading one to become overconfident in their likelihood of success. Strategies may also be locally successful in the simulation but violate broader pedagogical and doctrinal concerns and lead to failure in larger, real-world contexts. For example, while eliminating political opposition may succeed in a local urban simulation, it may profoundly violate doctrine by leading to very negative consequences in a more global context.

As these simulations increase in size and richness, it becomes harder to verify (let alone guarantee) that they accurately encode the pedagogy. Human subject playtesting provides accurate data. But it explores only a limited subspace of simulation paths due to the high cost, in time and money. Although multiagent systems support automatic exploration of many more paths than is possible with real people, the enormous space of possible simulation paths in any nontrivial training simulation prohibits an exhaustive exploration of all contingencies.

However, many of these contingencies are very unlikely to ever be realized by a student. Specifically, a student is highly unlikely to perform actions randomly without regard to their effects. Consequently, presuming a student is sampling from a uniform distribution of all possible action sequences is a poor starting point for evaluating a complex multiagent based social simulation.

We present an automated mechanism that instead tests only those paths that we can expect from real human behavior. We first analyze a multiagent training simulation already deployed in classrooms. The result shows that, while the vast majority of students received appropriate feedback from the multiagent system, some students were able to succeed despite violating the pedagogy. Given this motivating example, we then present an automated mechanism for using the human data to generate a distribution appropriate for sampling simulation paths. Our combined mechanism can thus systematically explore a much larger space of possible training paths and verify the degree to which a multiagent simulation adheres to its intended pedagogy.

2 PsychSim and UrbanSim

While our methodology applies to many agent-based simulations, we use PsychSim as our example architecture [7,11]. PsychSim is a social simulation tool for modeling a diverse set of entities (e.g., people, groups, structures), each with its own goals, private beliefs, and mental models about other entities. Each agent generates its beliefs and behavior by solving a partially observable Markov decision problem (POMDP) [4].

Multiple training simulations use PsychSim to generate behavior for the people, groups, and environment that students interact with to practice skills in a safe but realistic setting. The Tactical Language Training System helps students acquire communicative skills in foreign languages and cultures, where PsychSim agents represented villagers with whom the student develops rapport through conversation [13]. BiLAT uses PsychSim agents to engage students in bilateral negotiations in face-to-face meetings within a specific cultural context [5]. PsychSim agents also teach people to avoid risky behavior by simulating situations with pressure to engage in such behavior [6,9].

In this paper, we focus on UrbanSim, a simulation-based training system that has been deployed to teach stabilization operations in post-conflict urban environments [8]. The student directs multiple military units to execute operations in the context of a fictional urban scenario. The student's goal is to make progress along multiple dimensions (e.g., economic, political, security), called Lines Of Effort (LOEs). PsychSim agents generate the behavior for people, groups, and structures, as well as computing the effects of the students' decisions on their states. In the scenario used in this paper, there were 88 such agents and 6 real-valued LOEs derived from their states. The students give commands to 11 units under their control, after which PsychSim agents observe the commands' effects, choose their own counteractions, and observe those counteractions' effects. This cycle repeats for 15 rounds, with the students getting feedback each round through their LOE scores and a partial view of the scenario state.

3 Evaluation of Pedagogy

Although UrbanSim has been successfully deployed in classroom, the question remains about how well the multiagent component correctly encodes the intended pedagogy. That pedagogy relates to the strategies in selecting commands to give to units based on current state of the world and phase of the mission. The goal of this training simulation is for the students' scores to be positively correlated with how well their action choices satisfy the intended pedagogy. UrbanSim gives students more than 3000 possible ways to deploy their 11 units for each of the 15 rounds, thus producing 10^{26} possible strategies. Given the impracticality of exhaustive enumeration of that strategy space using agent-based simulation, we instead used playtesting to explore only a subset.

3.1 Study Population

We recruited 58 participants (56 male, 2 female) from a US metropolitan area. 35% of them are between 18 and 35, 14% are between 36 and 45 and 16% are above 45 years of age. 11% of the participants have high school education or GED, 79% have some college education or college degree, 10% have some graduate education or a graduate degree. 21% of the participants spend 1-4 hours using computer daily, 79% spend more than 5 hours. 6% of the participants have not or only played video games several times in the past year, 9% play video games monthly, 28% play weekly and 58% play video games daily. 70% of the participants did not spend any time in active military duty.

3.2 Experiment Manipulation and Procedure

When UrbanSim is deployed in the classroom, students are first shown a usability video about basic operations in UrbanSim and then a pedagogy video on the desirable strategies to use in UrbanSim. In the pedagogy video, participants are taught to:

1. Consider a non-aggressive approach as an alternative to the oft-preferred aggressive approach. For example, attacking a group is an aggressive action while hosting a meeting with the local mayor is a non-aggressive action.

2. Direct units under command to carry out Clear actions first, then Hold actions and finally Build actions. Clear, Hold and Build are not types of actions, but effects of an action. The Clear effect of an action is to remove potential danger in an area. The Hold effect is to protect an area that has danger already removed. The Build effect is to help a secured area recover and prosper. Each action has a weighted effect on Clear, Hold and Build, e.g. advising a local mayor can affect both Hold and Build.
3. Plan ahead instead of being purely reactionary, e.g discouraging “Whack-a-Mole”.

To encourage a greater diversity of strategies, one group of participants watches only the usability video (NoInstruction) and a second group watches both videos (WithInstruction). NoInstruction participants first fill out a consent form and a demographic background questionnaire, then watch the usability video. Next, they practice basic operations in UrbanSim for 15-20 minutes. After that, the participants interact with UrbanSim for 2 hours. Finally, they fill out the post-questionnaire. The procedure for the WithInstruction group is identical except that participants watch the pedagogy video following the usability video. There are 32 participants in the NoInstruction group and 26 participants in the WithInstruction group.

3.3 Measures

Demographic background questionnaire: asked questions about participant’s age, education, video game experience, computer use experience and military background.

Post questionnaire: contains questions regarding the strategies that participants used in UrbanSim, perceived importance of people and groups in the scenario (e.g. police, tribes), perceived importance of the LOEs, self-efficacy of improving LOEs and their assessment of the effect of the training simulation actions on LOEs, e.g. the impact of patrolling a neighborhood on the economy, security, etc.

Training Simulation logs: captures the actions chosen by each participant for each unit for each turn, LOE scores before each turn was committed, final score of popular support, and final score for LOEs. We categorized participants’ actions for each turn as whether they are Clear, Hold or Build actions and which LOEs they address.

3.4 Results

One participant’s data was excluded from the analysis because the participant had no experience using a computer. A total of 57 participants’ data are included in the analysis.

Encoding of Pedagogy. The first aspect of the pedagogy is to consider non-aggressive action as an alternative to aggressive actions. So overall, we should observe participants performing more non-aggressive actions than aggressive actions.

Pedagogy 1: *Number of Non-aggressive Actions > Number of Aggressive Actions*

The second aspect of the pedagogy is to follow a Clear → Hold → Build strategy. We summed up the number of actions carried out by the 11 units during the first third (turns

1 to 5), second third (turns 6 to 10) and last third (turns 11 to 15) of the game. We then ranked the Clear, Hold and Build effect of all the actions in each third. If the effect on Clear is higher than Hold and Build, we then categorize that third as Clear focused. There are 171 thirds from 57 participants. Only 3% of the thirds are Hold focused, so we chose to ignore Hold and instead categorized only the Clear and Build effects of the actions of the first half of the game (turn 1 to 7) and second half (turn 8 to 15) of the game. Following this categorization, the pedagogy is still very clear: a student should secure an area through Clear actions before performing Build actions in that area.

Pedagogy 2: Clear → Build

Effect of Experiment Manipulation. We conducted an ANOVA test on the percentage of non-aggressive actions participants took, and a CHI-Squared Goodness of Fit test on whether participants adhered to the two pedagogies, using the NoInstruction and WithInstruction groups as independent variables. Additionally, we compared the score on LOEs between two experiment groups using the ANOVA test. Results show that there was no significant difference between our two experiment groups on participants’ use of non-aggressive (NA) actions ($N = 57, p = .45$), whether they followed the pedagogy ($N = 54, p = .53$) and their performance on LOEs ($N = 47, p = .78$).

Table 1. Mean percentage of non-aggressive actions, number of participants following pedagogy, and mean LOE scores

		No Instruction	With Instruction
NA Actions		0.788	0.802
Followed Pedagogy	Yes	20	14
	No	10	10
LOE Score		361.4	358.1

Effect of Pedagogy. Because there are no significant differences between the two experiment groups on the variables we are interested in, we combined the data from the two groups for the following analysis. Overall, we found that all the participants overwhelmingly adopted Pedagogy 1, choosing more non-aggressive actions (79%) than aggressive actions (21%). This means that we do not have data to compare scores between participants who followed Pedagogy 1 and those who did not. We will focus on Pedagogy 2 for the remainder of the analysis.

We then conducted an ANOVA test on performance on LOEs between participants who followed Pedagogy 2 and those who did not. Overall, there is a significant difference on performance on LOEs between participants who followed the pedagogy and those who did not. People who followed the intended pedagogy (Clear → Build) performed better on the LOE scores than those who did not ($M_{NotFollow} = 330.4, M_{Follow} = 377.1, N = 45, p < .001$).

Figure 1a shows that the distribution of LOE scores from participants who did not follow Pedagogy 2 is a lot more spread out compared to the distribution from those who followed the pedagogy. This implies that some participants who did not follow the pedagogy got high LOE scores. This issue is clearly illustrated in Figure 1b where we dichotomize the performance on LOE into High and Low. In Figure 1b, the left column represents the participants who did not follow the pedagogy, and the right column represents the ones who did. The lighter color represents low LOE scores and the darker

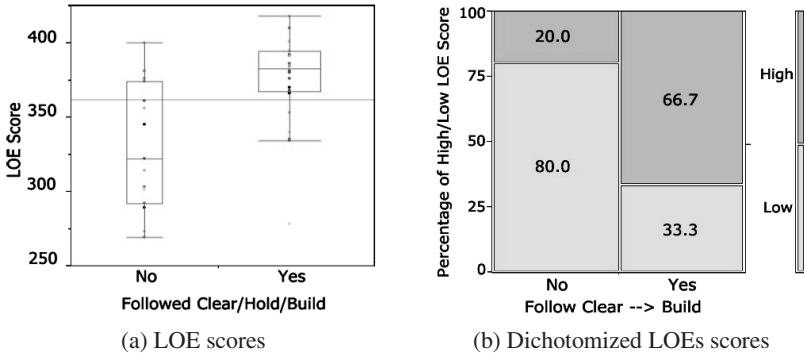


Fig. 1. Comparison of performance on (a) LOE scores and (b) dichotomized scores between participants who followed Pedagogy 2 and those who did not

color represents high LOE scores. The numbers on the graph represents the percentage of participants in that particular case, e.g. followed pedagogy and got a high LOE score.

We can see that a significant percentage of participants achieved high LOE scores despite not following the intended pedagogy (Clear \rightarrow Build). In fact, this group of participants all followed the Build \rightarrow Build strategy, which worked just as well as the Clear \rightarrow Build strategy. This could be problematic in a training simulation because following the Build \rightarrow Build strategy would have severe consequences in the real world, e.g. early builds will be destroyed if an area was not secured first through Clear/Hold.

Figure 1b also shows a region of participants who followed the pedagogy but received low scores (the lower right). While this is also indicative of an error, our procedure for identifying Clear \rightarrow Build strategies is subject to false positives, in that strategies that we identified as following Clear \rightarrow Build may still be violating Pedagogy 2. For example, while a student’s Build actions may be restricted to only the second half of the game, they may have been executed in regions that had not been previously cleared. Our purely temporal classification would not detect such an error. On the other hand, a strategy that does not satisfy Clear \rightarrow Build in our crude classification definitely violates Pedagogy 2, so the upper left region of Figure 1b (and the rewarded Build \rightarrow Build strategy within) corresponds to clearly undesirable outcomes.

4 Simulation-Based Verification of Pedagogy

Section 3’s experimental results demonstrate that the simulation generally encourages the correct behavior, thanks to the rounds of playtesting and model editing that had already occurred. However, the results also identified one pedagogically incorrect strategy (namely, Build \rightarrow Build) that was also rewarded by the simulation. The encouragement of such a strategy suggests the need for changes to the underlying scenario model to bring the simulation more in line with the intended pedagogy. Unfortunately, it is prohibitively costly to playtest after each such change, making it impossible to use human subjects in a tight iterative refinement cycle. Moreover, the playtesting results from

Section 3 represent only 57 possible simulation paths. However, the training scenario provides the student with over 10^{26} possible simulation paths in trying to capture the complexity of real-world urban stabilization. Thus, even if playtesting were feasible, it could explore only an infinitesimal portion of the possible space.

On the other hand, a student actively trying to succeed in the simulation would never try many of the 10^{26} possible simulation paths. For example, a student would not deliberately choose to devote resources to repair a structure that was already operating at full capacity. Although it is possible that a student might do so in error, the likelihood of such errors is so low that we may safely ignore such a possibility in our verification process. Of greater concern are errors like Build \rightarrow Build that show up in multiple cases even within the relatively small data set of Section 3. Our goal in this section is to use this data acquired as the basis for an automatic method for exploring simulation paths that is sensitive to the likelihood of behaviors by real students.

4.1 Markov Chain Monte Carlo Simulation

Our proposed automatic method generates a plot like Figure 1b by randomly generating paths through the training simulation that give us a final score and that allow us to determine whether they followed the pedagogy. Markov chain Monte Carlo (MCMC) simulation provides such a method, in that we can translate a distribution over student actions into simulation path samples [2,3]. To apply MCMC to a training simulation, we must first represent the evolving state (both observed by the student and hidden inside the system) as a Markov chain, X_t . In the multiagent system underlying our simulation, the complete state (S from the POMDP) of the UrbanSim scenario is already represented as a set of 1452 features (e.g., a structure’s capacity), each a real-valued number from -1 to 1 (e.g., 1 means that the structure is functioning at 100% of capacity). While we wish to capture the evolution of the overall simulation state, the states in the Markov chain must represent the student’s decision-making inputs as well. The student sees very little of the 1452 features and is instead informed mainly by the LOE scores (which in this scenario, are derived deterministically from the simulation state). We thus augment the simulation state with the observable LOE scores to capture both the state of the simulation and the factors that influence the student’s choice of action. In addition to capturing all of the relevant factors, the Markov chain representation must also capture the transition from the current state to the next as a function of the student’s action, but independent of prior state history. However, our survey data identified that students often reacted to *changes* in their score, not just the current value. Therefore, to account for this factor and to preserve the Markovian property, we add the latest change in LOE score to the state as well. In summary, the set of possible states for our Markov chain is defined over the possible simulation states, observable values, and changes in reward values: $X = S \times \Omega \times \Delta R$.

4.2 Sampling Distribution

Given this representation of the current state, we must represent the Markovian state transitions in terms of the distribution over possible student’s actions and their effects.

The underlying simulation dynamics (T) can generate the effects of actions, the observation function (O) can generate what the student sees of that state, and the scoring function (R) can generate the changes in rewards. However, all three functions require the student’s action choices as input. Therefore, the only new component we need for the dynamics of our Markov chain ($\Pr(X_t|X_{t-1})$) is the students’ decision-making. The current state has sufficient information to motivate different students’ choices, which we can thus model as a function, $\pi : \Omega \times \Delta R \rightarrow \Pi(A)$, that maps from observation and change in reward to a probability distribution over action choices.

For complex training scenarios, students may have too many possible choices for limited data to generate a meaningful distribution over their decision-making. For example, in the UrbanSim scenario, there are more than 3000 possible actions, so we would require a prohibitively large data set to learn a distribution over the original fine-grained action space, $|A| > 3000$. Instead, we propose clustering the original actions based on their effect on the game scores (e.g., the 6 different LOEs). For a given state, we can sum the cumulative effect of the student’s actions on the game score (e.g., the effect of all 11 subordinates’ actions on the 6 LOEs).

We can now examine the playtesting data in these terms to compute a frequency count of actions chosen as a function of possible score changes. Table 2 shows the expected rate of different types of actions as a function of changes in one of the score dimensions (labeled *LOE 2*). The probability distribution in this table is based on data collected from 57 participants. Students are roughly half as likely to choose an action to increase LOE 2 if there has been no

change in its value, and that actions LOE 1 are more common regardless. Note that the numbers in Table 2 are obviously highly domain-dependent, but the method of acquiring them generalizes quite easily. By clustering the actions according to the scores they immediately increase, one can automatically analyze the logs to compute such frequency counts in a straightforward manner.

Table 2. Expected probability of action types given most recent change in LOE 2

Action	Decrease	Increase	No Change
LOE 1	0.36	0.32	0.38
LOE 2	0.25	0.22	0.12
LOE 3	0.00	0.01	0.01
LOE 4	0.13	0.10	0.10
LOE 5	0.05	0.09	0.08
LOE 6	0.19	0.22	0.25

4.3 Simulation Paths

Now that we have the abstract strategy, $\hat{\pi}$, for the students’ actions, we can compute the dynamics of our Markov chain:

$$\Pr(X_t = \langle s_t, \omega_t, \Delta r_t \rangle | X_{t-1}) = \langle s_{t-1}, \omega_{t-1}, \Delta r_{t-1} \rangle \\ = \sum_{\hat{a} \in \hat{A}} \hat{\pi}(\omega_{t-1}, \Delta r_{t-1}, \hat{a}) \hat{T}(s_{t-1}, \hat{a}, s_t) O(s_t, \hat{a}, \omega_t) \Pr(\Delta r_t = R(s_t, \hat{a}) - R(\omega_{t-1}))$$

where we assume that the previous reward is extractable from the previous observation, ω_{t-1} . For training simulations where the students do *not* observe their scores along the way, we can simply explicitly encode the score as an additional component of our Markov chain state, X . The final missing piece is the abstract transition probability,

\hat{T} , over our abstract actions, \hat{A} . The underlying simulation provides the fine-grained transition function, T , which we will use to derive its abstract counterpart. In particular, for each abstract action, \hat{a} , we will define its effect as a uniform distribution over its possible corresponding fine-grained actions, a :

$$\hat{T}(s_{t-1}, \hat{a}, s_t) = \sum_{a|C(a)=\hat{a}} T(s_{t-1}, a, s_t) / |\{a|C(a)=\hat{a}\}|$$

We can now run the simulation engine and substitute actions sampled according to Section 4.2 instead of the student actions. Each such run requires only 4 minutes (as opposed to the hour required by the typical human subject), and we were able to generate 316 paths in 21 hours of computation time. The end result of each path is a run of the exact same form as used in playtesting and, thus, amenable to the evaluation procedure of Section 3. Thus, we determined whether the generated actions satisfied the intended pedagogy, and we extracted the score achieved by those actions. Finally, we generated the graph in Figure 2 (of exactly the same form as Figure 1b) to identify the degree to which the pedagogy is satisfied. Of the paths that violated Pedagogy 2, most received an appropriately low score, but the simulation identified 143 paths where an incorrect strategy received a high score, far exceeding the incorrect paths found among the 57 student paths in Figure 1b. Given that the simulation was able to generate Figure 2 overnight, as opposed to the weeks required to schedule the human subjects for Figure 1b, our automated exploration method has greatly accelerated our ability to verify the simulation underlying our training system.

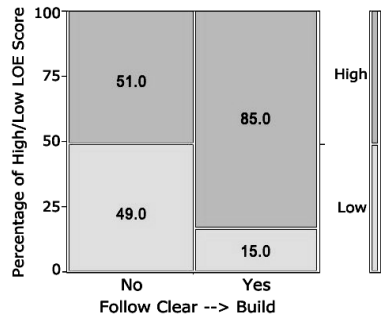


Fig. 2. Results from simulation-based verification

5 Discussion

The methodology presented in this paper provides a mechanism for automatic verification of an agent-based training simulation using limited human user data. The true test of a training simulation is in a thorough pedagogical evaluation of student learning when using the system, and our proposed methodology is in no way a replacement of such an evaluation. Our methodology instead seeks to give the simulation designer feedback during the authoring process. In particular, a graph like Figure 1b identifies paths through the simulation that violate the intended pedagogy, directing the designer to possible modeling errors. Section 4’s automatic method for generating such graphs can then give the simulation designer similar feedback for the refined models, without requiring further playtesting. Furthermore, the systematic exploration of a larger space of possible student strategies can give the simulation designer greater confidence in the agent models before proceeding to the overall pedagogical evaluation and deployment.

Going beyond the reactive strategies of our MCMC approach to modeling the student's behavior, there is the potential to use PsychSim's POMDP-based behavior-generation mechanism to provide more sophisticated models of student moves. In the post-questionnaire, we collected information about how students ranked the various LOEs in priority and how they thought different actions affected those LOEs. The former gives us insight into how the students' subjective reward function deviated from the "rational" student's. The latter gives us insight into how the students' model of the simulation dynamics deviated from the correct transition probability function, T . Thus, we can potentially learn PsychSim models for different students and use these models to generate more deliberative strategies than the reactive strategies of our MCMC approach.

Finally, our verification methodology can be a key component to facilitating the overall authoring process for training simulations. This paper presents a novel method for automatically finding simulation paths that are inconsistent with intended pedagogy. Given the output of our method, we can then use existing algorithms [10] to help automate the modification of the simulation to bring it more in line with that intended pedagogy. Thus, the methodology and algorithms presented in this paper represent a critical step toward greatly reducing the burden of authoring agent-based training simulations while simultaneously improving their pedagogical fidelity.

References

1. Calder, R.B., Smith, J.E., Courtemanche, A.J., Mar, J.M.F., Ceranowicz, A.Z.: ModSAF behavior simulation and control. In: Proceedings of the Conference on Computer-Generated Forces and Behavioral Representation, pp. 347–356 (1993)
2. Gilks, W.R., Richardson, S., Spiegelhalter, D.J.: Markov chain Monte Carlo in practice. Chapman and Hall, London (1996)
3. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57(1), 97–109 (1970)
4. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101, 99–134 (1998)
5. Kim, J.M., Randall, J., Hill, W., Durlach, P.J., Lane, H.C., Forbell, E., Core, M., Marsella, S., Pynadath, D., Hart, J.: BiLAT: A game-based environment for practicing negotiation in a cultural context. *IJAIED* 19(3), 289–308 (2009)
6. Klatt, J., Marsella, S., Krämer, N.C.: Negotiations in the Context of AIDS Prevention: An Agent-Based Model Using Theory of Mind. In: Vilhjálmsón, H.H., Kopp, S., Marsella, S., Thórisson, K.R. (eds.) IVA 2011. LNCS, vol. 6895, pp. 209–215. Springer, Heidelberg (2011)
7. Marsella, S.C., Pynadath, D.V., Read, S.J.: PsychSim: Agent-based modeling of social interactions and influence. In: ICCM, pp. 243–248 (2004)
8. McAlinden, R., Gordon, A., Lane, H.C., Pynadath, D.: UrbanSim: A game-based simulation for counterinsurgency and stability-focused operations. In: AIED Workshop on Intelligent Educational Games (2009)
9. Miller, L.C., Marsella, S., Dey, T., Appleby, P.R., Christensen, J.L., Klatt, J., Read, S.J.: Socially optimized learning in virtual environments (SOLVE). In: ICIDS (2011)
10. Pynadath, D.V., Marsella, S.C.: Fitting and compilation of multiagent models through piecewise linear functions. In: AAMAS, pp. 1197–1204 (2004)

11. Pynadath, D.V., Marsella, S.C.: PsychSim: Modeling theory of mind with decision-theoretic agents. In: IJCAI, pp. 1181–1186 (2005)
12. Rickel, J., Johnson, W.L.: Integrating pedagogical capabilities in a virtual environment agent. In: Agents, pp. 30–38. ACM Press (1997)
13. Si, M., Marsella, S.C., Pynadath, D.V.: THESPIAN: An architecture for interactive pedagogical drama. In: AIED, pp. 595–602 (2005)
14. Tambe, M., Johnson, W.L., Jones, R.M., Koss, F., Laird, J.E., Rosenbloom, P.S., Schwamb, K.: Intelligent agents for interactive simulation environments. *AI Magazine* 16, 15–39 (1995)

Using State Transition Networks to Analyze Multi-party Conversations in a Serious Game

Brent Morgan¹, Fazel Keshtkar¹, Ying Duan¹,
Padraig Nash², and Arthur Graesser¹

¹ University of Memphis, Psychology, Institute for Intelligent Systems,
365 Innovation Drive, Memphis, TN 38152, USA

² University of Wisconsin-Madison, Educational Psychology, Educational Sciences Building,
Room 1078D, 1025 West Johnson Street, Madison, WI 53711
brent.morgan@memphis.edu

Abstract. As players interact in a serious game, mentoring is often needed to facilitate progress and learning. Although human mentors are the current standard, they present logistical difficulties. Automating the mentor's role is a difficult task, however, especially for multi-party collaborative learning environments. In order to better understand the conversational demands of a mentor, this paper investigates the dynamics and linguistic features of multi-party chat in the context of an online epistemic game, Urban Science. We categorized thousands of player and mentor contributions into eight different speech acts and analyzed the sequence of dialogue moves using State Transition Networks. The results indicate that dialogue transitions are relatively stable with respect to gameplay goals; however, task-oriented stages emphasize mentor-player scaffolding, whereas discussion-oriented stages feature player-player collaboration.

Keywords: collaborative learning, epistemic games, natural language processing.

1 Introduction

Serious games are increasingly becoming a popular, effective supplement to standard classroom instruction [1]. Some classes of serious games provide microworlds [2] that allow players to explore a virtual environment. These simulations have ideal and often simple problems with targeted scaffolding to help users identify important concepts and think critically about them. Multi-party chat is pervasive in serious games and crucial to success in multi-player recreational games, including the epistemic games [3, 4, 5] that will be addressed in the present study.

Epistemic games and collaboration can be effective environments for learning [6], but a critical element for success in these environments is access to some form of directed help. A substantial body of research suggests that mentoring is needed in order to facilitate learning tools, such as reflection, elaboration, scaffolding, modeling, and so forth [7, 8, 9]. Without this, student learning is minimal.

While mentoring is a necessary element for learning in epistemic games, this role is almost exclusively provided by a human at the present time. However, the cost incurred with training a human mentor, as well as logistics (e.g., availability), represent a critical barrier for widespread use of a collaborative epistemic game. Consequently, if the role of the mentor could be automated, it would allow an established epistemic game to be scaled up for widespread use. Although great strides have been made in automating one-on-one tutorial dialogues [10], multi-party chat presents a significant challenge for natural language processing. The goal of this paper, then, is to provide a preliminary understanding of player-mentor conversations in the context of an epistemic game, specifically *Urban Science*.

Urban Science is an epistemic game created by education researchers at the University of Wisconsin-Madison, designed to simulate an urban planning practicum experience [7]. During the game, players communicate with other members of their planning team, as well as with an adult mentor role-playing as a professional planning consultant. Urban Science consists of 19 distinct stages, each of which has one of two functions, *task-oriented* or *discussion-oriented* (with 13 and 6 stages, respectively). The task-oriented stages have more concrete actions to perform. Discussion-oriented stages have high interactivity, discussion, and reflection.

It is plausible that the different educational goals of each stage type may have corresponding differences in the conversational patterns between players and mentors. To investigate these patterns, the conversations between the mentor and players were analyzed with respect to meaning, syntax, and discourse function by speech act classification. These categorized speech acts were analyzed to identify speech act sequences in the conversations, represented as State Transition Networks (STN).

1.1 Speech Act Classification and State Transition Networks

Analyses of a variety of corpora, including chat and multiparty games, have converged on a set of speech act categories that are both theoretically justified and that also can be reliably coded by trained judges [11, 12]. Our classification scheme has 8 broad categories: Statements, Requests, Questions, Reactions, Expressive Evaluations, MetaStatements, Greetings, and Other. After classifying individual speech acts, pairs of speech acts can be joined in STNs. STNs specify the speech act transitions both within and between conversation participants with respect to specific speakers and the associated speech act categories.

Discourse acts in educational contexts have been documented in great detail in the context of classroom discourse [13, 14] and human tutoring [15, 16]. For example, a common three-step sequence in classrooms is: “Teacher Question → Student Answer → Teacher Feedback Response” [17]. The goal of this paper is to identify the conversational patterns in multi-party conversations in an epistemic game (such as Urban Science) with the ultimate objective of automating the mentor’s role.

1.2 Hypotheses

First, we predict that our analyses will identify speech acts and transitions common to both task- and discussion-oriented stages. For example, aforementioned research

indicates that mentoring is critical to maximize learning [7, 8, 9]. Thus, mentor contributions should constitute the most pivotal nodes in the STNs of both types of stages. The research also suggested that mentor questions often initiate conversational sequences, which are followed by player responses and then feedback on the response. This dynamic is well-established and should be evident across both formats.

In addition to commonalities, we also seek to pinpoint some differences between stage types. In the task-oriented stages, goal achievement is a priority, suggesting that mentor requests would be more relevant. Similarly, task-based stages should also feature questions by the players about how to proceed. Most importantly, we expect two distinct epistemic networks to emerge: scaffolding and collaboration. Scaffolding occurs when mentor responses to player contributions help guide players to the next step. This should be essential to facilitating goal completion in task-oriented stages. Conversely, collaboration represents meaningful interactions between players. This should be more evident in the discussion format, as players interact and reflect upon their previous actions.

2 Methods

Twenty-one high school-aged participants and two mentors played Urban Science for ten hours over three days. Players communicated with each other and the mentor via a chat window. Player and Mentor chat contributions were automatically categorized into speech acts using the Naive Bayes classification algorithm on word features. The classification compares favorably to trained human coders with a kappa of 0.677, compared to a kappa of 0.797 between two humans [18].

STNs were created by calculating the conditional probability of each transition between speech acts as well as the overall frequency of each speech act in the corpus. For example, a mentor statement might be followed by a player reaction 28% of the time, and a player reaction might constitute 0.8% of the entire corpus. For each transition, a minimum conditional probability threshold of 15% was used for inclusion in the network, as well as an overall frequency of 0.3%. Additionally, although there are only two roles in the game (player and mentor), one crucial piece of information that the STNs can provide is the identity of the speaker. Specifically, in the case of adjacent player contributions, it is critical to distinguish whether the response is a follow-up from the same player (“P → P”) or whether it is a reply by some Other Player (“P → OP”). This distinction helps in identifying player collaborations.

3 Results

Our analysis of the Urban Science data initially classified contributions into individual speech acts, then calculated the conditional probabilities for each transition, and then the overall frequencies/likelihoods for each speech act category. We expected to find some commonalities and differences between two different types of interactions during gameplay, namely task-oriented and discussion-oriented stage types. We found that the correlation between transition conditional probabilities was quite large, r

(318) = 0.63, $p < .001$, which supports the notion that conversation dynamics are largely stable. Inspection of the STNs for both formats unveils these common patterns, but also highlights some transitions that distinguish the two. The STNs for task- and discussion-oriented stages are shown in Figures 2 and 3, respectively.

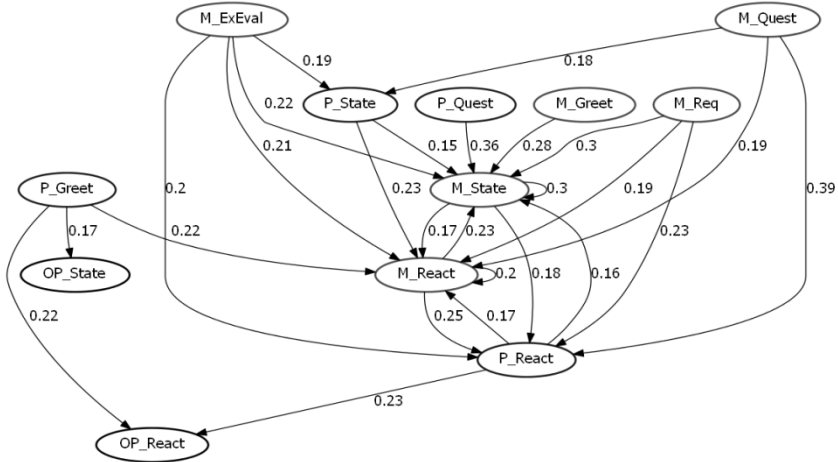


Fig. 1. State Transition Network for Task-oriented stages

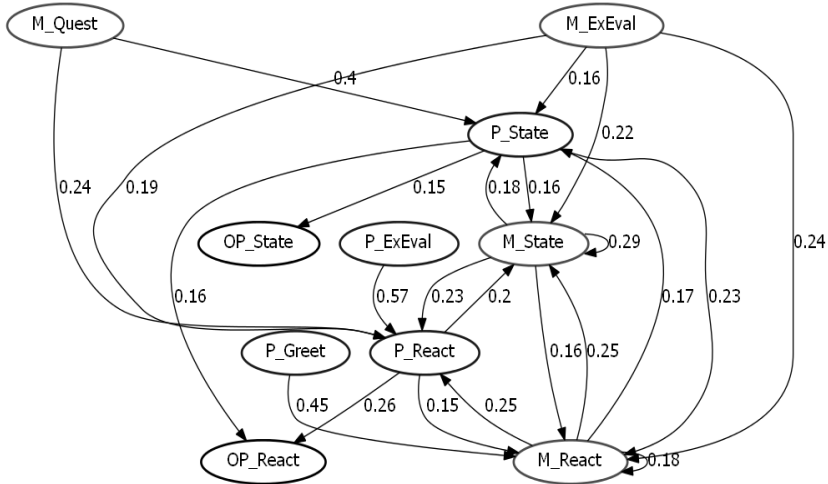


Fig. 2. State Transition Network for Discussion-oriented stages

Our first prediction was that mentor contributions would constitute the most pivotal nodes in the STNs of both stage types, reflecting the importance of the mentor in student learning. This was supported by the relative importance of mentor statements and mentor reactions in both STNs. We also predicted that mentor questions would

initiate conversational sequences. Figures 2 and 3 both suggest that mentor questions (along with expressive evaluations) were crucial in triggering dialogue progressions. Additionally, mentor questions in both networks were typically followed by player statements or reactions, which were in turn followed by reactions or statements from the mentor or another player. This suggests the “Question → Response → Feedback” sequence that was discussed previously [17].

With respect to differences in stage type, mentor requests and player questions played a larger role in the task-oriented stages, in line with our predictions, whereas in the discussion-oriented stages, player statements and expressive evaluations had a higher impact. We also expected distinct epistemic patterns to emerge between the two stage types, namely scaffolding and collaboration. The distinguishing feature of these patterns is the relative frequency of a mentor response to a player contribution versus a response by some other player. Whereas the task-oriented STN features more mentor nodes (indicating scaffolding), the discussion-oriented STN produced similar OP nodes. However, the OP contributions in the discussion-oriented stages were more likely to be a response to player statements and reactions (i.e., the final link in the “Question → Response → Feedback” chain), as opposed to responses to greetings, which are unlikely to be meaningful. These observations support the prediction that scaffolding is more important to facilitate the goal achievement for task-based gameplay, whereas the discussion-based format emphasizes student collaboration.

4 Conclusion and Future Work

We expected that particular transitions between speech acts would be common within both types of gameplay in Urban Science, task- and discussion-oriented. The correlation of transition between the two stage types was surprisingly strong, indicating that transitions are relatively stable across different modes of gameplay. Despite the overlap in transition frequencies between task- and discussion-oriented stages, we were able to identify some crucial differences between the two types. Mentor requests and player questions reflected the goal-driven activities of the task-oriented stages, whereas the discussion-oriented stages showed greater emphasis on player statements and expressive evaluations as they reflected on previous game actions. The two stages also differed in the final link of the “Question → Response → Feedback” sequence, where the feedback was more likely to be provided by the mentor in the task-oriented stage (indicating scaffolding), but in the discussion format, other players were increasingly likely to respond (suggesting collaboration).

The results of the presented analyses are applicable to a number of current and future investigations. First, we are currently analyzing additional chat room interactions in order to replicate these findings and assist in automating the role of the mentor. This includes predicting points in the conversation where a mentor should provide a contribution, as well as the appropriate speech act at a given point.

References

1. Ritterfeld, U., Cody, M., Vorderer, P. (eds.): *Serious games: Mechanisms and effects*. Routledge, New York (2009)
2. Hoyles, C., Noss, R., Adamson, R.: Rethinking the Microworld idea. *Journal of Educational Computing Research* 27(1-2), 29–53 (2002)
3. Dieterle, E., Clarke, J.: Multi-user virtual environments for teaching and learning. In: Pagan, M. (ed.) *Encyclopedia of Multimedia Technology and Networking*, 2nd edn. Idea Group, Hershey (in press)
4. Ketelhut, D., Dede, C., Clarke, J., Nelson, B., Bowman, C.: Studying situated learning in a multi-user virtual environment. In: Baker, E., Dickieson, J., Wulfek, W., O’Neil, H. (eds.) *Assessment of Problem Solving Using Simulations*, pp. 37–58. Earlbaum, Mahwah (2007)
5. Shaffer, D.W.: *How Computer Games Help Children Learn*. Palgrave, New York (2007)
6. Millis, K., Forsyth, C., Butler, H., Wallace, P., Graesser, A.C., Halpern, D.: Operation ARIES! A serious game for teaching scientific inquiry. In: Ma, M., Oikonomou, A., Lakhmi, J. (eds.) *Serious Games and Edutainment Applications*, pp. 169–195. Springer, London (2011)
7. Bagley, E.S., Shaffer, D.W.: When people get in the way: Promoting civic thinking through epistemic gameplay. *International Journal of Gaming and Computer-Mediated Simulations* 1(1), 36–52 (2009)
8. Kirschner, P.A., Sweller, J., Clark, R.E.: Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist* 41(2), 75–86 (2006)
9. Nash, P., Shaffer, D.W.: Mentor modeling: The internalization of modeled professional-thinking in an epistemic game. *Journal of Computer Assisted Learning* 27(2), 173–189 (2011)
10. Graesser, A.C., D’Mello, S.K., Cade, W.: Instruction based on tutoring. In: Mayer, R.E., Alexander, P.A. (eds.) *Handbook of Research on Learning and Instruction*, pp. 408–426. Routledge, New York (2011)
11. Moldovan, C., Rus, V., Graesser, A.C.: Automated Speech Act Classification For Online Chat. In: *The 22nd Midwest Artificial Intelligence and Cognitive Science Conference* (2011)
12. D’Andrade, R.G., Wish, M.: Speech act theory in quantitative research on interpersonal behavior. *Discourse Processes* 8(2), 229–259 (1985)
13. Gee, J.P.: *An introduction to discourse analysis: Theory and method*. Routledge, New York (1999)
14. Nystrand, M.: Research on the role of classroom discourse as it affects reading comprehension. *Research in the Teaching of English* 40, 392–412 (2006)
15. Graesser, A.C., D’Mello, S.K., Cade, W.: Instruction based on tutoring. In: Mayer, R.E., Alexander, P.A. (eds.) *Handbook of Research on Learning and Instruction*. Routledge Press, New York (in press)
16. Graesser, A.C., Person, N.K.: Question asking during tutoring. *American Educational Research Journal* 31, 104–137 (1994)
17. Sinclair, J., Coulthart, M.: *Towards an analysis of discourse: The English used by teachers and pupils*. Oxford University Press, London (1975)
18. Rus, V., Moldovan, C., Witherspoon, A., Graesser, A.C.: Automatic Identification of Speakers’ Intentions in A multi-Party Dialogue System. In: *21st Annual Meeting of the Society for Text and Discourse* (2011)

How to Evaluate Competencies in Game-Based Learning Systems Automatically?

Pradeepa Thomas, Jean-Marc Labat, Mathieu Muratet, and Amel Yessad

Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie 4, Place Jussieu,
75005 Paris, France
{pradeepa.thomas, jean-marc.labat,
mathieu.muratet, amel.yessad}@lip6.fr

Abstract. Serious games are increasingly used in schools, universities or in vocational training. When they are used in the classroom, teachers often have to deal with the lack of tools for monitoring the students during the game and assessing them after the game. So they often tend to add assessment questionnaires to the fun sequence of “learning by playing”, to ensure that students have learned during the session. Our goal is to enable the teacher to do without this type of questionnaires by providing them an automated tool for monitoring and analyzing the actions performed by learners. The system combines an “expert Petri Net” and a domain and game action ontology. Our first experiment conducted on a sample of fifteen students showed that the diagnostic tool gives relatively close results to those of an online assessment questionnaire proposed by the teacher.

Keywords: Serious games, Game-based learning, Assessment, Petri Nets, User tracking.

1 Introduction

The question of learning through serious games is often asked. Much research has been carried out [1], [2], [3]. When the serious games are used in the classroom, teachers often have to deal with the lack of tools for monitoring and assessing students. So they often tend to use assessment questionnaires to ensure that students have learned during the session. This practice interrupts the game dynamics created by game-based learning systems. Our contribution is a tool for teachers to monitor and analyze the progress of the player (from traces of the game). The system uses indicators inspired from Hollnagel’s analysis of human errors [4]. The tool is based on an “expert” Petri net and a domain and game action ontology. Petri Nets are used to model the expert rules of the domain and to diagnose the non-compliance of these rules. The ontology represents the domain concepts and their equivalent in terms of game actions, relations between game actions and between concepts and actions.

After having highlighted the difficulties of assessing learning in game-based learning systems we explain in detail the algorithms used by showing how the properties

and tools of a Petri Net (PN) can be used to label the behavior of the player. We then present the first encouraging results of a comparative evaluation between our system and an assessment using an online questionnaire.

2 Assessment in Game-Based Learning Systems

Several studies have considered the issue of automatic assessment of learning in serious games. Thus, in [5] the authors added a system of state machines to the game, i.e. predefined situations that the teacher wants to watch. The teacher sets the states he wants to trace in the monitoring system. This approach is interesting but requires the teacher a great effort of interpretation to analyze the provided game indicators. In [6], the system compares the student causal graph to the teacher’s one and highlights the missing and erroneous link. In [7], the authors use plan space exploration to generate a suitable game play for the learner. The game is automatically adapted to the actions of the player. Our approach has the same objectives as these approaches but uses different techniques and is more interested in labeling the errors performed by the player: the main goal is to assist the teacher in his evaluation of the “learning player”.

3 Automatic Monitoring and Game Action Analysis System

3.1 Diagnostic Indicators

Drawing on the work of Hollnagel [8], we defined a classification of actions made or not made by the learner, using the CREAM method (Cognitive Reliability and Error Analysis Method). In the case of error analysis in game-based learning systems, we present in the figure 1 an evaluation of the actions of the player.

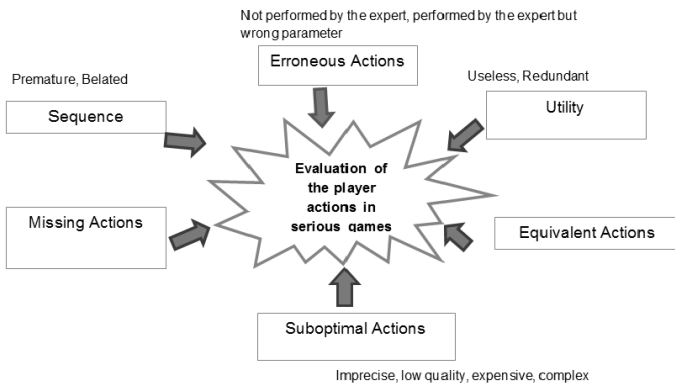


Fig. 1. Evaluation of the player's actions in the game-based learning systems according to the CREAM method

Each label is explained in [9]. For example, suboptimal actions enable progress in the game but the multiplication of these actions by the learner reveals a poor mastery of the field and imperfect skills. The player manages to overcome difficulties without finding the right solution. Belated and premature actions are necessary to resolve the current problem but do not happen in the right sequence. Equivalent actions are not performed by the expert, but produce exactly the same result as those of the expert.

3.2 Combining an Expert Petri Net and a Game Action and Domain Ontology

As detailed in [10] [9], we use a Petri net to follow the progress of the learner step by step. The idea is to analyze every "pedagogically significant" action performed by the player and to label each one according to the headings defined in Figure 1. The transitions of the Petri Net are game actions and the places represent game properties. The Petri net describes the expert behavior in the game: it performs the actions that the expert uses to solve the problem. The Petri net is built using a reverse engineering process on the game engine and by extracting domain rules from experts. Once the network is initialized by marking the places that describe the data of a problem to solve, the Petri Net will list all the solutions, i.e. the graph of the actions leading to an expert resolution of the problem (Petri Net reachability graph). Petri nets have been used in the field of game-based learning systems but rather to design games and to validate and verify the consistency of scenarios [11][12][13]. The ontology of game actions completes the approach. We use it to link game actions to the domain competencies and to represent the equivalence and sub-optimality relations between game actions. Ontologies have been used for the diagnosis of errors in learning systems [14] as well as for knowledge diagnosis in serious games [15].

3.3 Game Actions Labeling Algorithm

The goal of this algorithm is to analyze the actions performed by the player step by step and to compare them with the « expert » Petri net. The system provides the teacher with an overview by presenting the percentage of each label defined in Figure 1. Thus, the list of missing actions allows the teacher to identify blocking points. Moreover, even if a student passes the level, the multiplication of erroneous actions demonstrates a process of trial and error to reach the solution. Finally, belated and premature actions reveal a lack of optimization in the sequence of actions. The player performs the correct actions but not at the most opportune moment.

The diagnosis algorithm works as follows:

1. Expert Petri Net loading and reachability graph calculation
2. Player's traces loading and sub-optimal / equivalent actions research: the ontology is queried first to detect these error categories. These actions are labeled and then replaced by the corresponding expert action.
3. With the reachability graph, identification of :
 - (a) Right actions : firable transitions
 - (b) Erroneous actions : transitions that don't appear in the reachability graph

- (c) Redundant actions: live transitions (available in the expert Petri Net) but all the output places are marked (the player already has the information he requested)
 - (d) Premature actions : live transitions but not firable because prerequisite transitions are missing
 - (e) Missing actions : transitions that appear in the reachability graph but not in the traces of the player
4. Finally, the belated actions are obtained as follows: each time the player performs a non firable transition, the system calculates and stores the expected ones. Thus, when the player performs an “expected action” , it is a belated one.

4 Case Study

4.1 The Game

Ludiville has been developed by KTM Advance, for the “Banque Populaire Caisse d’Epargne” Group. It is designed for fledgling account managers. The goal of the player is to meet the demand of a customer by handling a more or less pre-filled loan file by performing domain linked game actions. One of the particularities of the game is to allow the player to use generic action when he doesn’t know what type of information to ask the client for. For instance, “ask for document” action can be used instead of “Pay slips”. These generic actions ensure that the learning player does not get frustrated. He can move forward inside the game without being held up by a lack of knowledge about some domain aspects. However, these cards yield fewer points than the specific cards, which reveal *a priori* core competencies.

4.2 The Experiment

The aim of the experiment is to compare the results of the diagnostic tool with that of an online questionnaire. The game has been tested on fifteen Higher National Certificate students. The part of the course on mortgages has not been addressed by the teacher beforehand. After a quick presentation of the game interface, the students played independently for about an hour. They all finished the first level. The traces containing all the actions performed for each client and each attempt were collected in XML files. At the end of the game, students responded to an online questionnaire developed by the teacher, referring to the concepts covered in the game. For example, they were asked to name the key documents to identify the personal characteristics of the client. The questions were classified according to four tabs defined in the game: client, project, loan and finalizing. Students had not been warned initially that they would be assessed at the end of the session. We chose to analyze in detail the last customer case of the first level. This is an assessment case that contains most of the skills used in the previous cases. This validation is considered as the “boss” at the end of a level in “traditional” video games. Some students had to go through several attempts to complete this case. We chose to analyze the latest. Take the average of all trials would have penalized those who started several times. However, the number of attempts has been passed on to the teacher to give a more accurate evaluation.

For each student, an overall average and average per competency was calculated by coding the responses to the questionnaire as follows: right (2 pts), approximate (1pt), no answer (0pt), wrong (-1pt). The diagnostic indicators (right action, too early, too late, sub-optimal, equivalent erroneous) were also related to each sub area of expertise and coded as follows : right (1pt), sub-optimal (1,5 pts), premature and belated (1 pt), erroneous (-0,5pt), missing (1pt).

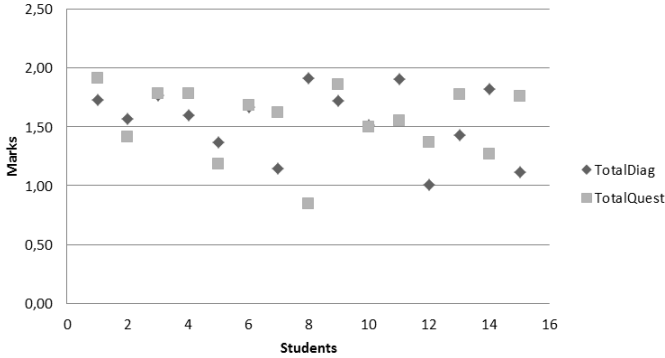


Fig. 2. Results comparison

4.3 Discussion et Perspectives

Comparing the two point clouds in Figure 2 shows that the results coincide for two-thirds of the students. The average is 1.55 for the diagnosis tool while it is 1.56 for the questionnaire. The Wilcoxon signed rank test gives p-value at 0.69. The two series are quite close. For student 8 and student 15, the differences are explained by a misunderstanding in the questionnaire (the same question). For student 8, the importance of the gap is due to the multiplication of non-expert actions. In our tool, we can see hesitations, trials and errors. Thus, at the rating of game actions, students who had thoughtful behavior have clearly an advantage compared with those who have adopted a process of trial and error, by multiplying the attempts. Moreover, it is not because they have increased the errors that they did not finally learn from their mistakes: this explains why their results in the questionnaires are good. We should refine the labeling of non-expert actions in order to isolate those that specifically reveal misconceptions.

5 Conclusion

From several experiments, it will be possible to identify players' behavior patterns using data mining techniques such as clustering. In the rest of our work, we also plan to analyze in detail the various attempts on the same mission: how does the player adjust his strategy when he starts again a mission? In this regard, the Petri Net-based approach when implemented in real time on a game, allows for automatic and

appropriate guidance. Indeed, when the player is blocked because he did not perform an action, the system can send a clue. The authors want to thank the French government who funded this research.

References

1. Johnson, W.L., Wu, S.: Assessing Aptitude for Learning with a Serious Game for Foreign Language and Culture. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 520–529. Springer, Heidelberg (2008)
2. Virvou, M., Katsionis, G., Manos, K.: Combining Software Games with Education: Evaluation of its Educational Effectiveness. *Education Technology & Society* 8 (2005)
3. Defreitas, S., Oliver, M.: How can exploratory learning with games and simulations within the curriculum be most effectively evaluated? *Computers & Education* 46, 249–264 (2006)
4. Hollnagel, E.: Barriers and accident prevention. Ashgate Publishing, Ltd. (2004)
5. del Blanco, Á., Torrente, J., Marchiori, E.J., Martínez-Ortiz, I., Moreno-Ger, P., Fernández-Manjón, B.: Easing assessment of game-based learning with e-adventure and LAMS. In: Proceedings of the Second ACM International Workshop on Multimedia Technologies for Distance Learning, pp. 25–30. ACM, New York (2010)
6. Shute, V.J., Masduki, I., Donmez, O.: Conceptual Framework for Modeling, Assessing and Supporting Competencies within Game Environments. *Science* 8, 137–161 (2010)
7. Thomas, J.M., Young, R.M.: Annie: Automated Generation of Adaptive Learner Guidance for Fun Serious Games. *IEEE Transactions on Learning Technologies* 3, 329–343 (2010)
8. Hollnagel, E.: Cognitive reliability and error analysis method: CREAM. Elsevier (1998)
9. Thomas, P., Yessad, A., Labat, J.M.: Petri nets and ontologies: tools for the « learning player » assessment in serious games. In: Proceedings of the 11th IEEE International Conference on Advanced Learning Technologies, pp. 415–419. IEEE Computer Society, Athens (2011)
10. Yessad, A., Thomas, P., Capdevila, B., Labat, J.-M.: Using the Petri Nets for the Learner Assessment in Serious Games. In: Luo, X., Spaniol, M., Wang, L., Li, Q., Nejdil, W., Zhang, W. (eds.) ICWL 2010. LNCS, vol. 6483, pp. 339–348. Springer, Heidelberg (2010)
11. Natkin, S., Vega, L.: Petri Net Modelling for the Analysis of the Ordering of Actions in Computer Games. In: Présenté à 4th International Conference on Intelligent Games and Simulation, GAME-ON 2003 (2003)
12. Araujo, M., Roque, L.: Modeling Games with Petri Nets. In: *Breaking New Ground: Innovation in Games, Play, Practice and Theory*, West London, United Kingdom (2009)
13. Brom, C., Šisler, V., Holan, T.: Story Manager in ‘Europe 2045 ’ Uses Petri Nets
14. Abou Assali, A., Lenne, D., Debray, B.: Case Retrieval in Ontology-Based CBR Systems. In: Mertsching, B., Hund, M., Aziz, Z. (eds.) KI 2009. LNCS, vol. 5803, pp. 564–571. Springer, Heidelberg (2009)
15. Conlan, O., Hampson, C., Peirce, N., Kickmeier-Rust, M.D.: Realtime Knowledge Space Skill Assessment for Personalized Digital Educational Games. In: 2009 Ninth IEEE International Conference on Advanced Learning Technologies, pp. 538–542. IEEE Computer Society, Riga (2009)

Sense Making Alone Doesn't Do It: Fluency Matters Too!

ITS Support for Robust Learning with Multiple Representations

Martina A. Rau¹, Vincent Aleven¹, Nikol Rummel^{1,2}, and Stacie Rohrbach³

¹ Human-Computer Interaction Institute, Carnegie Mellon University

² Institute of Educational Research, Ruhr-Universität Bochum, Germany

³ School of Design, Carnegie Mellon University

{marau, aleven}@cs.cmu.edu, nikol.rummel@rub.de, stacie@cmu.edu

Abstract. Previous research demonstrates that multiple representations of learning content can enhance students' learning, but also that students learn deeply from multiple representations only if the learning environment supports them in making connections between the representations. We hypothesized that connection-making support is most effective if it helps students *make sense* of the content across representations and in *becoming fluent* in making connections. We tested this hypothesis in a classroom experiment with 599 4th- and 5th-grade students using an ITS for fractions. The experiment further contrasted two forms of support for sense making: *auto-linked* representations and the use of *worked examples* involving one representation to guide work with another. Results confirm our main hypothesis: A combination of worked examples and fluency support lead to more robust learning than versions of the ITS without connection-making support. Therefore, combining different types of connection-making support is crucial in promoting students' deep learning from multiple representations.

Keywords: Multiple representations, fractions, intelligent tutoring system, connection making, classroom evaluation.

1 Introduction

Multiple representations, such as charts and diagrams in mathematics, are universally used in instructional materials because they can emphasize important aspects of the learning content. Representations as learning tools may be especially beneficial when incorporated in intelligent tutoring systems (ITSs): rather than working with static representations, students can interact with virtual manipulatives [1], and they can be tutored on their interactions with them. There is extensive evidence in the educational psychology literature that learning with multiple representations can enhance students' deep understanding of the domain [2,3]. However, research has also shown that, in order to benefit from multiple representations, students need to make connections between them [2,4,5]. Yet, students find it difficult to make these

connections [2] and tend not to make them spontaneously [2,6]. Therefore, they need to be supported in doing so [7].

In the domain of fractions, multiple representations such as circles, rectangles, and number lines are commonly used [8]. Each representation provides a different conceptual view on fractions [9]. In order to gain a deep understanding of fractions, students need to understand the conceptual views presented by each representation, and they need to relate the representations to one another [8,10]. Being able to relate these different representations is key to developing a deep understanding of fractions (e.g., as numbers that have magnitudes), which is an important educational goal [10].

A crucial question when designing learning environments that use multiple representations is therefore what kind of connection-making support will promote deep learning. Following the KLI theoretical framework for robust learning [11], we distinguish between two types of learning processes: *sense-making* processes and *fluency-building* processes. *Making sense of connections* means (in the case of fractions) that students conceptually understand how different representations relate to each other (e.g., *why* two representations show the same fraction). *Fluently making connections* means to fast and effortlessly relate different representations (e.g., representations that show the same value). Prior research on how best to support students in making connections between multiple representations has focused only on supporting sense-making processes, for instance, by supporting students in relating corresponding elements of representations at a structural level [12]. However, both types of learning processes may be necessary in order to develop competence in a complex domain [11]. Applying this notion to learning with multiple representations, we hypothesize that students learn most robustly when, in addition to being supported in making sense of connections between multiple representations, they are supported in fluently making connections between multiple representations.

A crucial question regarding sense-making support is further: how much automated support should students receive from the system [2]? On the one hand, providing students with auto-linked representations (AL), in which the system, rather than the student, connects and updates representations, has been shown to enhance learning in complex domains [5]. On the other hand, research has demonstrated that students should actively create connections between representations, rather than passively observing correspondences [13]. Thus, we compare two ways of sense-making support, one in which the tutor demonstrates connections (i.e., auto-linked representations, AL), one in which more of that burden falls on the student. A well-researched way of supporting active sense-making processes is to provide students with worked examples (WEs), that is, solved problems with solution steps shown [14]. WEs have been shown to be effective in many domains [14], and have been used in ITSs (e.g., [15]). Berthold and Renkl [16] compared students' learning from multi-representational WEs to single-representation WEs and found that multiple representations can enhance students' learning from WEs. However, to our knowledge, WEs have not yet been used as a means to support students in making connections between multiple representations. In our study, students use a WE that uses a more familiar representation as a guide to solve an isomorphic problem that involves a less familiar representation. As they integrate the example problem and the

new problem, they can make connections between the two representations. We hypothesize that WE support (compared to AL support) will be the more effective type of sense-making support in promoting students' learning of fractions, since students have to engage more actively in making connections.

We address these hypotheses in the context of a proven ITS technology, namely, Cognitive Tutors [17]. The Fractions Tutor has been tested and iteratively improved based on five experimental studies with almost 3,000 students. Although Cognitive Tutors have been widely researched with middle- and high-school students [18] (e.g., Rittle-Johnson and Koedinger [19] report on a study in which 6th-graders used a Cognitive Tutor for fractions), the effectiveness of Cognitive Tutors and other ITSs for elementary-school students remains under-researched.

We conducted a classroom experiment to investigate the effects of sense-making support for connection making and of fluency support for connection making on students' understanding of fractions. 599 4th- and 5th-grade students worked with the Fractions Tutor during their regular mathematics class. Students either received sense-making support for connection making (AL or WE) or not. This factor was crossed with a second experimental factor, namely, whether or not students received fluency support for connection making. Since many education researchers and practitioners emphasize the importance of helping students understand number lines [8,10], we included a version of the Fractions Tutor that provides only a number line as a control condition.

2 Methods


2.1 Fractions Tutor

The ITS used in the present study used three different interactive representations of fractions: circles, rectangles, and number lines. Each representation emphasizes certain aspects of different conceptual interpretations of fractions [9]. The circle as a part-whole representation depicts fractions as parts of an area that is partitioned into equally-sized pieces. The rectangle is a more elaborate part-whole representation as it can be partitioned vertically and horizontally. At the same time, it does not have a standard shape for the unit, like the circle does. Finally, the number line is considered a measurement representation and thus emphasizes that fractions can be compared in terms of their magnitude, and that they fall between whole numbers.

The Fractions Tutor covers a comprehensive set of ten topics including interpreting representations, reconstructing the unit of fraction representations, improper fractions from representations, equivalent fractions, fraction comparison, fraction addition and subtraction. In our classroom study, students in all conditions first worked on six introductory problems that introduced the representations. They then worked on eight problems per fractions topic, yielding a total of 80 tutor problems. The sequence of tutor problems included both single-representation problems and (in the connection-making support conditions) multiple-representation problems.

Making Fractions

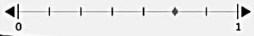
A Let's review a circle as an example to make a fraction!



Let's show $\frac{5}{7}$ on the circle.

- 1 Into how many equal sections must the unit be partitioned?
- 2 How many blue sections do you need to show $\frac{5}{7}$?
- 3 Look at the circle above to see the fraction.

B Let's use a number line to make a different fraction!



Let's show $\frac{5}{7}$ on the number line.

- 1 Into how many equal sections must the unit be partitioned?
- 2 How many sections do you need to show $\frac{5}{7}$?
- 3 Place a dot on the number line that shows $\frac{5}{7}$.

C What did we learn about the circle and the number line?

- 1 In the circle and the number line, the unit is partitioned into sections, and the number of sections is the .
- 2 The circle and the number line each show sections out of the unit, and that is the of the fraction.

? Hint

Students review a worked-out example with an area-model representation.

Then, students complete the same steps using a number line.

Finally, students are prompted to reflect on correspondences between representations.

Fig. 1. Example of sense-making support: worked-example problem

To support students in making connections between the different representations, we created three new types of tutor problems. WE problems and AL problems were designed to provide sense-making support. Each was designed to emphasize conceptual correspondences between the two representations. In the WE problems (see Fig. 1), an example of a solved problem with a familiar representation (i.e., circle or rectangle) was displayed on the left. This worked example contained filled-in answers for all except for the last step. After the student filled in the last step of the worked example, an isomorphic problem with a less familiar representation (number line) showed up on the right. The worked example served to guide students' work on this problem. To solve the problem, students manipulated the interactive number line. The AL problems followed the same side-by-side format with problem steps lined up, but there was no WE. Rather, as students completed the steps in the number line problem, the area model representation updated automatically to mimic the steps the student performed on the number line. In this sense, the more familiar representation provided feedback on the work with the less familiar representation. (To make this work at a technical level, we extended the CTAT tools [20] so that the number line component could serve as a controller for the area model component.) The WE and the AL problems included self-explanation prompts at the end of each problem (see bottom of Fig. 1) which asked students to identify correspondences of the two given representations.

The third type of connection-making problems, mixed representation problems (Mix; see Fig. 2), were designed to help students become fluent in connecting representations. Given a set of representations of fractions, students grouped them (through drag-and-drop) according to the fraction they represent. Students had to drag each individual graphical representation into the correct drop area labeled with a symbolic fraction. Students could drag-and-drop the fraction representations in any order. The drop area was able to detect which graphical representation the student drag-and-dropped into it, and could thereby give error feedback accordingly, when necessary. In each problem, multiple representations matched the same symbolic fraction.

Students received error feedback and hints on all steps. Hint messages and error feedback messages were designed to give conceptually oriented help, often in relation to the representations. The single-representation problems included prompts to help students relate the representations to the symbolic fractions. We had found these prompts to be effective in an earlier experimental study [3].

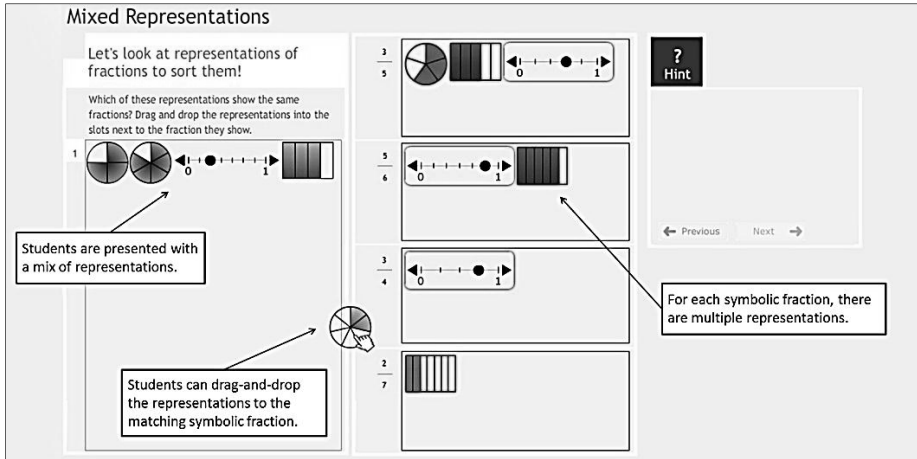


Fig. 2. Example of fluency support: mixed representations problem

2.2 Test Instruments

We assessed students’ knowledge of fractions at three test times. We created three equivalent test forms. Based on data from a pilot study with 61 4th-grade students, we made sure that the difficulty level of the test was appropriate for the target age group, and that the different test forms did not differ in difficulty. In our classroom study, we randomized the order in which the different test forms were administered.

The tests targeted two knowledge types: procedural and conceptual knowledge. The conceptual knowledge scale assessed students’ principled understanding of fractions. The test items included reconstructing the unit, identifying fractions from graphical representations, proportional reasoning questions, and verbal reasoning questions about comparison tasks. The procedural knowledge scale assessed students’ ability to solve questions by applying algorithms. The test items included finding a fraction between two given fractions using representations, finding equivalent fractions, addition, and subtraction. The theoretical structure of the test (i.e., the two knowledge types just mentioned) was based on a factor analysis with the pretest data from the current experiment. We validated the resulting factor structure using the data from the immediate and the delayed posttests.

2.3 Experimental Design and Procedure

In the present paper, we report the data from 599 4th- and 5th-grade students from one school district with 5 different elementary schools (25 classes) in the United States. Students participated in the study as part of their regular mathematics instruction. All students worked with versions of the Fractions Tutor designed and created specifically for this study. Students were randomly assigned to one of the conditions shown in Table 1. We used a 2 (fluency support) x 3 (sense-making support) + 1 (NL control condition) experimental design to investigate the effects of connection making support on students' learning of fractions. The fluency support factor had two levels: students either received Mix problems as fluency support, or no fluency support. The sense-making support factor had three levels: students either received WE problems or AL problems as sense-making support, or no sense-making support.

We assessed students' knowledge of fractions three times. On the first day, students completed a 30-minute pretest. They then worked on the Fractions Tutor for about ten hours, spread across consecutive school days. The day following the tutor sessions, students completed a 30-minute posttest. About one week after the posttest, we gave students an equivalent delayed posttest.

Table 1. Experimental conditions¹ included in the experimental study

		Sense-making support			Control
		None	Auto-linked representations	Worked example	
Fluency support	None	MGR	AL	WE	
	Mixed representations	Mix	AL-Mix	WE-Mix	
Control					NL

3 Results

Students who completed all tests, and who completed their work on the tutoring system were included in the analysis, yielding a total of $N = 428$. The number of students who were excluded from the analysis did not differ between conditions, $\chi^2(6, N = 169) = 4.34, p > .10$. Table 2 shows the means and standard deviations for the conceptual and procedural knowledge scales by test time and condition.

A hierarchical linear model (HLM; [21]) with four nested levels was used to analyze the data. HLMs are regression models that take into account nested sources of variability [21]. HLMs allow for significance testing in the same way as regular regression analyses do. We modeled performance for each of the three tests for each student (level 1), differences between students (level 2), differences between classes (level 3), and between schools (level 4). More specifically, we fit the following HLM:

¹ MGR = multiple graphical representations, AL = auto-linked representations, WE = worked examples, Mix = mixed representations, NL = number line.

$$\text{score}_{ij} = \text{test}_j + \text{sense}_k + \text{fluency}_1 + \text{sense}_k * \text{fluency}_1 + \text{pre}_i * \text{sense}_k + \text{pre}_i * \text{fluency}_1 + \text{student}(\text{class})_i + \text{class}(\text{school})_i + \text{school}_i, \tag{1}$$

with the dependent variable score_{ij} being student_i 's score on the dependent measures at test_j (i.e., immediate or delayed posttest). Sense_k indicates whether or not student_i received sense-making support, and fluency_1 indicates whether student_i received fluency support. In order to analyze whether students with different levels of prior knowledge benefit differently from connection-making support, we included students' pretest scores as a covariate (pre_i), and modeled the interaction of pretest score with sense-making support ($\text{pre}_i * \text{sense}_k$), and with fluency support ($\text{pre}_i * \text{fluency}_1$). $\text{Student}(\text{class})_i$, $\text{class}(\text{school})_i$, and school_i indicate the nested sources of variability due to the fact that student_i was in a particular class of a particular school. The reported p -values were adjusted for multiple comparisons using the Bonferroni correction. We report partial η^2 for effect sizes on main effects and interactions between factors, and Cohen's d for effect sizes of pairwise comparisons. An effect size partial η^2 of .01 corresponds to a small effect, .06 to a medium effect, and .14 to a large effect. An effect size d of .20 corresponds to a small effect, .50 to a medium effect, and .80 to a large effect.

Table 2. Proportion correct: means (and standard deviation) for conceptual and procedural knowledge at pretest, immediate posttest, delayed posttest. Min. score is 0, max. score is 1.

		pretest	immediate posttest	delayed posttest
conceptual knowledge	MGR	.33 (.20)	.45 (.23)	.48 (.26)
	AL	.38 (.20)	.49 (.23)	.51 (.26)
	WE	.36 (.22)	.43 (.20)	.49 (.26)
	Mix	.31 (.21)	.37 (.22)	.44 (.24)
	AL-Mix	.36 (.20)	.43 (.24)	.49 (.25)
	WE-Mix	.39 (.21)	.52 (.24)	.58 (.26)
	NL	.37 (.20)	.43 (.25)	.48 (.20)
procedural knowledge	MGR	.25 (.25)	.30 (.28)	.30 (.26)
	AL	.21 (.18)	.26 (.24)	.26 (.24)
	WE	.26 (.21)	.29 (.24)	.31 (.27)
	Mix	.19 (.17)	.23 (.20)	.25 (.22)
	AL-Mix	.20 (.18)	.25 (.21)	.26 (.21)
	WE-Mix	.26 (.20)	.32 (.26)	.33 (.26)
	NL	.21 (.20)	.25 (.22)	.27 (.23)

3.1 Effects of Connection-Making Support

We had expected that a combination of fluency support and sense-making support for connection making would lead to better results than either sense-making or fluency support alone. The results confirm our hypothesis for conceptual knowledge: we found a significant interaction effect between sense-making and fluency support on conceptual knowledge, $F(2, 351) = 3.97, p < .05, p. \eta^2 = .03$, such that students who received both types of support performed best on the conceptual knowledge posttests. The main effects of sense-making and fluency support were not significant ($F_s < 1$). There was no significant interaction effect on procedural knowledge ($F < 1$).

We had further predicted that WE problems would be the more effective type of sense-making support compared to AL problems. The results confirm this hypothesis for the conditions that received fluency support. Effect slices for the effect of sense-making support (i.e., a test of the effect of sense-making support for each level of the fluency support factor) showed that there was a significant effect of sense-making support within the conditions with fluency support on conceptual knowledge, $F(2, 343) = 4.34, p < .05, p. \eta^2 = .07$, but not within the conditions without fluency support ($F < 1$). *Post-hoc* comparisons between the Mix, AL-Mix, and the WE-Mix conditions confirmed that the WE-Mix condition significantly outperformed the Mix condition, $t(341) = 2.82, p < .01, d = .32$, and the AL-Mix condition $t(342) = 2.20, p < .05, d = .26$, on conceptual knowledge. In summary, WE problems are more effective in supporting sense-making of connections than AL problems, provided that students also receive fluency support.

Finally, to verify the advantage of receiving connection-making support over the NL control condition, we compared the most successful condition (WE-Mix) to the NL condition using *post-hoc* comparisons. The advantage of the WE-Mix condition over the NL was significant on conceptual knowledge, $t(115) = 2.41, p < .05, d = .27$.

3.2 Learning Effects

To investigate whether students learned from the pretest to the immediate posttest and to the delayed posttest across conditions, we modified the HLM and treated pretest scores as dependent variables, not as covariates (i.e., $pre_i, pre_i * sense_k,$ and $pre_i * fluency_l$ were excluded from the model in equation 1). The main effect for test was significant on procedural knowledge, $F(2, 842) = 43.04, p < .01, p. \eta^2 = .01$, and conceptual knowledge, $F(2, 842) = 98.56, p < .01, p. \eta^2 = .11$. Students in all conditions performed significantly better at the immediate posttest than at the pretest on conceptual knowledge, $t(842) = 9.15, p < .01, d = .40$ and on procedural knowledge, $t(842) = 7.15, p < .01, d = .20$. Similarly, students performed significantly better at the delayed posttest than at the pretest on conceptual knowledge, $t(842) = 13.80, p < .01, d = .60$ and on procedural knowledge, $t(842) = 8.70, p < .01, d = .24$.

4 Discussion and Conclusion

We had hypothesized that students would learn most robustly about fractions when being supported both in making sense of connections and in fluently making connections between multiple representations. Our results confirm this hypothesis for students' conceptual understanding of fractions: robust conceptual learning with multiple representations is enhanced by a combination of fluency support and sense-making support for connection making. We did not find effects of connection-making support on procedural knowledge. This finding is not surprising: it is conceivable that making connections between multiple representations benefits students' principled understanding of fractions but not their algorithmic knowledge of operations.

The fact that we did not find main effects of sense-making support and fluency support for connection making, on the other hand, is surprising: it shows that each type of connection-making support alone is not effective, but that the combination of both is needed to enhance students' conceptual understanding of fractions. This finding is particularly interesting because prior research on connection making has mostly focused on sense-making processes by supporting connection making of structurally equivalent elements. Our results suggest that standard sense-making support for connection making should be extended by also supporting fluency in making connections. It is possible that fluency activities allow students to deepen the conceptual knowledge about connections they acquired through sense-making activities.

With respect to *how* best to support sense making, our finding that WE support leads to better learning than AL support demonstrates, in line with earlier research on connection making [13], that students need to actively create connections between representations. We show that a novel application of WEs is effective in supporting active connection making. This finding extends the existing literature on WEs by showing that they can help students benefit from multiple representations when used as a means to support sense-making of connections.

As predicted, the advantage for combining fluency and sense-making support for connection making was also significant compared to the control condition who worked only with number lines. Number lines are often considered the most important graphical representation of fractions [10], which may lead teachers to use only number lines in fractions instruction. However, our findings show that with effective connection-making support, multiple representations of fractions can facilitate the acquisition of conceptual knowledge more so than practicing only the number line.

Finally, our results demonstrate significant learning gains for students who worked with the Fractions Tutor during their regular mathematics class. The gains persist at least until one week after the study when we administered the delayed posttest. This finding extends the ITS literature by demonstrating the effectiveness of a Cognitive Tutor for elementary-school students. Evaluation studies with ITSs have focused far more on high schools and middle schools than elementary schools [18,19]. Furthermore, the substantial and robust learning gains are encouraging, given that fractions are a difficult topic for elementary and middle-school students – a fact that provides a major obstacle for later mathematics learning, such as in algebra [8]. Our ITS for fractions is effective in helping students overcome some of these difficulties.

In conclusion, the present experiment extends the ITS and educational psychology literature on learning with multiple representations in several ways. First, our findings show that, although prior research has conceived of connection making as primarily a sense-making process, effective connection making involves fluency processes and therefore requires activities aimed at supporting sense making *and* activities aimed at supporting fluency. Second, we demonstrate that students need to be *active* in making connections between representations, and that a novel application of *worked examples* is effective in helping students to accomplish this difficult task. Third, the study provides insight into *the type of knowledge* for which connection-making support is beneficial. Connection-making support does not benefit students in learning to apply

algorithms to solve procedural tasks, but it helps them acquire conceptual knowledge of domain principles. Finally, our findings extend the findings on the effectiveness of Cognitive Tutors to the younger population of elementary school students.

Acknowledgements. This work was supported by the National Science Foundation, REESE-21851-1-1121307. We thank Ken Koedinger, Mitchell Nathan, Kathy Cramer, Peg Smith, Jay Raspat, Michael Ringenberg, Brian Junker, Howard Seltman, Cassandra Studer, the students, teachers, and principals, the CTAT and the Datashop teams, especially Mike Komisin and Alida Skogsholm for their efforts in retrieving the data on time.

References

- [1] Suh, J., Moyer, P.S.: Developing Students' Representational Fluency Using Virtual and Physical Algebra Balances. *Computers in Mathematics and Science Teaching* 26, 155–173 (2007)
- [2] Ainsworth, S.: DeFT: A conceptual framework for considering learning with multiple representations. *Learning and Instruction* 16, 183–198 (2006)
- [3] Rau, M.A., Alevan, V., Rummel, N.: Intelligent tutoring systems with multiple representations and self-explanation prompts support learning of fractions. In: 14th International Conference on Artificial Intelligence, pp. 441–448. IOS Press, Amsterdam (2009)
- [4] Bodemer, D., et al.: Supporting learning with interactive multimedia through active integration of representations. *Instructional Science* 33, 73–95 (2005)
- [5] van der Meij, J., de Jong, T.: Supporting Students' Learning with Multiple Representations in a Dynamic Simulation-Based Learning Environment. *Learning and Instruction* 16, 199–212 (2006)
- [6] Rau, M.A., et al.: How to schedule multiple graphical representations? A classroom experiment with an intelligent tutoring system for fractions. In: To appear in the Proceedings of ICLS 2012 (accepted, 2012)
- [7] Bodemer, D., et al.: The Active Integration of Information during Learning with Dynamic and Interactive Visualisations. *Learning and Instruction* 14, 325–341 (2004)
- [8] National Mathematics Advisory Panel: Foundations for Success: Report of the National Mathematics Advisory Board Panel, U.S. Government Printing Office (2008)
- [9] Charalambous, C.Y., Pitta-Pantazi, D.: Drawing on a Theoretical Model to Study Students' Understandings of Fractions. *Educational Studies in Mathematics* 64, 293–316 (2007)
- [10] Siegler, R.S., et al.: Developing effective fractions instruction: A practice guide. In: National Center for Education Evaluation and Regional Assistance. IES, U.S. Department of Education, Washington, DC (2010)
- [11] Koedinger, K.R., et al.: Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. *Cognitive Science* (in press)
- [12] Seufert, T., Brünken, R.: Cognitive load and the format of instructional aids for coherence formation. *Applied Cognitive Psychology* 20, 321–331 (2006)
- [13] Bodemer, D., Faust, U.: External and mental referencing of multiple representations. *Computers in Human Behavior* 22, 27–42 (2006)

- [14] Renkl, A.: The worked-out example principle in multimedia learning. In: Mayer, R. (ed.) *Cambridge Handbook of Multimedia Learning*, pp. 229–246. Cambridge University Press, Cambridge (2005)
- [15] Salden, R.J.C.M., Koedinger, K.R., Renkl, A., Alevan, V., McLaren, B.M.: Accounting for beneficial effects of worked examples in tutored problem solving. *Educational Psychology Review* 22, 379–392 (2010)
- [16] Berthold, K., Eysink, T., Renkl, A.: Assisting self-explanation prompts are more effective than open prompts when learning with multiple representations. *Instructional Science* 37, 345–363 (2009)
- [17] Koedinger, K.R., Corbett, A.: Cognitive Tutors: Technology Bringing Learning Sciences to the Classroom. In: Sawyer, R.K. (ed.) *The Cambridge Handbook of the Learning Sciences*, pp. 61–77. Cambridge University Press, New York (2006)
- [18] Koedinger, K.R., Corbett, A.: Cognitive tutors: Technology bringing learning sciences to the classroom. Cambridge University Press, New York (2006)
- [19] Rittle-Johnson, B., Koedinger, K.R.: Designing Knowledge Scaffolds to Support Mathematical Problem Solving. *Cognition and Instruction* 23, 313–349 (2005)
- [20] Alevan, V., et al.: A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education* 19, 105–154 (2009)
- [21] Raudenbush, S.W., Bryk, A.S.: *Hierarchical Linear Models: Applications and Data Analysis Methods*. Sage Publications, Newbury Park (2002)

Problem Order Implications for Learning Transfer

Nan Li, William W. Cohen, and Kenneth R. Koedinger

School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh PA 15213 USA
{nli1,wcohen,koedinger}@cs.cmu.edu

Abstract. The order of problems presented to students is an important variable that affects learning effectiveness. Previous studies have shown that solving problems in a blocked order, in which all problems of one type are completed before the student is switched to the next problem type, results in less effective performance than does solving the problems in an interleaved order. While results are starting to accumulate, we have little by way of precise understanding of the cause of such effect. Using a machine-learning agent that learns cognitive skills from examples and problem solving experience, SimStudent, we conducted a controlled simulation study in three math and science domains (i.e., fraction addition, equation solving and stoichiometry) to compare two problem orders: the blocked problem order, and the interleaved problem order. The results show that the interleaved problem order yields as or more effective learning in all three domains, as the interleaved problem order provides more or better opportunities for error detection and correction to the learning agent. The study shows that learning when to apply a skill benefits more from interleaved problem orders, and suggests that learning how to apply a skill benefits more from blocked problem orders.

Keywords: learning transfer, learner modeling, interleaved problem order, blocked problem order.

1 Introduction

One of the most important variables that affects learning effectiveness is the order of problems presented to students. While most existing textbooks organize problems in a blocked order, in which all problems of one type (e.g. learning to solve equations of the form $S_1/V=S_2$) are completed before the student is switched to the next problem type, it is surprising that problems in an interleaved order often yields more effective learning. Numerous studies have experimentally demonstrated this effect (e.g., [18,6,2,9,23,4,17,7]). However, the cause of the the effect is still unclear. A computational model that demonstrates such behavior would be a great help in better understanding this widely-observed phenomena, and might reveal insights that can improve current education technologies.

- Skill divide (e.g. $-3x = 6$)
- Perceptual information:
 - Left side ($-3x$)
 - Right side (6)
- Precondition:
 - Left side ($-3x$) does not have constant term
- Operator sequence:
 - Get coefficient (-3) of left side ($-3x$)
 - Divide both sides with the coefficient (-3)

Fig. 1. A production rule for divide

In this paper, we conducted a controlled-simulation study using a machine-learning agent, SimStudent. SimStudent was trained on real-student problems that were of blocked orders or interleaved orders. We then tested whether the advantages of interleaved problem orders over blocked problem orders are exhibited in all three domains. After that, we carefully inspected what causes such effect by inspecting SimStudent’s learning processes and learning outcomes, which are not easily obtainable from human subjects.

2 A Brief Review of SimStudent

SimStudent is a machine-learning agent that inductively learns skills to solve problems from demonstrated solutions and from problem solving experience. It is an extension of programming by demonstration [8] using inductive logic programming [13] as an underlying learning technique. In the rest of this section, we will briefly review the learning mechanism of SimStudent. For full details, please refer to [10].

SimStudent learns production rules as skills to solve problems. During the learning process, given the current state of the problem (e.g., $-3x = 6$), SimStudent first tries to find an appropriate production rule that proposes a plan for the next step (e.g., (*coefficient* $-3x$ *?coef*) (*divide* *?coef*)). If it finds a plan and receives positive feedback, it continues to the next step. If the proposed next step is incorrect, negative feedback and a correct next step demonstration are provided to SimStudent. The learning agent will attempt to learn or modify its production rules accordingly. If it has not learned enough skill knowledge and fails to find a plan, a correct next step is directly demonstrated to SimStudent for later learning.

Figure 1 shows an example of a production rule learned by SimStudent in a readable format¹. A production rule indicates “where” to look for information in the interface, “how” to change the problem state, and “when” to apply a rule. For example, the rule to “divide both sides of $-3x=6$ by -3 ” shown in Figure 1 would

¹ The actual production rule uses a LISP format.

be read as “given a left-hand side ($-3x$) and a right-hand side (6) of the equation, when the left-hand side does not have a constant term, then get the coefficient of the term on the left-hand side and divide both sides by the coefficient.”

As there are three main parts in a production rule, SimStudent’s learning mechanism also consists of three parts: a “where” learner, a “when” learner, and a “how” learner. The “where” learner acquires knowledge about where to find useful information in the GUI. For example, for the step *divide -3, -3x* and 6 are the useful information, the GUI elements associated with them are *Cell 21* and *Cell 22*. The learning task is to find paths that identify such elements. All of the elements in the interface are organized in a tree structure. For instance, if the GUI has a table in it, the table node has columns as children, and each column has multiple cells as children. For each cell, SimStudent uses a *deep feature* learning mechanism that acquires knowledge on how to further parse the content in each cell into a cell parse tree. When given a set of positive examples (i.e., GUI elements associated with useful information in the steps), the learner carries out a specific-to-general learning process (e.g., from *Cell 21* to *Cell ?1* to *Cell ??*). It finds the most specific paths that cover all of the positive examples.

The “when” learner acquires the precondition of the production rule that describes the desired situation to apply the rule (e.g. (*not (has-constant ?var1)*)) given a set of *feature predicates*. Each predicate is a boolean function of the arguments that describes relations among objects in the domain. For example, (*has-coefficient -3x*) means $-3x$ has a coefficient. The “when” learner utilizes FOIL [15] to acquire the precondition as a set of feature tests. FOIL is an inductive logic programming system that learns Horn clauses from both positive and negative examples expressed as relations. If a step is either demonstrated to SimStudent or receives positive feedback, that step is a positive example for FOIL; otherwise, a negative example.

The last component is the “how” learner which acquires knowledge about how to change the problem state. Given all of the positive examples and a set of *basic operator functions* (e.g., (*divide ?var*)), the “how” learner attempts to find a shortest operator function sequence that explains all of the training examples using iterative-deepening depth-first search.

3 Problem Order Study

To get a better understanding of how and why problem orders affect learning efficiency, we carried out a controlled simulation study on SimStudent given different problem orders.

3.1 Methods

To ensure the generality of the results, we selected three math and science domains: fraction addition, equation solving, and stoichiometry. Both the training and testing problems were selected from problems solved by human students in classroom studies. SimStudent was tutored by interacting with automatic tutors that simulate the automatic tutors used by human students.

Fraction Addition: In the fraction addition domain, SimStudent was given a series of fraction addition problems of the form

$$\frac{\text{numerator}_1}{\text{denominator}_1} + \frac{\text{numerator}_2}{\text{denominator}_2}$$

All numerators and denominators are positive integers. The problems are of three types in the order of increasing difficulty: 1) *easy problems*, where the two addends share the same denominators (i.e., $\text{denominator}_1 = \text{denominator}_2$, e.g., $1/4 + 3/4$), 2) *normal problems*, where one denominator is a multiple of the other denominator (i.e., $\text{GCD}(\text{denominator}_1, \text{denominator}_2) = \text{denominator}_1$ or denominator_2 , e.g., $1/2 + 3/4$), 3) *hard problems*, where no denominator is a multiple of the other denominator (e.g., $1/3 + 3/4$). In this case, students need to find the common denominator (e.g. 12 for $1/3 + 3/4$) by themselves. Both the training and testing problems were selected from a classroom study of 80 human students using an automatic fraction addition tutor. The number of training problems is 20, and the number of testing problems is 6.

Equation Solving: The second domain in which we tested SimStudent is equation solving. Equation solving is a more challenging domain since it requires more complicated prior knowledge to solve the problem. For example, it is hard for human students to learn what is a coefficient, and what is a constant. Also, adding two terms together is more complicated than adding two numbers.

In this experiment, we evaluated SimStudent based on a dataset of 71 human students in a classroom study using an automatic tutor, CTAT [1]. The problems are also in three types: 1) problems of the form $S_1 + S_2V = S_3$, 2), $V/S_1 = S_2$, 3) $S_1/V = S_2$, where S_1 and S_2 are signed numbers, and V is a variable. Note that the terms in the above problem forms can appear in any order, and surrounded with parenthesis. There were 12 training problems, and 11 testing problems in the experiment.

Stoichiometry: Lastly, we evaluated SimStudent in a chemistry domain, stoichiometry. Stoichiometry is a branch of chemistry that deals with the relative quantities of reactants and products in chemical reactions. We selected stoichiometry because it is different from equation solving and fraction addition in nature. In the stoichiometry domain, SimStudent was asked to solve problems such as “How many moles of atomic oxygen (O) are in 250 grams of P_4O_{10} ? (Hint: the molecular weight of P_4O_{10} is 283.88 g P_4O_{10} / mol P_4O_{10} .)”. 8 training problems and 3 testing problems were selected from a classroom study of 81 human students using an automatic stoichiometry tutor [11].

To solve the problems, SimStudent needs to acquire three types of skills: 1) unit conversion (e.g. $0.6 \text{ kg H}_2\text{O} = 600 \text{ g H}_2\text{O}$), 2) molecular weight (e.g. There are 2 moles of P_4O_{10} in $283.88 \times 2 \text{ g P}_4\text{O}_{10}$), 3) composition stoichiometry (e.g. There are 10 moles of O in each mole of P_4O_{10}). The problems are of three types ordered in increasing difficulty, where each later type adds one more skill comparing with its former type.

Measurement: To measure learning gain, the production rules learned by SimStudent were tested on the testing problems each time tutoring was done on a single training step. For each step in the testing problems, we measure a *step score* for it. In math and science problems, there is often more than one way to solve one problem. Hence, at each step, there is usually more than one production rule that is applicable. In this case, among all possible correct next steps, we count the number of correct steps that are actually proposed by some applicable production rule, and report the step score as the number of correct next steps covered by learned rules divided by the total number of correct next steps plus the number of incorrect next steps proposed by SimStudent, i.e.,

$$\frac{\#OfCorrectNextStepsProposed}{Total\#OfCorrectNextSteps + \#OfIncorrectNextStepsProposed}$$

For example, if there are four possible correct next steps, and SimStudent proposes three, of which two are correct, and one is incorrect, then only two correct next steps are covered, and thus the step score is $2/(4 + 1) = 0.4$. We report the average step score over all testing problem steps for each curriculum.

3.2 Blocked vs. Interleaved Problem Orders

To manipulate the order of problems given to SimStudent, for each domain, we first grouped the problems of the same type together. Since there were three types of problems, we had three groups in each domain: *group - 1*, *group - 2*, and *group - 3*. Then, there were six different orders of these three groups. For each order (e.g. [*group - 1*, *group - 2*, *group - 3*]), we generated one blocked-ordering curriculum by repeating the same type of problems² in each group right after that group’s training was done (e.g., [*group - 1*, *group - 1'*, *group - 2*, *group - 2'*, *group - 3*, *group - 3'*]). To generate the interleaved-ordering curriculum, the same types of problems will be repeated once the whole set of problems were done (e.g. [*group - 1*, *group - 2*, *group - 3*, *group - 1'*, *group - 2'*, *group - 3'*]).

Table 1. 12 curricula of different orders for each domain

Blocked-Ordering Curricula	Interleaved-Ordering Curricula
1, 1', 2, 2', 3, 3'	1, 2, 3, 1', 2', 3'
1, 1', 3, 3', 2, 2'	1, 3, 2, 1', 3, 2'
2, 2', 1, 1', 3, 3'	2, 1, 3, 2', 1', 3'
2, 2', 3, 3', 1, 1'	2, 3, 1, 2', 3', 1'
3, 3', 1, 1', 2, 2'	3, 1, 2, 3', 1', 2'
3, 3', 2, 2', 1, 1'	3, 2, 1, 3', 2', 1'

After this manipulation, we ended up having 12 curricula of different orders for each domain as shown in Table 1. Six of them were blocked-ordering curricula, whereas the other six were interleaved-ordering curricula. SimStudent was

² The problems will be of the same form, but with different values. For example, $3x = 6$ may be replaced by $4x = 8$.

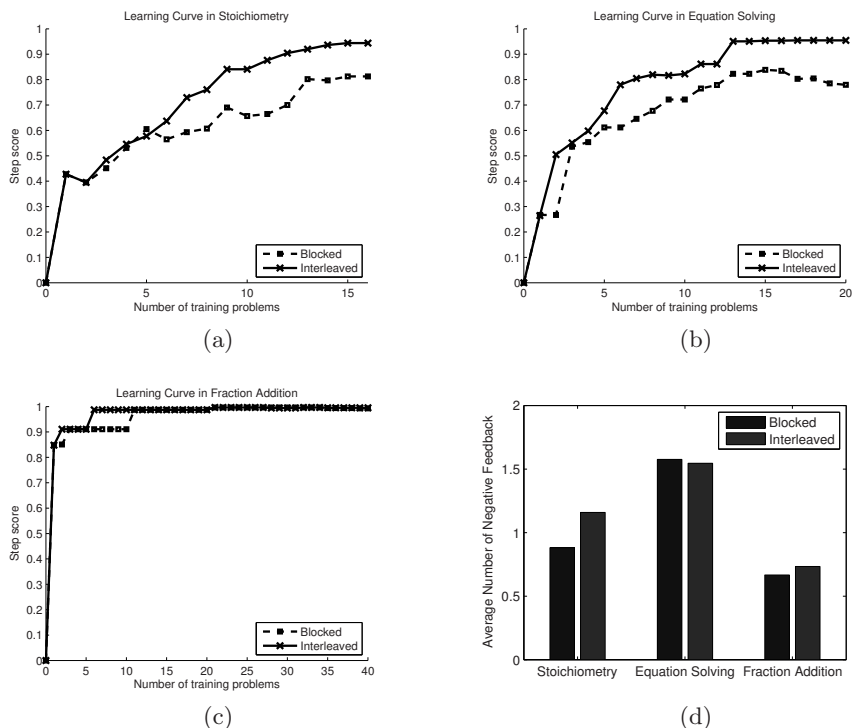


Fig. 2. Learning curves of blocked-ordering curricula vs. interleaved-ordering curricula in three domains, a) stoichiometry, b) equation solving, c) fraction addition, and the average number of times SimStudent receives negative feedback for each skill across three domains

trained and tested on all these curricula, the results are the average step scores over curricula of the same type (blocked or interleaved).

3.3 Results

Figure 2 shows the learning curves of SimStudent trained on blocked-ordering or interleaved-ordering curricula. As we can see in the graph, in all three domains, the interleaved-ordering curricula yielded as or more effective learning than the blocked-ordering curricula.

In the domain of stoichiometry, the step score of the interleaved-ordering curricula was 0.944, whereas the step score of the blocked-ordering curricula was 0.813. A sign test between pairs of step scores achieved by the associated interleaved-ordering and blocked-ordering curricula (e.g., $[group - 1, group - 2, group - 3, group - 1', group - 2', group - 3']$ vs. $[group - 1, group - 1', group - 2, group - 2', group - 3, group - 3']$) showed that, after trained on 40 problems, the interleaved-ordering curricula is significantly ($p < 0.05$) more effective than the blocked-ordering curricula.

Similar results were also observed in the equation solving domain. The interleaved-ordering curricula again showed a benefit (0.955 vs. 0.858) over blocked-ordering curricula. The sign test also demonstrated significant ($p < 0.05$) advantages of interleaved-ordering curricula over the blocked-ordering curricula.

In fraction addition, SimStudent got an average step score of 0.995 when trained with interleaved-ordering curricula, which is slightly higher than the step score SimStudent received (0.993) when trained with blocked-ordering curricula. There was no significant difference between the two conditions.

3.4 Implications for Instructional Design

We can inspect the data more closely to get a better qualitative understanding of why the SimStudent model is better and what implications there might be for improved instruction. In two of three domains, interleaved-ordering curricula are more advantageous than blocked-ordering curricula. These results provide theoretical support for the hypothesis that when teaching human students in math and science domains, an interleaved problem order yields better learning than a blocked problem order.

To better understand the cause of the advantages of interleaved-ordering curricula, we further measured the amount of negative feedback received by SimStudent, as it is one of the important factors in achieving effective learning. The amount of negative feedback is assessed by the average number of times SimStudent received negative feedback for each skill. As presented in Figure 2(d), the SimStudent given interleaved-ordering problems receives significantly ($p < 0.05$, 31.5%) more negative feedback than the SimStudent trained on blocked-ordering problems in stoichiometry, and 10.0% more negative feedback in fraction addition.

One possible explanation for this is when problems are of an interleaved order, SimStudent may incorrectly apply the production rules learned from previous problem types to the current problem, even if the current problem is of another type. In this case, SimStudent receives explicit negative feedback from the tutor. In contrast, when trained on blocked-ordering curricula, SimStudent has fewer opportunities for incorrect rule applications, and thus receives less negative feedback. Since the negative feedback serves as negative training examples of the “when” learning, more negative feedback in the interleaved problem order case enables SimStudent to yield more effective “when” learning compared to blocked problem orders. Although SimStudent received approximately the same amount of negative feedback ($p = 1$, -1.9%) in the blocked problem order case and interleaved problem order case, a careful inspection shows that negative examples from other problem types are sometimes more informative than those from the same problem type. For example, in algebra, during the acquisition of the skill “subtract”, the SimStudent given blocked-ordering problems learned that when there is a constant term in either side of the equation (e.g., term S_2 is a number in $S_1 V + S_2 = S_3$), subtract both sides with that number (e.g., (*subtract* S_2)). But it failed to learn that there must be a plus sign before S_2 . In the interleaved condition, SimStudent received negative feedback when it tried to subtract both sides with S_2 when given problems of type $S_1 / V = S_2$. Then, the SimStudent given

interleaved-ordering problems modified its when-part. The updated production rule became, “when there is a constant term that follows a plus sign in either side of the equation, subtract both sides with that number.”

We conjecture that the frequent use of blocked examples in textbooks might relate to perceived memory limitations of students. SimStudent currently does not have any severe memory (or retrieval) limitations (e.g., it remembers all past examples no matter how long ago). SimStudent would need to have some memory limitations if it were to have a bigger knowledge base or to better model humans. If it did, the benefits for blocking may go up, and in particular for “how” learning. Let’s consider a fixed memory size for SimStudent, which means SimStudent is only able to remember a fixed number of most recent training examples. SimStudent receives training examples of “how” learning only when the current step is demonstrated or SimStudent applies a production rule correctly. Hence, in the blocked problem order case, SimStudent maintains all the training examples of the current problem type unless the number of training examples exceeds the memory limit. In contrast, when trained on interleaved-ordering curricula, SimStudent needs to remember training examples for multiple problem types. For any specific production rule, the number of training examples will be smaller than that given a blocked-ordering curricula, which could result in less effective learning than the blocked-ordering case.

This also relates to VanLehn’s work on “learning one subprocedure per lesson” [20]. If a subprocedure is achieved in the same way, that is, with the same how-part in the production rule, then as Vanlehn suggested, problems of blocked orders are more beneficial. However, for production rules/procedures to differentiate across subgoals, the when-part needs to be acquired and in that case, interleaving problems of different types is important.

In summary, the study shows that learning when to apply a skill benefits more from interleaved problem orders, and suggests that learning how to apply a skill benefits more from blocked problem orders. Therefore, when tutoring students in domains that are more challenging in “how” learning, we suggest that the problems presented to students should be of blocked orders. If the learning task requires more rigorous “when” learning, interleaved-ordering problems should be preferred.

4 Related Work

The main objective of this work is to better understand how and why problem orders affect learning outcome using a learning agent. A considerable amount of research has demonstrated the effectiveness of interleaved problem orders. Shea and Morgan [18] were the first that showed problems of a random order yields better performance in retention and transfer tests than students trained on problems of a blocked order, and named this effect as the *contextual interference (CI) effect*. The CI effect compares random problem orders and blocked problem orders, not interleaved problem orders and blocked orders, but the results should be similar since the main point is whether consecutive problems should be of the same or different types. That is, random problem orders have lots of interleaving. After that, a growing number of studies (e.g., [6,2,9,23,4,17,7]) have repeatedly

observed the CI effect in different tasks. Other studies on relatively complex tasks (e.g., [19]) or novices (e.g., [5]) have yielded mixed results. To explain the CI phenomenon, researchers have proposed several hypothesis including the elaboration hypothesis [18], the forgetting or reconstruction hypothesis [9], etc. More details on these hypotheses are available in [22], however, all are described in fairly ambiguous language and none have the precision of a computational theory. In contrast, SimStudent provides a precise, unambiguous implementation of how and why interleaving may be effective.

Research on task switching [12] shares a resemblance with our work. It shows that subjects' responses are substantially slower and more error-prone immediately after a task switch. Our work differs from this research in that we focus on learning tasks. During the learning process, switching among problems of different types also increases the cognitive load, but causes more effective learning.

Other research on creating simulated students [21,3,14] and simulating expert memory [16] also share some resemblance to our work. VanLehn [21] created a learning system and evaluated whether it was able to learn procedural "bugs" like real students. To the best of our knowledge, none of the above approaches made use of the models to simulate the advantage of interleaved or random problem orders over blocked problem orders.

5 Concluding Remarks

In spite of the promising results, there remain several fruitful future steps. First, the current study used only one set of problems in each domain. To evaluate the generality of the claim, we should carry out the same set of experiments using other problem sets or in other domains. Second, we would like to carry out more studies in which SimStudent has limited memory, and validate whether "how" learning gains more from blocked problem orders in this case. Last, future research could apply the theoretical implications in a study on human students, and evaluate the validity of the recommended tutoring strategy.

In this paper, we carried out a controlled simulation study to gain a better understanding of why interleaved problem orders generate more effective learning than blocked problem orders. We measured the learning effectiveness of a machine-learning agent, SimStudent, in three domains given different problem orders. The results show that since the interleaved problem order yields more opportunities for error detection and correction, the SimStudent trained by interleaved-ordering curricula achieved better performance than the SimStudent trained by blocked-ordering curricula.

References

1. Alevan, V., McLaren, B.M., Sewall, J., Koedinger, K.R.: A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education* 19, 105–154 (2009)
2. Carnahan, H., Van Eerd, D.L., Allard, F.: A note on the relationship between task requirements and the contextual interference effect. *Journal of Motor Behavior* 22(1), 159–169 (1990)

3. Chan, T.-W., Chou, C.-Y.: Exploring the design of computer supports for reciprocal tutoring. *International Journal of Artificial Intelligence in Education* 8, 1–29 (1997)
4. Del Rey, P.: Effects of contextual interference on the memory of older females differing in levels of physical activity. *Perceptual and Motor Skills* 55(1), 171–180 (1982)
5. French, K.E., Rink, J.E., Werner, P.F.: Effects of contextual interference on retention of three volleyball skills. *Perceptual and Motor Skills* 71, 179–186 (1990)
6. Gabriele, T.E., Hall, C.R., Buckolz, E.E.: Practice schedule effects on the acquisition and retention of a motor skill. *Human Movement Science* 6, 1–16 (1987)
7. Jelsma, O., Pieters, J.M.: Practice schedule and cognitive style interaction in learning a maze task. *Applied Cognitive Psychology* 3(1), 73–83 (1989)
8. Lau, T., Weld, D.S.: Programming by demonstration: An inductive learning formulation. In: *Proceedings of the 1999 International Conference on Intelligence User Interfaces*, pp. 145–152 (1998)
9. Lee, T.D., Magill, R.A.: The locus of contextual interference in motor-skill acquisition. *Journal of Experimental Psychology. Learning Memory and Cognition* 9(4), 730–746 (1983)
10. Li, N., Cohen, W.W., Koedinger, K.R.: Integrating representation learning and skill learning in a human-like intelligent agent. Tech. Rep. CMU-MLD-12-1001, Carnegie Mellon University (January 2012)
11. McLaren, B.M., Lim, S.-J., Koedinger, K.R.: When and how often should worked examples be given to students? new results and a summary of the current state of research why isn't the science done? *Cognitive Science*, 2176–2181 (2008)
12. Monsell, S.: Task switching. *Trends in Cognitive Sciences* 7(3), 134–140 (2003)
13. Muggleton, S., de Raedt, L.: Inductive logic programming: Theory and methods. *Journal of Logic Programming* 19, 629–679 (1994)
14. Pentti Hietala, T.N.: The competence of learning companion agents. *International Journal of Artificial Intelligence in Education* 9, 178–192 (1998)
15. Quinlan, J.R.: Learning logical definitions from relations. *Mach. Learn.* 5(3), 239–266 (1990)
16. Richman, H.B., Staszewski, J.J., Simon, H.A.: Simulation of expert memory using epam iv. *Psychological Review* 102(2), 305–330 (1995)
17. Sekiya, H., Magill, R.A., Anderson, D.I.: The contextual interference effect in parameter modifications of the same generalized motor program. *Research Quarterly for Exercise and Sport* 67(1), 59–68 (1996)
18. Shea, J.B., Morgan, R.L.: Contextual interference effects on the acquisition, retention, and transfer of a motor skill. *Journal of Experimental Psychology Human Learning Memory* 5(2), 179–187 (1979)
19. Tsutsui, S., Lee, T.D., Hodges, N.J.: Contextual interference in learning new patterns of bimanual coordination. *Journal of Motor Behavior* 30(2), 151–157 (1998)
20. VanLehn, K.: Learning one subprocedure per lesson. *Artificial Intelligence* 31, 1–40 (1987)
21. VanLehn, K.: *Mind Bugs: The Origins of Procedural Misconceptions*. MIT Press, Cambridge (1990)
22. Wulf, G., Shea, C.H.: Principles derived from the study of simple skills do not generalize to complex skill learning. *Psychonomic Bulletin Review* 9(2), 185–211 (2002)
23. Young, D.E., Cohen, M.J., Husak, W.S.: Contextual interference and motor skill acquisition: On the processes that influence retention. *Human Movement Science* 12(5), 577–600 (1993)

Knowledge Component Suggestion for Untagged Content in an Intelligent Tutoring System

Mario Karlovčec¹, Mariheida Córdova-Sánchez², and Zachary A. Pardos³

¹ Department of Computer Science, Jozef Stefan Institute, Slovenia

² Department of Computer Science, Purdue University, USA

³ Department of Computer Science, Worcester Polytechnic Institute, USA

mario.karlovcec@ijs.si, cordovas@purdue.edu, zpardos@wpi.edu

Abstract. Tagging educational content with knowledge components (KC) is key to providing useable reports to teachers and for use by assessment algorithms to determine knowledge component mastery. With many systems using fine-grained KC models that range from dozens to hundreds of KCs, the task of tagging new content with KCs can be a laborious and time consuming one. This can often result in content being left untagged. This paper describes a system to assist content developers with the task of assigning KCs by suggesting knowledge components for their content based on the text and its similarity to other expert-labeled content already on the system. Two approaches are explored for the suggestion engine. The first is based on support vector machines text classifier. The second utilizes K-nearest neighbor algorithms employed in the Lemur search engine. Experiments show that KCs suggestions were highly accurate.

Keywords: Intelligent Tutoring Systems, Text Mining, Knowledge Components, TextGarden, Lemur, Bag-of-Words.

1 Introduction

When designing exercises within the learning software, appropriate knowledge components should be assigned to them. “A knowledge component is a description of a mental structure or process that a learner uses, alone or in combination with other knowledge components, to accomplish steps in a task or a problem.” [1] The process of assigning knowledge components to the exercises can be a time consuming job, since the number of possible knowledge components can be very large. In order to help the tutor or course designer in writing exercises we have proposed two approaches that suggest knowledge components. The first approach is based on text mining [2] and SVM classification algorithm and the second is based on a search engine with a KNN classification algorithm [3]. These two approaches can be used for a system that could encourage the course designers to assign knowledge components to new exercises they design, as well as to existing exercises that do not have knowledge components assigned to them.

2 Related Work

This work continues the line of research proposed by Rose *et al.* [4] and expands on the prior art by applying a variety of optimizations as well as evaluating the algorithms on numerous KC models of varying granularity. The work by Rose *et al.* presented KC prediction results on a model of 39 KCs but skill models have since increased in complexity. We investigate how KC prediction accuracy scales with larger KC models and which algorithms adequately meet this challenge.

The necessity of associating knowledge components with problem solving items is shared by a number of tutoring systems including The Andes physics tutor [5], The Cognitive Tutors [6] and the ASSISTments Platform [7]. The Andes and Cognitive tutors use student modeling to determine the amount of practice each individual student needs for each KC. The student model that these tutors use is called Knowledge Tracing [6], which infers student knowledge over time from the history of student performance on items of a particular KC. This model depends on the quality of the KC model to make accurate predictions of knowledge.

The KC association with items in a tutor is typically represented in an *Item* \times *KC* lookup table called a Q-matrix [8]. Methods such as Learning Factors Analysis [9] have been proposed to automate the improvement of this Q-matrix in order to improve the performance of the student model. Recently, non-negative matrix factorization methods have been applied in order to induce this Q-matrix from data [10]. The results of this work are promising but its applications so far are limited to test data where there is no learning occurring and only to datasets with only around five KCs, where these KCs represent entirely different high level topic areas such as Math and English which do not intersect. All the student modeling and Q-matrix manipulation methods have so far not tapped any information in the text of the items they are evaluating. This paper will make the contribution of looking at this source of information for making accurate KC predictions. While this paper focuses on text mined KC suggestion to aid content developers, this technique is relevant to those interested in Q-matrix improvement as well.

3 The ASSISTments Platform

The dataset we evaluated comes from The ASSISTments Platform. The ASSISTments platform is a web based tutoring system that assists students in learning, while it gives teachers assessment of their students' progress. The system started in 2004 with a focus on 8th grade mathematics, in particular helping students pass the Massachusetts state test. It has since expanded to include 6th through 12th grade math and scientific inquiry content.

A feature that sets ASSISTments apart from other systems is its robust web based content building interface [7] that is designed for rapid content development by system experts and teachers alike. Teachers are responsible for a growing majority of the content in ASSISTments. While the content has been vetted and verified as being of educational value by ASSISTments system maintainers, the content often lacks meta

information such as KC tagging as this is an optional step in content creation. An ASSISTments administrator must add this tagging or leave it blank which can cause a lack of accuracy in student model analysis of the data and also inhibits the system from reporting KC information to teachers. The tagging has to be performed by selecting from the large list of KC, which are organized into 5 categories and sorted alphabetically within the categories. Untagged content in ASSISTments is a growing phenomenon with only 29% of the content possessing KC tags as of this writing. Accurate KC suggestion would expedite the processes of content tagging and encourage external content builders to tag their content.

4 Data

The dataset used for testing the performance of the proposed approaches was taken from tagged content on the ASSISTments Platform during the 2005-2006 school year. The ASSISTments Platform has three KC models consisting of varying degrees of granularity. The first two models, containing 5 and 39 KCs, use KC names corresponding to the Massachusetts state math standards. The system's finest-grained KC model contains 106 KCs which were created in-house [11]. The KCs from the 106 model have a hierarchical relationship to the 39 KC and 5 KC models. This allows content to be tagged only with the 106 KCs and then inherit the KCs from the other models in the hierarchy. While tagging with the 106 model is preferred, content builders can choose from KCs from any model to tag their content.

5 Approaches

In order to solve the problem of assigning appropriate KCs by providing automatic suggestion system in the process of exercise design, two approaches are suggested: a text mining approach using the SVM classifier and the search engine based approach with the KNN classifier.

5.1 Text Mining Approach with SVM Classifier

One approach was based on text mining and building SVM classification model using the Text Garden [12] utility. It has been shown [2] that the SVM is an appropriate method for text classification. The main reasons include the ability to handle high dimensional input space and suitability for problems with dense concepts and sparse instances. The classification model was built based on set of labeled exercises. We wanted to test the influence of stop words removal and stemming on the classification problem, so four different classification models were built, covering all combinations of applying these standard text processing techniques.

5.2 Search Engine and KNN Approach

The second approach was to use the Lemur Toolkit [3] with a K Nearest Neighbors (KNN) classification algorithm. KNN is a commonly used algorithm that finds the K documents closest (most similar) to the document being tested. The Lemur Toolkit is an open source search engine. The questions in the training set were indexed using Lemur. The text of the test set questions were then used as queries against the indexed questions. The top k (in this case k=200) most relevant search results, most relevant to the query, were retrieved (along with their KC tag). Each retrieved document was assigned a score based on its rank (e.g. the score of the top document is 200, the score of the second retrieved document is 199, and so on). We calculate a score for a tag as

$$tag_score(t) = \frac{a}{\sum score(t) + b}$$

where $\sum score(t)$ is the summation of all document scores with tag t , and a and b were both chosen to be two times the KC model size. This is done to predict KCs using a weighted measure of the frequencies of tags (i.e. KCs) and their retrieval ranks. Lastly, for each unlabeled question (query), the tag with the highest tag_score is assigned to it.

6 Results

Testing was performed using the 5 folds cross validation method. All experiments used accuracy as the goodness metric. For both approaches, testing was performed on the three different knowledge component models: the largest model with 106 KCs, the 39 KC model and the 5 KC model. In, [13] the automatic text generation of mathematical word problems is performed. The paper shows that leaving out the common text processing techniques, namely stop word removal and stemming, can increase the performance of text categorization. To take into account the findings of that paper, we tested each dataset with four different text processing setting: (1) without applying stop-words removal and stemming (SVM); (2) applying only stop-words removal; (3) applying only stemming; and (4) applying both stop-words removal and stemming.

Table 1. Experimental results of proposed approaches for suggesting knowledge components

Dataset	Number of suggestions									
	SVM					KNN				
	1	2	3	4	5	1	2	3	4	5
106 KC	0.607	0.739	0.784	0.809	0.823	0.574	0.736	0.796	0.835	0.865
106 KC ST	0.621	0.749	0.798	0.824	0.842	0.567	0.728	0.795	0.834	0.866
39 KC	0.683	0.815	0.863	0.895	0.914	0.666	0.815	0.854	0.898	0.914
39 KC ST	0.689	0.818	0.870	0.901	0.916	0.653	0.829	0.865	0.907	0.914
5 KC	0.814	0.943	0.969	0.981	1.000	0.762	0.919	0.976	0.993	1.000
5 KC ST	0.815	0.938	0.969	0.983	1.000	0.784	0.923	0.976	0.996	1.000

The experimental results of both approaches are shown in Table 1. The table shows accuracy results given KC suggestions ranging from 1 to 5. The accuracy when suggesting 5 KCs, for example, is the percentage of exercises where the correct KC was among the top 5 suggested KCs. For the 5 KC model, 5 suggestions always results in 100% accuracy. Each row represents a different dataset with classification algorithm and text processing settings used in the experiment. 106KC, 39KC and 5KC are labels for the different knowledge components models. SVM and KNN are labels for the two different classification algorithms. ST indicates applying stemming. Each column of the table represents different number of suggestions. Performance of stop-words removal did worse than stemming. Performance of stop-words removal in addition to stemming also did worse than just stemming. These results were not shown in the table for space reasons. The results in this table are represented with sensitivity. Sensitivity in information retrieval is recall for the binary classification problems. It is the probability that a relevant KC is suggested for the exercise. The Kappa value for the 39 KC and 106 KC model was 0.669 and 0.610 respectively.

7 Discussion

Results of the experimental testing indicate that the proposed approaches are suitable for practical usage. Table 1 show the results, which are grouped according to the model of KC used for testing. The SVM classifier with stemming performs the best for every KC model. The dataset with 106 KC model is the hardest challenge for the proposed approach, but this is the KC model for which the system can be mostly useful in practical application. If only one KC is suggested for the 106 KC model, it would be the correct one in 62.1% of the cases. Suggestion systems usually suggest more than one option, if the number of these suggestions is 5; the correct KC is among these 5 in 84.2 % of the cases, or in 88.9% of the cases if there are 10 suggestions. If the number of suggestions increases, the probability that the correct KC is among them naturally grows, but the effort required from the user to choose the correct KC among the suggested also increases. Comparing the results with all four combinations of typical text processing procedures applied (stop-word removal and stemming), not removing stop-words and performing stemming improves the accuracy of the system as suggested by [13]. However this improvement is much less significant than in the referenced paper. The improvement for the best options in comparison with the worst option - removing stop-words and no stemming, were around 2%.

Results indicate that both suggested approaches are suitable for practical usage, since they would decrease significantly the number of KCs to be used for labeling, without compromising much on efficiency (i.e. failing to show the correct labels).

Acknowledgements. This research was supported by the National Science Foundation via the “Graduates in K-12 Education” (GK-12) Fellowship, award number DGE0742503. We would like to thank the additional funders of the ASSISTments Platform found here: <http://www.webcitation.org/5ym157Yfr>. We would also like to thank Geoff Gordon, Ken Koedinger, John Stamper and the other organizers of the Pittsburgh Science of Learning Center EDM summer program.

References

1. Pittsburgh Science for Learning Center, "LearnLab", http://www.learnlab.org/research/wiki/index.php/Knowledge_component (accessed November 15, 2011)
2. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
3. University of Massachusetts and Carnegie Mellon University, "The Lemur Project", <http://www.lemurproject.org>
4. Rosé, C., Donmez, P., Gweon, G., Knight, A., Junker, B.: Automatic and Semi-Automatic Skill Coding with a View Towards Supporting On-Line Assessment. In: Proceedings of the 2005 Conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology, Amsterdam (2005)
5. Gertner, A.S., VanLehn, K.: Andes: A Coached Problem Solving Environment for Physics. In: Gauthier, G., VanLehn, K., Frasson, C. (eds.) ITS 2000. LNCS, vol. 1839, pp. 133–142. Springer, Heidelberg (2000)
6. Koedinger, K.R., Anderson, J.R., Hadley, W.H., Mark, M.A.: Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education* 8(1), 30–43 (1997)
7. Razzaq, L., Patvarczki, J., Almeida, S.F., Vartak, M., Feng, M., Heffernan, N.T., Koedinger, K.R.: The ASSISTment builder: Supporting the Life-cycle of ITS Content Creation. *IEEE Transactions on Learning Technologies Special Issue on Real-World Applications of Intelligent Tutoring Systems* 2(2), 157–166 (2009)
8. Birenbaum, M., Kelly, A.E., Tatsuoaka, K.K.: Diagnosing knowledge states in algebra using the rule-space model. *Journal for Research in Mathematics Education* 24(5), 442–459 (1993)
9. Cen, H., Koedinger, K.R., Junker, B.: Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 164–175. Springer, Heidelberg (2006)
10. Desmarais, M.C.: Conditions for effectively deriving a Q-Matrix from data with Non-negative Matrix Factorization. In: Proceedings of Educational Data Mining 2011, Eindhoven, Netherlands (2011)
11. Razzaq, L., Heffernan, N.T., Feng, M., Pardos, Z.A.: Developing Fine-Grained Transfer Models in the ASSISTment System. *Journal of Technology, Instruction, Cognition, and Learning* 5(3), 289–304 (2007)
12. Artificial Intelligence Laboratory - Institute Jozef Stefan, "Artificial Intelligence Laboratory", <http://ailab.ijs.si/tools/text-garden/> (accessed November 15, 2011)
13. Cetinas, S., Si, L.S., Ping, Y.X., Zhang, D., Young, J.P.: Automatic Text Categorization of Mathematical Word Problems. In: Proceedings of the Twenty-Second International FLAIRS Conference, Florida (2009)

Automating Next-Step Hints Generation Using ASTUS

Luc Paquette, Jean-François Lebeau, Gabriel Beaulieu, and André Mayers

Université de Sherbrooke, Québec, Canada
{Luc.Paquette, Andre.Mayers}@USherbrooke.ca

Abstract. ASTUS is an authoring framework designed to create model-tracing tutors with similar efforts to those needed to create Cognitive Tutors. Its knowledge representation system was designed to model the teacher's point of view of the task and to be manipulated by task independent processes such as the automatic generation of sophisticated pedagogical feedback. The first type of feedback we automated is instructions provided as next step hints. Whereas next step hints are classically authored by teachers and integrated in the model of the task, our framework automatically generates them from task independent templates. In this paper, we explain, using examples taken from a floating-point number conversion tutor, how our knowledge representation approach facilitates the generation of next-step hints. We then present experiments, conducted to validate our approach, showing that generated hints can be as efficient and appreciated as teacher authored ones.

Keywords: Hint generation, knowledge representation, model-tracing tutors.

1 Introduction

The “intelligence” of intelligent tutoring systems (ITS) results from their ability to offer relevant pedagogical feedback tailored to the learner's needs. In order to achieve this objective, most systems offer different services [1] such as:

- An expert module that analyzes the task's model to assess the learner's progression towards a solution.
- A learner model that assesses the learner's mastery of the task's knowledge.
- A pedagogical module that provides relevant feedback.

Ideally, in order to reduce the development costs, those modules would be independent from the task. In this context, the creation of a tutor would only require modeling the knowledge relevant to the task and implementing the learning environment's graphical user interface (GUI). Unfortunately, it is difficult for a tutor to provide sophisticated feedback using only the task's model. For this reason, the pedagogical module usually relies on domain specific content integrated to the model. We designed the ASTUS framework and its knowledge representation approach as a step towards solving this problem for model-tracing tutors (MTTs) [2].

Our objective is to offer a framework [3] that can take advantage of the content of the task's model in order to generate different types of sophisticated pedagogical

feedback [4]. This approach is inspired by Ohlsson's learning mechanisms theory [5]. According to this theory, learning can be achieved using nine different mechanisms. Each of these mechanisms can be more or less effective according to the learning context and can be activated by different types of learning activities and feedback. Tutors, such as those created using ASTUS, would greatly benefit from being able to generate feedback targeting a maximum number of those mechanisms.

In order to apply Ohlsson's [5] theory to our framework, we first focused our efforts on a mechanism classically used by MTTs: instructions provided as next-step hints. Whereas most MTT provide next-step hints [6, 7], they are usually authored by a teacher and integrated to the knowledge units contained in the task's model. Barnes and Stamper [8] worked on associating teacher authored hints to automatically generated task models, but few efforts have been made to automate the generation of the hints themselves. The automation of this feedback would contribute to the reduction of the efforts required to author MTTs.

The work presented in this paper describe how, using the ASTUS framework's knowledge representation system, we are able to automatically generate next-step hints. We describe, and illustrate using examples, the processes of generating hints and we present the results of experiments conducted in order to validate our approach.

2 ASTUS

ASTUS is an authoring framework for the creation of MTTs similar to the Cognitive Tutors [9]. One of its main differences is the use of a novel knowledge representation system instead of the more traditional production rule based ones. This system was designed to facilitate the manipulation of the task model by task independent processes such as the automatic generation of pedagogical feedback.

Rather than modeling the cognitive processes used by learners to execute a task, ASTUS's knowledge representation models the teachers' point of view of the task. The format used to model the task is designed to make the content of each knowledge unit explicit. This property allows the manipulation of the model by the framework and is crucial for the generation of feedback such as next-step hints.

In this section, we present a summary of the main structures of ASTUS's knowledge representation system [2]. Semantic knowledge is modeled using concepts: task specific abstractions that are pedagogically relevant. Each concept defines a set of essential features that can refer to other concepts or primitive values (integer, decimal number, symbol, boolean).

Procedural knowledge is modeled using goals and procedures that together form a procedural graph. Figure 1 (left) shows part of the procedural graph in the case of our floating-point tutor. Goals are shown as rectangles and procedures as ovals.

Goals can be achieved by the execution of a procedure (primitive or complex). Primitive procedures model skills that are considered already mastered by the learners. They are reified as atomic interactions in the learning environment's GUI. Complex procedures specify sets of sub goals the learner has to achieve. Those sub goals are arranged according to dynamic plans specific to the procedure's type (a

sequence, a selection or iteration). Both procedures and goals can specify variables (parameters) used to refine their behavior.

During the tutor’s execution, goals and procedures are instantiated in order to produce an episodic tree (right of figure 1). This tree contains all of the completed (C) or currently executing (E) goals and procedures as well as goals that will be expanded in the future (W). The episodic tree is used to match the learner’s steps and indicate whether they are valid or not. This is achieved by using the complex procedures’ scripts to expand the tree up to each of the possible next-steps.

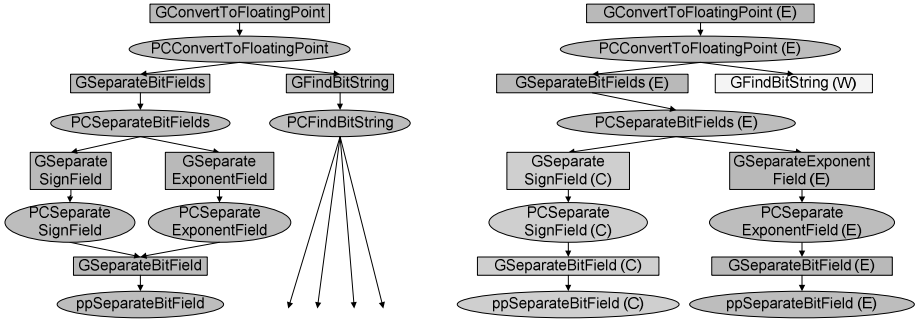


Fig. 1. Examples of part of the procedural graph (left) and its instantiation as an episodic tree (right) for our floating-point number conversion tutor

3 Hint Generation

Since the task’s model is defined using structures that the framework can manipulate, it is possible to automatically generate pedagogical feedback such as next-step hints. To achieve this, the framework mainly benefits from the information contained in the procedural graph and the episodic tree.

We distinguish two main features for hints: their structure (independent from the task) and their content (specific to the task). We defined the structures of our hints as text templates to be filled using task specific content extracted from the knowledge units defined in the task’s model.

We illustrate the process of next-step hint generation using a conditional procedure from our floating-point conversion tutor. More precisely, this example is taken from the sub-task of converting a decimal number to a binary format. The following procedure is used while converting the integer part of a decimal number:

```
Conditional procedure 'PCDivideInt' achieves 'GDivideInt' {
  if 'current_line' instanceof 'FirstLine'
    goal 'GDivideInitialInt' with 'int', 'current_line'
  if not ('current_line' instanceof 'FirstLine')
    goal 'GDividePrevQuotient' with 'current_line'
}
```

The definition of this procedure contains information that can be used by the framework in order to generate next-step hints. The header contains the procedure's identifier (*PCDivideInt*) and the identifier of the goal it achieves (*GDivideInt*). The body of the procedure specifies two sub goals that are available to the procedure (*GDivideInitialInt* and *GDividePrevQuotient*) and the parameters that will be used to instantiate them. The body also specifies two conditions (one for each sub goals). In addition, the procedure's type (conditional) specifies how it will be executed: the conditions will be evaluated and the sub goal associated with the first condition evaluated as true will be instantiated.

Once we have determined the content available for the generation of hints, we can choose the structure of the desired hint. For instance, the definition of a conditional procedure could be used to generate a "pointing-hint":

You need to [parent goal name].

This message can be instantiated using the information contained in the parent goal to become:

You need to *divide the integer*.

The text used in the message comes from the name associated to the *GDivideInt* goal:

```
Goal 'GDivideInt' eng-name 'divide the integer' {
  parameter 'int' type 'Integer' eng-name 'integer'
  parameter 'current_line' type 'IntLine' eng-name 'current line'
}
```

In fact, all of the domain specific text used to generate hint messages comes from the name associated to the knowledge units. This approach requires less effort than asking a teacher to write each hint, especially if multiple hints are associated to the same knowledge unit or the hints have to be translated in multiple languages.

The first "pointing-hint" message is abstract and does not provide much help regarding how to execute the procedure. In fact, this template could be used for any type of procedure. Producing more helpful messages requires more specific content. We can examine how a conditional procedure is executed (select the appropriate sub goal) and combine this information with the knowledge of the available sub goals to produce the following hint:

In order to [parent goal name], you must either [sub goal name] or [sub goal name].

Which would be instantiated as:

In order to *divide the integer*, you must either *divide the initial integer* or *divide the previous quotient*.

While this hint is more explicit regarding how to execute the procedure than the “pointing-hint”, it could still be made more specific. In order to solve this problem, the tutor can refer to the conditions associated to each of the sub goals.

The conditions are explicitly defined using a combination of logical expressions (and, or, not, exists, isInstance, equals). This information can be used to generate hints by starting from a condition’s root expression and generating hints for each of its sub expressions. During this process, the “not” expression can be used as a modifier for a “positive” attribute that impacts the templates used to generate each expression’s hint. Table 1 presents the different templates we used for each expression types.

Table 1. Templates associated to the conditions’ expressions. The bracketed text indicates a sub template and the parenthesis indicates whether a sub expression is positive (T) or not (F).

Positive	not	and	or	exists	isInstance	equals
True		$[expr_1(T)]$	$[expr_1(T)]$	a	[var] is a	[var1]
	$[expr(F)]$	and	or	[concept]	[concept]	equals
False		$[expr_2(T)]$	$[expr_2(T)]$	exists		[var2]
		$[expr_1(F)]$	$[expr_1(F)]$	no	[var] is	[var1] does
	$[expr(T)]$	or	and	[concept]	not a	not equal
		$[expr_2(F)]$	$[expr_2(F)]$	exists	[concept]	[var2]

Using those templates, the previous hint can be modified to provide additional instruction regarding when to apply each of the procedure’s sub goals. The condition expressions described in the “PCDivideInt” procedure (defined previously) can be used to generate messages that are integrated to the hint:

In order to divide the integer, you must either divide the initial integer, if current line is a first line, or divide the previous quotient, if current line is not a first line.

This hint takes advantage of all of the information contained in the procedure’s definition, but can still be modified by using the information contained in the episodic tree regarding the current state of the problem being solved. This can be used to reduce the size of the hint and to focus the learner’s attention on the correct sub goal:

In order to [parent goal name], you must [active sub goal name] since [active condition].

Which would be instantiated as:

In order to divide the integer, you must divide the initial integer since current line is a first line.

This last template is the one currently used by our framework, but this decision is specific to how we decided to provide next-step hints. Any combination of one or more templates (those given as examples or new templates using the available information) can be used to provide next-step hints.

The examples given in this section show how the information contained in the definition of knowledge units can be used to generate next-step hints. Those hints can be customized according to the desired pedagogical strategy: they can provide different amounts of instruction and they can be contextualized using the current state of the learning environment. The current implementation uses text templates in order to generate the hints, but could be improved by using natural language techniques. For instance, in our previous examples, the condition expression “*current line* is a *first line*” could be rewritten as “the current line is the first line”. Such small modifications would greatly improve the readability of the generated hints.

In this paper, we only described how next-step hints can be generated for conditional procedures. A similar process has been applied to every type of procedural knowledge units. Among them are inferences, expressions that model mental skills applied to fill in the parameters of goals and procedures. They can be used to further contextualize next-step hints by specifying how a parameter is deduced from known ones, recalled from memory or perceived in the learning environment’s GUI.

4 Experiments

In order to validate our hint generation approach, we conducted multiple experiments during a computer science course at the University of Sherbrooke. We used a floating-point number conversion tutor designed using ASTUS. The objective of our first experiment was to evaluate the learning gains and the students’ appreciation of next-step hints generated by our framework. This first experiment is detailed in [10], but we present here a summary of its methodology and the analysis of its results. In this first study, 34 students were separated in two groups: 19 students received teacher authored hints (TH) and 15 received framework generated hints (FH). Statistical analyses of the results did not show a significant difference in learning gain when comparing pretest and posttest scores (left of figure 2) for both conditions and showed that framework generated hints can be as appreciated as equivalent teacher authored hints (right of figure 2) for different types of complex procedures: while iteration, conditional, sequence with N sub goals and sequence with 1 sub goal.

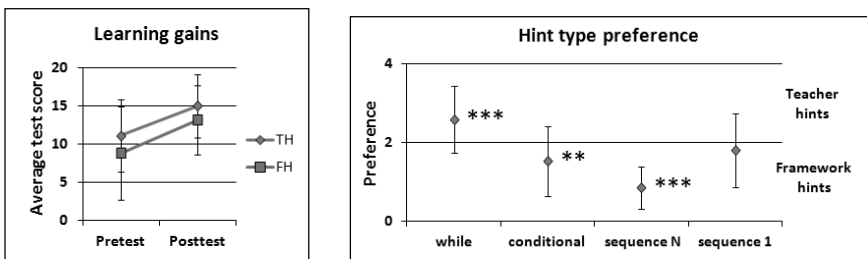


Fig. 2. Graphs illustrating the results of our first experiment. The ‘*’ character indicates the statistical significance (** for $p < 0.01$ and *** for $p < 0.001$).

Table 2. Summary of the statistical analysis for our second experiment

	Stat	<i>p</i>	Effect size	Power
Pretest scores	$t(23.629) = 0.576$	0.570	$d = 0.20$	8.50%
Learning gain (NH)	$t(15) = 6.213$	< 0.001***	$d = 1.35$	99.89%
Learning gain (WH)	$t(15) = 5.550$	< 0.001***	$d = 1.37$	99.91%
ANCOVA	$F(1, 29) = 3.057$	0.046*	$\eta^2_p = 0.091$	39.40%

While our first experiment did not find any significant difference between framework generated and teacher authored hints in the context of our floating-point tutor, it does not validate that the learning gains can be attributed to the received hints. Indeed, the observed gains could simply be caused by the activity of solving problems using a tutor. In order to determine if next-step hints were helpful while solving problems with our tutor, we conducted a second experiment comparing the learning gains of a tutor without next-step hints (only flag feedback) to those of a tutor also providing framework generated next-step hints. A group of 32 students was separated, at random, in two sub-groups: 16 students used a tutor that did not provide next-step hints (NH) and 16 students used one that provided framework generated hints (WH). The students were first asked to complete a pretest (20 minutes), then use the tutor (40 minutes) and finally complete a posttest (20 minutes). There were two versions of the test (graded on a total of 20). Half the students received the first version as pretest and the second as posttest while the order was reversed for the other half. Table 2 summarizes the results of our analysis.

A two-sample t-test showed no statistically significant differences between the participants' pretest scores for the NH ($M = 8.13$; $SD = 2.34$) and the WH ($M = 8.82$; $SD = 4.16$) conditions. Although no significant differences were found, the standard deviation of the WH condition is much higher than the one for the NH condition.

The learning gains between the pretests and posttests were validated using paired t-tests. Both conditions showed significant gains. The NH condition's pretest ($M = 8.13$; $SD = 2.34$) and posttest ($M = 12.09$; $SD = 3.30$) scores indicate a large effect size, and so do the WH condition's pretest ($M = 8.81$; $SD = 4.16$) and posttest ($M = 14.44$; $SD = 4.06$) scores. The effect sizes are very similar even though the mean learning gain is higher for the WH (5.63) condition when compared to the NH (3.96) condition. This lack of difference results from the difference in standard deviations between the two conditions. The effect size for the WH condition would have been higher if its standard deviations were closer to those of the NH condition.

A one-tailed analysis of covariance (ANCOVA), with the pretest scores as the covariate, showed a significant differences between the posttest scores for the NH ($M_{aj} = 12.31$) and WH ($M_{aj} = 14.22$) conditions. This suggests that the use of next-step hints during problem solving allows learners to achieve higher learning gains. In order to further validate this result, we conducted a third experiment using the same methodology as the second one. In this experiment there were 16 learners for the NH condition and 17 for the WH condition. Its results are summarized in table 3.

Table 3. Summary of the statistical analysis for our third experiment

	Stat	<i>p</i>	Effect size	Power
Pretest scores	$t(31) = 0.318$	0.753	$d = 0.11$	61.00%
Learning gain (NH)	$t(15) = 2.970$	0.010**	$d = 0.52$	49.46%
Learning gain (WH)	$t(16) = 4.401$	< 0.001***	$d = 0.77$	86.64%
ANCOVA	$F(1, 30) = 2.818$	0.052	$\eta^2_p = 0.086$	36.40%

A two-sample t-test showed no statistically significant differences between the learners’ pretest scores for the NH ($M = 10.97$; $SD = 4.28$) and the WH ($M = 11.50$; $SD = 5.24$) conditions.

The learning gains between the pretests and posttests were validated using paired t-tests. Both conditions showed significant gains. The NH condition’s pretest ($M = 10.97$; $SD = 4.28$) and posttest ($M = 13.19$; $SD = 4.25$) scores indicate a medium effect size, and the WH condition’s pretest ($M = 11.50$; $SD = 5.24$) and posttest ($M = 15.26$; $SD = 4.43$) scores indicate a large effect size. The higher effect size for the WH condition suggests it yielded higher learning gains than the NH condition.

A one-tailed analysis of covariance (ANCOVA), with the pretest scores as the covariate, showed no significant differences between the posttest scores for the NH ($M_{aj} = 13.83$) and WH ($M_{aj} = 15.26$) conditions. The results of the test were very close to a statistically significant difference ($p = 0.052$). This, combined with the differences in effect size for the paired t-tests and the higher adjusted mean score for the WH, suggests that the WH condition yielded higher learning gains.

While neither our second nor our third experiments yielded strong statistical results, the results of both suggest that the WH condition leads to higher learning gains. Figure 3 shows the results of those two experiments in graphic form. In both, the steeper slopes for the WH conditions illustrate how the students in the WH conditions improved their posttest results by a greater amount than those in the NH condition. Those graphs can be compared to the equivalent graph for our first experiment (left of figure 2) for which the two slopes (FH and TH) are very similar, which is consistent with the absence of a significant difference.

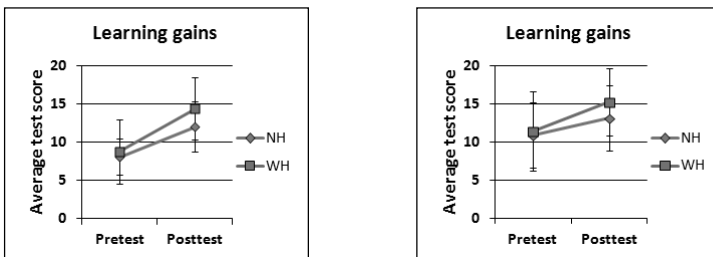


Fig. 3. Graphs illustrating the results of our second (left) and third (right) experiments

5 Discussion

The results of our most recent experiments show how framework generated next-step hints yielded higher learning gains compared to the use of a tutor offering only flag feedback. This shows that the floating-point number conversion task is complex enough for the use of next-step hints to improve learning gains, but it does not evaluate the efficiency of framework generated hints. A previous experiment [10] showed that framework generated hints can be as efficient and as appreciated as teacher authored ones in the context of our floating-point tutor.

Additional experiments could be used to improve our empirical validation by reproducing similar results for different tasks. Such results would suggest that the efficiency of framework generated hints can be generalized to multiple types of task of different complexity. These experiments would also benefit from bigger groups of learner to increase the statistical power of their result. Additionally, they would benefit from learners with no background in computer science. In our experiments, the learners were all computer science students that are well suited to understanding computer generated messages. Reproducing similar results with learners from more varied backgrounds would support our hypothesis that efficient hints can be automatically generated by a framework regardless of the task taught.

Our experiments have shown that our framework has access to the information required to generate efficient next-step hints. It would be interesting to research how the hints' efficiency can be improved by modifying how they present this information. The use of natural language techniques might impact the hints' efficiency by improving their readability, thus fostering better communication between the learner and the tutor. Their efficiency might also be improved by the use of learning theories optimizing the content and the format of the hints provided to the learners.

The use of generated next-step hints is useful in order to reduce the authoring efforts required to create a tutor by only having to associate readable names to knowledge units instead of complete message templates. They could also be used to customize the hints to groups of learners, specific learners within that group or even specific learning situation. For example, hints could be generated in different languages, they could be made culturally aware [11] and they could consider the learner's current emotional state [12]. The content of the hints would remain the same but their presentation would vary according to those parameters. Although it would be possible for a teacher to author multiple versions of every hint to account for those parameters, having the framework generate the hints would require much less efforts.

In addition to reducing the efforts of authoring hints, being able to generate them can be essential for situations where it is not possible to enumerate all the possible hints. An example of such a situation is negative feedback on errors. In order to provide such feedback, model-tracing frameworks usually require the tutor's author to model erroneous procedural knowledge. This process requires a lot of efforts due to the very high number of different errors. In order to reduce the required efforts, we are currently working on a model, based on Sierra's theory of procedural error [13], to allow our framework to diagnose as many of those errors as possible without modeling additional erroneous knowledge [14]. Since the errors are automatically

diagnosed by the framework while a learner solves a problem, they are not explicitly defined in the task and it is not possible to enumerate all the required hints. It is thus essential for the framework to be able to generate efficient hints in order to provide feedback regarding the diagnosed errors.

The example of providing negative feedback on errors illustrates how being able to generate next-step hints is a first step toward achieving our objective of developing a framework able to provide feedback for many of Ohlsson's learning mechanisms. We started by automating the generation of next-step hints feedback for instruction, but our work will also be extended to support other mechanisms such as negative feedback on error, a type of feedback usually provided by constraint-based tutor [15].

6 Conclusion

In this paper, we explained how the ASTUS framework generates next-step hints using domain independent knowledge structures. We presented experiments showing that these hints can be as effective and as appreciated as teacher-authored hints in the context of our floating-point number conversion tutor.

Future work will focus on expanding the number of different types of feedback the framework can generate in order to take advantage of as many of Ohlsson's learning mechanisms [5] as possible. Our hypothesis is that the same characteristics that allow the generation of next-step hints will be helpful when generating other types of feedback. Our next objective is to diagnose and offer negative feedback regarding the learners' errors without requiring the modeling of knowledge marked as erroneous.

Acknowledgements. We would like to thank Richard St-Denis for allowing the use of our tutor in his class, Mikaël Fortin for authoring the "teacher authored hints" used during our experiment and François Bureau du Colombier and Jean Pierre Mbungira for their help with the implementation of our tutor.

References

1. Wenger, E.: *Artificial Intelligence in Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Morgan Kaufmann Publishers Inc. (1987)
2. Paquette, L., Lebeau, J.-F., Mayers, A.: *Authoring Problem-Solving Tutors: A Comparison between ASTUS and CTAT*. In: Nkambou, R., Bourdeau, J., Mizoguchi, R. (eds.) *Advances in Intelligent Tutoring Systems*. SCI, vol. 308, pp. 377–405. Springer, Heidelberg (2010)
3. Lebeau, J.-F., Paquette, L., Fortin, M., Mayers, A.: *An Authoring Language as a Key to Usability in a Problem-Solving ITS Framework*. In: Alevén, V., Kay, J., Mostow, J. (eds.) *ITS 2010, Part II*. LNCS, vol. 6095, pp. 236–238. Springer, Heidelberg (2010)
4. Paquette, L., Lebeau, J.-F., Mayers, A.: *Integrating Sophisticated Domain-Independent Pedagogical Behaviors in an ITS Framework*. In: Alevén, V., Kay, J., Mostow, J. (eds.) *ITS 2010, Part II*. LNCS, vol. 6095, pp. 248–250. Springer, Heidelberg (2010)

5. Ohlsson, S.: *Deep Learning: How the Mind Overrides Experience*. Cambridge University Press, New York (2011)
6. VanLehn, K.: The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education* 16(3), 227–265 (2006)
7. Alevin, V.: Rule-Based Cognitive Modeling for Intelligent Tutoring Systems. In: Nkambou, R., Bourdeau, J., Mizoguchi, R. (eds.) *Advances in Intelligent Tutoring Systems*. SCI, vol. 308, pp. 33–62. Springer, Heidelberg (2010)
8. Barnes, T., Stamper, J.: Toward Automatic Hint Generation for Logic Proof Tutoring Using Historical Student Data. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) *ITS 2008*. LNCS, vol. 5091, pp. 373–382. Springer, Heidelberg (2008)
9. Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R.: Cognitive Tutors: Lessons Learned. *The Journal of the Learning Sciences* 4(2), 167–207 (1995)
10. Paquette, L., Lebeau, J.-F., Mbungira, J.P., Mayers, A.: Generating Task-Specific Next-Step Hints Using Domain-Independent Structures. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011*. LNCS, vol. 6738, pp. 525–527. Springer, Heidelberg (2011)
11. Blanchard, E.G., Mizoguchi, R.: Designing Culturally-Aware Tutoring Systems: Toward an Upper Ontology of culture. In: Blanchard, E., Allard, D. (eds.) *Culturally Aware Tutoring Systems, CATS 2008*, pp. 23–24 (2008)
12. Woolf, B., Burleson, W., Arroyo, I., Dragon, T., Cooper, D., Picard, R.: Affect-Aware Tutors: Recognising and Responding to Student Affect. *Journal of Learning Technology* 4, 129–163 (2009)
13. VanLehn, K.: *Mind Bugs: The Origin of Procedural Misconceptions*. MIT Press (1990)
14. Paquette, L., Lebeau, J.F., Mayers, A.: Modeling Learner’s Erroneous Behaviours in Model Tracing Tutors. In: *Proceedings of UMAP 2012* (accepted, 2012)
15. Mitrovic, A.: Modeling Domains and Students with Constraint-Based Modeling. In: Nkambou, R., Bourdeau, J., Mizoguchi, R. (eds.) *Advances in Intelligent Tutoring Systems*. SCI, vol. 308, pp. 63–80. Springer, Heidelberg (2010)

The Effectiveness of Pedagogical Agents' Prompting and Feedback in Facilitating Co-adapted Learning with MetaTutor

Roger Azevedo¹, Ronald S. Landis², Reza Feyzi-Behnagh¹, Melissa Duffy¹, Gregory Trevors¹, Jason M. Harley¹, François Bouchet¹, Jonathan Burlison³, Michelle Taub¹, Nicole Pacampara¹, Mohamed Yeasin⁴, A.K.M. Mahbubur Rahman⁴, M. Iftekhhar Tanveer⁴, and Gahangir Hossain⁴

¹ McGill University, Dept. of Educational and Counselling Psychology, Montreal, Canada
roger.azevedo@mcgill.ca

² Illinois Institute of Technology, College of Psychology, Chicago, IL, USA
rlandis@iit.edu

³ University of Memphis, Dept. of Psychology, Memphis, TN, USA
jdburlisn@yahoo.com

⁴ University of Memphis, Dept. of Electrical and Computer Engineering, Memphis, TN, USA
myeasin@memphis.edu

Abstract. Co-adapted learning involves complex, dynamically unfolding interactions between human and artificial pedagogical agents (PAs) during learning with intelligent systems. In general, these interactions lead to effective learning when (1) learners correctly monitor and regulate their cognitive and metacognitive processes in response to internal (e.g., accurate metacognitive judgments followed by the selection of effective learning strategies) and external (e.g., response to agents' prompting and feedback) conditions, and (2) pedagogical agents can adequately and correctly detect, track, model, and foster learners' self-regulatory processes. In this study, we tested the effectiveness of PAs' prompting and feedback on learners' self-regulated learning about the human circulatory system with MetaTutor, an adaptive, multi-agent learning environment. Sixty-nine ($N=69$) undergraduates learned about the topic with MetaTutor, during a 2-hour session under one of three conditions: prompt and feedback (PF), prompt-only (PO), and no prompt (NP) condition. The PF condition received timely prompts from several pedagogical agents to deploy various SRL processes and received immediate directive feedback concerning the deployment of the processes. The PO condition received the same timely prompts, without feedback. Finally, the NP condition learned without assistance from the agents. Results indicate that those in the PF condition had significantly higher learning efficiency scores than those in both the PO and control conditions. In addition, log-file data provided evidence of the effectiveness of the PA's timely scaffolding and feedback in facilitating learners' (in the PF condition) metacognitive monitoring and regulation during learning.

Keywords: self-regulated learning, metacognition, pedagogical agents, co-adaptation, multi-agent systems, learning, product data, process data.

1 Objectives and Theoretical Framework

When learning about complex science topics such as the human circulatory system, research indicates that individuals can gain deep conceptual understanding through effective use of self-regulated learning (SRL). The successful use of cognitive and metacognitive SRL processes involves setting meaningful goals for one's learning, planning a course of action for attaining these goals, deploying a diverse set of effective learning strategies in pursuit of the goals, continuously monitoring one's own understanding of the material and the appropriateness of the current information, and making adaptations to one's goals, strategies, and navigational patterns based on the results of such monitoring processes and resulting judgments [1,2,3,4]. Although learners should attempt to follow these guidelines when attempting difficult topics, exploration of typical learning has demonstrated that few learners, in fact, engage in effective self-regulated learning. Although motivation and affect play a role in determining learners' willingness to self-regulate, we assume a lack of self-regulatory skills is the main obstacle to adequate regulation and, subsequently, deficient learning gains and conceptual understanding [5,6]. Therefore, the current research makes use of pedagogical agents (PAs) to assist learners during interactions with MetaTutor, a multi-agent adaptive hypermedia learning environment that models, scaffolds, and fosters learners' use of cognitive and metacognitive SRL processes during learning about the human circulatory system.

Learners attempting to self-regulate often face limitations in their own metacognitive skills, which, when compounded with lack of domain knowledge, can result in cognitive overload in open-ended learning environments [7,8,9]. One method of relieving the cognitive burden placed on learners in this situation is to provide assistance in the form of adaptive scaffolding. Previous experiments conducted by Azevedo and colleagues [e.g., 10,11] established that adaptive scaffolding provided by a human tutor leads to greater deployment of sophisticated planning processes, metacognitive monitoring processes, and learning strategies as well as larger shifts in mental models of the domain. The purpose of the current work is to determine if adaptive scaffolding provided by PAs within an adaptive, intelligent hypermedia learning environment is also capable of producing the same, or better, learning outcomes and increased use of effective SRL processes.

The current experiment used a mixed-methodology design that combined product and process data to examine the effect of various types of SRL prompting and scaffolding delivered by PAs in an adaptive intelligent hypermedia learning environment. Three learning conditions were used to determine the efficacy of scaffolding SRL through pedagogical agents: 1) prompting with feedback condition (PF), 2) prompting only condition (PO), and 3) no prompting condition (NP). Participants were randomly assigned to one of the three conditions and asked to learn about the human circulatory system using MetaTutor during a two-session experiment. This experiment included the collection of concurrent think-aloud protocols, eye-tracking data, human-agent dialogue, learning outcome measures, log-file data, metacognitive judgments during learning, embedded quizzes, and facial recognition data for affect classification. Due to the complexity of the data analyses, we only report the learning outcomes (i.e.,

learning efficiency) and a few of the log-file variables that are indicative of learners' use of SRL processes.

2 Method

2.1 Participants

Participants were 69 undergraduate students (75% females) from a large public university in North America. The mean age of the participants was 23 and their mean GPA was 2.84. All participants were paid \$10 per hour, up to \$40 for completion of the 2-day, 4-hour experiment.

2.2 Materials and MetaTutor

Materials consisted of several computerized elements. The pretest and posttest each included 25 multiple-choice items each with four foils. Items on the pretest and posttest included text-based items (which could be answered by directly referring one sentence within the content) and inferential items (which required integrating information from at least two sentences within the content). Two equivalent forms of the test were created using a total of 50 items and the forms used for pretest and posttest were counterbalanced across participants.

The learning environment used by all participants, MetaTutor, is an adaptive hypermedia learning environment including 41 pages of text and static diagrams, organized by a table of contents displayed in the left pane of the environment (see Figure 1). The version of MetaTutor used in this experiment includes material related to the human circulatory system. Along with the table of contents, the environment includes a timer indicating time remaining, an SRL palette which learners may use to instantiate an interaction with the pedagogical agent (e.g., indicate that they want to take notes), and an overall learning goal (which was the same for all participants) and sub-goals (which were created by all participants at the beginning of the learning session with the assistance of one of the PAs). Additionally, four distinct pedagogical agents (Gavin, Pam, Mary, and Sam) are displayed in the upper right-hand corner of the environment, which provide varying degrees of prompting and feedback throughout the learning session designed to scaffold students' SRL skills and content understanding.

2.3 Instructional Conditions

We designed and tested three versions of the MetaTutor environment. In the Prompt and Feedback (PF) version, participants were prompted by PAs to use specific self-regulatory processes (e.g., metacognitively monitor their emerging understanding of the topic), and given immediate feedback about their use of those processes. In the Prompt only (PO) version, participants received the same prompts as the ones provided to those in the PF version. However, the agents in the PO version did not

provide feedback. The timing of the prompts used in both the PF version and the PO version was adaptive to the individual learner and was determined using various factors of learner interaction, including time on page, time on current sub-goal, number of pages visited, relevancy of the current page for the current sub-goal, etc. In the No Prompt (NP) version, participants did not receive prompts or feedback. All three versions (PF, PO, NP) provided an SRL palette, which allowed participants to self-select any SRL processes they wanted to use during the learning session.

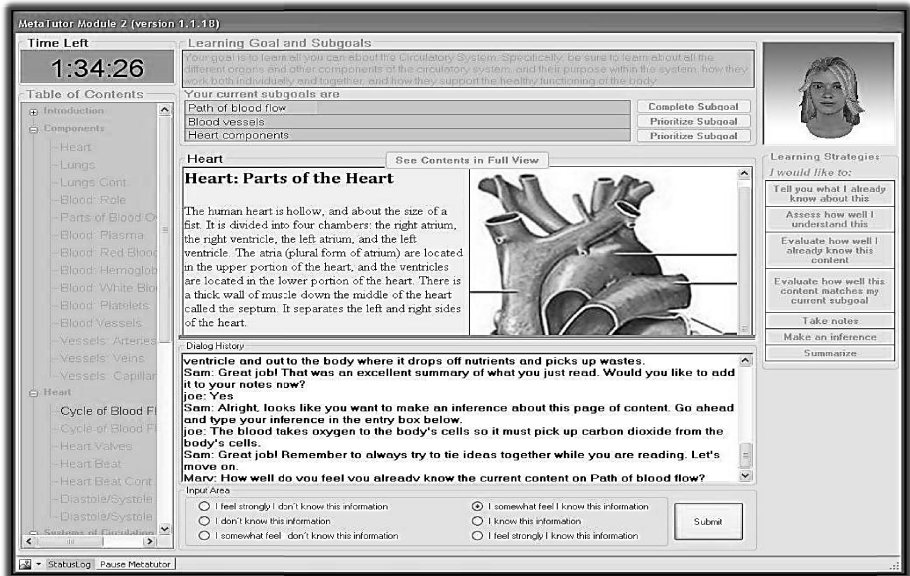


Fig. 1. Screenshot of the MetaTutor Interface

2.4 Experimental Procedure

On day one of the experiment, participants completed a demographics questionnaire and the pretest on the human circulatory system. Learners were given up to 20 minutes to complete the pretest. On day two, participants engaged in the learning session and completed the posttest on the human circulatory system. Before beginning the learning session, the Tobii T60 eye-tracker was calibrated to each participant individually. All participants were then instructed in the think-aloud procedure and shown a short video demonstrating thinking aloud. Next, each participant was shown another short video explaining and demonstrating the various functionalities of MetaTutor and providing the learners with their overall learning goal (see Figure 1). This introductory video also demonstrated the use of an electronic note-taking feature within the environment and instructed the participants to use the peripheral drawing pad if and when they chose to draw. Following the introductory videos, the learners were given two hours to learn about the human circulatory system using MetaTutor. All participants were provided the opportunity to take a short break (5 minutes) during the two

hours, although not all chose to do so. During the learning session, participant verbalizations and facial expressions were recorded using a Microsoft Lifecam(TM) within the eye-tracker monitor. Immediately after the learning session, participants were given up to 20 minutes to complete the posttest. Finally, all participants were paid and debriefed before leaving the lab.

3 Results

In this section we present the learning outcomes (expressed as learning efficiency) and a subset of the log-file data.

Learning Time with the Science Content. Learning time was calculated by summing the amount of time spent viewing the instructional content (i.e., text and diagrams). Interactions with the agents, in which the instructional content was not visible, were not included in learning time. One-way analysis of variance (ANOVA) indicated a significant difference between the groups in learning time, $F(2,66) = 40.71, p < .001$. LSD post-hoc analyses indicated that the Control group had a longer total learning time ($M = 87.94, SD = 12.42$) when compared to both the PO condition ($M = 68.31, SD = 11.18$) and the PF condition ($M = 56.84, SD = 11.82$), $p < .001$. Additionally, the PO condition had a significantly longer learning time compared to the PF condition, $p < .01$.

Number of Content Pages Visited. One-way ANOVA also indicated a significant difference between the groups in the mean number of pages visited (out of 41 possible¹) during the learning session, $F(2,66) = 22.17, p < .001$. LSD post-hoc analyses revealed that the Control group visited significantly more pages ($M = 38.87, SD = 3.84$) than both the PO condition ($M = 33.26, SD = 8.39; p < .05$) and the PF condition ($M = 23.56, SD = 10.07; p < .001$). Additionally, the PO condition visited significantly more pages than the PF condition, $p < .001$.

Amount of Time Spent Reading Pages and Inspecting Diagrams. Results indicated that students did not differ significantly in the amount of time spent on each page (see Table 1). On average, students spent between 60 seconds to 90 seconds on each page ($p > .05$). By contrast, one-way ANOVA revealed a statistically non-significant difference between groups in the mean time spent viewing individual diagrams within the environment, $F(2,66) = 3.02, p = .052$. Given the observed level of marginally significant differences, LSD post-hoc analyses were conducted and revealed that mean diagram view time was greater for the PF condition ($M = 1.05 \text{ min}, SD = 0.99$) compared to the Control condition ($M = 0.54 \text{ min}, SD = 0.46$), $p = .016$. The PO condition did not differ significantly from the remaining two conditions ($M = 0.75 \text{ min}, SD = 0.51$).

Number of Sub-Goals Generated during Learning. One-way ANOVA indicated a significant difference between the groups in the number of sub-goals generated during

¹ Subsequent revisits to the same page were not counted in the total.

the learning session, $F(2,66) = 8.74, p < .001$. LSD post-hoc analyses revealed that the PO condition ($M = 4.13, SD = 1.29$) and the Control condition ($M = 4.70, SD = 1.72$) both attempted significantly more sub-goals than the PF condition ($M = 3.04, SD = 0.98$), $p < .01$. There was not a significant difference between the PO condition and the Control condition. One-way ANOVA indicated a significant difference between the groups in the mean time spent on each individual sub-goal during the learning session, $F(2,66) = 10.31, p < .001$. LSD post-hoc analyses revealed that the PF condition ($M = 41.39, SD = 18.62$) spent significantly longer on each sub-goal compared to both the PO condition ($M = 27.77, SD = 9.96$) and the Control condition ($M = 23.30, SD = 12.18$), $p < .01$.

Learning Efficiency². One-way ANOVA on the learning efficiency scores indicated a significant effect of learning condition on learners learning efficiency ($F[2,66] = 6.64, p < .01$). Post-hoc comparisons revealed that the Prompt and Feedback (PF) condition significantly outperformed the No Prompt (NP) condition ($d = 0.84$). Non-significant differences were demonstrated for each of the remaining two comparisons ($p > .05$). See Table 1 for descriptive statistics.

Table 1. Means (and Standard Deviations) for Various Measures by Condition

	NP Condition (No Prompt Condition) M (SD)	PO Condition (Prompt Only) M (SD)	PF Condition (Prompt and Feedback) M (SD)
*Overall Learning Time (with instructional material only) (min.)	87.94 (12.42)	68.31 (11.18)	56.84 (11.82)
*Number of Pages Visited	38.87 (03.84)	33.26 (08.39)	23.56 (10.07)
Overall Mean Time on Page (min.)	1.07 (00.66)	0.99 (00.50)	1.32 (01.06)
Overall Mean Time on Diagrams (min.)	0.54 (00.46)	0.75 (00.51)	1.05 (00.99)
*Number of Sub-Goals Set During Learning Session	4.70 (01.72)	4.13 (0.1.29)	3.04 (00.98)
*Mean Time Spent on Self-Set Sub- Goal (min.)	23.30 (12.18)	27.77 (09.96)	41.39 (18.60)
*Learning Efficiency (%)	23.10 (06.00)	28.90 (10.40)	34.30 (13.60)

Note: * $p < .05$

² Each participant received one point for each correct answer selected on the pretest and posttest. From this value, a *learning efficiency score* was calculated by dividing the raw posttest score by the number of minutes the participant was actually learning (*time on task*). Time on task was defined as the sum of all of the time spent viewing domain-related content (text and/or diagram). During certain periods of the learning session, the learning content was hidden from view due to interactions with the agent. To account for differential *learning time*, the time each participant spent viewing the learning content was factored in to the learning efficiency score (Faw & Waller, 1976; Simons, 1983).

4 Discussion

Current results show that college students' learning about a challenging science topic with hypermedia can be facilitated if they are provided with adaptive prompting and feedback scaffolding designed to regulate their learning. More importantly, we have demonstrated that PAs are effective in facilitating students' SRL processes by providing timely prompting and feedback. Their effectiveness stems from the system's ability to determine optimal times during a learning session (e.g., prompting learners to activate their prior knowledge at the beginning of each generated sub-goal; prompting students to assess whether the current text and diagram are relevant for the current sub-goal). We have demonstrated the effectiveness of prompting and feedback by showing that students in this condition (i.e., PF condition) read less material and navigated through fewer hypermedia pages during the learning task. They also tended to spend more time on each page and spend more time inspecting each diagram presented in MetaTutor. Those in the PF condition also set fewer sub-goals but they spent more time on each sub-goal. Overall, the data support existing theoretical frameworks and models of SRL [e.g., 1,3] related to the use of computers as Meta-Cognitive tools [1,2]. Subsequent analyses of the verbal protocols, metacognitive judgments, emotions data, and log-file data will allow us to extend current models of SRL and build more sophisticated intelligent multi-agent technology-learning environments designed to detect, trace, model, and foster students' SRL.

Our study contributes to an emerging field that merges educational, cognitive, learning, and computational sciences by addressing issues related to learning about complex science topics with multi-agent environments [1,5,6,8,9,12]. Our study also contributes to an emerging body of evidence which illustrates the critical role of SRL in students' learning with hypermedia [1,2,6,8,11], and extends recent research regarding the role of intelligent, adaptive scaffolding in facilitating students' learning with hypermedia [13]. Converging temporally-aligned, multi-level data will allow us to examine the critical role of PAs as external regulatory agents whose scaffolding methods facilitate students' self-regulated learning [1,8,12]. Lastly, both our product and process data can be applied to inform the design of intelligent multi-agent hypermedia environments as Metacognitive tools to foster learners' self-regulated learning of challenging science topics by providing adaptive scaffolding [1,5,6,8,14].

5 Current and Future Directions

In this paper we presented a few product measures to assess the effectiveness of agents' prompting in supporting learners' SRL processes during learning with MetaTutor. We are currently analyzing huge amounts of data collected from several methods (i.e., eye-tracking, log-file, affect classification, concurrent think-alouds, notes and drawings, learner-agents dialogue, metacognitive judgments, on-line summaries, use of SRL palette). In this section, we present several directions we're currently exploring to enhance our understanding of the various conceptual, theoretical,

methodological, and analytical issues related to SRL and the potential of multi-agent learning environments.

Measuring SRL with Multi-agent Learning Environments. Multi-agent technology-based learning environments have become popular educational and research tools [12]. Researchers are using them as *educational tools* to foster learning about complex and challenging topics and domains since embodied pedagogical agents can be programmed to detect, track, model, and foster students' self-regulatory processes, such as planning, metacognitive monitoring, strategy selection and deployment, regulation of affect, motivational beliefs, and reflection [1,9]. In addition, agent-based environments are also being used as *research tools* to measure the deployment of self-regulatory processes by allowing researchers to collect rich, multi-stream data, including self-report measures of self-regulated learning (SRL), on-line measures of cognitive and metacognitive processes, dialogue moves regarding agent-student interactions, natural language processing of help-seeking behavior, physiological measures of motivation and emotions, emerging patterns of effective problem solving behaviors and strategies, traces of inquiry cycles, etc. In addition, collecting various data streams is critical to enhancing our understanding of when, how, and why students regulate or don't regulate their learning and adapt their regulatory behaviors [15,16,17].

Unique Measurement and Data Analytic Challenges. The current experimental protocol provides a rich source of data through multiple, temporally connected channels. Although our reported analyses relied exclusively on comparisons between experimental groups separately for particular process and outcome variables, the nature of our data is substantially more complex. For example, because SRL processes unfold temporally, we ultimately want to map emotional and or cognitive reactions at one point in time to responses within and across channels at later points in time. Such processes will provide a much more comprehensive picture of the learning process and will allow us to not only identify pre-post performance differences, or simple mean differences across groups, but also to model the intraindividual growth trajectories that underlie learning.

Using MetaTutor to Measure Temporal Dynamics of SRL during Complex Learning. We are synthesizing the results, emphasizing issues and insights that relate to the strengths and weaknesses of collecting, coding, analyzing, and interpreting process data [e.g., see 1]. One issue is the importance of the classification of these processes at various *levels of granularity* and *valence*. For example, macro-level (e.g., monitoring process) and micro-level classifications (e.g., monitoring process such as judgment of learning [JOL]) supplemented with valence (i.e., positive or negative [e.g., JOL+]) are key to understanding the multi-level nature of these processes (and inter-related feedback mechanisms) and serve to augment current conceptions and theoretical frameworks of SRL [3]. We are also dealing with the *temporal alignment* of several data streams (e.g., concurrent think-alouds with eye-tracking data), which are key to understanding the unfolding of the processes in real time and providing evidence of behavioral signatures associated with specific SRL processes. For example, some on-line measures need to be *augmented* with other measures and methods in

order to provide converging evidence. The use of log-file data to *generate hypotheses* regarding fundamental assumptions about SRL (e.g., agency, individual agent's adaptations, and co-adaptations between human and artificial agent during learning). We are also exploring ways in which on-line measures can be *converged with other process, product, and self-report data* to provide a comprehensive understanding of SRL measurement during learning with multi-agent learning environments.

Co-Regulated Learning between Human and Artificial Pedagogical Agents in the Context of a Multi-agent Adaptive Hypermedia Environment. Co-adaptation between human and artificial agents is a core issue in the ITS community [see 19]. Contemporary research on multi-agent learning environments has focused on SRL while relatively little effort has been made to use *co-regulated learning* as a guiding theoretical framework. This oversight needs to be addressed given the complex nature that self-and other-regulatory processes play when human learners and artificial pedagogical agents interact to support learners' internalization of SRL processes [see 19]. For example, learning with a multi-agent hypermedia environment such as MetaTutor involves having a learner interact with four artificial pedagogical agents. Each agent plays different roles including modeling, prompting, and scaffolding SRL processes (e.g., planning, monitoring, and strategy use) and providing feedback regarding the appropriateness and accuracy of learners' use of SRL processes. Accordingly, we are dealing with the challenges and opportunities of our methodological and analytical approaches. One challenge involves determining how our (current study and) research can be re-conceptualized within the framework of co-regulated learning. By doing so, we will extend the human and computerized theoretical models typically used in this research area.

Acknowledgements. The research presented in this paper has been supported by funding from the National Science Foundation (DRL 0633918 and IIS 0841835) awarded to the first author and (DRL 1008282) awarded to the second author.

References

1. Azevedo, R., Moos, D., Johnson, A., Chauncey, A.: Measuring cognitive and metacognitive regulatory processes used during hypermedia learning: Issues and challenges. *Educational Psychologist* 45, 210–223 (2010)
2. Azevedo, R., Witherspoon, A., Chauncey, A., Burkett, C., Fike, A.: MetaTutor: A Meta-Cognitive tool for enhancing self-regulated learning. In: Pirrone, R., Azevedo, R., Biswas, G. (eds.) *Proceedings of the AAAI Fall Symposium on Cognitive and Metacognitive Educational Systems*, pp. 14–19. Association for the Advancement of Artificial Intelligence (AAAI) Press, Menlo Park (2009)
3. Winne, P.H., Nesbit, J.C.: Supporting self-regulated learning with cognitive tools. In: Hacker, D.J., Dunlosky, J., Graesser, A.C. (eds.) *Handbook of Metacognition in Education*. Erlbaum, Mahwah (2009)
4. Zimmerman, B., Schunk, D.: *Handbook of self-regulation of learning and performance*, pp. 102–121. Routledge, New York (2011)

5. Schwartz, D.L., Chase, C., Chin, D.B., Oppezzo, M., Kwong, H., Okita, S., et al.: Interactive metacognition: Monitoring and regulating a Teachable Agent. In: Hacker, D.J., Dunlosky, J., Graesser, A.C. (eds.) *Handbook of Metacognition in Education*, pp. 340–358. Routledge, New York (2009)
6. White, B., Frederiksen, J., Collins, A.: The interplay of scientific inquiry and metacognition: More than a marriage of convenience. In: Hacker, D.J., Dunlosky, J., Graesser, A.C. (eds.) *Handbook of Metacognition in Education*, pp. 175–205. Routledge, New York (2009)
7. Azevedo, R., Cromley, J.G., Moos, D.C., Greene, J.A., Winters, F.I.: Adaptive content and process scaffolding: A key to facilitating students' self-regulated learning with hypermedia. *Psychological Testing and Assessment Modeling* 53, 106–140 (2011)
8. Leelawong, K., Biswas, G.: Designing learning by teaching agents: The Betty's Brain System. *International Journal of Artificial Intelligence in Education* 18, 181–208 (2008)
9. Robison, J., Rowe, J., McQuiggan, S., Lester, J.: Predicting User Psychological Characteristics from Interactions with Empathetic Virtual Agents. In: Ruttkay, Z., Kipp, M., Nijholt, A., Vilhjálmsón, H.H. (eds.) *IVA 2009. LNCS*, vol. 5773, pp. 330–336. Springer, Heidelberg (2009)
10. Azevedo, R., Johnson, A., Chauncey, A., Graesser, A.: Use of hypermedia to convey and assess self-regulated learning. In: Zimmerman, B., Schunk, D. (eds.) *Handbook of Self-Regulation of Learning and Performance*, pp. 102–121. Routledge, New York (2011)
11. Azevedo, R., Witherspoon, A.M.: Self-regulated use of hypermedia. In: Hacker, D.J., Dunlosky, J., Graesser, A.C. (eds.) *Handbook of Metacognition in Education*, pp. 319–339. Routledge, New York (2009)
12. Graesser, A.C., McNamara, D.S.: Self-regulated learning in learning environments with pedagogical agents that interact in natural language. *Educational Psychologist* 45, 234–244 (2010)
13. Vanlehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems and other tutoring systems. *Educational Psychologist* 46(4), 197–221 (2011)
14. Aleven, V., Roll, I., McLaren, B.M., Koedinger, K.R.: Automated, unobtrusive, action-by-action assessment of self-regulation during learning with an intelligent tutoring system. *Educational Psychologist* 45, 224–233 (2010)
15. Calvo, R., D'Mello, S.K. (eds.): *New perspectives on affect and learning technologies*. Springer, New York (2011)
16. Azevedo, R., Aleven, V. (eds.): *International handbook of metacognition and learning technologies*. Springer, Amsterdam (in press)
17. D'Mello, S.K., Graesser, A.C.: Dynamics of affective states during complex learning. *Learning and Instruction* 22, 145–157 (2012)
18. Kinnebrew, J., Biswas, G., Sulcer, B., Taylor, R.: Investigating self-regulated learning in Teachable Agent environments. In: Azevedo, R., Aleven, V. (eds.) *International Handbook of Metacognition and Learning Technologies*. Springer, Berlin (in press)
19. Hadwin, A.F., Järvelä, S., Miller, M.: Self-regulated, co-regulated, and socially-shared regulation of learning. In: Zimmerman, B.J., Schunk, D.H. (eds.) *Handbook of Self-Regulation of Learning and Performance*, pp. 65–84. Routledge, New York (2011)

Noticing Relevant Feedback Improves Learning in an Intelligent Tutoring System for Peer Tutoring

Erin Walker¹, Nikol Rummel², Sean Walker³, and Kenneth R. Koedinger³

¹School of Computing, CIDSE, Arizona State University, USA

²Institute of Psychology, Ruhr-Universität Bochum, Germany

³Human-Computer Interaction Institute, Carnegie Mellon University, USA

erin.a.walker@asu.edu, nikol.rummel@rub.de,

walker.sean.m@gmail.com, koedinger@cmu.edu

Abstract. Intelligent tutoring techniques can successfully improve student learning from collaborative activities, but little is known about why and under what contexts this support is effective. We have developed an intelligent tutor to improve the help that peer tutors give by encouraging them to explain tutee errors and provide more conceptual help. In previous work, we have shown that adaptive support from this “tutor” tutor improves student learning more than randomly selected support. In this paper, we examine this result, looking more closely at the feedback students received, and coding it for relevance to the current situation. Surprisingly, we find that the amount of relevant support students receive is not correlated with their learning; however, there is a positive correlation with learning and students noticing relevant support, and a negative correlation with learning and students ignoring relevant support. Designers of adaptive collaborative learning systems should focus not only on making support relevant, but also engaging.

Keywords: intelligent tutoring, computer-supported collaborative learning, adaptive collaborative learning systems, peer tutoring.

1 Introduction

Intelligent tutoring systems (ITSs) successfully improve domain learning by tracking problem-solving progress, providing tailored help and feedback, and selecting problems that target misconceptions [1]. However, many of the successful ITSs have been in domains that have well-defined rules such as math and physics (e.g., [2]). Early ITSs were criticized for over-constraining student problem-solving, overemphasizing shallow procedural knowledge, and thus not properly addressing higher-order skills like collaboration, critical thinking, and creativity. In recent years, several ITSs have been developed in response to these criticisms, focusing on metacognition [3], affective modeling and detection [4], and interpersonal interaction [5]. This new wave of ITSs represents an important step towards personalization at all levels of learning: cognitive, metacognitive, motivational, and social [6].

We contribute to this effort by improving the abilities of ITSs for providing adaptive support to collaborative learning. Students benefit from group work, but only when they exhibit productive behaviors [7]. In theory, adaptive collaborative learning

support (ACLS) would be an improvement over nonadaptive forms of support for collaboration, which overstructure the activity for some students while providing insufficient support for others [8, 9]. Indeed, early empirical results suggest that ACLS is an improvement over fixed support and no support at all [10, 11]. However, it is not yet clear why and when ACLS is effective at improving learning. Our work takes a step towards understanding the conditions under which ACLS is effective.

In [12], we proposed two hypotheses for why adaptive support may be effective: 1) Students benefit from receiving relevant support that they can apply to their interaction; and 2) Students who believe support is adaptive feel more accountable for their collaborative actions. To test these hypotheses, we developed an intelligent tutor that assists peer tutors in giving more correct help and higher quality help. In a controlled study, using pre-post measures of learning and surveys of student perceptions, we found evidence that it is the actual adaptivity of support that matters, rather than whether students perceive support as adaptive. However, our conclusions were limited because our analysis did not include process data.

This paper examines why ACLS is effective by looking directly at the *relevance* of each feedback message peer tutors received from the computer, and at the way peer tutors reacted to each message. There have been several ACLS systems that have not been tested in a classroom, but have been evaluated by verifying the validity of the collaborative model used [13], or the applicability of the feedback given [14]. The construction and evaluation of these systems rest upon two hypotheses: Adaptive support systems increase the amount of relevant support given to collaborating students (*H1*), and the more relevant support students receive, the more they will learn (*H2*). Further, research on individual learning from ITSs suggests that it's important that students pay attention to support at the right moments [4]. One reason why relevant support on its own may not be effective is if students fail to notice and engage with the support. Thus, we also examine the relationship between peer tutors' *noticing* of feedback given by the ITS and their domain learning, by including opportunities in the interface for peer tutors to rate support. We hypothesize that peer tutors who notice more relevant support will learn more (*H3*; see Table 1).

2 The Adaptive Peer Tutoring Assistant

Our system builds on the Cognitive Tutor Algebra (CTA), a successful intelligent tutoring system for high school mathematics [2], and allows students to tutor each other using the same interface. The Adaptive Peer Tutoring Assistant (*APTA*) is modeled after traditional novice peer tutoring scripts, where one student tutors another student of the same ability. These scenarios have been successful in classroom environments [e.g., 15], primarily because students of all abilities benefit from giving help [16]; peer tutors engage in *reflective* processes, where they reflect on their partners' errors and notice their own misconceptions, and *elaborative* processes,

Table 1. Hypotheses investigated in this paper

Name	Description
<i>H1</i>	An adaptive system increases the relevant support collaborating students receive.
<i>H2</i>	The more students receive relevant support, the more they will learn.
<i>H3</i>	The more students notice relevant support, the more they will learn.

where they build on their knowledge as they construct explanations [17]. *APTA* encourages peer tutors to engage in these processes, focusing on three skills:

Skill 1: Necessary help. Peer tutors respond to tutee errors and requests for help. This skill leads peer tutors to reflect on the errors and requests.

Skill 2: Targeted help. Peer tutors ask tutees to self-explain and directly address tutee misconceptions in dialogue. Again, this skill leads peer tutors to reflect on misconceptions.

Skill 3: Conceptual help. Peer tutors give conceptual help, prompting them to engage in elaborative behaviors as they construct explanations.

Because help-giving is an important component of many collaborative scenarios [7], we believe that testing our hypotheses within *APTA* will generalize to other *ACLS*.

In the learning environment, students are given a problem like “Solve for y ,” for an equation like “ $ay + by + m = n$ ”. They are grouped into pairs and are seated at different computers at opposite sides of the same classroom. For the remainder of this paper, we refer to the student acting as the tutor in the learning activity as the *peer tutor*, and the student being tutored as the *tutee*. Tutees solve the problem using menus, selecting options like “Subtract from both sides” and typing in a term like m . For some problems, the computer performs the operation; for other, more advanced problems, the student must type in the result of the operation themselves. Peer tutors

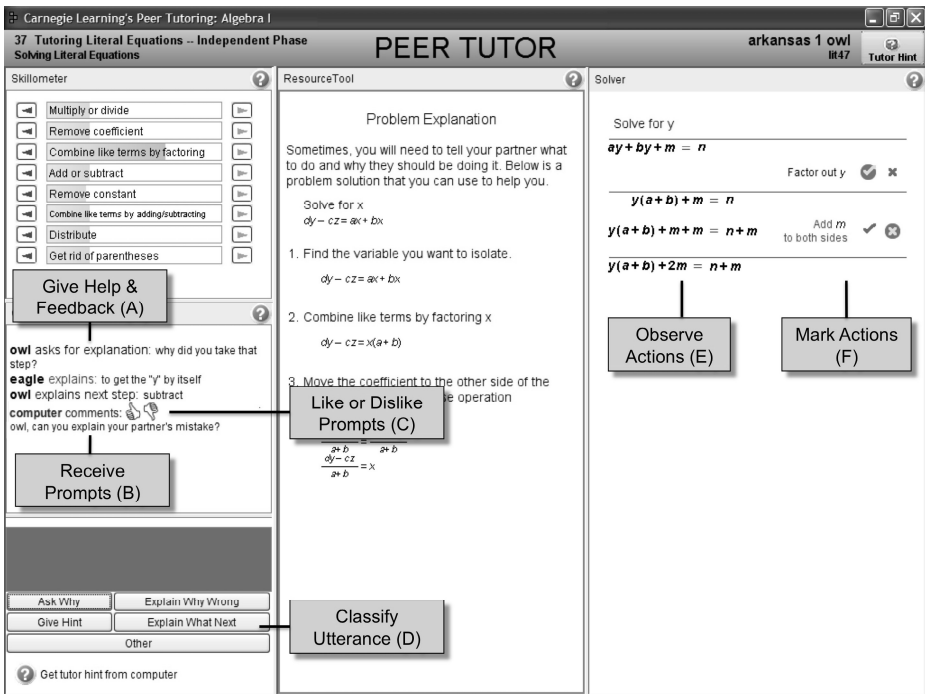


Fig. 1. Peer tutor’s interface in *APTA*. The peer tutor watches the tutee take problem-solving actions (E), and marks the actions right or wrong (F). Students can talk in the chat window (A), where they receive prompts from the computer (B), and can choose to like them, dislike them, or ignore them (C).

can see their peer tutee's actions, but cannot solve the problem themselves (see *E* in Figure 1). Instead, they mark the peer tutee's actions right or wrong (*F* in Figure 1), and receive feedback from the cognitive tutor on whether their marks are correct (described more in [12]). Peer tutors can also interact with tutees in a chat tool, where they give help and feedback (*A* in Figure 1). We augmented the chat tool with sentence classifiers (*D* in Figure 1), asking peer tutors to label their utterances prior to submitting them. Encouraging students to use sentence classifiers correctly was an additional system goal (*Skill 4: Use of Classifiers*).

APTA supports peer tutors in giving better help using reflective prompts visible to both students in the chat window (*B* in Figure 1). For example, after peer tutor instrumental help like "subtract x ", the computer might say "[Tutor], why do you say that? Can you explain more?" The reflective prompts were adaptive in terms of content and timing, based on knowledge tracing of the four skills described above (necessary help, targeted help, conceptual help, and use of classifiers). In response to each relevant peer tutor or tutee action, *APTA* followed a four-step process. First, in Step 1, the problem state was assessed based on the tutee problem-solving action, a machine classification of the peer tutor chat, and a self-classification of the tutor chat. Next, in Step 2, *APTA* used a 20-rule production model to classify the action as effective, somewhat effective, somewhat ineffective, or ineffective, as it related to each of the skills relevant to the particular action. The system assessments of each relevant skill were adjusted using Bayesian Knowledge Tracing. In Step 3, all skills whose assessments had been adjusted based on the previous action were compared to pre-defined thresholds related to the rules that had been fired, to determine if feedback should be given based on the skill. Each threshold had a priority, and the activated threshold with the highest priority was selected as a candidate for feedback. Finally, in Step 4, a feedback message was selected randomly from all possible messages associated with a given threshold. Table 2 displays sample positive and negative feedback related to each skill. We did not give positive feedback for use of classifiers because we considered it to be more distracting than valuable.

Table 2. Positive and negative feedback messages for the four skills traced by *APTA*. Positive feedback was given in response to firing of effective or somewhat effective rules, while negative feedback was given in response to firing of ineffective or somewhat ineffective rules.

Skill	Positive Feedback	Negative Feedback
Necessary help	Keep at it! When your partner asks for help, it's a good chance to explain how to solve the problem.	[Tutor], if you don't know how to help your partner ask the computer for a hint.
Targeted help	Good work! Remember, exploring what your partner is doing wrong can help them not make the same mistake on future problems.	[Tutor], can you explain your partner's mistake?
Conceptual help	Keep it up! Talking about concepts behind the problems can help you to understand them better.	[Tutor], when you explain a step to your partner tell them why they should be doing the step.
Use of classifiers	None	[Tutor], think about whether "ask why", "explain why wrong", "hint", or "explain next step" best describes what you last said.

3 Method

In the study discussed in this paper, described more fully in [12], we compared three conditions. In the *real adaptive* condition, students received adaptive support and were told it was adaptive. In the *real nonadaptive* condition, students received nonadaptive support and were told it was nonadaptive. In the *told adaptive* condition, students received nonadaptive support but were told it was adaptive. As we noticed from previous studies that much nonadaptive support was still plausible feedback that could be applied to the interaction context, the inclusion of the *told adaptive* condition was, in part, to evaluate if students who believed support was adaptive would benefit from nonadaptive support that they received. If students thought the system was adapting to their behaviors, they may be more likely to attend to the support and apply it to their interaction. The *real adaptive* condition used APTA, as described above, while the two nonadaptive conditions received prompts selected as follows. Every time students would have received a reflective prompt were they in the *real adaptive* condition, they did not receive a prompt in the nonadaptive conditions. However, they received a prompt within the next three turns, thus *yoking* the nonadaptive prompt to the adaptive prompt. We randomly selected the content of the nonadaptive prompt, with one exception: we did not choose content related to the skill addressed in the yoked adaptive prompt. Nevertheless, there were many situations where the randomly selected prompt could be perceived as relevant.

Participants were 130 high school students (49 males, 81 females) from one high school, currently enrolled in Algebra 1, Geometry, or Algebra 2. The study was run at the high school, either immediately after school or on Saturdays. Students participated in sessions with up to eight other students (M group size = 7.41, SD = 1.35). Each session was randomly assigned to condition, and then within each pair, students were randomly assigned to the role of tutor or tutee. For the most part, students came with partners they had chosen. For ease of scheduling, we sometimes assigned an extra student to a given session, and 8 students worked alone. 1 dyad was excluded due to a logging error with the computer prompts. Thus, 120 students participated in the collaborative activity. Since our goal was to improve the help that peer tutors give, our discussion in this paper focuses on the 60 students who were assigned the role of peer tutor. An analysis of tutee learning is presented in [12].

Students first took a 20-minute domain pretest, and then spent 20 minutes working individually using the CTA to prepare for tutoring. They were then assigned either the tutor or tutee role. Students spent 60 minutes in a tutoring phase, with one student tutoring another student. Finally, students took a 20 minute domain posttest. Pre- and posttests were counterbalanced, and assessed knowledge of literal equation solving.

We used process data from the study to measure two variables: relevance of computer support and peer tutor noticing of support. First, we coded each instance of support delivered by the computer tutor for whether it was relevant to the current context, as defined by the tutee-tutor interactions spanning the last instance of tutee dialogue, tutor dialogue, and tutee problem step. To be relevant, negative feedback had to meet three criteria:

1. *Not contradict the current situation.* E.g., feedback that referred to an error contradicts the situation if tutees had not made an error.

2. *Refer to something students were not currently doing.* E.g., feedback that prompted for more conceptual help would only be relevant if students were not giving conceptual help.
3. *If students were to follow the help, their interaction would be improved, based on the four skills.* E.g., feedback that tells the tutor to give help would improve the interaction if the tutee had asked for help and not received it.

For positive feedback to be relevant, students had to be doing something to merit positive feedback, and then the advice given by the feedback had to meet the above criteria #1 and #3. To calculate interrater reliability, two raters independently coded 30% of the data, with a kappa of 0.70. Conflicts were resolved through discussion.

The second construct, peer tutor noticing of support, came from an interface feature we added to allow students to give us feedback on the computer prompts. As each prompt was given in the chat window, students could choose to rate the feedback (by clicking thumbs up or thumbs down, see C in Figure 1), or ignore it completely. Students were told that this action would help us determine which feedback was useful. We coded students as noticing the feedback if they rated the feedback, suggesting that they had read and reflected on the feedback. Not rating the feedback gave us no information on their response. We further discuss the implications of this measure in the discussion.

4 Results

For the purposes of this paper we focus on an analysis of how relevant feedback and noticing feedback influenced peer tutor learning. As reported in [12], we conducted a one-way ANCOVA to examine the effects of condition on peer tutor learning, with posttest score as the dependent measure, condition as a between subjects variable, and pretest score as a covariate (see Table 3). Condition had a significant effect on posttest score ($F[2,56] = 4.10, p = 0.022$), and pretest was also significantly predictive of posttest score ($F[1,56] = 31.49, p < 0.001$). We found that providing real adaptive support led peer tutors to learn more. According to an ANOVA, total feedback did not differ between the three conditions ($F[2,57] = 0.591; p = 0.557$; see Table 3), suggesting that the nature of the feedback led to the improvement.

We first examined *H1*, to verify using the process data that the implementation of the adaptive support condition indeed had the intended effect, in that the amount of relevant feedback differed between adaptive and nonadaptive conditions (see Table 3 for means). We conducted a linear regression with relevance as the dependent variable. We included two dummy coded condition variables in the regression, one representing the told adaptive condition and one representing the real nonadaptive condition. We also controlled for total feedback given by the computer, adding it as a predictor variable, and including the two interaction terms between each dummy coded condition variable and total feedback. Because we included interaction terms, we centered the total feedback variable by subtracting the mean. We found that the model that included the interaction terms was a better fit for the data (F Change $[2,54] = 20.62, p < 0.001$). The results of the regression are presented in Table 4. The model was significant ($R^2 = 0.902, F[5,54] = 99.95, p < 0.001$). All variables entered were significant in the model. When all else is held constant, the real adaptive

Table 3. Mean pretest scores, posttest scores, and amounts of total feedback given by the computer, relevant feedback given by the computer, and attended feedback given by the computer. Standard deviations are in parentheses.

Condition	Pretest	Postttest	Total Feedback	Relevant Feedback	Noticed Feedback
Real Adaptive	0.27 (0.17)	0.39 (0.18)	15.53 (11.28)	12.84 (10.83)	7.16 (6.56)
Told Adaptive	0.24 (0.12)	0.27 (0.14)	17.50 (9.37)	7.45 (5.50)	4.73 (5.91)
Real Nonadaptive	0.30 (0.15)	0.29 (0.18)	14.26 (8.00)	5.68 (4.44)	4.21 (4.12)

Table 4. Regression results comparing the effects of condition and total feedback on relevant feedback given by the computer

Variable	β	$t(55)$	p
Told Adaptive	-0.402	-8.04	<0.001
Real Nonadaptive	-0.400	-7.97	<0.001
Total Feedback	1.149	17.62	<0.001
Total Feedback* *Told Adaptive	-0.327	-5.66	<0.001
Total Feedback* Real Nonadaptive	-0.263	-4.98	<0.001

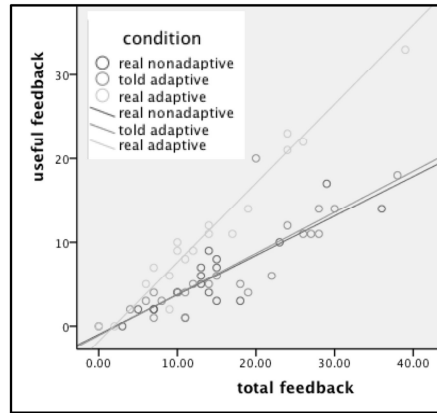


Fig. 2. Graph representing the interaction between total feedback given by the computer, useful feedback, and condition

condition was responsible for significantly more instances of relevant feedback (76%) than the told adaptive (41%) and real nonadaptive conditions (40%). The interaction terms show that the more total instances of feedback, the greater the difference between the real adaptive condition and other conditions (see Figure 2).

We then examined $H2$, looking at whether total relevant feedback was related to learning. We conducted a linear regression, with posttest as the dependent variable, and told adaptive, real nonadaptive, pretest, and relevant feedback as predictor variables ($R^2 = 0.44$; $F(4,55) = 10.97$; $p < 0.001$). While as before condition and pretest were significantly predictive of learning, the total amount of relevant feedback was not ($\beta = -0.180$, $t(54) = -1.65$, $p = 0.104$). Despite the real adaptive condition containing more relevant help, this alone did not explain learning gains found.

Next, we looked at $H3$, examining whether the amount of relevant support students rated affected their learning. We had divided support into two categories: support that peer tutors noticed (by pressing the like or dislike button), and support that peer tutors

ignored. Given that we had also coded support for whether it was relevant or irrelevant, we then had four categories: noticed relevant support, ignored relevant support, noticed irrelevant support, and ignored irrelevant support (see Table 5 for means). We conducted a linear regression, with posttest as the dependent variable, and several predictor variables: pretest, noticed relevant feedback, ignored relevant feedback, noticed irrelevant feedback, and ignored irrelevant feedback. The overall model was significant ($R^2 = 0.512$, $F[5,54] = 11.32$, $p < 0.001$). Noticing relevant feedback was significantly positively related to learning, while ignoring relevant feedback was significantly negatively related to learning (see Table 6). On the other hand, student interactions with irrelevant feedback did not relate to learning.

Because noticing or ignoring relevant feedback related to learning, we explored how those variables differed between conditions. We conducted a MANCOVA with noticed relevant and ignored relevant feedback as dependent variables, and condition and total feedback as predictor variables. Condition significantly affected the amount of noticed relevant feedback ($F[2,56] = 7.10$, $p = 0.002$) and ignored relevant feedback ($F[2,56] = 3.46$, $p = 0.038$). This relationship was strongest for the noticed relevant variable, where post-hoc pairwise comparisons revealed that the real adaptive condition was significantly different from both the real nonadaptive condition ($p = 0.009$) and the told nonadaptive condition ($p = 0.003$). For ignored relevant feedback, the adaptive condition was marginally different from the told adaptive condition ($p = 0.06$) and not significantly different from the real nonadaptive condition ($p = 0.105$).

As noticing feedback played a role in tutor learning, we further examined whether students noticed different amounts of feedback across conditions. A one-way ANCOVA with noticed feedback as the dependent variable, condition as the independent variable, and controlling for total feedback, revealed that students did not notice different amounts of feedback across conditions ($F[2,56] = 1.78$, $p = 0.178$; means of noticed feedback are in Table 3). Students noticed similar numbers of feedback across conditions, but because there was more relevant feedback in the real adaptive condition, students noticed more relevant feedback in that condition.

Table 5. Means of variables relating to attention and relevant feedback. Standard deviations are in parentheses.

Condition	Noticed Relevant	Ignored Relevant	Noticed Irrelevant	Ignored Irrelevant
Real Adaptive	5.63 (5.33)	7.21 (9.02)	1.53 (1.98)	1.16 (1.54)
Told Adaptive	2.05 (3.65)	5.41 (4.53)	2.68 (3.31)	7.36 (5.18)
Real Nonadaptive	1.79 (2.10)	3.89 (4.46)	2.42 (2.48)	6.16 (5.48)

Table 6. Regression results for the effects of relevant and attended feedback on posttest score

Variable	β	$t(55)$	p
Pretest	0.550	5.66	<0.001
# Noticed Relevant	0.324	2.81	0.007
# Ignored Relevant	-0.279	-2.64	0.011
# Noticed Irrelevant	-0.088	-0.84	0.407
# Ignored Irrelevant	-0.070	-0.61	0.543

5 Discussion and Conclusions

In this paper, we examined when adaptive collaboration support might be effective. We discovered that our adaptive system indeed provided students with more relevant support than a nonadaptive system, and this difference became more apparent the more feedback students received. However, relevant support alone was not related to student learning. Instead, students had to notice relevant support in order to benefit from the support. Students noticed support at similar rates across all three conditions, but because there was more relevant support in the adaptive condition, students noticed more relevant support when the system was adaptive.

Our results depend heavily on our measure of relevance and our measure of noticing. The coding scheme we developed for feedback relevance took several iterations, and we found that many feedback messages could be interpreted as relevant in several different situations. The nonadaptive conditions had relatively high incidences of relevant help, even though we tried to select messages that were not relevant. It is possible that a carefully designed nonadaptive system may be able to mimic the performance of an adaptive support system. Our second measure tracked whether students liked or disliked particular feedback messages as an indication of whether students noticed feedback. This measure of noticing implies that students had read the feedback, and had potentially reflected on how it related to their interaction. This method has limitations; if students did not respond to a feedback message, it is impossible to be certain that they did not notice it. However, as a rough measure, it provided insight on how students reacted. Including these types of measures in other ITSs may provide useful online information on how students react to support.

One interpretation of the results is causal: The adaptive system led students to notice more relevant support, and students who noticed relevant support learned more. This interpretation might explain why students in the real adaptive condition learned the most. However, the adaptive system also caused students to *ignore* more relevant support (albeit to a lesser degree) and students who ignored more relevant support learned less. It is possible that students who ignored relevant support were struggling the most with the learning activity, and also learning less because of their difficulties. While we are limited in our ability to draw causal conclusions from this analysis, we do know that the amount of relevant support played a factor in student learning; noticing relevant support related to learning, while noticing irrelevant support did not. Encouraging students to notice more support, while continuing to work on making support more relevant, may be one key to maximizing the benefits of ACLS.

Thus, the next step in this work will be to examine why students notice support, and determine how to encourage more students to attend to and reflect on support. It is likely that individual differences affect the degree to which students notice help (although noticing relevant support was not correlated with pretest score). A promising approach might be to use data mining techniques to improve the design of feedback messages, improving student likelihood of noticing those messages. The timing of messages might have an influence: In initial exploration, we found that feedback messages that appeared when peer tutors were struggling and distracted were more likely to be ignored. Content might also have an influence: Feedback messages that were specific and easily implemented appeared to be more engaging.

Our work makes a contribution to the study of ACLS by showing that producing more relevant support alone is not sufficient for improving learning. Students who benefit from relevant support must notice the support. While this finding is intuitive, and has been discussed in individual learning, it had not previously been demonstrated in learning from collaborative systems or discussed in the design of ACLS. ACLS systems are often designed and evaluated with the ultimate goal of creating more relevant support. Future designers of such systems will have to explore how to improve student noticing of support in order to have a significant impact.

Acknowledgments. This work was supported by the Pittsburgh Science of Learning Center, NSF Grant #SBE-0836012, and a Computing Innovations Fellowship, NSF Grant #1019343. Thanks to Ruth Wylie and Kasia Muldner for their comments.

References

1. VanLehn, K.: The behavior of tutoring systems. *IJAIED* 16(3), 227–265 (2006)
2. Koedinger, K., Anderson, J., Hadley, W., Mark, M.: Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education* 8, 30–43 (1997)
3. Muldner, K., Burleson, W., VanLehn, K.: “Yes!”: Using Tutor and Sensor Data to Predict Moments of Delight during Instructional Activities. In: De Bra, P., Kobsa, A., Chin, D. (eds.) *UMAP 2010*. LNCS, vol. 6075, pp. 159–170. Springer, Heidelberg (2010)
4. Roll, I., Alevan, V., McLaren, B.M., Koedinger, K.R.: Improving students’ help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction* 21, 267–280 (2011)
5. Ogan, A., Alevan, V., Jones, C., Kim, J.: Persistent Effects of Social Instructional Dialog in a Virtual Learning Environment. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011*. LNCS, vol. 6738, pp. 238–246. Springer, Heidelberg (2011)
6. du Boulay, B., Avramides, K., Luckin, R., Martinez-Miron, E., Rebolledo-Mendez, G., Carr, A.: Towards Systems That Care: A Conceptual Framework based on Motivation, Metacognition and Affect. *International Journal of Artificial Intelligence in Education* 20(3), 197–229 (2010)
7. Johnson, D.W., Johnson, R.T.: Cooperative learning and achievement. In: Sharan, S. (ed.) *Cooperative Learning: Theory and Research*, pp. 23–37. Praeger, NY (1990)
8. Dillenbourg, P.: Over-scripting CSCL: The risk of blending collaborative learning with instructional design. In: Kirschner, P.A. (ed.) *Three worlds of CSCL : Can we support CSCL?*, pp. 61–91 (2002)
9. Kollar, I., Fischer, F., Slotta, J.D.: Internal and external collaboration scripts in web-based science learning at schools. In: Koschmann, T., Suthers, D., Chan, T.-W. (eds.) *The next 10 years! Proc. CSCL 2005*, pp. 331–340. Lawrence Erlbaum Associates, Mahwah (2005)
10. Baghaei, N., Mitrovic, A., Irwin, W.: Supporting Collaborative Learning and Problem Solving in a Constraint-based CSCL Environment for UML Class Diagrams. *International Journal of Computer-Supported Collaborative Learning* 2(2-3), 159–190 (2007)
11. Kumar, R., Rosé, C.P., Wang, Y.C., Joshi, M., Robinson, A.: Tutorial dialogue as adaptive collaborative learning support. In: *Proc. AIED 2007*, pp. 383–390. IOS Press (2007)
12. Walker, E., Rummel, N., Koedinger, K.R.: Adaptive support for CSCL: Is it feedback relevance or increased accountability that matters? In: *Proc. CSCL 2011*, pp. 334–342 (2011)

13. Suebnukarn, S., Haddawy, P.: Modeling Individual and Collaborative Problem-Solving in Medical Problem-Based Learning. *User Modeling and User-Adapted Interaction* 16(3-4), 211–248 (2006)
14. Constantino-González, M.A., Suthers, D., Escamilla de los Santos, J.: Coaching web-based collaborative learning based on problem solution differences and participation. *IJAIED* 13, 263–299 (2003)
15. Fantuzzo, J.W., Riggio, R.E., Connelly, S., Dimeff, L.A.: Effects of reciprocal peer tutoring on academic achievement and psychological adjustment: A component analysis. *Journal of Educational Psychology* 81(2), 173–177 (1989)
16. Ploetzner, R., Dillenbourg, P., Preier, M., Traum, D.: Learning by explaining to oneself and to others. In: Dillenbourg, P. (ed.) *Collaborative Learning: Cognitive and Computational Approaches*, pp. 103–121. Elsevier Science Publishers (1999)
17. Roscoe, R.D., Chi, M.: Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions. *Review of Educational Research* 77(4), 534–574 (2007)

Multi-paradigm Generation of Tutoring Feedback in Robotic Arm Manipulation Training

Philippe Fournier-Viger¹, Roger Nkambou², André Mayers³,
Engelbert Mephu-Nguifo^{4,5}, and Usef Faghihi²

¹Dept. of Computer Sciences, University of Moncton, Canada

²Dept. of Computer Sciences, University of Quebec in Montreal, Canada

³Dept. of Computer Sciences, University of Sherbrooke, Canada

⁴Clermont Université, Université Blaise Pascal, LIMOS, F-63000 Clermont-Ferrand

⁵CNRS, UMR 6158, LIMOS, F-63173 Aubière

philippe.fv@gmail.com, nkambou.roger@uqam.ca,
andre.mayers@usherbrooke.ca, mephu@isima.fr, jfaghihi@yahoo.com

Abstract. Building an intelligent tutoring system requires to define an expertise model that can support appropriate tutoring services. This is usually done by adopting one of the following paradigms: building a cognitive model, specifying constraints, integrating an expert system and using data mining algorithms to learn domain knowledge. However, for some ill-defined domains, the use of a single paradigm could lead to a weak support of the user in terms of tutoring feedback. To address, this issue, we propose to use a multi-paradigm approach. We illustrate this idea in a tutoring system for robotic arm manipulation training. To support tutoring services in this ill-defined domain, we have developed a multi-paradigm model combining: (1) a data mining approach for automatically building a task model from user solutions, (2) a cognitive model to cover well-defined parts of the task and spatial reasoning, (3) and a 3D path-planner to cover all other aspects of the task. Experimental results indicate that the multi-paradigm approach allows providing assistance to learners that is much richer than what is offered with each single paradigm.

Keywords: tutoring services, expertise model, ill-defined domains.

1 Introduction

To assist learners during problem-solving activities, an intelligent tutoring system (ITS) needs to be equipped with domain knowledge that can support appropriate tutoring services. However, modelling the domain knowledge can be quite time-consuming and difficult especially for *ill-defined domains* [1]. According to Lynch et al. [1], domains containing *ill-structured problems* are ill-defined. Simon [2] defines an ill-structured problem as one that is complex, with indefinite starting points, multiple and arguable solutions, or unclear strategies for finding solutions. To provide domain knowledge to an ITS, three popular paradigms have been widely used in the ITS community. The first one is cognitive task analysis, which consists of observing

expert and novice users (e.g. [3]) to produce effective problem spaces or task models. However, cognitive task analysis is very time-consuming [3]. Furthermore, for ill-defined domains, it is not always possible to define a complete or partial task model by hand. The second paradigm is constraint-based modeling (CBM) [4]. It consists of specifying sets of constraints on a correct behavior instead of providing a complete task description. Though, this approach was shown to be effective for some ill-defined domains, it can be very challenging to design a complete set of constraints for some domains. The third paradigm consists of integrating an expert system into an ITS (e.g. [5, 6]). However, developing an expert system can be difficult and costly, especially for ill-defined domains, and expert systems sometimes do not generate explanations in a form that is appropriate for learning. Recently, a fourth paradigm [7, 8] used data mining algorithms to automatically extract partial task models from users interactions with an ITS. The partial task models can then be used to offer assistance to learners. Even though the approach was proven to be efficient in procedural ill-defined domains, the task models extracted are partial and are not useful for unseen situations.

We assume that a good integration of these different paradigms could help maximize the benefits associated with each of them in specific conditions. To validate this hypothesis, we have implemented the multi-paradigm model within *CanadarmTutor*, an ITS for training astronauts to the *Canadarm2* robot manipulation in various situations. Our preliminary experiments have shown promising results.

This paper is organized as follows. Section 2 introduces *CanadarmTutor* and the three paradigms we have implemented into it for representing the domain expertise. Section 3 explains how we have combined them in a multi-paradigm expert model. Section 4 presents an experimental evaluation of *CanadarmTutor* equipped with the multi-paradigm model, followed by some concluding remarks in section 5.

2 *CanadarmTutor*

CanadarmTutor [9] (cf. Figure 1.a) is a simulation-based tutoring system for coaching astronauts how to operate *Canadarm2* (cf. Figure 1.b), a 7 degrees of freedom robotic arm deployed on the International Space Station (ISS). The main learning activity in *CanadarmTutor* is to move the arm from a given configuration to a goal configuration. Such activity is usually done in various complex tasks including inspecting the ISS and moving payloads. The arm movements are performed by astronauts inside the ISS. Maneuvering *Canadarm2* on the ISS is difficult since there is a limited view of the environment. The environment is rendered through three monitors, each showing the view obtained from a single camera while about ten cameras are mounted at different locations on the ISS and on the arm. To move the arm, the operator must select at every moment the best cameras for viewing the scene of operation. Moreover, an operator has to select and perform appropriate joint rotations for moving the arm, while avoiding collisions and dangerous configurations. Operators also have to follow an extensive security protocol that comprises numerous steps because a single mistake, such as neglecting to lock the arm into position can lead to catastrophic and costly consequences. Operating *Canadarm2* is an ill-defined task (according to the definition of Simon [2]) because there exist a huge number of ways to move the arm to a goal configuration and it is very difficult to formalize how to select the moves that a

human would execute. The reason is that some arm movements are preferable to others depending on criteria that are hard to be formalized such as the view of the arm given by the cameras, the relative position of obstacles on the ISS to the arm and the familiarity of the user with certain manipulations. In practice, skills to operate the arm are mainly learned by practice. Because of this, it is hard to model the domain expertise in CanadarmTutor.

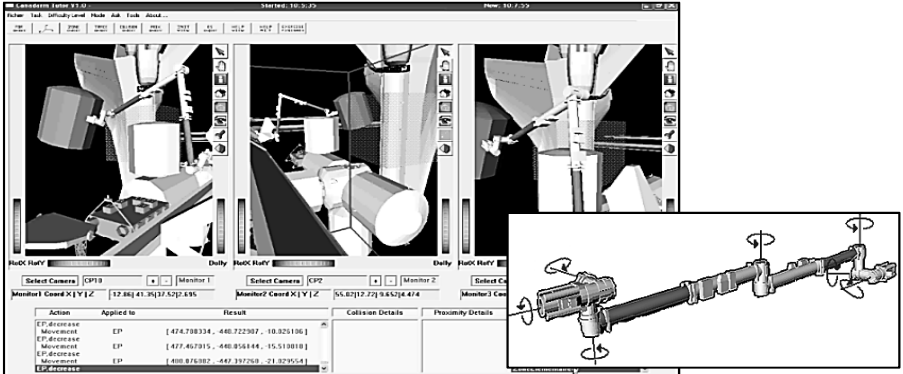


Fig. 1. (a) CanadarmTutor, (b) a 3D representation of Canadarm2

2.1 Integrating a Path-Planner for Automatic Path Generation

To implement the domain expertise in CanadarmTutor, we first based our work on the expert system approach. A custom path-planner named FADPRM was integrated into CanadarmTutor [9]. FADPRM is an efficient algorithm for robot path-planning in constrained based environments. It can calculate a trajectory (e.g. Figure 2.a) between any two robotic arm configurations while avoiding obstacles and considering constraints such as dangerous and desirable zones. Integrating FADPRM in CanadarmTutor provides the following benefits. First, in a training session, CanadarmTutor uses FADPRM to automatically produce demonstrations of correct arm maneuver on the ISS by generating a path between two arm configurations, while considering the obstacles (the ISS modules) and predefined constraints. Second, for a given task, CanadarmTutor automatically generates paths and estimates the distance with the learner solution to evaluate it. Although the path-planner can provide useful tutoring services, our experiments with learners show that the generated paths are not always realistic, as they are not based on human experience. Moreover, they do not cover some important aspects of the task such as selecting cameras and adjusting their parameters. Furthermore, given that the path-planner has no representation of knowledge and skills, it cannot support important tutoring services such as estimating learners' knowledge gaps.

2.2 Integrating a Cognitive Model to Assess Skills and Spatial Reasoning

Facing these problems, we applied the cognitive task analysis paradigm [3]. To understand how astronauts operate Canadarm2, we attended two-week training with

astronauts at the Canadian Space Agency and also interviewed the training staff. To encode how users operates the robotic arm, we used a custom cognitive model [10], similar to the one used in CTAT [1], the reference model for building “model-tracing tutors”. The main difference between CTAT and our model is that ours is designed to also evaluate spatial reasoning, a key issue for manipulating Canadarm2. To take into account the spatial dimension, our review of the literature on spatial cognition has shown that most researchers in psychology and neurosciences agree that spatial knowledge is declarative and is necessary for complex spatial reasoning (“allocentric representations”) [11, 12, 13]. Furthermore, spatial knowledge could be represented by relations of the form “a r b”, where “a” and “b” are symbols designating objects and “r”, a spatial relationship between the objects [14].

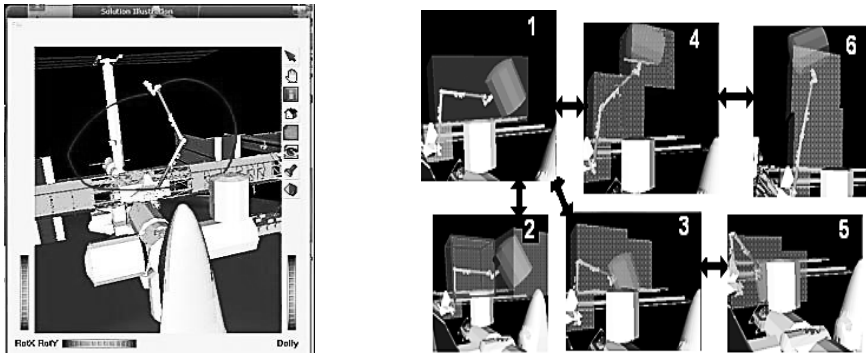


Fig. 2. (a) The FADPRM Path-Planner (b) Six Elementary Spaces

Based on these facts, to model the spatial knowledge in CanadarmTutor, we discretized the 3D space into 3D subspaces that we name elementary spaces (ESP). This allows us to represent the continuous space as discrete symbols. In Canadarm2 manipulation, it was determined that the most realistic types of ESP for mental processing are ESs configured with an arm shape. Figure 2b illustrates 6 of the 30 ESs that we defined. For example, one can move the arm from ESP 1 to ESP 2, ESP 3 and ESP 4. ESP 5 can be reached from ESP 3, and ES6 can be reached from ES4. Each ESP is represented by three cubes. Spatial knowledge was then encoded as four types of relationships such as (1) a camera can see an ESP or an ISS module, (2) an ESP contains an ISS module, (3) an ESP is next to another ESP and (4) a camera is attached to an ISS module. The procedural knowledge of how to move the arm to a goal configuration was modeled as a loop where the learner, before any arm movements, must recall a set of cameras for viewing the ESPs containing the arm, select the correct cameras, adjust their parameters, retrieve a sequence of ESPs to go from the current ESP to the goal, and then start moving the arm to the next ESP.

This task model allowed us to integrate six new tutoring services in CanadarmTutor. First, a learner can explore the task model to learn how to operate the arm and learn about properties of the ISS, the cameras and Canadarm2. Second, model-tracing capability allows the system to evaluate the learner knowledge during arm manipulation exercises. After a few exercises CanadarmTutor automatically

builds a detailed learner profile that shows the strength and weakness of the learner in terms of mastered, missing and buggy knowledge. This is done by comparing the task model with a learner solution to see which knowledge is used by the learner. Third, CanadarmTutor uses the declarative knowledge linked to the task model to generate and provide the learner with direct questions such as “Which camera can be used to view the Node02 ISS module?”. The fourth tutoring service is to assist the learners by providing useful hints and demonstrations during arm manipulation exercises. Suggesting the next step and generating demonstrations is done thanks to the model-tracing capability of this paradigm. The fifth tutoring service is to generate personalized exercises based on the student model. By using the student model, CanadarmTutor can generate exercises that involve knowledge not yet mastered by the learner. The sixth and last tutoring service is to offer proactive help to the learner. For instance, if Canadarm2 is moved without performing camera adjustment, CanadarmTutor warns the learner to check if cameras are well adjusted. This type of help which is also implemented based on model-tracing is particularly appreciated by beginners and intermediate learners. However, the cognitive model also has some limitations. Although it models the main steps of the manipulation task in detail, it does not go into details about how to select joint rotations for moving Canadarm2. The reason is that for a given arm movement problem, there is a huge number of possibilities and choosing one of them requires considering criteria that are hard to formalize such as the safety and ease of manoeuvres. It is thus not possible to define a complete and explicit task model for this task, making it an ill-defined task according to Simon’s definition [2]. The path-planner could generate paths to provide help at the level of joint rotation. But they are sometimes too complex and difficult to be executed by users, as they are not based on human solutions.

2.3 Using Data Mining Techniques to Learn Partial Task Models

Given the aforesaid drawbacks with other paradigms, we applied the fourth paradigm, which is the automatic acquisition of partial task models [8]. It consists of applying data mining algorithms on user solutions to automatically extract a partial task model instead of defining it by hand. The goal is to provide tutoring services for parts of the task of operating the arm that are ill-defined and could not be represented easily with the cognitive model (e.g. how to select the joint rotations to move Canadarm2). An advantage of this approach over the path-planner is that it is based on real user data.

To apply this approach, we first recorded a set of user solutions for each exercise [8]. In CanadarmTutor, an exercise consists of moving the robotic arm from an initial configuration to a goal configuration. For each attempt, a *sequence of actions* is created in a database. We defined 112 actions that can be recorded including (1) applying a rotation value to one of the seven arm joints (2) selecting a camera and (3) performing an increase or decrease of the pan/tilt/zoom of a camera. An example of a partial action sequence recorded for a user in CanadarmTutor is $\langle (0, rotateSP\{2\}), (1, selectCP3), (2, panCP2\{4\}), (3, zoomCP2\{2\}) \rangle$ which represents decreasing the rotation value of joint SP by two units, selecting camera CP3, increasing the pan of camera CP2 by four units and then its zoom by two units. Furthermore, we annotated

sequences with contextual information called “dimensions”. Table 1 shows an example of a toy database containing six solutions annotated with five dimensions. In this Table, *a*, *b*, *c*, and *d* denote actions. The dimension “*Solution state*” indicates if the learner solution was successful. Values for this dimension are assigned by CanadarmTutor. The four other dimensions are examples of dimensions that can be added manually. The dimension “*Expertise*” denotes the expertise level of the learner who performed a sequence. “*Skill_1*”, “*Skill_2*” and “*Skill_3*” indicate whether any of these three specific skills were demonstrated by the learner when solving the problem. This example illustrates five dimensions. However, any kind of learner information or contextual information can be encoded as dimensions. In CanadarmTutor, we used 10 skills that we selected to be the most important, and the “*solution state*” and “*expertise level*” dimensions to annotate sequences.

To generate a partial task model from the user solutions, we then applied a custom sequential pattern mining algorithm [8] on the database of user solutions. The algorithm takes as input a sequential database and a threshold named *minsup*. The algorithm then extracts subsequences of actions that are common to at least *minsup* learners. We have designed the custom algorithm specifically to accept dimensions and also different types of constraints useful in our context [8]. Table 2 shows some subsequences (also called patterns) found from the database shown in Table 1 with *minsup* = 2. Consider pattern P3. This pattern represents doing action *b* one time unit (immediately) after action *a*. The pattern P3 appears in sequences S1 and S3 of Table 1. It has thus a *support* of two. Moreover, the annotations for P3 tell us that this pattern was performed by experts who possess skills 1, 2 and 3 and that P3 was found in plan(s) that failed, as well as plan(s) that succeeded.

Table 1. An example toy database containing 6 user solutions

ID	Dimensions					Sequence of actions
	Solution state	Expertise	Skill_1	Skill_2	Skill_3	
S1	successful	Expert	yes	yes	yes	<(0,a),(1,b,c)>
S2	successful	novice	no	yes	no	<(0,d)>
S3	buggy	expert	yes	yes	yes	<(0,a),(1,b,c)>
S4	buggy	intermediate	no	yes	yes	<(0,a),(1,c),(2,d)>
S5	successful	expert	no	no	yes	<(0,d),(1,c)>
S6	successful	novice	no	no	yes	<(0,c),(1,d)>

Table 2. Some frequent patterns extracted from the dataset of Table 1 with a *minsup* of 2

ID	Dimensions					Sequence of actions	Support
	Solution State	Expertise	Skill_1	Skill_2	Skill_3		
P1	*	expert	yes	yes	yes	<(0,a)>	2
P2	*	*	*	yes	yes	<(0,a)>	3
P3	*	expert	yes	yes	yes	<(0,a),(1,b)>	2
P4	successful	*	no	*	*	<(0,d)>	3

We have then implemented three tutoring services in CanadarmTutor that use the partial task models. First, CanadarmTutor can assess the profile of the learner (expertise level, skills, etc.) by looking at the applied patterns. If for example a learner applies patterns with the value "intermediate" for the dimension “*expertise*” 80 % of the time, then CanadarmTutor asserts that the learner expertise level is "intermediate".

In the same way, CanadarmTutor can diagnose mastered and missing/buggy skills for users who demonstrated a pattern by looking at the “skills” dimensions of the applied patterns (e.g. “Skill_1” in Table 2).

The second tutoring service consists in determining the possible actions from the set of patterns and proposing one or more actions to the learner. In CanadarmTutor, this functionality is triggered when the student select “What should I do next?” in the interface menu. CanadarmTutor then checks the matching patterns to make a recommendation to the learner. For example, if the learner performed a rotation of the joint SP followed by a rotation of the joint EP and ask “What Should I do next?”, CanadarmTutor will look for patterns that match with SP, EP to suggest what next action the learner should do.

The third tutoring service is to let learners explore patterns by themselves to find out about ways to solve problems. CanadarmTutor provides an interface that lists the patterns and their annotations, and provides sorting and filtering functions.

The paradigm of learning partial task models from user solutions has several advantages. Unlike the path-planner, it allows us to provide tutoring services based on real users’ arm manipulations (multiple profile users). Moreover, it allows us to assist learners about how to choose a joint rotation –which was impossible to achieve with the cognitive model. However, an important limitation with the partial task model paradigm is that no help can be offered to learners for unexplored solution paths. Thus each of the three paradigms that we have separately tested into CanadarmTutor has its own advantages and limitations. Based on this observation, we decided to combine them to create a multi-paradigm expertise model.

3 Combining the Three Paradigms

The goal is to provide a model that can switch from one paradigm to another in order to take advantages of each one’s strength in situations where it is the best. The proposed multi-paradigm model works as follows.

During arm manipulation exercises, CanadarmTutor performs model-tracing to update the student model. The student model is a list of knowledge units from the cognitive model. Each unit is annotated with a probability that indicates if the knowledge is mastered by the learner. Moreover, the student model is also updated when a learner answers questions asked by CanadarmTutor (cf. section 2.2).

When an exercise is completed (fail or success), the solution is added to a sequence database of user solutions for that exercise (a database similar to the one shown in Table 1). The solution is then annotated with the dimension “Solution State” to indicate the success or failure. Moreover, the skills from the cognitive model are used to annotate sequences as dimensions (if the mastery level is higher than 0.8 in the student model, the skill is considered mastered). Thereafter, when a minimum of 10 sequences have been recorded for an exercise, the data mining algorithm is applied for extracting a partial task model for the exercise.

When CanadarmTutor detects that a learner follows a pattern during an exercise from the corresponding partial task model, dimensions of the pattern are used for updating the student model. For example, if a learner applies a pattern common to learners possessing “Skill_1”, the mastery level of “Skill_1” in the student model will

be heightened by a small increment (we use 0.05 in CanadarmTutor). In this way, the partial task models are also used for updating the student model (the student model is shared by the cognitive model and the partial task model approach).

During a learning session, CanadarmTutor uses the student model for generating exercises that progressively involves new knowledge or knowledge that is judged not yet mastered by the learner (this is done as explained in section 2.2). The exercises that are generated are either questions about declarative knowledge of the cognitive model or robotic arm manipulation exercises.

During an arm manipulation exercise, when a learner asks for help about what should be done next, the system generates a solution using the three aforementioned approaches (cf. Figure 3). First, the cognitive model gives the general procedure that should be followed for moving the arm such as “You should select a camera and then adjusts its parameter for monitor 2” (cf. Figure 3.A). This help is generated by performing model-tracing with the cognitive model. Then, in the same window, the patterns from the partial task model that match the current user solution are displayed to the learner. For example, three patterns are presented in Figure 3.B. The learner can view a pattern as an animation by using the arrow buttons. Patterns give mainly the information about the joint rotations that should be performed for moving the arm. If no pattern matches the current learner solution, a demonstration is generated by the path-planner that demonstrates possible paths as solutions (cf. Figure 3.C).

Furthermore, CanadarmTutor can provide proactive help to learners such as assisting the learners to choose the best cameras thanks to the cognitive model (cf. section 2.2). CanadarmTutor can also let the learner explore patterns from the partial task models (cf. section 2.3) or the cognitive model (cf. section 2.1) to learn about different ways to solve problems or about the general procedure for moving the arm. The learner can also request demonstrations at any time from the path-planner (cf. section 2.1) or the cognitive model (cf. section 2.2).

Table 1 summarizes the different tutoring services supported by each paradigm and the multi-paradigm model is provided in Figure 3. It shows that the tutoring services supported by the multi-paradigm approach are much richer.

4 Experimental Evaluation

We performed an evaluation with ten users to evaluate the multi-paradigm version of CanadarmTutor. The goal of the evaluation was twofold: (1) to measure if the tutoring services help the learners to learn and (2) if, during an exercise, CanadarmTutor’s interventions are relevant to the current solution. To make sure that for each exercise some patterns are extracted by our data mining algorithms, we recorded at least 30 solutions for each robotic arm manipulation exercise.

Experimental Procedure. We explained to each participant the procedure of the experiment and what kind of data will be collected. Then, we asked each participant to perform fifteen procedural exercises. Completing the exercises took about one hour for each participant. During this session, we allowed participants to use all tutoring services. We set CanadarmTutor to record all solutions so that they can be examined after the experiment. During the experiment, we observed the participant and took notes to evaluate (1) if the tutoring services gave relevant help when they were used

and (2) whether the learners corrected their mistakes after using the tutoring services or they were more confused. Finally, we performed a five minute interview with each learner to see their opinion on the same two aspects, and also their general opinion about the tutoring services and how CanadarmTutor could be improved.

Experimental Results. All participants completed the fifteen exercises. Most participants used all tutoring services. We found that participants relied more on the tutoring services for the most difficult exercises, which is what we expected. All participants mentioned that they found the tutoring services very useful and that the tutoring services helped them learn how to manipulate Canadarm2. Our observation was that learners using the tutoring services did not repeat their mistakes after receiving feed-back. Users also agreed that the set of tutoring services would be less interesting if some were removed, which confirm that the multi-paradigm model is superior to using each individual approach.

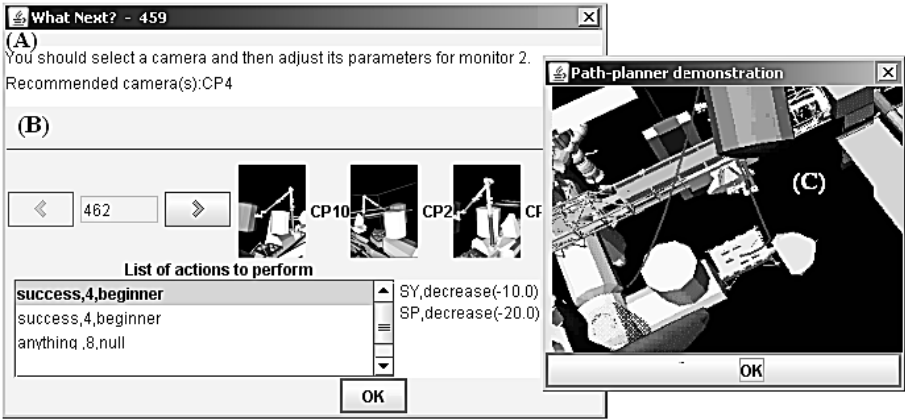


Fig. 3. A Hint Offered by the Multi-Paradigm Approach

Table 3. Tutoring services offered with each paradigm

	Path-planner	Cognitive model	Data mining approach	Multi-paradigm
Generate path demonstrations and evaluate the path followed by the learner	Yes			Yes
Free exploration of the knowledge, demonstrations, hints, proactive help, skill evaluation (for well-defined parts of the task)		Yes		Yes
Evaluate declarative knowledge with questions (including spatial knowledge)		Yes		Yes
Free exploration of the knowledge, hints, skill evaluation (for ill-defined parts of the task)			Yes	Yes
Integrated help covering all aspects of the task				Yes

5 Conclusion

In this paper, we have argued for the use of multi-paradigm approaches for supporting tutoring services in procedural and ill-defined domains. The motivation is that different approaches are sometimes better suited for different parts of the same ill-

defined task. We have presented this idea using CanadarmTutor. We have first described how we have tested three different approaches to support tutoring services in CanadarmTutor. We then discussed their respective limitations and explained how the multi-paradigm approach combines the three approaches in the latest version of CanadarmTutor to overcome limitations of each paradigm. The result is tutoring services that greatly exceed what all previous versions of CanadarmTutor offered. An experimental evaluation confirmed that the multi-paradigm model allows us to provide relevant and helpful tutoring services that are appreciated by users.

Acknowledgments. Our thanks go to the FQRNT and NSERC for their logistic and financial support. We also thanks all members of the GDAC/PLANIART involved in this project.

References

- [1] Lynch, C., Ashley, K., Aleven, V., Pinkwart, N.: Defining Ill-Defined Domains; A literature survey. In: Proc. Ill-Defined Domains Workshop, pp. 1–10 (2006)
- [2] Simon, H.A.: Information-processing theory of human problem solving. In: Estes, W.K. (ed.) *Handbook of Learning and Cognitive Processes. Human Information*, vol. 5, pp. 271–295. John Wiley & Sons, Inc. (1978)
- [3] Aleven, V., McLaren, B.M., Sewall, J., Koedinger, K.R.: The Cognitive Tutor Authoring Tools (CTAT): Preliminary Evaluation of Efficiency Gains. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006. LNCS*, vol. 4053, pp. 61–70. Springer, Heidelberg (2006)
- [4] Mitrović, A., Mayo, M., Suraweera, P., Martin, B.: Constraint-Based Tutors: A Success Story. In: Monostori, L., Váncza, J., Ali, M. (eds.) *IEA/AIE 2001. LNCS (LNAI)*, vol. 2070, pp. 931–940. Springer, Heidelberg (2001)
- [5] Clancey, W.: Use of MYCIN’s rules for tutoring. In: Buchanan, B., Shortliffe, E.H. (eds.) *Rule-Based Expert Systems*. Addison-Wesley (1984)
- [6] Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N.: Using Latent Semantic Analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments* 8, 149–169 (2000)
- [7] Barnes, T., Stamper, J.: Toward Automatic Hint Generation for Logic Proof Tutoring Using Historical Student Data. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) *ITS 2008. LNCS*, vol. 5091, pp. 373–382. Springer, Heidelberg (2008)
- [8] Fournier-Viger, P., Nkambou, R., Mephu Nguifo, E.: Learning Procedural Knowledge from User Solutions To Ill-Defined Tasks in a Simulated Robotic Manipulator. In: Romero, et al. (eds.) *Handbook of Educational Data Mining*, pp. 451–465. CRC Press (2010)
- [9] Belghith, K., Kabanza, F., Hartman, L., Nkambou, R.: Anytime Dynamic Path-planning with Flexible Probabilistic Roadmaps. In: Proc. ICRA 2006, pp. 2372–2377 (2006)
- [10] Fournier-Viger, P., Nkambou, R., Mayers, A.: Evaluating Spatial Representations and Skills in a Simulator-Based Tutoring System. *IEEE Trans. Learn. Tech.* 1(1), 63–74 (2008)
- [11] Burgess, N.: Spatial memory: how egocentric and allocentric combine. *Trends in Cognitive Sciences* 10(12), 551–557 (2006)
- [12] Nadel, L., Hardt, O.: The Spatial Brain. *Neuropsychology* 18(3), 473–476 (2004)
- [13] Tversky, B.: Cognitive Maps, Cognitive Collages, and Spatial Mental Models. In: Campari, I., Frank, A.U. (eds.) *COSIT 1993. LNCS*, vol. 716, pp. 14–24. Springer, Heidelberg (1993)
- [14] Gunzelmann, G., Lyon, D.R.: Mechanisms for Human Spatial Competence. In: Barkowsky, T., Knauff, M., Ligozat, G., Montello, D.R. (eds.) *Spatial Cognition 2007. LNCS (LNAI)*, vol. 4387, pp. 288–307. Springer, Heidelberg (2007)

User-Centered Design of a Teachable Robot

Erin Walker and Winslow Burleson

School of Computing, CIDSE, Arizona State University, Tempe, AZ, 85282
{erin.a.walker,winslow.burleson}@asu.edu

Abstract. Robotic learning environments may benefit if combined with intelligent tutoring technologies, but it is unclear how best to integrate the two types of systems. We explore this integration using a tangible teachable agent paradigm, where students teach a robot about geometry concepts. To identify potential design directions, we employ a user-centered method called Speed Dating, involving construction of several scenarios probing student needs, and then orchestration of user enactments of the scenarios. We found that students seek activities that provide them with an appropriate level of challenge, feelings of discovery, opportunity for physicality, and a sense of responsibility for the robot. We discuss the implications of these findings with respect to building a tangible teachable robot. By employing HCI methods underutilized in learning, we gain traction on an important research challenge in education technology.

Keywords: teachable agents, tangible learning environments, robotic learning environments, intelligent tutoring systems.

1 Introduction

Intelligent tutoring systems (ITSs) have been successful at improving classroom learning, due to their personalized hints, feedback, and problem selection. However, most mainstream educational software has been designed for personal computers, and this paradigm creates an artificial separation between the input device, system output, and underlying real-world representation [1]. Tangible learning environments (TLEs), where students interact in physical spaces with digitally augmented devices, facilitate sensory engagement, experiential learning, and collaborative exploration [2]. There is growing evidence that for the “learning-by-doing” activities associated with TLEs to be effective, they need to be combined with explicit goals, a structure that provides students with support, and mechanisms that encourage students to persist in the face of failure [3]. Integrating TLEs and ITSs may improve their effectiveness. We explore the intersection between TLEs and ITSs using a robotic teachable agent for middle school mathematics. The robot will adjust its behaviors in ways that demonstrate what the student has taught it, highlight potential misconceptions, and provide students with feedback and encouragement. The combination of robotic tangibles and teachable agents presents a difficult design problem: How does one leverage advantages of tangible environments while retaining benefits of structured learning with teachable agents? We employ a user-centered methodology called Speed Dating to identify student needs relating to a tangible teachable robot.

2 Background

There are many platforms that support learning through human-robot interaction. For example, as part of turtle geometry, students programmed a robotic logo turtle to turn and move certain distances [4]. Other platforms allow students to build their own robots, such as Lego Mindstorms, where students attach motors and sensors to programmable bricks [5]. There is evidence that these activities are indeed successful at improving programming and robotics skills [6]. Whether mathematics and science outcomes are directly improved is less clear, with only ten quantitative evaluations of learning from robotics programs yielding mixed results [7]. As with other forms of inquiry learning, learning from robotics may require guidance, access to positive examples, and self-reflection [6]. Personalized learning techniques may improve robotic learning activities [8, 9], but this combination is mostly unexplored.

We combine teachable agent paradigms with learning from robotics. Peer tutoring literature suggests that students can learn by tutoring because they pay more attention to the material, reflect on misconceptions, and elaborate their knowledge when they construct explanations [10]. Following up on human-human results, some developers have designed educational technology systems so that students *teach* an agent about the subject they are learning [11]. As in peer-to-peer tutoring, peer-to-agent tutors benefit cognitively as they watch their teachable agents solve problems, noticing their misconceptions and elaborating on their knowledge [12]. Another large part of learning from teaching is motivational. Students feel responsible for their students, and as a result try harder to understand the material [13].

In the combination of teachable agents and robotic learning systems, there are several design directions that often conflict. Teachable agent systems have been mostly designed for individual learners on personal computers. They model student cognition by tracking how students teach the agent, and attempt to provide enough social presence to engender feeling of rapport and accountability. In teachable agent environments, the designer determines the learning objectives, and provides students with adequate scaffolding in achieving those objectives. There are several open questions in moving teachable agent paradigms to a tangible space. How does the physical space affect student interactions with the teachable robot? Are the same kinds of learning objectives and scaffolds appropriate?

There are many potential design directions in creating a teachable robot with cognitive and motivational scaffolding, and thus it is important to take a student-centered approach [14]. While students cannot necessarily tell us what leads them to learn, they can tell us what engages them during learning. These insights, in conjunction with scientific knowledge about learning and motivation, can inform critical design decisions in complex learning environments. In our work, we employ a modified version of an HCI method called Speed Dating, where the design team rapidly explores divergent design concepts in order to identify needs that users perceive in themselves [15]. Speed Dating has two phases. The first phase, need validation, involves presenting small groups of target users with several storyboards, and soliciting reactions. In the second phase, users role play particular scenarios in order to make abstract elements of their interaction with the system concrete. This approach places focus on how users feel a particular experience matches their needs.

3 Speed Dating Method

Our first step was to generate several concepts for tangible robotics activities that incorporate user needs. We constrained our ideation in two ways. First, we chose middle and high school geometry as our learning domain, with sample tasks ranging from plotting ordered pairs to proving two triangles are similar. Second, we chose an *iRobot Create* as the central piece of technology to use. Off the shelf, the *iRobot Create* can run simple programs that allow it to move and turn. As part of brainstorming, we relaxed most technological constraints on the robot. We assumed that users could interact with the robot using gestures or speech, and that the users and robot could interact with projected figures on the floor or walls of the learning environment. We named the robot “Rover.” We generated 24 scenarios spanning concepts we were interested in exploring. Each scenario had three storyboard panels, read in sequence from left to right, with explanatory captions. Figure 1 is a scenario where a hidden shape is revealed once students solve a problem in a physical space.

We brought students into the lab in groups for need validation sessions lasting two hours. Sessions consisted of four alternating periods of user-centered design and brainstorming. For the user-centered phases, we presented students with each sketch, and asked a discussion question. Once students were done discussing a sketch, we presented them with the next sketch. Students saw an average of six sketches in sequence prior to moving on to brainstorming. In brainstorming, some participants found it natural to sketch their ideas; for others participants simply discussed their ideas as a group. Once participants stopped generating ideas, we moved on to the next user-centered phase. We had three groups with a total of 11 participants. Participants were between 13 and 16 years old, and all but two participants had already taken geometry. Participants were split into three groups of 3, 3, and 5 participants. Students within a group knew each other. We audiotaped the sessions, and retained sketches students generated as data. In our analysis, we looked for strong reactions to elements of scenarios, as per need validation methodology. We also looked for links between what students said during brainstorming and their strong reactions.

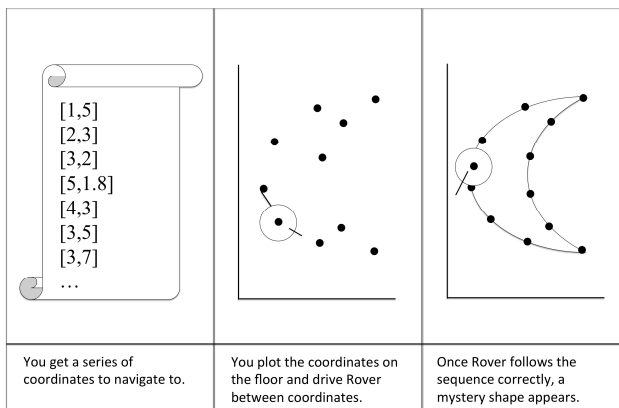


Fig. 1. Scenario prompting the need of discovery. The robot’s name is Rover.

For user enactments, we brought students into the lab in groups for sessions lasting one and a half hours. We explained the learning activity to students, and had them collaboratively work through problems they would be expected to teach the robot. Then we had students assume various roles: one student played the role of the robot, one played the role of the peer tutor, one played the role of the classroom teacher, and one acted as a helper to the peer tutor. Initially, we asked the peer tutor to teach the robot a particular concept, with the help of the classroom teacher. After, we gave students particular scenarios used in the need validation, and asked students to act them out. Students drew on paper on the floor to better simulate an embodied geometry environment. We had three groups with a total of 10 participants. Participants were between 11 and 14 years old, and most were taking middle school math. We videotaped the sessions, and retained scrap paper students used. In our analysis of the results of the design activity, we looked for mechanisms of interaction between students and the robot, paying attention to social and cognitive features.

4 Analysis of User Needs

One of the motivational elements of these environments we intended to probe was student feeling of *challenge*. Students had strong negative reactions to scenarios that supplied little support. They complained about doing work without perceiving the value, “*Hmm, this [has] happened to me... I did all of this, and I have to figure out where I went wrong*” (P6). Students also reacted to too much feedback. They commented: “*I don’t think Rover should tell them what they did, because, they have to, like, figure it out.*” (P9). Student comments focused on the motivational elements of challenge rather than on cognitive ones. Their resistance to feeling stuck and desire to have the solution within reach came up quite a bit: “*I feel like kids would be more prone to trying to figure it out if it were almost there...*” (P2). In user enactments, these themes reappeared. When the “robot” made careless errors the student could easily correct, such as forgetting to add a side when calculating the perimeter, peer tutors became excited and explained to the robot what to do.

The need of *discovery* resonated with students. In discovery, something previously hidden was revealed as part of learning activities. We illustrate this finding with student reactions to a connect-the-dots scenario, which was designed to prime discovery (see Figure 2). When the figure was revealed, P3 stated “*This one’s fun, with shapes... I would want to know what the dots would actually mean, like, the mystery factor.*” Part of the appeal of this need seemed to be the surprise and curiosity provoked by adding simple elements of interactivity into the learning environment, which we had not anticipated. When discussing the potential for projected geometric figures, one student said, “*You’d probably get color too... graph paper is boring. If it’s projected, you can try to make it fun*” (P8).

While we incorporated instructional principles into our scenarios that tapped in to students interacting in a physical space, we did not expect students to respond so strongly to physical motion. *Physicality* was a need that students identified, where the enjoyment students predicted over physical motion occurred across several scenarios. Students said moving around was useful for engagement, to break up the monotony of class interaction. P5 said, when talking about the sketches in general: “*We’re at*

school 7 hours a day, sitting in the classroom with, like, off-gray walls... it's like a prison... You get to like jump up and move around, because we sit down all the time... that's like great for your mind." Students also emphasized the importance of physical space for being able to visualize certain geometric concepts. The importance of physicality was further expressed in the ways students interacted during the user enactments. Students worked naturally around the same large physical canvas, turn-taking and grounding using pointing and other gestures. To get new perspectives, students would move to different positions around the canvas. All students were involved and attentive throughout the whole process. The nature of the interaction was qualitatively different than it would have been around a personal computer.

Responsibility for the agent was another theme that was brought up repeatedly. Students were excited by the idea that they could interact in pet-like ways with the robot. They responded enthusiastically to scenarios where Rover showed emotion, *"That's cute! You would be like, aww (P3)"*, and then expanded their brainstorming to further personify the robots, *"the whole concept of the dog is really appealing, you could make little clothes for it, they could be the antennas, if it were the dog, you could have it be the ears"*. While we had thought that most collaborative learning activities would be motivating to students, the ideas that resonated the most were the ones that specifically involved intergroup competition. A sketch that got one of the most positive reactions across all groups was one where groups would teach their robot different shapes, and then the robots would face off to see who could draw the most shapes. In reference to the sketch, P5 said, *"That's cool that different ones would face off, I like that"*, with P3 replying *"It would get everyone really excited"*. In many cases, students suggested similar ideas prior to seeing that sketch (*"...seeing a debate scene, both trying to get the answer right, one sends their team to have it do one thing, the other sends their Rover to do the other;"* P4).

5 Design Directions

In this paper, we employed the user-centered design method of Speed Dating as a way of making principled choices in the design of a teachable robot. Themes of challenge, discovery, physicality, and responsibility for the robot emerged. Creating a teachable agent that students can interact with in a physical space necessitates changes in teachable agent design. Students emphasized that they valued the physicality of interacting with a robot, in particular focusing on activities as simple as being able to map geometric concepts to physical motion. As we saw in the user enactments, properties of the physical learning environment facilitated students in accessing a shared workspace, changing location often, and working together. It also changed the nature of student responsibility for the robot: Students conceptualized the robot as being owned by a group. This shift towards a collaborative teachable agent paradigm presents particular modeling challenges, as it is more difficult to assess problem-solving and collaborative interactions in a physical space, rather than a digital one.

Our work guides decisions about the learning objectives and scaffolds that are appropriate in a tangible robot. Interactions in tangible environments are difficult to assess, and students are given freedom in defining learning objectives and pursuing their own goals. Our design results suggest that when we do define objectives, it will

be important to pay attention to the motivational elements of providing students with challenge. Allowing students to define their own agendas for what to teach the robot, providing them with suggestions for what to teach within their abilities, and encouraging them to challenge themselves at appropriate moments may be appropriate directions to explore in teachable robots. For example, having the robot make errors that students easily notice and correct may, when necessary, boost their confidence. We further found that the need of discovery resonated with students. Building scenarios where students discover aspects of robot behavior or physical space might engage students cognitively with the robot. In many ways, our results mirror ideas from the broader literature. However, a literature review of teachable agents offers several potential design directions, with little guidance for which ones are appropriate in a given scenario. User-centered design early in the construction of learning environments can help researchers attack difficult design problems.

Acknowledgements. This research was funded under a Computing Innovations Fellowship, NSF 1019343. Thanks to Ruth Wylie and the GALLAG group at ASU.

References

1. Ishii, H., Ullmer, B.: Tangible Bits: Towards seamless interfaces between people, bits and atoms. In: Proceedings of CHI 2007, pp. 234–241. ACM Press (1997)
2. Price, S.: A representation approach to conceptualizing tangible learning environments. In: Proc. 2nd International Conference on Tangible and Embedded Interaction, pp. 151–158. ACM Press (2008)
3. de Jong, T., van Joolingen, W.R.: Scientific discovery learning with computer simulations of conceptual domains. Review of Educational Research 68, 179–201 (1998)
4. Papert, S.: Mindstorms: children, computers, and powerful ideas. Basic Books, New York (1999)
5. Martin, F., Mikhak, B., Resnick, M., Silverman, B., Berg, R.: To mindstorms and beyond: evolution of a construction kit for magical machines. In: Robots for Kids: Exploring New Technologies for Learning. Morgan Kaufmann, San Francisco (2000)
6. Petre, M., Price, B.: Using Robotics to Motivate ‘Back Door’ Learning. Education and Information Technologies 9(2), 147–158 (2004)
7. Benitti, F.B.V.: Exploring the educational potential of robotics in schools: A systematic review. Computers & Education 58(3), 978–988 (2012)
8. Kanda, T., Hirano, T., Eaton, D., Ishiguro, H.: Interactive Robots as Social Partners and Peer Tutors for Children: A Field Trial. Human-Computer Interaction 19, 61–84 (2004)
9. Han, J.-H., Jo, M.-H., Jones, V., Jo, J.-H.: Comparative study on the educational use of home robots for children. Journal of Information Processing Systems 4(4), 159–168 (2008)
10. Roscoe, R.D., Chi, M.: Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors’ explanations and questions. Review of Educational Research 77(4), 534–574 (2007)
11. Matsuda, N., Keiser, V., Raizada, R., Tu, A., Stylianides, G., Cohen, W.W., Koedinger, K.R.: Learning by Teaching SimStudent: Technical Accomplishments and an Initial Use with Students. In: Alevan, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6094, pp. 317–326. Springer, Heidelberg (2010)

12. Leelawong, K., Biswas, G.: Designing learning by teaching agents: The Betty's Brain System. *International Journal of Artificial Intelligence in Education* 18(3), 181–208 (2008)
13. Chase, C., Chin, D., Oppezzo, M., Schwartz, D.: Teachableagents and the protégé effect: Increasing the effort towards learning. *Journal of Science Education and Technology* 18(4), 334–352 (2009)
14. Luckin, R., Underwood, J., du Boulay, B., Holmberg, B., Kerawalla, J., O'Connor, J., Smith, H., Tunley, H.: Designing educational systems fit for use: A case study in the application of human-centred design for AIED. *IJAIED* 16, 353–380 (2006)
15. Davidoff, S., Lee, M.K., Dey, A.K., Zimmerman, J.: Rapidly Exploring Application Design Through Speed Dating. In: Krumm, J., Abowd, G.D., Seneviratne, A., Strang, T. (eds.) *UbiComp 2007*. LNCS, vol. 4717, pp. 429–446. Springer, Heidelberg (2007)

An Intelligent Tutoring and Interactive Simulation Environment for Physics Learning

Lakshman S. Myneni and N. Hari Narayanan

Intelligent & Interactive Systems Research Laboratory, Computer Science
and Software Engineering Dept., Auburn University, Auburn, AL 36849, USA
{mynenls, naraynh}@auburn.edu

Abstract. This paper presents a learning environment called the Virtual Physics System (ViPS) that helps students learn physics concepts in the context of pulleys, a class of simple machines that are difficult to construct and experiment with in the real world. ViPS is novel in that it combines simulation and tutoring, identifies student misconceptions, customizes tutoring accordingly, and employs a pedagogical strategy of guiding students in problem solving through construction and simulation of pulley setups. An evaluation study showed that ViPS is effective in helping students learn and overcome their misconceptions.

Keywords: Intelligent Tutoring System, Physics Learning, Physics Simulation.

1 Introduction

Tutoring is known to improve student learning. When a human tutor is not available, the next best option maybe an Intelligent Tutoring System [e.g., 3, 5]. Another highly beneficial learning activity is problem solving through experimentation. It is a hands-on activity that involves designing and building an experimental setup, letting it perform its function, and collecting data from it in order to solve a problem and to better understand the underlying phenomena, or to test a scientific hypothesis. Computer modeling and simulation often take the place of physical experimentation in this learning activity. Many researchers have described the affordances and limitations of problem solving using physical experimentation and computer simulations in science education research. Zacharia and Anderson [6] investigated the effects of interactive computer-based simulations, presented prior to inquiry-based laboratory experiments, on students' conceptual understanding of mechanics. They found that the use of simulations improved students' ability to generate predictions and explanations of the phenomena in the experiments. Finkelstein and coworkers [1] looked at how students learned about electrical circuits differently with simulated or physical circuits. They reported that students who used simulations scored better on an exam and were able to build physical circuits more quickly than students who used physical circuits.

Our research combines tutoring and simulation-based experimentation in a single learning environment, the Virtual Physics System (ViPS). ViPS is an intelligent tutor that provides guided tutoring to a student as he or she solves physics problems

involving pulleys. ViPS also allows the student to construct, simulate and collect data from various pulley setups. Furthermore, ViPS is designed to detect and help address six common student misconceptions regarding pulleys (Table 1), obtained from a physics education researcher with years of experience in the field.

Table 1. Different misconceptions addressed by ViPS

Misconception 1	The more pulleys there are in a setup, the easier it is to pull to lift a load.
Misconception 2	The longer the string in a pulley setup, the easier it is to pull to lift a load.
Misconception 3	Pulling upwards is harder than pulling downwards.
Misconception 4	Having more pulleys in a pulley setup reduces the amount of work.
Misconception 5	Size (radius) of pulleys in a pulley setup affects the amount of work.
Misconception 6	Improper understanding of force and work.

ViPS detects which of these misconceptions a student has by asking the student to solve a set of problems at the beginning. The problem solving requires answering questions about pulley setups after constructing and running them in the simulation environment. Based on this, ViPS constructs a student model. This model, continually updated throughout a student session, is used for generating additional problems for the student to experiment with, and for providing hints and other kinds of feedback based on the student's knowledge state. As far as we know, ViPS is the first learning environment in which an intelligent tutoring system is integrated with an interactive simulation environment specifically tailored to address student misconceptions.

2 ViPS Architecture

ViPS, implemented in Java, consists of a graphical user interface that manages interaction with students, a simulation module that simulates the virtual pulley setups built by students, a feedback module that generates appropriate messages for the student during simulation and problem solving, a knowledge evaluator that evaluates the knowledge of the student from various tests administered during a session, a tutor module that tutors the student for misconceptions, a student model that includes the history of student interactions and various measures of student performance, a domain knowledge model that represents domain knowledge, a database of problems, and a procedural knowledge model that represents student solution paths within individual problems. Due to the page limitations of a short paper, we describe only the graphical user interface, tutor module and simulation module here. The interested reader is referred to [2] for information on the other components.

The graphical user interface is divided into two main parts: a tabbed work area for creating pulley setups and solving problems and an object pallet for selecting the components required to create a pulley setup (see Figure 1). Students can create a pulley setup by dragging the required components from the object pallet on to the work area and clicking on the thread button. Students can also interactively manipulate various parameters of the components, like the size of a pulley, value of the load etc. A problem is given to the student in the form of textual and pictorial

representations. The student is asked to solve the problem by creating the setups required to answer the question, running the simulations and comparing the simulation outputs of the setups created. The problems in ViPS were designed and checked by experienced physics educators. Currently, ViPS contains ten problems per misconception (60 in total, with more to be added in future) in its database. A web-based interface is available to teachers and experts to add or modify problems. One reason ViPS poses problems to a student is to identify his/her misconceptions in order to address them through problem solving.

The simulation module is responsible for simulating the setups created by the student. In particular, it provides a platform for running simulations of setups that are difficult or impossible to create in the physical world, such as running a simulation with zero friction or running a simulation with quintuple pulleys. The outputs generated by the simulation include graphs and real time values of variables like force, work done, potential energy, friction and mechanical advantage (Figure 2).

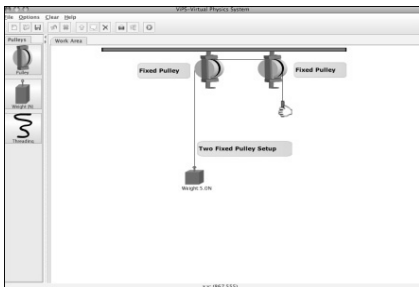


Fig. 1. ViPS Work Area

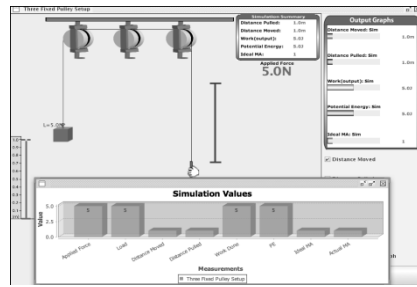


Fig. 2. ViPS Simulation Window

The student uses this module to simulate the different pulley setups he/she creates during problem solving. Domain knowledge regarding possible or valid pulley setups is represented in ViPS in the form of a Bayesian Belief Network. This network is used by ViPS to (1) find all possible setups that can be created using components that an individual student has assembled on the work area, (2) find components for creating a valid setup that are missing from the work area, and (3) generate dynamic hints regarding pulley setups to help the student. It is possible that the components assembled by the student do not lead to a unique pulley setup, and instead can be used to produce several possible setups. If this happens, ViPS infers and displays a list of possible setups based on the probabilities of creating each setup as determined by the Bayesian network, and ranked by an algorithm that we developed. This algorithm uses four attributes to rank order possible setups: (1) the number of components needed by a setup that are missing from the work area; (2) the number of grooves in each pulley in the setup; (3) the total number of components in the setup; and (4) the number of times this setup was created by the student previously. Then the student is asked about which of these setups most closely matches his or her intention. Based on the students' selection, the simulation module generates dynamic hints to guide the student towards the completion of the intended setup in the work area.

The Tutor module of ViPS employs the instructional technique of Coached Problem Solving [4]. It is responsible for overseeing the process of tutoring a student for the misconceptions he/she might have, and it is also responsible for overseeing the process of student problem solving by using the information generated by the student model to select and present appropriate problems. It uses a decision algorithm to determine the level of coaching to provide, and interfaces with the feedback module to generate appropriate hints. The interaction between the tutor module and a student begins with the student attempting a “pre knowledge test” evaluated by the knowledge evaluator. This test helps ViPS detect any misconceptions the student might have about pulley systems at the outset. After detecting and recording misconceptions that are present, the tutor module helps the student resolve these misconceptions by asking them to solve problems related to each detected misconception. Depending on whether the student solves these problems correctly (or not), tutoring for that misconception is not (is) provided, as explained below. If the student doesn’t exhibit any misconception at the outset, no problems or tutoring will be given.

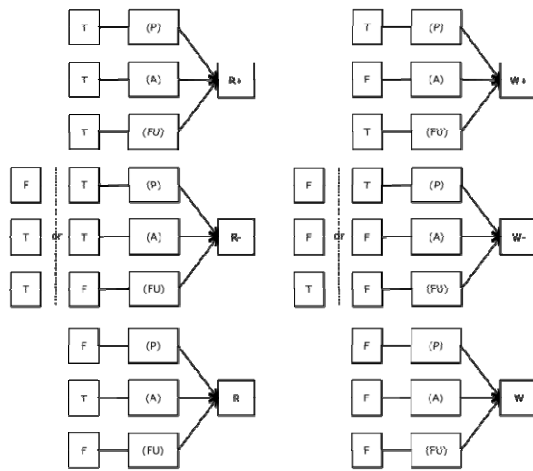


Fig. 3. Student problem solving performance classification

Table 2. Tutor action decision table

Problem 1	Problem 2	Problem 3	Action
T(R+, R-, R)	T(R+, R-, R)	N/A	Go to next misconception
T(R+, R-, R)	F(W+, W-, W)	T(R+, R-, R)	Go to next misconception
T(R+, R-, R)	F(W+, W-, W)	F(W+, W-, W)	Tutor for current misconception
F(W+, W-, W)	T(R+, R-, R)	T(R+, R-, R)	Go to next misconception
F(W+, W-, W)	T(R+, R-, R)	F(W+, W-, W)	Tutor for current misconception
F(W+, W-, W)	F(W+, W-, W)	N/A	Tutor for current misconception

For each misconception detected by the pre knowledge test, the tutor’s decision as to whether to tutor a student or not about that misconception depends on the student’s response to the problems specific to that misconception that he or she has been given to solve. For each problem, the student has to first enter a prediction (P), then his

answer (A) and finally answer a follow-up (FU) question. Based on these three answers, each of which could be correct (T) or wrong (F), the student's performance on the problem is classified into one of six categories R+, R, R-, W-, W, or W+ (see Figure 3). ViPS concludes that the student successfully solved a problem (marked T in Table 2) if the outcomes are R+, R-, or R, else it is concluded that the student failed to solve the problem (marked F in Table 2). The tutor module presents two problems per misconception, and a third problem depending on the outcomes of the first two problems, to verify whether a student indeed has that particular misconception detected from the pre knowledge test. The problem solutions are used to decide whether to tutor the student for that misconception, or move on to address the next misconception of the student using another set of three problems, as shown in Table 2. If the student solves the first two problems correctly, then she is determined not to have the corresponding misconception, so the tutor will move on to the next misconception (Table 2, row 1). If the first problem is not correctly solved, the system will present the student with a second problem. If its solution is incorrect as well, the student will be tutored for that misconception (Table 2, row 6). If she solves the first problem correctly but errs in the second one (or vice versa), the tutor will present a third problem, and depending on its outcome will either move to the next misconception or start tutoring actions to clear the current misconception (Table 2, rows 2-5). Tutoring actions consist of spoken (by an avatar) and written textual explanations and pictures.

3 Evaluation of ViPS

Evaluation focused on two questions. Does ViPS help students learn and clear their misconceptions? Is working with ViPS more effective than working with real pulleys? Fifty seven students (engineering majors from one university and pre-service elementary teachers from a second university) were assigned to one experimental condition: the ViPS group, in which participants took a pre-test, worked with ViPS individually and then took a post-test. One hundred and fifty eight pre-service elementary teachers from the second university were randomly assigned to two additional experimental conditions: (1) the Physical-Virtual (PV) group in which participants took a pre-test, worked in pairs with physical pulleys, then took a mid-test, next worked in pairs with ViPS, and finally took a post-test, and (2) the Virtual-Physical group (VP) in which participants took a pre-test, worked in pairs with ViPS, then took a mid-test, next worked in pairs with physical pulleys, and finally took a post-test.

A paired-sample t-test was performed on the pre-and-post test scores of students in the ViPS group ($n=57$) to evaluate their learning gain after using ViPS. There was a score increase from pre-test to post-test with statistical significance ($t(56)=-17.66$, $p=0.001$). Scores increased by 300% from an average of 4.57 to 13.71 (max score = 18). Clearly, ViPS is effective in teaching students. Linear regression found a significant positive correlation ($N=57$, $R=0.756$, $R^2=0.571$, $p=0.03$, Standardized Beta=0.792) between learning gain and number of problems solved by the ViPS group. On average, each student solved eight problems. Linear regression also found a positive but non-significant correlation between learning gain and number of simulations created ($N=57$, $R=0.039$, $R^2=0.002$, $p=0.83$). On average, students created and ran 14 simulations. A repeated measures mixed analysis of variance test was performed on pre-test to mid-test scores of the VP Group and the PV Group (158 students or 79

pairs in both groups solved problems related to the same misconception, but the VP group used ViPS between the pre- and mid-tests, whereas the PV group used actual pulleys) to compare their learning gains. Results showed that the learning gain was higher for the VP group that used ViPS, with statistical significance ($F(1,156)=4.54$, $p=0.035$, $\eta^2=0.28$, and $\text{power}=0.563$). Thus, students learned more from ViPS than from physical pulleys.

The most common misconception among students was Misconception 2 (Table 1), followed by Misconceptions 1 and 4. Sixty students exhibited all the six misconceptions. A paired-sample t-test was conducted to compare the number of misconceptions identified in the pre-test and post-test in the ViPS group. There was a significant reduction in number of misconceptions from pre-test to post-test ($t(54)=16.6$, $p=0.001$). On average, each student exhibited five misconceptions after pre-test and two misconceptions after post-test. The number of misconceptions decreased significantly after working with ViPS. These results indicate that ViPS is effective.

4 Conclusion

The contribution of this research is an intelligent simulation and tutoring system called ViPS for learning physics concepts through simulating and getting tutored on a class of simple machines, which has several features that together make it unique. It employs the Coached Problem Solving approach to detect and effectively tutor for common student misconceptions in physics. It is able to dynamically infer valid pulley setups from the components that a student selects and places on the workspace, and to adaptively generate hints based on student actions. It is a tool for creating, exploring and simulating pulley setups that are difficult to construct and manipulate in the physical world. The interface of ViPS is designed to help students connect abstract concepts of physics with tangible pictorial representations. ViPS integrates virtual experimentation through simulation with intelligent tutoring. An evaluation of ViPS with over 200 students showed that it was effective in helping students learn and clear their misconceptions, and more beneficial than working with real pulleys.

References

1. Finkelstein, N.D., Adams, W.K., Keller, C.J., Kohl, P.B., Perkins, K.K., Podolefsky, N.S.: When learning about the real world is better done virtually: A study of substituting computer simulations for laboratory equipment. *Physical Review Special Topics – Physics Education Research* 1(1), 010103 (2005)
2. Myneni, L.S.: An Intelligent and Interactive Simulation and Tutoring Environment for Exploring and Learning Simple Machines. Doctoral Dissertation, Auburn University (2011)
3. Ritter, S., Anderson, J., Koedinger, K., Corbett, A.: The Cognitive Tutor: Applied research in mathematics education. *Psychonomics Bulletin & Review* 14(2), 249–255 (2007)
4. VanLehn, K.: Conceptual and Meta Learning During Coached Problem Solving. In: Lesgold, A.M., Frasson, C., Gauthier, G. (eds.) *ITS 1996*. LNCS, vol. 1086, pp. 29–47. Springer, Heidelberg (1996)
5. Woolf, B.P.: Building Intelligent Interactive Tutors: Student-centered Strategies for Revolutionizing E-learning. Morgan Kaufmann (2008)
6. Zacharia, Z., Anderson, O.R.: The effects of an interactive computer-based simulation prior to performing a laboratory inquiry-based experiment on students' conceptual understanding of physics. *American Journal of Physics* 71(6), 618–629 (2003)

Guru: A Computer Tutor That Models Expert Human Tutors

Andrew M. Olney¹, Sidney D'Mello², Natalie Person³, Whitney Cade¹, Patrick Hays¹,
Claire Williams¹, Blair Lehman¹, and Arthur Graesser¹

¹ University of Memphis

[aolney, wlcade, dphays, mcwilliams, balehman, a-graesser]@memphis.edu

² University of Notre Dame

sdmello@nd.edu

³ Rhodes College

person@rhodes.edu

Abstract. We present *Guru*, an intelligent tutoring system for high school biology that has conversations with students, gestures and points to virtual instructional materials, and presents exercises for extended practice. *Guru*'s instructional strategies are modeled after expert tutors and focus on brief interactive lectures followed by rounds of scaffolding as well as summarizing, concept mapping, and Cloze tasks. This paper describes the *Guru* session and presents learning outcomes from an in-school study comparing *Guru*, human tutoring, and classroom instruction. Results indicated significant learning gains for students in the *Guru* and human tutoring conditions compared to classroom controls.

Keywords: intelligent tutoring system, expert tutor, biology, conversation.

1 Introduction

Guru is a dialogue-based intelligent tutoring system (ITS) in which an animated tutor agent engages the student in a collaborative conversation that references a multimedia workspace displaying and animating images that are relevant to the conversation. *Guru* provides short lectures on difficult biology topics, models concepts, and asks probing questions. *Guru* analyzes typed student responses via natural language understanding techniques and provides formative feedback, tailoring the session to individual students' knowledge levels. At other points in the session, students produce summaries, complete concept maps, and perform Cloze tasks. To our knowledge, *Guru* is the first ITS that covers an entire high school biology course.

Guru is distinct from most dialogue-based ITSs, such as AutoTutor [1] or Why-Atlas [2], because it is modeled after 50-hours of *expert* human tutor observations that reveal markedly different pedagogical strategies from previously observed novice tutors [3]. Our computational models of expert tutoring are multi-scale, from tutorial modes (e.g. scaffolding), to collaborative patterns of dialogue moves (e.g. information-elicitation), to individual moves (e.g. direct instruction) [4]. However, the

importance of tutoring expertise has recently been called into question. In a meta-analysis, VanLehn [5] examined the effectiveness of step-based ITSs and human tutoring compared to no tutoring learning controls matched for content. He reported that the effect sizes of human tutoring are not as large as Bloom's two sigma effect [6]. Instead, the effect sizes for human tutoring are much lower ($d = .79$), and step-based systems ($d = .76$) are comparable to human tutoring. Even so, the *relative* influence of expertise on learning outcomes remains unclear and requires more research.

The present study addresses the effectiveness of Guru in promoting learning gains. Specifically, how do learning gains obtained from classroom instruction + Guru compare to classroom + human tutoring and classroom instruction alone? We begin with a sketch of Guru followed by an experiment designed to evaluate the effectiveness of Guru in an authentic learning context, namely an urban high school in the U.S.

2 Brief Description of Guru

Guru covers 120 biology topics aligned with the Tennessee Biology I Curriculum Standards, each taking from 15 to 40 minutes to cover. Topics are organized around *concepts*, e.g. *proteins help cells regulate functions*. Guru attempts to get students to articulate each concept over the course of the session. In this study, a Guru session is ordered in phases: Preview, Lecture, Summary, Concept Maps I, Scaffolding I, Concept Maps II, Scaffolding II, and Cloze Task. Guru begins with a **Preview** making the topic concrete and relevant to the student, e.g. "Proteins do lots of different things in our bodies. In fact, most of your body is made out of proteins!" Guru's **Lectures** have a 3:1 (Tutor:Student) turn ratio [4, 7] in which the tutor asks concept completion questions (e.g., Enzymes are a type of what?), verification questions (e.g., Is connective tissue made up of proteins?), or comprehension gauging questions (e.g., Is this making sense so far?). At the end of the lectures, students generate **Summaries**; summary quality determines the concepts to target in the remainder of the session. For target concepts, students complete skeleton **Concept Maps** generated from concept text [8]. In **Scaffolding**, Guru uses a Direct Instruction → Prompt → Feedback → Verification Question → Feedback dialogue cycle to cover target concepts. A **Cloze** task requiring students to fill in an ideal summary ends the session.

Guru's interface (see Figure 1) consists of a multimedia panel, a 3D animated agent, and a response box. The agent speaks, gestures, and points using motion capture and animation. Throughout the dialogue, the tutor gestures and points to images on the multimedia panel most relevant to the discussion, and images are slowly revealed as the dialogue advances. Student typed input is mapped to a speech act category (e.g., Answer, Question, Affirmative, etc.) using regular expressions and a decision tree learned from a labeled tutoring corpus [9,10]. Guru uses speech act category and multiple models of dialogue context to decide what to do next. Thus an affirmative in the context of a verification question is interpreted as an Answer, while an affirmative in the context of a statement like "Are you ready to begin?" is not. Guru uses a general model of dialogue (e.g., feedback, questions, and motivational dialogue) and specific models representing the *mode* of the tutoring session, including

Lecture and Scaffolding. The mode models contain specific logic for answer assessment, feedback delivery (positive, neutral, or negative), and student model maintenance consisting of the concepts associated with each topic. A full description of the system is beyond the scope of the current paper.

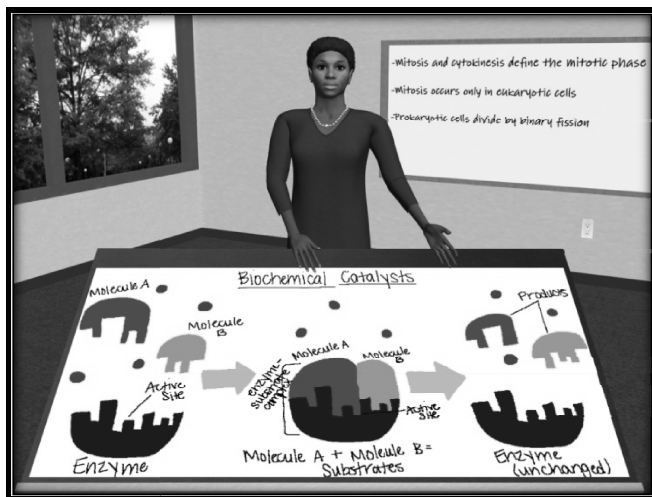


Fig. 1. Guru interface

3 Method

Thirty-two tenth graders enrolled in Biology I in an urban U.S. high school participated once a week for three weeks in a three condition repeated-measures study where students interacted with both Guru and a human tutor *in addition* to their regular classroom instruction. Tutored topics were covered in class in the previous week. Space limitations prevent listing the intricate details of the methods. What is important to note is that (1) there were four topics in the study (topics A: Biochemical Catalysts, B: Protein Function, C: Carbohydrate Function, D: Factors Affecting Enzyme Reactions), (2) students received classroom instruction on all four topics, (3) students received additional tutoring for two out of the four topics (A and B), (4) some students were tutored by Guru for topic A and a human tutor for topic B, whereas other students received Guru tutoring for topic B and human tutoring for topic A, (5) tutoring topic (e.g., A or B) was counterbalanced across Guru and the human tutor (6) all students completed pretests, immediate posttests, and delayed posttests on all topics. This design allowed us to (1) compare Guru with human tutoring (e.g., learning gains for topic A vs. B, where topic is counterbalanced across tutors), (2) compare learning gains from tutoring + classroom with learning gains from classroom instruction only (gains for A and B vs. C and D), and (3) assess if there are any benefits to classroom instruction alone (i.e., do learning gains for C and D exceed zero).

Knowledge assessments were multiple-choice tests; twelve item pre- and posttests were administered at the beginning and end of each tutoring session to assess prior knowledge and *immediate learning gains*, respectively. Test items were randomized across pre- and posttests, and the order of presentation for individual questions was randomized across students. Students also completed a 48-item *delayed* posttest the final week. Half of test items were previously used on the immediate pre or posttests, and half were new, with randomized order across students. The researcher who prepared the knowledge tests had access to the topics, the concepts for each topic, the biology textbook, and existing standardized test items. Content from the lectures, scaffolding moves, and other aspects of Guru were *not* made available to the researcher. The researcher was also blind to the tutored condition.

Students and parents provided consent prior to the start of the experiment. Students were tested and tutored in groups of two to four. The procedure for each tutorial session involved (a) students completing the pretest for 10 minutes (b) a tutorial session with either Guru or the human tutor for 35 minutes, and (c) the immediate posttest for 10 minutes. The four human tutors were provided with the topic to be tutored, the list of concepts, and the biology textbook. Each tutor was an undergraduate major or recent graduate in biology. Prior to the study, each tutor participated in a one day training session provided by a nonprofit agency that trains volunteer tutors for local schools. Thus while our tutors might be considered experts in the biology domain, they were not expert tutors.

4 Results

The pretest and immediate and delayed posttests were scored and proportionalized. A repeated measures ANOVA did not yield any significant differences on pretest scores, $F(2, 56) = 1.49, p = .233$, so students had comparable knowledge prior to tutoring. Separate proportionalized learning gains for immediate and delayed posttest were computed as follows: (proportion posttest - proportion pretest) / (1 - proportion pretest). This measure tracks the extent to which students acquire knowledge from pre to post. Two scores beyond 3.29 SD from the mean were removed as outliers.

A repeated measure ANOVA on proportional learning gains for the *immediate posttest* was significant, $F(2, 54) = 5.09, MSe = .212$, partial eta-square = .159, $p = .009$. Planned comparisons indicated that immediate learning gains for Guru ($M = .385, SD = .526$) and human tutoring ($M = .414, SD = .483$) did not differ from each other ($p = .846$) and were significantly ($p < .01$) greater than the classroom control ($M = .060, SD = .356$). The effect size (Cohen's d) for Guru vs. classroom was 0.72 sigma, while there was a 0.83 sigma effect for the human vs. classroom comparison.

This pattern of results was replicated for the *delayed posttest* (see Figure 2). The ANOVA yielded a significant model, $F(2, 54) = 5.80, MSe = .219$, partial eta-square = .177, $p = .005$. Learning gains for Guru ($M = .178, SD = .547$) and human tutoring ($M = .203, SD = .396$) were equivalent ($p = .860$) and significantly greater ($p < .01$) than the no-tutoring classroom control ($M = -.178, SD = .203$). The Guru vs. classroom effect size was 0.75 sigma, the human vs. classroom effect size was 0.97 sigma.

Paired samples t -tests indicated that learning gains on the delayed posttests were significantly lower ($p < .05$) than gains on the immediate posttests for all three conditions, which was expected. There was considerable learning on the delayed posttests for the Guru and human conditions, but not the classroom condition: one-sample t -tests indicated that proportional learning gains on the delayed posttests for Guru and human tutoring was significantly greater than 0 (zero is indicative of no learning) but was significantly *less* than zero for the classroom condition.

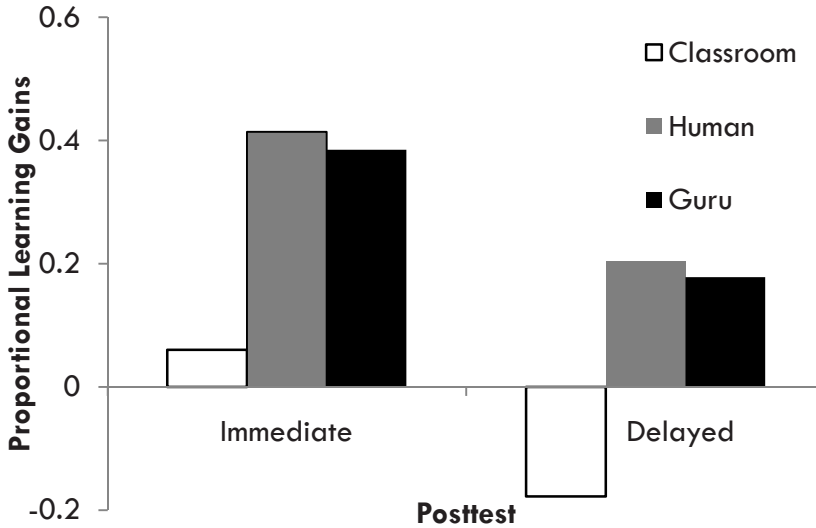


Fig. 2. Proportional learning gains

5 General Discussion

These results suggest that Guru is as effective as novice tutors and more effective than classroom instruction only. More importantly, the benefits of tutoring continue after a delay of one to two weeks. Although no differences between Guru and the human tutors were found, there were some limitations to this comparison. First, the human tutors were not able to work one-on-one with 32 students, and so they worked with two to four students simultaneously whereas students worked with Guru individually. However, prior work suggests that the group size may not have detracted from the human tutor condition: Bloom's 2 sigma effect was achieved with groups of 1-3 [6].

Another limitation is that the present human tutors do not meet the same criteria of expertise as the expert tutors on which Guru is modeled, e.g. licensed teachers with considerable tutoring experience (see [11]). Thus the lack of difference between Guru and human tutoring does not clarify Guru's effectiveness vis-à-vis expert human tutors. The .79 effect size for human tutoring reported by VanLehn [5] is highly comparable to the effect size of both Guru and human tutors in the present study, so it is unclear whether an expert tutor under these same conditions would generate

significantly greater learning gains. Nonetheless, we are very encouraged by these findings and have preliminary evidence of Guru's efficacy.

Acknowledgment. This research was supported by the National Science Foundation (NSF) (HCC 0834847 and DRL 1108845) and Institute of Education Sciences (IES), U.S. Department of Education (DoE), through Grant R305A080594. Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF, IES, or DoE.

References

1. Graesser, A.C., Lu, S.L., Jackson, G., Mitchell, H., Ventura, M., Olney, A.: AutoTutor: A tutor with dialogue in natural language. *Behavioral Research Methods, Instruments, and Computers* 36, 180–193 (2004)
2. VanLehn, K., Jordan, P.W., Penstein Rosé, C., Bhembé, D., Böttner, M., Gaydos, A., Makatchev, M., Pappuswamy, U., Ringenberg, M.A., Roque, A.C., Siler, S., Srivastava, R.: The Architecture of Why2-Atlas: A Coach for Qualitative Physics Essay Writing. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363, pp. 158–167. Springer, Heidelberg (2002)
3. Person, N.K., Lehman, B., Ozbun, R.: Pedagogical and Motivational Dialogue Moves Used by Expert Tutors. In: 17th Annual Meeting of the Society for Text and Discourse, Glasgow, Scotland (2007)
4. D'Mello, S.K., Olney, A.M., Person, N.K.: Mining collaborative patterns in tutorial dialogues. *Journal of Educational Data Mining* 2(1), 1–37 (2010)
5. VanLehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist* 46(4), 197–221 (2011)
6. Bloom, B.: The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher* 13(6), 4–16 (1984)
7. D'Mello, S., Hays, P., Williams, C., Cade, W., Brown, J., Olney, A.: Collaborative Lecturing by Human and Computer Tutors. In: Alevén, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6095, pp. 178–187. Springer, Heidelberg (2010)
8. Olney, A.M., Cade, W.L., Williams, C.: Generating Concept Map Exercises from Textbooks. In: Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 111–119. Association for Computational Linguistics, Portland (2011)
9. Olney, A.M.: GnuTutor: An Open Source Intelligent Tutoring System Based on AutoTutor. In: Proceeding of 2009 AAAI Fall Symposium on Cognitive and Metacognitive Educational Systems, pp. 70–75. AAAI Press (2009)
10. Razor, T., Olney, A.M., D'Mello, S.K.: Student Speech Act Classification Using Machine Learning. In: McCarthy, P.M., Murray, C. (eds.) Proceedings of 24rd Florida Artificial Intelligence Research Society Conference, pp. 275–280. AAAI Press, Menlo Park (2011)
11. Olney, A.M., Graesser, A.C., Person, N.K.: Tutorial Dialog in Natural Language. In: Nkambou, R., Bourdeau, J., Mizoguchi, R. (eds.) Advances in Intelligent Tutoring Systems. SCI, vol. 308, pp. 181–206. Springer, Heidelberg (2010)

Developing an Embodied Pedagogical Agent with and for Young People with Autism Spectrum Disorder

Beate Grawemeyer¹, Hilary Johnson¹,
Mark Brosnan², Emma Ashwin², and Laura Benton¹

¹Department of Computer Science, University of Bath, Bath BA2 7AY, UK
{b.grawemeyer, h.johnson, l.j.benton}@bath.ac.uk

²Department of Psychology, University of Bath, Bath BA2 7AY, UK
{m.j.brosnan, e.l.ashwin}@bath.ac.uk

Abstract. This paper describes how we developed an embodied pedagogical agent (EPA) with and for young people with autism spectrum disorder (ASD). ASD is characterised by impairments in social communication, imagination, and perspective-taking, which can compromise design and collaboration. However, if an ASD preference for visual processing can be supported by providing images of design ideas as they develop, these difficulties may be overcome. We describe a methodology that successfully supports the visualisation and development of EPAs using our prototype visualisation tool (EPA DK), enabling ASD users to function as active design participants.

Keywords: Embodied Pedagogical Agent, participatory design, autism spectrum disorder.

1 Introduction

Our aim is to include young people with ASD in the design and development process of educational software. To achieve this, particular impairments that are associated with ASD, including social communication, creativity, imagination, and perspective taking, need to be overcome. However, there can be enhanced visual processing abilities. It may be possible to support these preferences for visual material to allow young people with ASD to contribute effectively to design and collaboration processes by visually supporting the externalisation of ideas.

We have developed an intelligent tutoring system for mathematics, which includes an educational pedagogical agent (EPA). The benefits of EPAs have been widely documented. They can enhance motivation, understanding and attitudes in learners (e.g. [3, 5, 7]). However, this research was based upon a typically developing population and may not be generalised to users with ASD, given their social communication deficits. Therefore this paper outlines the process whereby a pedagogical agent was designed and developed with and for young people with ASD. We offer a contribution to methodology in this area: a simple tool which enables us to visualise, develop, and code EPAs dynamically, on-the-fly, in design sessions with the active participation of young people with ASD.

2 EPA Development Kit (EPA DK)

Yamamoto & Nakakoji [10] state that a design process involves the externalisation of partial solutions to a problem (for example, sketching an idea), which will be constantly revised in order to gain a better understanding of the problem, whilst aiming at a solution. Thus, a key issue is how do we develop appropriate means of supporting externalisation for ASD users.

A computer system can be seen as a tool able to define a user's externalisation space and the ways in which the user can interact with it [9]. Different types of software models and tools have been developed that are able to support ideas generation and collaboration. As described in [8] the success of such tools may depend upon their use. The tools encourage the externalisation of ideas as well as the manipulation and / or management of external representations.

For individuals with ASD, an effective tool for idea generation and integration needs to provide an externalisation space that is narrowed down and restricted to what is currently relevant in the design process, but still allows enough space for the integration and refinement of ideas. Current tools are unable to provide such an externalisation space as they are too generic. Thus, our EPA development toolkit (EPA DK) supports the process of both developing and visualising EPAs. The tool specifically supports the process of externalising and refining ideas, by transforming EPA sketches into a functional prototype directly, on-the-fly. It also provides a means of demonstrating different layout and EPA feedback / interaction options; and of changing an EPA's appearance using different media, such as screen printouts, and / or a software drawing package.

In order to investigate how the ASD preference for visual processing can be used to overcome difficulties in imagination and social communication, we applied the EPA DK tool in our system design and development process.

3 EPA Design and Development Process

The design process adopted Druin's [4] and Guha's et al. [6] work. It includes a three-stage participatory design process of individual idea generation, mixing of ideas, and integration into a 'big idea'. We complement this with on-the-fly rapid prototyping facilitated by EPA DK in the design and development sessions.

3.1 Study Aims

In order to involve young people with ASD as active participants, the study investigated whether difficulties in imagination [2], social communication and collaboration [1], could be overcome by supporting the externalisation of ideas to help make things concrete and also by providing a foundation for visually processing ideas of others.

3.2 Design Teams

For our design sessions, six high-functioning young people with ASD (all male, 11-15 of age) were divided into two groups. Each session included three young people with

ASD; a specialist ASD teaching assistant; and three researchers, who took different roles, including facilitator, designer, and note-taking observer. The studies took place at the school to provide an environment that was familiar to participants.

3.3 Procedure

Idea Generation and Mixing of Individual Ideas. Participants in each design group were asked to individually design an EPA for a mathematics tutor using different coloured pens, pencils, and blank A4 paper sheets. Participants were instructed that the role of the character was to encourage the student to perform certain exercises and would give feedback on answers. Further, the character's appearance and interaction could be decided by the participants, including its different emotional responses.

At the end of the individual idea session, participants were asked to explain each idea to other group members. This was followed by combining the individual ideas into one group idea, together with a drawing of this group idea on paper.

Big Idea. The next part of the EPA design process involved combining the two group ideas into a 'big EPA idea'. Here, all six participants were instructed to generate a 'big and even better' EPA design idea. A group spokesperson explained his group's idea to members of the other group, while a researcher noted the main features of the EPA design on a whiteboard. A mark was placed on features that were particularly liked by participants from the other group. Participants then decided on a 'big EPA idea' that conjoined the 'best' and 'most liked' features of the two group designs. This was followed by building the EPA idea using art materials.

EPA DK. All six participants were involved in a day-long prototyping session, where the EPA design was further refined using the EPA DK. The session was divided into three phases. The externalisation space given to participants was specifically tailored to particular design tasks, which changed and built up across the three phases.

The **first phase** demonstrated an idea using EPA DK, looking at the effect of transferring an idea into a concrete prototype, including different interaction options. Participants' feedback was used to change the prototype on-the-fly, with the resultant EPA prototype including only preferred EPA responses. The externalisation space was narrowed down to the specific idea, which placed the idea into its relevant context (an EPA prototype) and allowed its visual exploration.

The **second phase** investigated how the process of refining an existing idea could be visually supported. An electronic version of an EPA idea was given to participants that showed an external representation of an idea that could be refined. Participants were asked to change the EPA as they preferred, and the resulting image was then uploaded into EPA DK.

In the **third phase**, screen printouts of the EPA prototype were given to participants, as a medium where new ideas could be integrated. Participants were asked to externalise ideas for the verbal feedback a character could give for a positive and negative response to maths questions. The screen printouts showed the existing EPA idea with empty speech bubbles, in order to encourage participants to externalise ideas about the EPA's feedback.

4 Results

4.1 Idea Generation and Mixing of Individual Ideas

Participants in both groups were able to generate individual ideas that were then combined and mixed into a group idea. Figure 1 shows the evolution of the EPA ideas from the individual to the group ideas (shown in bold frames). One participant in the second group was unable to attend this session, hence only 2 individual ideas are shown in the second group.



Fig. 1. Examples of participants' EPA ideas and combined group ideas (bold frame)

The first group decided on a car, where you could see two characters from the back. Instead of showing emotions through facial expressions, the characters (shown from the back) would have a conversation about the student's progress on learning performance.

The idea of the second group included a 'pac man' character, which would dance, smile and jump when getting answers right. Emotions were expressed through the character changing colour: for example, yellow to express happiness, blue for sadness, or orange for pride.

4.2 Big Idea

The groups met to discuss the 'big EPA idea'. It was decided amongst participants that the EPA design should include characters that were sitting in a car. The characters would change colour to express emotions. Using art materials, participants then undertook different roles in building certain parts of the big idea. However, participants focussed on their original individual ideas from the previous sessions without actually integrating them: instead of building an EPA design based on a combination of their group ideas, participants referred back to their own individual ideas.

4.3 EPA DK

In this session the EPA DK tool was used to both visualise and develop the EPA idea. As a basis, we used the central idea from the 'big EPA ideas' session - the car design - with two characters sitting in the front seats, from a back-seat passenger's perspective. The EPA prototype was shown to participants and different feedback options and interaction styles were demonstrated. Participants expressed preferred and non-preferred feedback / interaction options. Non-preferred options were removed directly on-the-fly during design session.

Participants were then asked to refine the EPA design according to their wishes, providing an electronic version of the external representation of the EPA shown in the prototype. The refined design was then included within the EPA prototype and demonstrated to participants.

Participants were finally asked to develop ideas for the character's verbal interaction using screen printouts showing an EPA design. Figure 2 shows examples of participants' ideas for the character's responses. Interestingly, in contrast to the study described above (Section 4.2), participants were not only able to externalise and integrate new ideas, but to integrate their individual EPA idea from previous sessions.



Fig. 2. Examples of participants' ideas for verbal agent interaction

5 Discussion

Contrary to impairments in autism in imagination [2], participants were able to express and externalise their individual ideas for an EPA and to mix their individual ideas within a small group. We need to investigate further whether participants were able to mix their individual ideas within the smaller groups based on the ability to look at the other participant's drawings. Transforming an EPA idea into a concrete prototype enabled participants to visually explore different designs. By narrowing down the externalisation space, participants were able to visualise an idea (which might be someone else's idea).

The 'big ideas' session showed that when participants were asked to externalise and build an idea that was based on a combination of both group ideas, they reverted back to their individual ideas. This result supports the theories of autism outlining difficulties in social communication and collaboration [1]. However, these problems may be overcome if the externalisation space restricts participants to a specific collaborative issue or provides opportunities for adding further detail collaboratively.

6 Conclusion and Future Work

It is important to include users in the design of software, especially if the software is targeted at a special needs user group.

ASD is associated with social communication difficulties and imagination deficits, which may relate to problems in imagining the ideas of other participants. Those

difficulties can be overcome by using our computerized tool (EPA DK), which allows participants to view and experience different design ideas.

The next stage in the research agenda is to evaluate our intelligent tutoring system. This will include an assessment of the effectiveness, for engagement, motivation and learning, of the EPA design, created with and for young people with ASD.

Acknowledgements. We are especially grateful to the participants who willingly gave their time, and to their parents who gave consent for their children to take part in this study. The authors gratefully acknowledge Brislington Enterprise College (BEC) in Bristol (especially the ASD unit). The support of the Engineering and Physical Sciences Research Council (EPSRC, EP/G031975/1) is also gratefully acknowledged.

References

1. American Psychiatric Association: Diagnostic and Statistical Manual of Mental Disorders, 4th edn. American Psychiatric Publishing, Inc., Arlington (2000)
2. Craig, J., Baron-Cohen, S.: Creativity and imagination in autism and Asperger syndrome. *Journal of Autism and Developmental Disorders* 29(4), 319–326 (1999)
3. Dehn, D.M., van Mulken, S.: The impact of animated interface agents: a review of empirical research. *Int. J. Human-Computer Studies* 52, 1–22 (2000)
4. Druin, A.: Cooperative Inquiry: Developing New Technologies for Children with Children. In: *Proc. CHI 1999*, pp. 592–599. ACM Press (1999)
5. Girard, S., Johnson, H.: What Do Children Favor as Embodied Pedagogical Agents? In: Aleven, V., Kay, J., Mostow, J. (eds.) *ITS 2010. LNCS*, vol. 6094, pp. 307–316. Springer, Heidelberg (2010)
6. Guha, M.L., Druin, A., Chipman, G., Fails, J.A., Simms, S., Farber, A.: Mixing Ideas: A New Technique for Working with Young Children as Design Partners. In: *Proc. IDC 2004*, pp. 35–42. ACM Press (2004)
7. Gulz, A.: Benefits of Virtual Characters in Computer Based Learning Environments: Claims and Evidence. *Int. J. Artif. Intell. Ed.* 14(3), 313–334 (2004)
8. Johnson, H., Carruthers, L.: Supporting creative and reflective processes. *Int. J. Human-Computer Studies* 64, 998–1030 (2006)
9. Norman, D.: *Things That Make Us Smart*. Addison-Wesley (1993)
10. Yamamoto, Y., Nakakoji, K.: Interaction design of tools for fostering creativity in the early stages of information design. *Int. J. Human-Computer Studies* 63, 513–535 (2005)

WEBsistments: Enabling an Intelligent Tutoring System to Excel at Explaining Rather Than Coaching

Yue Gong, Joseph E. Beck, and Neil T. Heffernan

Computer Science Department, Worcester Polytechnic Institute
100 Institute Road, Worcester, MA, 01609, USA
{ygong, josephbeck, nth}@wpi.edu

Abstract. Most step-based Intelligent Tutoring Systems (ITS) are well suited for providing problem solving practice, and are well-tailored to help students solve specific items. Consequently, many ITS typically fail to perform as strong media for conveying conceptual and procedural *instruction*, rather than coaching. In order to overcome this deficiency, we leverage existing web-based resources, as many existing resources are well-designed for providing instruction. By combining external web pages with the ASSISTments tutoring system, we have created a stronger intervention that we have dubbed *WEBsistments*. A preliminary study found that students who were wrong on a problem and received a web page as assistance, improved more, relative to students who did not see a web page. In addition, our results suggest that weaker students seem to benefit more from using web pages as extra help.

Keywords: WEBsistments, Web-based Resources, Conceptual Instruction.

1 Introduction

After their first emergence over two decades ago, Intelligent Tutoring Systems (ITS) have attracted researchers from a variety of disciplines. Many research studies have been showing that ITSs resulted in substantial successes in improving student learning in different domains, such as mathematics [1], physics [2], and reading [3]. These systems' two major advantage over traditional classroom practicing is that students can get immediate feedback on correctness, and the ability to request help [4].

A common type of ITS is step-based [5]. Once a student enters a step, the tutor can provide feedback or help. Following this architecture, many ITS were designed to help students solve problems step by step. They generally provide several different forms of help, such as worked examples, hint messages and scaffolding questions (e.g. [1, 2]). Such assistance, independent of the many forms it takes on, is tactical with the goal of directing students to the solution for this problem. Thus the systems are well suited for coaching students. On the other hand, we notice that due to such design, the systems lack the ability to perform as a strong medium for conveying instruction for students who lack the background knowledge to benefit from coaching. We seek to address this problem of low-knowledge students not by authoring new

content, but by instead utilizing web-based resources which are already on the Internet. There are three reasons we see a benefit from this integration.

First, ITS are effective in assisting students with problem-solving practice. Web-based resources, however, are often not designed for a specific problem, but rather illustrate concepts, introduce vocabulary, and explain procedural solutions of a skill at a more general level. Including such material extends the repertoire of instruction that an ITS can deliver to students.

Second, web-based resources cheaply extend the range of media available for tutoring. Traditionally, for reasons of cost and expertise, much assistance in an ITS is text-based. Web-based resources are able to convey knowledge in a variety of modalities such as videos of a human teaching the skill that serve as a human tutor within a computer tutor, or animations that allow students to manipulate some components in order to teach students interactively. Intuitively, these new features could possibly help students learn by broadening the types of interactions.

Third, there are lots of good educational resources on Internet already. Rather than spending effort to create such resources, it is more cost effective to search for existing content, select content that appears to be effective, and integrate it into the ITS.

It appears that computer tutors and web-based recourses each address one aspect of education: coaching on problem-solving and general instruction, respectively. However, neither of them alone offers a complete solution. Towards the goal of finding an efficient means of constructing an intervention that covers both aspects of education, this paper presents our early-stage effort of combining web pages with ASSISTments: WEBsistments.

2 Methodology

The ASSISTments (www.assistments.org) system is a web-based tutor, primarily used for middle school mathematics by tens of thousands of students. Its standard method of instruction is to provide hints to help the student solve the problem, or scaffolding, which breaks the problem down into smaller steps. We enhanced its functionality to enable it to provide a button “Show me a web page,” which allows students to request a web page while solving a problem.

Students are allowed to request a web page in any stage of problem-solving, even before their first attempts. When a student clicks the request button, WEBsistments displays a web page associated with the skill tested by the problem. When there are multiple skills required in a problem, the web pages associated with the most advanced skill will be used to select a web page. A student cannot ask for multiple web pages while solving a problem, but he can use original assistance (hints and scaffolding) for the problem. WEBsistments collects information, such as how long the student spent on a web page, his next immediate action after seeing a web page was, whether he got the question correct right after seeing a web page, etc.

In the 2011-2012 version of WEBsistments, web resources were selected by two Worcester Polytechnic Institute undergraduates and a few volunteer middle school Math teachers. They ensured that each of the 147 Math skills that ASSISTments

tracks had 2-5 web pages that provide instruction on the skill. Then they tagged those web pages with the skill, indicating that the web page is relevant to that skill. Most problems in ASSISTments have already been tagged with one (or more) of 147 skills by domain experts. Therefore, through the skill mapping, there is a connection established between a problem and a set of relevant webpages. Since this is our first implementation of WEBSistments, we do not have a basis to prefer any page that has passed our screening process. Therefore, when deciding which web page to show, WEBSistments uses random selection.

WEBSistments has been used by 1121 8th grade (approximately 13 years old) students since July 2011. Since not all students chose to see a web page, we had to decide upon a comparison group, and selected students who were classmates of those who did request web pages. We also restricted our comparison set to those problems on which a student requested a web page, and only considered cases where students made an incorrect response. As a result of these restrictions, our data set consists of 9,983 problems solved by the students. The Web group includes cases where the student requested a web page (1104 problems); the No-web group includes the cases where students did not request a web page (8879 problems). Note that a student can be a member of both groups, if, for example, he requested a web page in one instance but decided not to in another.

3 Results

Each instance in our data set represents a student's wrong response to an initial problem, which we denote as P_1 . We then measure the student's performance on the next item using the same skill; we denote this problem as P_2 . To measure the learning gain, rather than just taking the difference of $P_2 - P_1$, we instead normalize the result by the population's average performance on each item. If P_2 is extremely easy, we should give not treat that as strong evidence of learning relative to a student getting a difficult question correct. In addition, we also considered the easiness of P_1 . This is because it tells whether one group has lower incoming knowledge than the other, as they may fail to respond to P_1 correctly even if P_1 was an easier problem. Therefore, we used the percent correctness of a problem across the entire population of ASSISTments students within the 2011-2012 school year to represent its easiness.

We used a performance score, shown in Equation 1, to represent how well a student performed in a problem. $Correctness_{i,j}$ is a binary value, 1 representing a correct response of student i to problem j and 0 representing incorrect. Problem easiness also ranges from 0 to 1 and a higher value means an easier problem. A performance score credits a student more when he successfully solves a harder problem, while punishes a student more when he fails an easier problem. Using performance scores, we calculated a gain score of a student between P_2 and P_1 by subtracting performance score $_{p_1}$ from performance score $_{p_2}$. We then calculated a gain score for each of the instances in the web and no-web groups.

$$\text{performance score } (i \in \text{students}, j \in \text{problems}) = correctness_{i,j} - easiness_j. \quad (1)$$

3.1 Overall Trend from Web Pages

In this section, we present our preliminary analyses of the data, aiming to examine whether there are any trends suggesting the effectiveness of web pages.

Table 1. Comparisons of the mean gain scores between the web and no-web groups

	Web group			No-web group		
	Mean	95% C.I.	N	Mean	95% CI	N
Overall	0.50	0.49 - 0.51	1104	0.40	0.39 - 0.40	8879
No Bottom-out	0.60	0.58 - 0.62	518	0.49	0.48 - 0.49	5336
Bottom-out	0.41	0.39 - 0.43	586	0.26	0.25 - 0.27	3543

Table 1 compares the statistics of the gain scores of the two groups. First, we observed that there were fewer cases, 1104, where students requested web page resources. In most cases, students still only sought for the traditional assistances of the tutor when they were stuck in the problems as there are three times as many (3543) cases where students solely used bottom-out hints. This result possibly suggests that the students preferred receiving the answer to learning, and raises issues of whether the group requesting web pages differs in desire to learn.

Second, we found that the mean gain score of the web group is 0.1 higher the no-web group, and the 95% confidence intervals have no overlap in values, indicating that the means are different at a significance level of 0.05. This result suggests that overall students who saw a web page learned more.

In addition, we extended our study to examine how web pages work for students with different proficiencies in Math. We included a new factor, “bottom-out hint” and used that to indicate a student’s proficiency. A bottom-out hint is presented as the last message in a sequence of hints for a problem, in which the answer to the problem is explicitly given. Due to its functionality, in the ITS research field, requesting a bottom-out hint presumably suggests that the student is weaker so as to need more help. We present the statistics of the two-way factorial in the last two rows of Table 1. The four means are corresponding to the factorial combinations of the use of web pages and the use of bottom-out hints. Consistent to the overall effects, at the factor level of “bottom-out”, in its two levels, each mean of the web group is higher than that of the no-web group. It suggests that web page support is generally helpful for both stronger and weaker students.

We found that the impact of using web page resources may be more effective for those who request bottom-out-hints. The difference between the means of the two sub groups is 0.15 (i.e. $0.41 - 0.26 = 0.15$), somewhat larger than the overall effect. Perhaps weaker students benefited more from getting extra web-based instruction? Moreover, the average gain for bottom-out-hints without web page support is just 0.26, suggesting that hint messages are a relatively slow means of instruction.

3.2 Modeling Effects of Web Pages

There are two issues which potentially impact the results of the previous statistical analyses. First, we did not consider whether the student saw a web page in P_2 .

Consider an example where a student does not request a web page in P_1 , requests a page in P_2 before responding, and as a result of the page gets P_2 correct. This student would show learning from P_1 to P_2 , and the no-web group would benefit since the student saw no web page on P_1 . Second, students certainly vary in their mathematics proficiency, which our first comparison did not account for.

To address these issues, we trained a model that considered multiple relevant factors simultaneously. For each instance, we used a binary value to indicate whether the student has seen a web page in P_1 and in P_2 . We used how many correct responses and incorrect responses have been produced by the student for the required skill to represent student proficiency. These two variables are used in the PFA model [6] and have been shown to effectively represent student proficiency [7].

Table 2. The logistic regression model of impacts on correctness of P_2

Independent variables	β
Saw a web page in P_1	0.393
Saw a web page in P_2	-1.693
Problem easiness of P_1	-0.983
Problem easiness of P_2	4.808
Number of prior corrects on the skill	0.010
Number of prior incorrects on the skill	-0.023
Reached the bottom-out hint in P_1	-0.635
Intercept	-1.992

Table 2 shows the result of the multinomial logistic regression run in SPSS to create a model to predict the correctness of P_2 . The regression model generated $r^2=0.17$, and all of the independent variables are reliable at $p<0.05$. Observing the coefficient value of “Saw a web page in P_1 ”, 0.393, we found that the model suggests the same trend as our prior statistical analyses. Considering the effects of all the relevant factors together, the model still acknowledges the positive effect of seeing a web page on helping students respond correctly to the next problem.

4 Contributions, Future Work, and Conclusions

This paper discusses a common issue across many ITSs (e.g.[1, 2, 3]): most step-based Intelligent Tutoring Systems focus predominantly on problem-solving. However, in order for students to benefit from problem-solving practice, sufficient declarative knowledge is essential [8], but ITS generally leave this task to teachers. We proposed and have pilot tested a solution to the problem: using web page resources on Internet as a complementary medium. We built WEBsistments to enhance an ITS to have the best of both worlds of coaching and instruction. We have found a promising trend of the effectiveness of this solution. This solution could be easily applied for most computer tutors, and is a low-cost option for ITS designers.

There are steps that could make WEBsistments better. First, students appear reluctant to request instruction; perhaps a tutorial policy that is proactive for students the tutor observes struggling to master the material would make sense? Our current on-demand policy could also cause a selection bias of students, and is certainly a

potential confound in our result, as instances in the web group are likely to be those done by motivated students who may be more eager to learn. However, the statistical model accounts for some of these individual differences. Second, a more intelligent method of selecting web pages is desired as it is likely that some web pages are more effective than others. In addition, individualizing web page recommendations is an interesting possibility. To prompt more learning and provide a web page to ensure that the most, a student's individual context could be considered as well. Possibly, student modeling and WEBSistments can make a strong join for this purpose.

In this paper, we presented our work, WEBSistments, to enhance a computer tutor to not only provide problem-solving practice, but also convey conceptual instruction to students. We conducted a pilot study to examine our hypothesis that students could learn more due to having this new form of assistance. Our results suggested that when web-based resources were used to help students in their problem-solving, it results in more gains in their performances in next problems. In a model where more factors were considered simultaneously, we also confirmed the positive effect of web pages. Moreover, bottom-out-hinting students, or weaker students in typical beliefs, seem to benefit more from receiving web-based resources as extra help.

References

1. Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K.R., Junker, B., et al.: The Assistentment project: Blending assessment and assisting. In: Looi, C.K., McCalla, G., Bredeweg, B., Breuker, J. (eds.) *Proceedings of the 12th Artificial Intelligence in Education*, pp. 555–562. ISO Press, Amsterdam (2005)
2. VanLehn, K., Lynch, C., Schultz, K., Shapiro, J.A., Shelby, R.H., Taylor, L., et al.: The Andesphysics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education* 15(3), 147–204 (2005)
3. Mostow, J., Aist, G.: Evaluating tutors that listen: An overview of Project LISTEN. In: Forbus, K., Feltovich, P. (eds.) *Smart Machines in Education*, pp. 169–234. MIT/AAAI Press (2001)
4. Mendicino, M., Razzaq, L., Heffernan, N.T.: A Comparison of Traditional Homework with Computer Supported Homework: Improving Learning from Homework Using Intelligent Tutoring Systems. *Journal of Research on Technology in Education (JRTE)* 41(3), 331–358 (2009)
5. VanLehn, K.: The Behavior of Tutoring Systems. *International Journal AI in Education* 16(3), 227–265 (2006)
6. Pavlik, P.I., Cen, H., Koedinger, K.: Performance Factors Analysis - A New Alternative to Knowledge. In: *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, Brighton, UK, pp. 531–538 (2009)
7. Gong, Y., Beck, J.: Items, Skills, and Transfer Models: Which Really Matters for Student Modeling? In: *Proceedings of the 4th International Conference on Educational Data Mining*, pp. 81–90 (2011)
8. Anderson, J.R.: *Rules of the Mind*. Lawrence Erlbaum Associates, Hillsdale (1993)

Automated Approaches for Detecting Integration in Student Essays

Simon Hughes^{1,*}, Peter Hastings¹, Joe Magliano²,
Susan Goldman³, and Kimberly Lawless³

¹ DePaul University

² Northern Illinois University

³ University of Illinois Chicago

Abstract. Integrating information across multiple sources is an important literacy skill, yet there has been little research into automated methods for measuring integration in written text. This study investigated the efficacy of three different algorithms at classifying student essays according to an expert model of the essay topic which categorized statements by argument function, including claims and integration. A novel classification algorithm is presented which uses multi-word regular expressions. Its performance is compared to that of Latent Semantic Analysis and several variants of the Support Vector Machine algorithm at the same classification task. One variant of the SVM approach worked best overall, but another proved more successful at detecting integration within and across texts. This research has important implications for systems that can gauge the level of integration in written essays.

Keywords: support vector machines, latent semantic analysis, multi-word regular expressions, integration, document classification.

1 Introduction

Researchers and teachers have recognized that a fundamental challenge for education is teaching students to be able to read with *deep understanding*. To thrive in society students need to learn how to select and evaluate multiple sources of information, make connections across sources (even when information is contradictory) and to apply what they discover to achieve their goals. These critical skills of reasoning within and across texts have been included in the U.S. Common Core Standards of education (<http://www.corestandards.org/in-the-states>).

Methods for teaching these skills will require the use of open-ended tasks like writing integrative essays. Previous work has explored the use of automated

* The project described in this article is funded, in part, by the Institute for Education Sciences, U.S. Department of Education (Grant R305G050091 and Grant R305F100007). The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Dept of Education. Correspondence may be sent to Peter Hastings, DePaul Univ. CDM, 243 S. Wabash, Chicago IL 60604, USA.

analysis of essays [1, for example], but mostly this has focused on summaries or analyses of single texts. Our overall goal is to teach students how to read and understand texts more deeply by having them write summaries that combine ideas from multiple texts. This poses two major challenges for automated essay analysis. The first is the semantic overlap of the texts. Some amount of overlap is necessary to help them make inferences. Yet this is problematic for automated techniques, particularly those that rely on word occurrences rather than text structure. The second challenge is cross-text inferences. Although a key goal of this project is to teach students to make such inferences, the broad variety of connections that students construct can make them harder to detect by automatic mechanisms. This paper analyses three different mechanisms that can be used as the evaluation component in a system to assist students to learn to integrate material across texts.

2 Document Classification Algorithms

2.1 Latent Semantic Analysis

Document classification techniques fall into two areas, ‘bag-of-words’ approaches which ignore word order, and order sensitive methods. This study investigates two bag of word approaches, Latent Semantic Analysis (LSA) and Support Vector Machines (SVM’s), and multi-word, which is order sensitive. LSA was initially developed as an Information Retrieval system, but was later found to closely model human lexical acquisition in a number of ways [8]. It creates a term-document co-occurrence matrix where each cell is weighted for the frequency of the term in the document relative to the entire corpus [6]. Then singular value decomposition re-orientes the data axes, ranked by their correlation with the data. The top K dimensions (typically 300–400) are used to compare texts. LSA has been used for text classification in applications such as comparing student answers to expected answers in an ITS [3] and grading student essays [2, for example]. For text classification, a threshold cosine value is chosen to achieve the best correlation with human similarity judgments.

2.2 Support Vector Machines

SVM’s were introduced as a binary classifier for classifying non-linearly separable classes. An SVM creates one or more hyperplanes in higher-dimensional space that allow linear separation of the data points into separate classes by selecting the hyperplane with the largest margin of separation. This minimizes generalization error. Multiclass SVM’s have subsequently been developed [5]. SVM’s have been successful at tackling a wide range of regression and classification problems, including text classification [11, for example]. Several authors have tried to improve SVM classification performance by combining them with techniques that take into account word order, with mixed results [11, for example].

2.3 Multi-Word

Ignoring word order when classifying text ignores useful semantic information, motivating research into the multi-word approach. There are 2 main variants, a syntactic and an n-gram approach. The syntactic technique extracts re-occurring phrases consisting only of nouns, adjectives and propositions that follow a particular syntactic structure [7,11]. The n-gram approach looks for the occurrence of any n-word phrase with a frequency above a threshold [10]. The extracted phrases are then typically used as features for some other classification approach, such as an SVM [11], or to enhance queries used to classify documents [10]. The approach has proven successful in a number of empirical studies [10,11].

3 Methodology

3.1 Data

In 2008 and 2009, students from grades 5–8 in two large urban public schools were asked to read three short articles (around 30 sentences each) about Chicago history, and then write essays about population growth in Chicago. 365 essays were written. The articles were created to be complementary, with minimal semantic overlap. One article covered “push” factors driving people to the city, another detailed “pull” factors pulling people to Chicago. The third described how advances in transportation enabled this migration. An *integrated model* was created to represent the conceptual content of the articles and likely connections that students might make between and within the articles, and between the articles and the overall question about population growth in Chicago. The conceptual content was hierarchically structured in the model into high-level claims, intermediate evidence supporting the claims, and low-level details about the evidence. Human annotators coded the correspondence between the student sentences and both the sentences of the articles and the (37) nodes of the integrated model. The inter-rater reliability for the two coders was 85%.

3.2 Metrics

Three metrics were used to measure classification performance across the different approaches, recall, precision and F_1 score, as described in [6, p. 578]. Recall measures false negatives, and thus Type II errors, while precision measures the number of false positives, and thus Type I errors. Typically, as recall increases, precision decreases and vice versa. A combined measure is commonly used to evaluate performance, the F measure, using a coefficient β to adjust the weighting of recall to precision [6, p. 578]. To evaluate the classification performance of each approach, we performed ten-fold cross-validation [9, p. 112].

3.3 LSA

We previously used LSA to identify how many sources the students were referring to [4]. We used the lsa.colorado.edu site to compare student sentences with the

sentences of the articles. A more important goal was to determine how well the students covered the concepts in the integrated model. To do this, we used the correspondences which were specified in the model between the nodes and the article sentences. Many of the nodes had multiple associated sentences, and many sentences had multiple associated nodes. The 7 “linking” nodes reflected an inference between part of an article to part of another article, or to the overall claim of the essay, and so had no corresponding article sentences. If the LSA cosine between a student’s sentence and an article sentence was above a threshold that we determined, the sentence was assigned the code(s) of the model node(s) associated with that sentence. We tested thresholds from 0.4 to 0.8, by 0.05 increments, and found a value of 0.7 had the highest overall F_1 score.

3.4 SVM

In prior work, we compared the performance of an SVM to a manual pattern matcher and LSA, and found that the patterns outperformed the SVM [4]. In that study, we used the multiclass SVM to choose the single most likely class for each test example. But many of our example sentences were assigned multiple codes resulting in these sentences appearing in the dataset multiple times, once for each code. This meant that at most one of these multi-code sentences could be coded correctly by the SVM, thereby limiting the overall performance.

In the current study, we evaluated two methods to overcome this. First we used an SVM binary classifier. For each of the 37 classes, a sentence was marked as a positive instance only if that class was in the set of codes assigned by the human raters. The sentences were represented by a *tfidf* weighted vector, as with LSA. We trained a different classifier for each code. The second SVM approach used the multiclass method, but in a different way. As well as the “best” prediction for each example, SVMlight gives a weight for each class. We established a threshold, and used it to assign (potentially) multiple classes to each example. To avoid bias in the choice of threshold, we calculated the average number of codes per sentence, then selected a threshold which would produce the same number. The threshold also depends on the C parameter (margin) that the SVM model was trained with, so we repeated the process for a range of C values. The best performance was achieved with a C value of 1000 and a threshold of 0.19. This method is marked as “SVM threshold” in the results.

3.5 Multiword

The multi-word approach used is closest to the n-gram approach and is a binary classifier. The algorithm extracts re-occurring expressions (one or more words long), as described in [10], and iteratively constructs a regular expression to classify each category. For each category, all multi-word phrases were extracted and converted into regular expressions. The category’s F_β score was then computed for each expression. The expression with the highest F_β score was removed along with all sentences matching the expression. This process was then repeated, and a composite regular expression was built iteratively by combining the highest

scoring expressions using the ‘or’ operator. Its classification performance was measured on the validation dataset after each iteration. The algorithm halted either when no expressions remained, or after ten consecutive iterations without improvement on the validation dataset to prevent over-fitting [9, p. 116]. β values of 0.25, 0.5, 1 and 2, were used with 0.25 producing the highest F_1 score.

4 Results and Discussion

Our main goal was to evaluate different methods of detecting integration between sources in sentences making up essays. To do this, the 37 model categories were separated into 5 groups corresponding to higher-level categories in the model, including 2 groups containing sentences showing integration between different texts (IR) and within the same text (RC). The IR category also contains inferred relations between a text and a top-level assertion. The other 3 categories consisted of sentences making top-level claims (CL), evidence for those claims (EV) and details surrounding the evidence (DET). Separating the integration categories from the other categories allows a direct evaluation of the techniques at detecting integration, and the other categorical groupings. These results along with the aggregate classification performance are shown in Figure 1 below.

The SVM binary classifier out-performed the other approaches overall and in the CL, EV, and DET categories, while the SVM threshold method demonstrated the best classification performance on the RC and IR categories and thus was the best approach for detecting integration. These 2 techniques showed significant improvement over the SVM multiclass. The multi-word method had the second highest classification performance on the IR category, although it did poorly on all other categories. LSA performed particularly poorly on the IR category. The LSA approach we adopted classified sentences based on their similarity to individual sentences in the source texts, and thus would perform poorly identifying sentences composed from multiple source sentences. The smallest category (IR) was a challenge for all of the algorithms. Machine learning algorithms often struggle with small datasets [9], which may explain this observation.

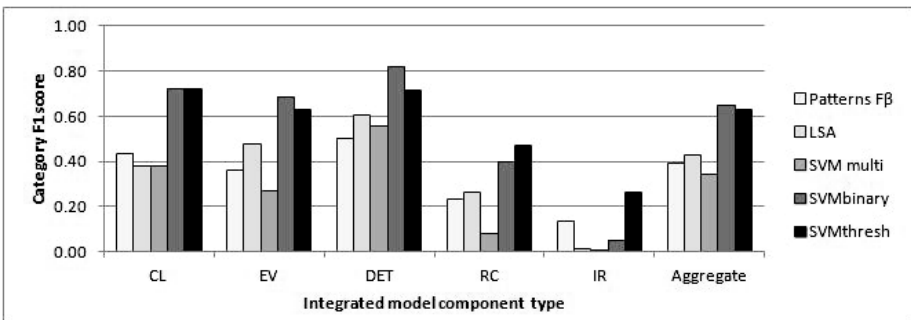


Fig. 1. Aggregate F_1 score by algorithm across different integration model categories

Although SVM's performed better overall, the strong performance of multi-word on the IR category indicates that a hybrid approach combining the threshold SVM method with multi-word may improve the performance at detecting integration. Several authors have used the multi-word approach to create features that were then used by an SVM for text classification [11, for example]. Such an approach may prove more successful at this task than either approach in isolation. Naive Bayes has been successfully applied to text classification and may also be effective at this task. Also, repeating the experiments with a larger dataset with more sentences in the IR category may yield better results. One limitation to this study was the need to create an integrated model of the topic, and manually code a dataset to this model. For this approach to be successfully applied to new domains, the manual effort required would need to be minimized. If multiple datasets on different topics were collected, each with their own integrated model, it may be possible to train a more general classifier that can detect integration in unseen datasets without the need for an integrated model.

References

1. Attali, Y., Burstein, J.: Automated essay scoring with e-rater R V.2. *Journal of Technology, Learning and Assessment* 4, 1–30 (2006)
2. Foltz, P., Britt, M., Perfetti, C.: Reasoning from multiple texts: An automatic analysis of readers' situation models. In: *Proceedings of the 18th Annual Conference of the Cognitive Science Society*, pp. 110–115. Erlbaum, Mahwah (1996)
3. Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N.: The Tutoring Research Group: Using Latent Semantic Analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments* 8(2), 129–147 (2000)
4. Hastings, P., Hughes, S., Magliano, J., Goldman, S., Lawless, K.: Text Categorization for Assessing Multiple Documents Integration, or John Henry Visits a Data Mine. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011. LNCS*, vol. 6738, pp. 115–122. Springer, Heidelberg (2011)
5. Joachims, T.: Making large-scale SVM learning practical. In: Schölkopf, B., Burges, C., Smola, A. (eds.) *Advances in Kernel Methods: Support Vector Learning*. MIT Press (1999)
6. Jurafsky, D., Martin, J.: *Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall, New York (2000)
7. Justeson, J.S., Katz, S.M.: Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering* 1(01) (1995)
8. Landauer, T., Dumais, S.: A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review* 104, 211–240 (1997)
9. Mitchell, T.: *Machine Learning* (Mcgraw-Hill International Edit), 1st edn. McGraw-Hill Education (ISE Editions) (October 1997)
10. Papka, R., Allan, J.: Document classification using multiword features. In: *Proceedings of the Seventh International Conference on Information and Knowledge Management, CIKM 1998*, pp. 124–131. ACM, New York (1998)
11. Zhang, W., Yoshida, T., Tang, X.: A comparative study of tf*idf, lsi and multi-words for text classification. *Expert Systems with Applications* 38(3) (2011)

On the WEIRD Nature of ITS/AIED Conferences

A 10 Year Longitudinal Study Analyzing Potential Cultural Biases

Emmanuel G. Blanchard

Department of Architecture, Design, and Medialogy
Aalborg University Copenhagen
emmanuel.g.blanchard@gmail.com

Abstract. Arnett (2008) confirmed that research production (authorship, samples) in major psychology journals is strongly dominated by Western societies that are not cognitively representative of the whole mankind (Henrich et al., 2010). In this paper, results from a ten-year analysis of paper production in ITS/AIED conferences suggest a similar bias in the AIED research field.

Keywords: Research production, cultural bias, AIED design and strategies.

1 Introduction

In an analysis of publications of six major journals of the American Psychology Association (APA), Arnett [1] shows that a huge majority of first authors are affiliated with academic institutions from *Western, Educated, Industrialized, Rich, and Democratic* (WEIRD) societies [2] that represent just 12% of the world population. This analysis further reveals that samples in considered journals are almost exclusively WEIRD ones, and that authors tend to easily broaden the applicability of their results to the whole of mankind. However, Henrich et al. [2] showed that WEIRD and non-WEIRD people cognitively differ to a great extent. This paper discusses if and how this WEIRD bias observed in psychology may be influencing AIED¹ research. First, Arnett [1] is presented. Henrich et al. [2] is then summarized and considered in the AIED context. Results of an analysis of full papers published in the AIED/ITS conferences are eventually reported and discussed by AIED senior members.

2 WEIRD Dominance on Psychology and Implications for ITS

The WEIRD dominance on psychology. The main contribution of Arnett [1] is an analysis of national affiliations of content of papers published in six premier APA journals between 2003 and 2007. Results for first authors and samples are summarized in Table 1, and show a very strong dominance of WEIRD first authors and a similarly large tendency to draw conclusions based only on WEIRD samples.

¹ In this paper, the AIED acronym refers to the field for which Intelligent Tutoring Systems (ITS) and Artificial Intelligence in Education (AIED) conferences are frequently acknowledged as premier events.

Table 1. National affiliations of first authors and of samples in six APA journals (see [1])

	DP	JPSP	JAP	JFP	HP	JEP	Total	DP	JPSP	JAP	JFP	HP	JEP	Total
Nb.	461	698	354	313	408	297	2531	466	721	334	273	371	287	2452
	1st Author (% per national affiliation)							Samples (% per national affiliation)						
USA	72%	65%	78%	85%	78%	66%	73%	64%	62%	73%	81%	76%	64%	68%
Eng.	17%	13%	12%	8%	16%	15%	14%	19%	12%	13%	8%	15%	14%	14%
Europe	9%	18%	9%	6%	6%	12%	11%	11%	19%	11%	8%	8%	13%	13%
Asia	1%	1%	1%	1%		4%	1%	4%	4%	2%	1%	1%	7%	3%
Latin A.								1%	1%					1%
Africa									1%					
Israel	2%	2%		1%	1%		1%	1%	2%		2%		2%	1%

Notes: The journals considered are *Developmental Psychology* (DP), *Journal of Personality and Social Psychology* (JPSP), *Journal of Abnormal Psychology* (JAP), *Journal of Family Psychology* (JFP), *Health Psychology* (HP), and *Journal of Educational Psychology* (JEP). In tables 1 and 2, ‘Latin A.’ refers to Latin America, and ‘Eng.’ to English-speaking countries i.e. the United Kingdom, Canada, Australia and New Zealand. Finally, according to [2], WEIRD societies refer to the ‘USA’, ‘Eng.’, ‘Europe’, and ‘Israel’ rows.

Arnett sees two main reasons for the dominance of WEIRD countries on psychology. The first one is economic, with governments of developing countries likely to dedicate their funds to more crucial expenses than research on psychology. However, this does not explain the low presence of research originating in non-WEIRD developed countries (e.g. Japan). Arnett thus suggests that the dominant philosophy of science in psychology remains “*on investigating fundamental processes, resting on the assumption – rarely stated, and rarely actually tested – that people anywhere can be taken to represent people everywhere, and that the cultural context of their lives can be safely ignored*”. This philosophy strongly favors the production of WEIRD-flavored content when considering that most psychology scholars are located in WEIRD societies and consequently have an easy access to WEIRD samples, and that they “*have extremely limited knowledge concerning the work of their international counterparts*” [3].

WEIRD People as Outliers in the World Population. Henrich et al. [2] extended [1] by investigating potential WEIRD cognitive biases through a four-level review: (i) Industrialized societies versus small scale societies. Variations in *visual perception*, *economic decision-making* (e.g. social motivation, fairness), *folk-biological reasoning*, and *spatial cognition* are reported between member of industrialized societies (frequent outliers) to members of various small-scale societies. Other variations in *decision-making* are also likely to exist. (ii) Western versus non-Western societies. Variations are reported with regards to *social-decision making* (e.g. fairness, cooperation, punishment), *reasoning strategies* (tendency of Westerners to be more analytic, and of others to be more holistic), *moral reasoning*, and *independent/interdependent self-concepts* (tendency of Westerners to be more individualistic, which has implications for features such as motivation or emotions). (iii) Contemporary US peoples versus the rest of the West. Reliance on US content is huge in contemporary psychology even when compared to other WEIRD societies (see Table 1). According to [2], US people have a higher tendency for *expressing strong individualism*, which may be the illustration of an ideology that “*particularly stresses the importance of freedom and self-sufficiency*”, and of “*various practices in education and childrearing*” that

enforce individualism. (iv) Typical contemporary American subjects versus other Americans. Much of American psychology relies on samples of college students. Variations are reported between them and other Americans with regards to *rationality of choices, individualism, conformity motivation, perception of racial diversity, structure of social networks, interdependence, pro-social behaviors*, etc. As test subjects, children are likely to have parents with a high socio-economic status (SES), while poor-SES and high-SES children show differences in processes such as *spatial reasoning*. Existing and reported similarities do not restrain Henrich et al. to state that WEIRD subjects “*are some of the most psychologically unusual people on Earth*”, and consequently “*may often be the worst population from which to make generalizations*”. The authors also warn that the demonstrated extreme reliance on WEIRD samples “*may cause researchers to miss important dimensions of variation, and devote undue attention to behavioral tendencies that are unusual in a global context*”.

[2] has been overwhelmingly supported by many researchers in [4]. These comments also bring additional elements to consider such as extending the suspicion of WEIRD biases to research on cognitive development, children’s social behavior, and parent-child interaction (p. 99-100), to philosophical production and intuitions (p. 110), and to experimental designs (p.84-85). Evidences of socio-cultural variations in brain functioning are also reported (p. 88-90), distortions on research resulting from the use of English and other WEIRD languages (p. 103) are also mentioned, and “*the promise of Internet in reaching more diverse samples*” (p.94-95) is also noticed.

WEIRD Biases Spreading to the ITS Research Field. The work of Arnett [1] convincingly demonstrates that contemporary psychology is WEIRD-dominated to a great extent. Furthermore, according to Henrich et al. [2], this situation is likely to produce ethnocentric biases in research since WEIRD societies are not cognitively representative of the world population, though there is a tendency among scholars to present results obtained on WEIRD samples incautiously as universalisms.

An initial conclusion can be drawn from this situation. Since AIED historically relies on research in psychology, the reported ethnocentric biases have most probably spread to this domain. Indeed, several features with reported variability between WEIRD and non-WEIRD societies (see [2, 4]) are genuine ITS topics of interest e.g. self-concepts, emotions (see [5] for an overview of cultural influences on the affective domain), reasoning strategies, decision making, cooperation, etc. However, concerns on potential WEIRD biases in AIED are not necessarily relevant if a tutoring system is tailored for a WEIRD audience, although even within WEIRD societies there may be large variations. Still, one has to be cautious when relying on theories established in a different socio-cultural context than the one of the targeted learners, and with growing educational needs of demographic giants such as China, India, Brazil, or Nigeria (all showing great market opportunities for AIED), alternative approaches could be envisioned for ITS to become more culturally-aware [6].

3 A Ten Year Analysis of Full Papers in ITS/AIED Conferences

While the influence of psychology-originated WEIRD biases on AIED is not really questionable, another point needs to be discussed: does AIED similarly produce

WEIRD-biased research results? To address this question, the full paper production of the ten last AIED/ITS conferences was analyzed. Similar to APA journals in psychology, both these conferences are seen as premier references by many members of the ITS community, especially when considering the limited number of long-term established journals dedicated to the discipline. Using the same regional categories as [1], the top part of Table 2 presents the distribution of first authors' national affiliations per conference. Results indicate nearly similar proportions of WEIRD first authors in ITS/AIED conferences as in results reported by Arnett (see Table 1).

Table 2. National affiliations of first authors and of samples in ITS and AIED conferences²

	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	Total
Nb.	93	40	73	89	67	60	62	68	61	49	662
1st Author (% per origin)											
USA	26%	40%	41%	46%	37%	70%	56%	49%	74%	63%	49%
Eng.	26%	20%	22%	26%	28%	17%	19%	30%	15%	18%	23%
Europe	40%	25%	21%	16%	16%	8%	13%	13%	5%	12%	19%
Asia	4%	10%	5%	10%	15%	3%	11%	9%	7%	4%	8%
Latin A.	4%	5%	11%	1%	3%	2%		1%			3%
Nb.	41	20	28	48	29	47	40	50	47	36	386
Considered Samples (% per origin)											
USA	34%	50%	61%	54%	55%	79%	75%	52%	81%	61%	61%
Eng.	37%	35%	29%	27%	28%	9%	13%	24%	6%	17%	21%
Europe	27%	10%	11%	13%	3%	9%	5%	10%	4%	14%	11%
Asia				4%	10%	4%	8%	10%	6%	8%	5%
Latin A.	2%	5%		2%	3%			4%	2%		2%

In order to make the analysis of AIED/ITS samples comparable to Arnett's results, further paper refinements were required. (i) Some ITS/AIED papers do not present any evaluations involving humans and had to be discarded. An analysis of this criterion revealed a strongly increasing tendency of ITS/AIED conferences content to include more and more human-related evaluations: in ITS2002 and AIED2003, papers with such content represented 50.5% and 57.5% respectively (lowest scores in the whole decade), whereas in ITS2010 and AIED2011, they represented 90.2% and 93.9% respectively (highest scores in the whole decade). (ii) Other papers use human-related data only to validate technical aspects³, and similarly had to be discarded. This further categorization showed that the rate of papers with (sometimes lousy) psychology-related features has also strongly increased, especially since AIED2007: in the second half of the decade, it lays between 64.5% and 79.6%, whereas it was between 38.5% and 54.1% in the first half. Various explanations can be envisioned to explain

² ITS2002 occurred in France/Spain, AIED2003 in Australia, ITS2004 in Brazil, AIED2005 in the Netherlands, ITS2006 in Taiwan, AIED2007 in USA, ITS2008 in French Canada, AIED2009 in the UK, ITS2010 in USA and AIED2011 in New Zealand.

³ Human-based evaluations were rated as purely technical only when there was a unique focus on validating the system/technique rather than on assessing the students, their behavioral or cognitive processes, or their appraisal of the system (in which cases they were perceived as including psychological features).

this evolution. Still, this evolution towards more systematic inclusions of psychology-related features makes AIED more sensible to WEIRD-biases affecting psychology.

Following this refinement, the national origins of samples from remaining papers (that include psychology-related features) were investigated. They are presented in the bottom part of Table 2. A significant proportion of samples were not clearly described, but it was possible most of the time to infer their origins by cross-checking indirect clues. Nevertheless, a few samples were discarded because of the impossibility of determining their origin with sufficient confidence. Results indicate a dominance of WEIRD samples that is comparable to Arnett's results. These results suggest that the AIED community may be producing similarly WEIRD-flavored research.

4 Discussion and Conclusion

In order to assess these results in a non-dogmatic way, seven AIED senior members, three of who are female, accepted to comment on them. Regarding their origin, one is from the USA, three are from English-speaking countries (one is a French-speaking Canadian), two are from Europe, and one is from Asia. Despite several attempts, no Latin American expert answered positively to the invitation. Regarding their academic background, two of the panel members have a PhD (or equivalent) in psychology, one in educational technology, and four in computer sciences and related disciplines. Due to space constraints, the following paragraphs only summarize some expert views and comments. Readers have to be aware that comments were collected individually. Hence, each expert may disagree with thoughts expressed by others.

All experts agreed to the existence of a WEIRD bias in AIED research with one expert even noticing a worrying “*strong tendency to blindness to that bias*” in some societies. However, for most of the experts (and for the author as well), it is important to insist that the bias is unintentional, that the selection of papers is only based on scientific criteria, and that the discussed bias can only be understood currently as an imbalance in author and sample origins since no results are actually provided on how it may be influencing the AIED research. Four experts insisted that the AIED field has several important differences with psychology that would lead this bias to have different incidences and implications on AIED production, which has to be thoroughly investigated in future work. One expert rightfully insisted on differentiating the fact that AIED research is mainly performed by WEIRD scholars, from the one that it is mainly grounded on WEIRD samples. These issues are not equally problematic and have to be considered separately. Another expert noted that other potential sampling biases should be investigated as well. Two experts insisted on the English language dominance in the academic world to partly explain the situation. Another expert stated that this imbalance would not be an issue if the AIED community correctly followed the ‘scientific paradigm’, which (s)he claims is not currently the case.

The author submitted several suggestions to the panel. Five experts agreed with the author that the main way to address the issue raised in this paper is to make the AIED community aware of it, which the current paper intends to achieve. Scholars could then self-regulate their work and the way they present their results. Six experts agreed with the author that papers including intercultural evaluations and collaborations should be encouraged, and more events should be dedicated to better understand

issues that may be culturally-variable and relevant for AIED development. Two experts further mentioned that the influence of culture on AIED should also be investigated in more master and doctoral projects. Five experts agreed with the author that conference reviewers should ensure that samples are correctly described and, consequently, sample description guidelines should be available on conference websites. The seventh expert did not see this point as a crucial solution.

Finally, two panel members suggest the AIED community to question itself about the current importance of human-based evaluations on paper acceptance/rejection decisions. Even when loosely done, they claim it has more impact on the acceptance decision than detailing a clever technical solution, which they consider a problematic situation.

As a conclusion, this paper attempts to make the community aware of an identified and quantified WEIRD bias in psychology research that is likely to have an indirect impact on the AIED research field. A ten years analysis of conference full papers production reveals similar WEIRD imbalances in the AIED research field, which suggests that it may be producing WEIRD-flavored research as well. Several AIED experts, while acknowledging the situation, have produced different interpretations and suggestions on how to address it in the future, and many other options could be investigated as well. Indeed, considering culture into AIED is not more of an ‘intractable problem’ than other ones our community has faced in the past. The true question is whether or not we want to embrace this challenge.

Acknowledgement. The author would like to thank Jacqueline Bourdeau, Benedict du Boulay, Bert Bredeweg, Monique Grandbastien, W. Lewis Johnson, Susanne P. Lajoie, Riichiro Mizoguchi, and the four ITS anonymous reviewers for their fruitful comments and suggestions, and Isabela Gasparini for her help in collecting conference data.

References

1. Arnett, J.J.: The neglected 95%. Why American psychology needs to become less American. *American Psychologist* 63(7), 602–614 (2008)
2. Henrich, J., Heine, S.J., Norenzayan, A.: The weirdest people in the world? *Behavioral and Brain Sciences* 33, 61–83 (2010)
3. Denmark, F.L.: Women and psychology: An international perspective. *American Psychologist* 53, 465–473 (1998)
4. Various authors: Open Peer Commentaries on “The weirdest people in the world?” *Behavioral and Brain Sciences* 33, 83–111 (2010)
5. Mesquita, B., Frijda, N.H., Scherer, K.R.: Culture and emotion. In: Dasen, P., Saraswathi, T.S. (eds.) *Handbook of Cross-Cultural Psychology. Basic Processes and Human Development*, vol. 2, pp. 255–297. Allyn & Bacon, Boston (1997)
6. Blanchard, E.G., Ogan, A.: Infusing Cultural Awareness into Intelligent Tutoring Systems for a Globalized World. In: Nkambou, R., Bourdeau, J., Mizoguchi, R. (eds.) *Advances in Intelligent Tutoring Systems*. *SCI*, vol. 308, pp. 485–505. Springer, Heidelberg (2010)

Goal-Oriented Conceptualization of Procedural Knowledge

Martin Možina, Matej Guid, Aleksander Sadikov,
Vida Groznik, and Ivan Bratko

Faculty of Computer and Information Science, University of Ljubljana, Slovenia
{martin.mozina,matej.guid,aleksander.sadikov,vida.groznik,
ivan.bratko}@fri.uni-lj.si

Abstract. Conceptualizing procedural knowledge is one of the most challenging tasks of building systems for intelligent tutoring. We present an algorithm that enables teachers to accomplish this task semi automatically. We used the algorithm on a difficult king, bishop, and knight versus the lone king (KBNK) chess endgame, and obtained concepts that could serve as textbook instructions. A pilot experiment with students and a separate evaluation of the instructions by experienced chess trainers were deemed very positive.

Keywords: domain conceptualization, procedural knowledge, goal-oriented rule learning, argument-based machine learning, chess.

1 Introduction

Domain conceptualization lies at the very core of building an intelligent tutoring system (ITS) [7],[10]. This involves the structuring of the domain and creating a vocabulary or ontology of key concepts. Domain conceptualization consists of declarative knowledge and procedural knowledge, which generally speaking is the knowledge exercised in the performance at some task. Procedural knowledge is usually implicit and not easily articulated by the individual. Due to its tacit nature this kind of knowledge is often very hard to conceptualize.

In this paper, we will consider symbolic problem solving domains where problem solving is based on reasoning with symbolic descriptions (like in physics, mathematics, or games like chess). A particular domain is defined with a basic domain theory (*e.g.*, the rules of chess) and a solution to be achieved (*e.g.*, checkmate the opponent in chess). The task is to find a sequence of steps that bring us from the starting state of the problem to the goal state.

The basic domain theory (or basic declarative knowledge of the domain) is usually simple and easy to remember. It is, in principle, sufficient for solving problems (*e.g.*, knowing rules of chess could in theory enable optimal play). However, finding a solution using only declarative knowledge would require far too extensive searching for a human. A human student is incapable of searching very deeply, therefore we need to teach him also the procedural knowledge – how to solve problems. The “complete” procedural knowledge would be a function

mapping from each problem state to a corresponding action that leads to the solution of the problem. In chess, such complete knowledge (called “tablebases”) is computed for some endgames. Tablebases effectively specify best moves for all possible positions. They logically follow from the rules of the game and can be viewed as a compilation of the rules into an extensive form. Tablebases can be used easily because they only require trivial amount of search. But now the problem is the space complexity – it is impossible for humans to memorize such tablebases that typically contain millions of positions.

There is a way, however, that enables humans to solve problems in such chess endgames quite comfortably. The key is that humans use some intermediate representation of the problem that lies between the rules of the game (or the corresponding tablebases) and solutions. We call such an intermediate representation a “conceptualized domain.” Powerful conceptualizations are sufficiently “small” so they can be memorized by a human, and they contain concepts that enable fast derivation of solutions. Such a domain conceptualization enables effective reasoning about problems and solutions [8].

In this paper, we propose a goal-oriented conceptualization of domains and explore how to semi-automatically construct such a conceptualization that can be effectively used in teaching problem-solving. To this end, we used argument-based machine learning (ABML) [6], an approach that combines learning from examples with learning from domain knowledge. Such a combination can be particularly useful in the problem of domain conceptualization, as it is consistent with data (accurate) and at the same time consistent with expert’s knowledge (understandable) [4]. A similar idea, however with a different goal, was explored in a system called *SimStudent* [2], where learning from examples and learning by tutored problem solving was interweaved. Another interesting and somewhat similar work comes from Tecuci et al. [9] who developed a series of systems called *Disciple* that combine different types of learning, such as learning from explanations provided by users or by generalizing learning examples.

2 Goal-Oriented Rules

A goal-oriented rule has the following structure:

IF preconditions THEN goal (depth)

The rule’s preconditions and goal are expressed in terms of attributes used for describing states. The *preconditions* is a conjunction of simple conditions specifying the required value of an attribute. For example, preconditions could contain $kdist = 3$ ($kdist$ being distance between kings in chess), or a threshold on an attribute value, *e.g.*, $kdist > 3$. Similarly, a goal is a conjunction of subgoals, where a subgoal can specify the desired value of an attribute (*e.g.*, $kdist = 3$) or any of the four possible qualitative changes of an attribute given the initial value: decrease, increase, not decrease, not increase or its optimization: minimize, maximize; *e.g.*, a subgoal can be “decrease $kdist$ ” (decrease distance between kings). The depth property of a rule specifies the maximum allowed number of steps in

Algorithm 1. Pseudo code of the goal-oriented rule learning method.

```

GOAL-ORIENTED RULE LEARNING (examples  $ES$ ,  $depth$ )
let  $allRules$  be an empty list
while  $ES$  is not empty do
  let  $seedExample$  be  $FindBestSeed(ES, ruleList)$ 
  let  $goals$  be  $DiscoverGoals(ES, seedExample, ruleList, depth)$ 
  if  $goals$  is empty then
    remove  $seedExample$  from  $ES$  and return to the beginning of while sentence
  end if
  let  $rule$  be  $LearnRule(ES, goals, ruleList)$ 
  add  $rule$  to  $allRules$ 
  remove examples from  $ES$  covered by  $rule$ 
end while
return  $allRules$ 

```

achieving the goal. It corresponds to the level of conceptualization, where higher depths lead to simpler rules with less conditions and less subgoals, however, these goals are more difficult to solve, because they require more search.

The complete proposed conceptualization of procedural knowledge is a decision list of ordered goal-oriented rules. In an ordered set of rules, the first rule that “triggers” is applied. Note the difference between goal-oriented rules and classical if-then rules. An if-then rule triggers for a particular state if the preconditions are true, while a goal-based rule triggers when the preconditions are true *and* the goal is achievable. For example, consider a rule: *IF edist > 1 THEN decrease kdist*. The correct interpretation of this rule is: “if black king’s distance from the edge is larger than 1 and a decrease in distance between kings is possible, then reach this goal: *decrease the distance between the kings*.”

If a goal is achievable, we would like to know how good it is in a given state. We evaluate the goal by its worst possible realization in terms of the distance-to-solution (*e.g.*, distance-to-mate in chess). Formally, a goal’s quality $q(g, s)$ in state s is defined as the difference between starting distance-to-solution and distance-to-solution after the worst realization of the goal g : $q(g, s) = dts(s_{worst}) - dts(s)$. We say that a goal is *good* for a state s if its worst realization reduces the distance to solution, *i.e.*, if $q(g, s) < 0$; otherwise the goal is *bad*.

The quality of a rule R is directly related to the quality of its goal on states covered by the rule. Let p be the number of covered examples where the goal is good and n number of all covered examples. Then, the quality is computed using the Laplacian rule of succession: $q(R) = (p + 1)/(n + 2)$.

3 Goal-Oriented Rule Learning Algorithm

The task of learning goal-oriented rules is stated as: given a set of problem solving states each labeled with a distance-to-solution, learn an ordered set of goal-oriented rules. As these states act as learning examples, we will use this term in the description of the algorithm. As mentioned above, each learning example is described with a set of attributes.

The pseudo code of our goal-oriented rule learning method is shown in Algorithm 1. It accepts two parameters; *ES* are the learning examples and *depth* is the maximum allowed search depth for achieving goals.

The learning loop starts by selecting a seed example, which is used in the following calls to procedures *DiscoverGoals* and *LearnRule*. The *DiscoverGoals* procedure finds *good* goals for the seed example and then *LearnRule* induces a rule covering this example. The idea of seed examples and learning rules from them was adopted from the AQ series of rule-learners developed by Michalski[3], and is especially useful here, since discovering a goal is a time consuming step. A learned rule is afterwards added to the list of all rules *allRules* and all examples covered by this rule are removed from the learning examples. The loop is stopped when all learning examples have been covered.

The *FindBestSeed* procedure selects as the seed example the one with the lowest distance-to-solution. The *DiscoverGoals* procedure searches for best goals in a given example. It starts with an empty goal and iteratively adds subgoals (selecting from all possible subgoals, see section 2) until we find a *good* goal. If there are several *good* goals having the same number of subgoals, then the method returns all *good* goals. The *LearnRule* procedure creates for each provided goal a data set containing all examples from *ES*, where this goal is achievable. Each example in the new data set is labeled as either a *good* goal or as a *bad* goal. Afterwards, *LearnRule* procedure learns a single rule from each data set and selects the best among them. We use the CN2 algorithm to learn a rule.

We extended the above algorithm with the capability to use *arguments* as in argument-based machine learning (ABML)[6]. Arguments are provided by an expert to explain single learning examples – we call such examples *argmented examples*. The task in ABML is to find a hypothesis that is consistent with learning examples and arguments. In goal-oriented rule learning, an argument has the following structure: “*argGoal* because *argConditions*,” where an expert expresses his or her opinion that the goal *argGoal* is *good* in the selected state, because the conditions *argConditions* hold.

We developed an iterative loop that asks the expert to explain only critical examples, *i.e.*, examples not covered by any sufficiently good rules. Such loop significantly decreased the required effort of the expert; he needed to explain only a few examples instead of all. Due to space limitations, we only presented an overview of the ABML extension (see [1] and [5] for more details).

4 Evaluation

We used our algorithm for the conceptualization of procedural knowledge required to deliver checkmate in the KBNK chess endgame. KBNK (king, bishop, and knight vs. a lone king) is regarded as the most difficult of the elementary chess endgames. The stronger side can always checkmate the opponent, but even optimal play may take as many as 33 moves. There are many recorded cases when strong players, including grandmasters, failed to win this endgame. In an interactive procedure between a chess teacher (a FIDE master of chess) and the

computer, we derived instructions in the form of goals for delivering checkmate from any given KBNK position (see [1] for details). The result of this procedure was an ordered set of eleven rules.

The rules were used to compile *teaching materials* for playing KBNK: textbook instructions, supplemented with five example games.¹ They were presented to three chess teachers (among them a selector of Slovenian women's squad and a selector of Slovenian youth squad) to evaluate their appropriateness for teaching chess-players. They all agreed on the usefulness of the presented concepts and found the teaching materials suitable for educational purposes. Among the reasons to support this assessment was that the instructions "clearly demonstrate the intermediate subgoals of delivering checkmate." [1]

We further assessed the teaching materials with the following pilot experiment with three students – chess beginners of slightly different levels – who played several KBNK games against a computer. The computer was defending "optimally," *i.e.*, randomly choosing among moves with the longest distance to mate (using chess tablebases). The time limit was 10 minutes per game. Each game started from a different starting position, all mate-in-30-moves or more. The moves and times spent for each move were recorded automatically.

At the beginning of the experiment, each student played three games against the computer, and they always failed to deliver checkmate. They clearly lacked procedural knowledge for successfully delivering checkmate in this endgame before seeing the teaching materials.

Next, the students were presented with the teaching materials. They were reading the instructions and observing the example games until they felt they are ready to challenge the computer once again. None of them spent more than 30 minutes at this second stage.

In the final stage of the experiment, the students were again trying to checkmate the optimally defending computer. The textbook instructions and example games were not accessible to the students during the games. Only if a game ended in a draw, the student was again granted the access to the teaching materials for up to ten minutes before starting a new game. While the first student (a slightly stronger chess player than the other two) successfully checkmated in the second game already, the other two students checkmated in games 5 and 6, respectively. Once they achieved the win the students had no problems at all achieving it again, even with the white bishop being placed on the opposite square color than in all previous games.

Although the goal of the conceptualized procedural knowledge included in the textbook instructions is not to teach students how to play "optimally," but merely to enable them to achieve a step-by-step progress towards delivering checkmate, it is particularly interesting that the second student in his third game of the final stage of the experiment played 22(!) optimal moves in a row – an achievement that a chess grandmaster could be proud of. Moreover, it happened in less than an hour after he was first given access to the textbook instructions and example games. This result would be very hard or even

¹ The teaching materials are available at <http://www.ailab.si/matej/KBNK>

impossible to achieve without an effective way of memorizing particular concepts of procedural knowledge required in order to master this difficult endgame.

5 Conclusions

We presented a novel algorithm for semi-automated conceptualization of procedural knowledge based on goal-oriented rules in symbolic domains. We applied the algorithm to the challenging KBNK chess endgame, and carried out a pilot experiment to evaluate whether the obtained concepts (instructions) could serve as a teaching tool. Somewhat surprisingly, even the beginner-level chess players were able to quickly grasp the concepts, and learn to deliver checkmate. A separate, subjective evaluation of the instructions by experienced chess trainers was also positive.

A more rigorous evaluation is an obvious task for further work. Apart from other domains, it should be evaluated whether the derived concepts could serve as the knowledge base of an ITS. To this end we plan to build such a system, and conduct an experiment on a much larger number of students.

References

1. Guid, M., Možina, M., Sadikov, A., Bratko, I.: Deriving Concepts and Strategies from Chess Tablebases. In: van den Herik, H.J., Spronck, P. (eds.) *ACG 2009*. LNCS, vol. 6048, pp. 195–207. Springer, Heidelberg (2010)
2. Matsuda, N., Keiser, V., Raizada, R., Tu, A., Stylianides, G., Cohen, W.W., Koedinger, K.R.: Learning by Teaching SimStudent: Technical Accomplishments and an Initial Use with Students. In: Alevan, V., Kay, J., Mostow, J. (eds.) *ITS 2010*. LNCS, vol. 6094, pp. 317–326. Springer, Heidelberg (2010)
3. Michalski, R.S.: A theory and methodology of inductive learning. *Artificial Intelligence* 20(2), 111–161 (1983)
4. Možina, M., Guid, M., Krivec, J., Sadikov, A., Bratko, I.: Fighting knowledge acquisition bottleneck with argument based machine learning. In: Patras, G. (ed.) *Proceedings of 18th European Conference on Artificial Intelligence (ECAI 2008)*, pp. 234–238. IOS Press, Patras (2008)
5. Možina, M., Guid, M., Sadikov, A., Groznic, V., Krivec, J., Bratko, I.: Conceptualizing procedural knowledge targeted at students with different skill levels (2010) (unpublished), <http://www.ailab.si/martin/abml/gorules.pdf>
6. Možina, M., Žabkar, J., Bratko, I.: Argument based machine learning. *Artificial Intelligence* 171(10/15), 922–937 (2007)
7. Murray, T.: Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education (IJAIED)* 10, 98–129 (1999)
8. Tadepalli, P.: Learning to solve problems from exercises. *Computational Intelligence* 24(4), 257–291 (2008)
9. Tecuci, G., Boicu, M., Boicu, C., Marcu, D., Stanescu, B., Barbulescu, M.: The disciple-RKF learning and reasoning agent. *Computational Intelligence* 21(4), 462–479 (2005)
10. Woolf, B.P.: *Building Intelligent Interactive Tutors: Student-centered strategies for revolutionizing e-learning*. Elsevier & Morgan Kaufmann, Burlington (2008)

Context-Dependent Help for Novices Acquiring Conceptual Systems Knowledge in DynaLearn

Wouter Beek¹ and Bert Bredeweg²

¹ Informatics Institute, University of Amsterdam
beek@uva.nl

² Informatics Institute, University of Amsterdam
b.bredeweg@uva.nl

Abstract. In Interactive Learning Environments for conceptual knowledge, novice learners need support with understanding the used qualitative vocabulary, interpreting the simulation results, and knowing which modeling tasks can be performed. We explain how we generate support automatically, without making a priori assumptions regarding the domain knowledge involved. Assistance is contextualized and generated on the fly, even for complex models and simulations.

Keywords: qualitative reasoning, conceptual modeling, support knowledge, help systems, causal explanation.

1 Introduction

DynaLearn [2] is an Integrated Learning Environment (ILE) for conceptual modeling that allows learning by constructing and simulating causal models.¹ Using Qualitative Reasoning (QR) techniques, DynaLearn provides domain-independent and formal means to externalize thought, thereby fostering the learner's beliefs about how a system behaves and why it behaves that way. Since the modeling language is very powerful, it introduces a host of concepts and tools, resulting in a steep learning curve for novice learners.

In order to support learners in their conceptual modeling attempt, we have implemented context-sensitive support facilities. A menu of questions that can be posed is dynamically generated. The answers to these questions are also automatically generated and include follow-up questions that disclose more in-depth information.[8] Having these basic help facilities integrated into the learning environment, and having them dynamically adapt to the learner's interactions, provides a scaffold for novice learners. The basic help is meant to be complementary to existing pedagogical instruments, and ensures a learner has sufficient foreknowledge in order to understand more complicated feedback facilities in DynaLearn. For these more basic support tasks, existing QR-based ILEs provide fixed learning resources (e.g. VModel [4], Betty's Brain [6], Model-It [3]). The goal and added value of our approach is to automate and custom-tailor this kind of support.

¹ This work is co-funded by the European Commission within the 7th Framework Programme project no. 231526. <http://www.dynalearn.eu>

2 Principles for Basic Help

Basic help is help that requires no foreknowledge regarding conceptual modeling or system dynamics thinking. It provides the propaedeutics for more advanced modes of assistance and tutoring. Since basic help must be blended with more advanced pedagogical modules, it is designed to be inherently complementary and non-obtrusive. In order to realize this the help modes are developed according to the following four principles:

Conciseness. Individual help messages must be short (i.e. one to three sentences) and focused. A message is focused if a novice learner is able to directly relate the help message's content to something s/he is working on at that very moment. Also, we want the cognitive load of processing the help message to be minimal, reducing the likelihood that processing help will obfuscate the learner's actual task.

Self-containment. A message is self-contained if its contents are self-descriptive. Self-descriptive contents can be understood without having to relate to resources that are not available in the message itself.

Completeness. Even though conciseness dictates that each individual communication be short and self-contained, help must allow *all* knowledge that is inside the learning environment or in the learner's model to be communicated upon further requests from the learner.

Context-dependence. Help message must relate directly to the learner-created model. As such, the automatically generated questions must only include requests that make sense within the current context.

3 Implementation of the Basic Help

This section explains the implementation of the basic help facilities, in line with the principles in section 2. We first discuss what all basic help modes have in common, and then zoom in on two of them.

Because of conciseness, each help message consists of 1 to 3 sentences. A message includes links to its related resources. The sentences are communicated in natural language by a virtual character that communicates the help results verbally (using text-to-speech), non-verbally, and in written form. The text is shown in a speech bubble and is displayed beside the content that the learner is working on. To relate the contents of a help message to on-screen elements in the ILE the character uses gesticulation, facial expression and a laser pointer.

The knowledge resources that the help modes use internally, as well as the generated messages themselves, are represented using Semantic Web techniques, allowing for explicit semantics. Every item that can be the onus of a help request is assigned a unique URI. The help modes are added on top of the legacy modeling environment Garp3. [1]

Messages are automatically generated, including links for possible follow-up requests. These follow-ups consist of URIs as well. Messages are self-contained RDF-documents with self-descriptive natural language contents, generated via

pattern matching or via a context-free grammar that is guided by the semantic relations of the RDF-graph. A glossary of QR and dynamic systems terminology is included as well.

The dynamically generated follow-up links put the knowledge a message expresses into a broader context of related help messages. The way in which the help messages and their interrelations are generated reflects the compositional nature of the QR modeling formalism. The compositional nature of QR models ensures that all knowledge can be reached by allowing the learner to traverse the graph of interconnected messages, thus ensuring completeness. The help modes also link to each other, making the network of interconnections more valuable by relating different types of knowledge.

As the learner uses the DynaLearn ILE, creating models and running simulations, applicable help requests are continuously generated inside a hierarchically structured menu. The learner can choose to access the various help modes from this menu. A basic help interaction is never forced on the learner, since this would conflict with its non-obtrusive and complementary purpose. Which questions are displayed in the menu depends on the state of the DynaLearn ILE and on the status of the learner-created model, which is extracted from the modeling environment and is transcribed to a First-Order Logic (FOL) representation.

We distinguish between three basic help modes related to different types of information: “How to?”, “What is?”, and “Why?”. The “What is?”-mode explains the modeling vocabulary in terms of the learner-created content. A conceptual model in DynaLearn consists of domain-specific assertions embedded in the generic modeling language vocabulary. Each expression created by a learner is a subtype and/or a refinement of the latter. The “What is?”-help mode is able to describe occurrences of every element that occurs in a model. Details of the workings of the “What is?”-mode are given in [9].

3.1 Model Building Task Support (How To?)

The “How to?”-mode explains how to perform tasks within the learning environment. Because there are many tasks (102 main tasks consisting of 2 to 8 subtasks), not all requests can be displayed at once. All tasks are relevant some of the time (e.g. save changes), and some tasks are relevant all of the time (e.g. quit the application), but not all tasks are relevant all of the time.

In order to not overwhelm the learner, only tasks that can be performed given the current state of the ILE are communicated in the “How to?”. Each model construction task S belongs to a static task ontology T and consists of a sequence of subtasks $\langle S_1, \dots, S_n \rangle$. Subtasks have associated preconditions $\text{Pre}(S_i)$ represented as FOL statements that are satisfied against the formal description of the learner-created model M . Task S can be performed if $\text{Pre}(S_i)$ is true for at least one of S 's subtasks S_i . Only tasks that can be performed are shown, and only subtasks that are not yet performed are included in the help message (see formula 1).

$$\{\langle S_{i+1}, \dots, S_n \rangle \mid \langle S_1, \dots, S_n \rangle \in T \wedge \exists!_{1 \leq i < n} M \vdash (\text{Pre}(S_i) \wedge \neg \text{Pre}(S_{i+1}))\} \quad (1)$$

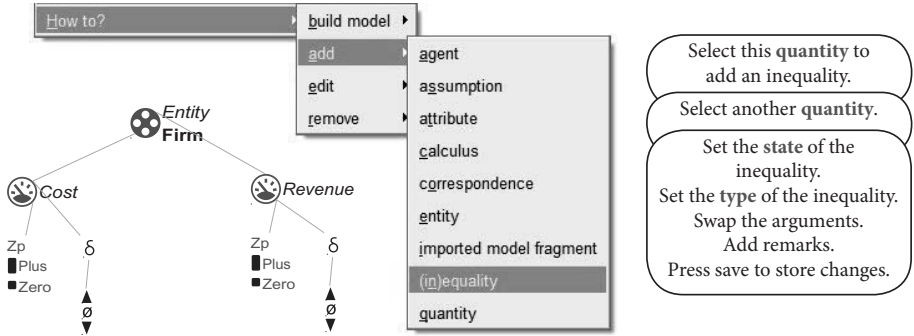


Fig. 1. The options for the “How to?”-functionality for the model fragment on the left. In this example the learner requests how to add an inequality relation (e.g. stating that costs > revenue). Three subtasks for this request appear to the right.

Besides preconditions (i.e. aspects of M that allow a subtask to be performed), each subtask also has post-conditions (i.e. changes to M due to performing a subtask) that ensure the formal description of the learner-model M stays up-to-date. A subtask is communicated once the learner has satisfied its preconditions. Performing the subtask brings about its post-conditions, triggering the preconditions of the next subtask (if any), etc. In this way information regarding individual subtasks (principle of conciseness) are communicated one-by-one at precisely the right moment (principle of context-dependence).

3.2 Causal Explanations (Why?)

The “Why?”-mode gives information about the simulation results (i.e. why some behavior occurs). These are distributed over time, and include ambiguous behavior represented as branching temporal states. Having learners understand the causal behavior of complex systems over multiple simulation states is difficult. This is one of the reasons why ILEs based on QR use single, within-state simulation (e.g. Betty’s Brain [6], VModel [4]). DynaLearn has taken a more advanced approach, making it possible to simulate causal behavior over an arbitrary number of states.

Questions can be posed as to why values (i.e. quantity values and quantity derivatives) and inequality relations (between quantities or between values) occur. Explanations can involve a causal chain of considerable length, making it difficult to meet both the conciseness and the completeness principles for all cases. Also, there may be multiple explanations, some of which are more important than others. Finally, explanations should be given at the right level of detail.

Each message explains a single reasoning step (conciseness principle). In order to ensure close resemblance with help provided by human experts, the experimentally established stock of reasoning steps identified by [5] is used. The reasoning

steps are embodied in reasoning components. A component explains the value of a single value or inequality (the component’s output or conclusion) using an arbitrary number of other values and/or inequalities (the component’s inputs or antecedents). The procedure that the component uses to calculate the output from its inputs is described by an additional procedural knowledge input.

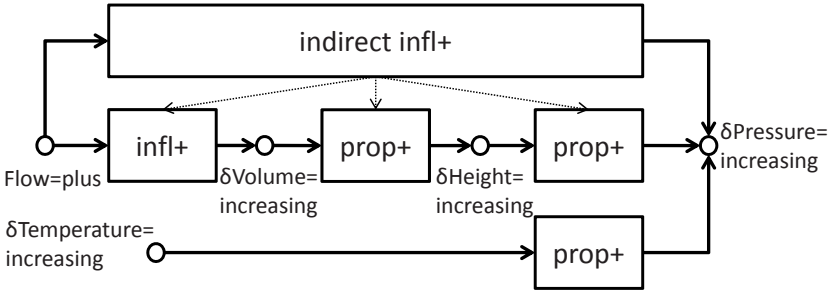


Fig. 2. An example of part of a point/component-representation that is used to answer “Why?”-requests. The boxes are components that represent causal inferences. The points are magnitude or derivative values of quantities. There are two explanations as to why the pressure is increasing (δ means ‘derivative’). Explanation one uses the aggregate component (at the top), stating that flow indirectly influences pressure (+ indicates a positive influence). Explanation two states that temperature propagates to pressure. Explanation one is decomposable (dotted lines indicate decomposition), stating that height propagates to pressure. For this last explanation a follow-up request exists that asks why height is increasing, etc., thus traversing the causal chain.

Based on the simulation results, a circuit-like representation is created. The values and inequalities are represented as points. The reasoning steps are represented as components. Points and components are connected forming an explanatory graph (see figure 2). In a help message the inputs deliver the premises and the output delivers the conclusion. Giving an more in-depth explanation amounts to traversing the point/component-circuit on a per-component basis, starting from the requested data point and reasoning backwards through the circuit. Follow-up requests take an input point from the previous component and explain it as the conclusion of a new component.

Multiple explanations of the same datum are represented as multiple components that are connected to a single point. Not all explanations are equally relevant for a learner and we do not want to communicate each of them (conciseness principle). Alternative explanations are ranked based on importance values assigned to each component type. For instance, influence components (causation) are ranked higher than correspondence components (additional constraints).

Explanations of varying abstraction levels are generated using aggregated components. An aggregate component explains the same output datum, but uses premises that are farther away (i.e. multiple decompositions deep). Aggregate components abstract the intermediary data points away. An example of an aggregation is a causal link that consists of one influence followed by an

arbitrary number of proportionalities. These are grouped into a single, aggregated causal component. The importance value of an aggregated component is based on the importance values of its subservient components, thereby favoring abstract explanations in case alternative explanations exist.

Follow-up requests can also be posed for each message that is generated based on an aggregate component. These follow-ups give a more detailed explanation of the same material. They are generated by ‘unpacking’ the aggregated component, resulting in multiple low-level components with additional in-between value and/or inequality points.

4 Concluding Remarks

We showed that it is possible to integrate automated assistance in the DynaLearn ILE such that communicated messages are concise and self-contained, all available basic knowledge is covered and the context is taken into account for both question and answer generation. These basic help facilities provide a scaffold for novice learners. Usage evaluation studies of DynaLearn with learners are reported in [7].

References

1. Bredeweg, B., Linnebank, F., Bouwer, A., Liem, J.: Garp3: Workbench for Qualitative Modelling and Simulation. *Ecological Informatics* 4(5-6), 263–281 (2009)
2. Bredeweg, B., Liem, J., Linnebank, F., Bühling, R., Wißner, M., Gracia del Río, J., Salles, P., Beek, W., Gómez Pérez, A.: DynaLearn: Architecture and Approach for Investigating Conceptual System Knowledge Acquisition. In: Aleven, V., Kay, J., Mostow, J. (eds.) *ITS 2010. LNCS*, vol. 6095, pp. 272–274. Springer, Heidelberg (2010)
3. Jackson, S., Stratford, S., Krajcik, J., Soloway, E.: Model-It: A case study of learner-centered design software for supporting model building. In: *Proc. from the Working Conference on Applications of Technology in the Science Classroom* (1995)
4. Forbus, K., Carney, K., Sherin, B., Ureel II, L.: VModel: A Visual Qualitative Modeling Environment for Middle-School Students. *AI Magazine* 26(3), 63–72 (2005)
5. De Koning, K., Bredeweg, B., Breuker, J., Wielinga, B.: Model-Based Reasoning about Learner Behaviour. *Artificial Intelligence* 117(2), 173–229 (2000)
6. Leelawong, K., Biswas, G.: Designing Learning by Teaching Agents. The Betty’s Brain System. *Int. J. of AIED* 18(3), 181–208 (2008)
7. Mioduser, D., Zuzovsky, R.: Evaluation of DynaLearn: First Phase Insights. Deliverable D7.2.6. DynaLearn, EC FP7 STREP project 231526 (2012)
8. Mittal, V., Moore, J.: Dynamic Generation of Follow-up Question Menus: Facilitating Interactive Natural Language Dialogues. In: *Proc. of the 8th NCAI*, pp. 90–97 (1995)
9. Wißner, M., Häring, M., Bühling, R., Beek, W., Linnebank, F., Liem, J., Bredeweg, B., André, E.: Basic Help and Teachable Agent. Deliverable D5.3. DynaLearn, EC FP7 STREP project 231526 (2010)

Towards an Ontology-Based System to Improve Usability in Collaborative Learning Environments

Endhe Elias¹, Dalgoberto Miquilino^{1,2}, Ig Ibert Bittencourt¹, Thyago Tenório¹, Rafael Ferreira³, Alan Silva¹, Seiji Isotani⁴, and Patrícia Jaques⁵

¹ Center of Excellence in Social Technologies, The Computing Institute, Federal University of Alagoas (UFAL), Brazil

² Maurício de Nassau Faculty, Brazil

³ Federal University of Pernambuco, Brazil

⁴ University of São Paulo (USP), Brazil

⁵ PIPCA, University of Vale do Rio dos Sinos (UNISINOS), Brazil
endhe.elias@ic.ufal.br, dalgoberto.pinho@mauriciodenassau.edu.br

Abstract. The systems usability has been the subject of increasing discussion for several decades and when this concerns to computer support collaborative learning (CSCL) environment, it becomes a harder task. In CSCL environments, the usability is evaluated based on the the technical and pedagogical aspects. Therefore, the literature discusses different methods and techniques to validate usability in such environments. However, these methods usually need to be applied manually. The manual process use to be very expensive, very specific and time consuming. In addition, in CSCL environments, the usability validation becomes even more difficult due the existence of several usability requirements and its high level of interaction. For this reason, there is a need to automate the process. On the one hand, the technical usability can be automated after a formal description of the environment features. On the other hand, the pedagogical usability is dependent on the user experience and it is not possible to be totally automated. This paper presents a semi-automatic validation system to improve usability in CSCL environments. It uses ontology to represent the usability knowledge and software agents to automate the process. Finally, a case study in a real environment is described to present the advantages of using the proposed system.

Keywords: Technical and Pedagogical Usability, CSCL Environments, Ontology, Software Agent.

1 Introduction

The teaching process has the potential to become more active, dynamic and personalized through computer support collaborative learning (CSCL) environments. Moreover, CSCL plays an important role in learners performance, for example, it has been suggested that CSCL helps students to facilitate high order cognitive processes and to create new knowledge [3].

When they have low usability these environments may hamper the interaction, causing a high degree of negative experiences. On the other hand, a good usability can emphasize high levels of participation of students and improve the learning process. Therefore, it is very important to take into account the usability during the development of CSCL systems. In CSCL environments, the evaluations should consider two types of usability: technical and pedagogical usability. On the one hand, technical usability addresses the technical interfaces enabling the development focused on the audience. On the other hand, pedagogical usability is associated with the educational materials and course preparation.

In this context, this paper presents a semi-automatic evaluation system to improve usability in CSCL environments. This system uses usability methods and techniques presented in the literature to create rules to deal with usability problems. This system considers the automatic inspection to evaluate the technical usability and questionnaires to evaluate the pedagogical usability. It also uses the user interaction to suggest new usability rules that can be added in real time. To accomplish this and to automate the process the system uses ontology and software agents. A case study in a real environment is described to present the advantages of using the proposed system.

2 System Proposal

This section aims to present the system features, describing each component of the system and the interaction among them. As presented in Figure 1, the proposed system has uses semantic technologies (i.e. ontologies) to represent the domain knowledge and software agents to automatically perform the inspection rules into the ontology. The ontology represents the CSCL environment features under the usability perspective. In addition, the system uses to pedagogical usability validation. With these components, the report generator component uses the agent and questionnaires output the generate a report in order for the team to improve the environment usability. Each system component is detailed in the next subsections.

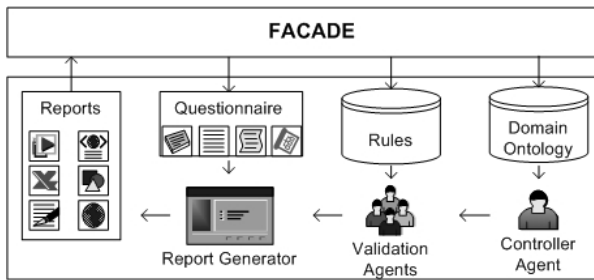


Fig. 1. System Components Overview

2.1 Domain Ontology

In order for the system to ensure the consistence of the inspection and the report generation, an ontology for representing usability into CSCL environments was modeled. The ontology represents concepts, relationships, and axioms related to the knowledge domain. The specification of the ontology is presented in Table 1.

Table 1. Domain Ontology

CSCL \sqsubseteq has_pages Pages CSCL \sqsubseteq presentation Boolean	CSCL \sqsubseteq color_consistency Boolean
Pages \sqsubseteq has_container Container Pages \sqsubseteq description String Pages \sqsubseteq has_alert Boolean Pages \sqsubseteq language String Pages \sqsubseteq title String	Pages \sqsubseteq presentation Boolean Pages \sqsubseteq everyday_expression Boolean Pages \sqsubseteq id String Pages \sqsubseteq layout String
Container \sqsubseteq has_item Item	Container \sqsubseteq layout_Container String
Item \sqsubseteq has_link Link Item \sqsubseteq has_form Form Item \sqsubseteq has_image Image	Item \sqsubseteq has_paragraph Paragraph Item \sqsubseteq has_buttons Buttons Item \sqsubseteq has_block Block
Block \sqsubseteq has_Item Item Block \sqsubseteq has_Tool Tool	Block \sqsubseteq has_Link Link

The main ontology concepts are described as follows: i) **CSCL**: it has a brief environment description and contains all the pages of the environment (*has_pages*); **Pages**: this entity contains page description (*presentation, description, language, id, title*), page design information (*layout*), alerts (*everyday_expression, has_alert*), and containers (*has_container*); **Container**: the container has the resources (*Item*) that can be used in the CSCL environment. It is important to say that a container may have more than one *Item*; **Item**: Items contain the features of the environment (e.g. text, forms, buttons, images, tools); **Block**: it contains item sets. The idea is to facilitate the grouping on the page.

2.2 Controller Agent and Validation Agent

The software agents are used to assess the environments at run time. If some changes occur in the domain ontology or in the rules the agents automatically perform the rules to inspect the usability of the environment. The system provides two types of software agents which are used to perform automatic verification.

The *Validation Agent* is responsible for validating the environment based on usability rules (described in the Section 2.4). After the validation, this agent generates a report with the usability issues and their status. As a result, the information obtained through this agent are used to produce reports to the system administrator.

The *Controller Agent* monitors the directory that stores the domain ontology and rules. Thus, for each new environment that is represented in the ontology, the controller agent creates a new validation agent. Therefore, when a CSCL administrator wants to create a new usability validation it is not necessary to run the system, he just needs to update the domain ontology. It is important to highlight that for each new environment a new validation agent is instantiated.

2.3 Questionnaire

In order to complete the assessment generated by the software agents the system uses a questionnaire, which should be answered by the students. The questionnaire was designed based on pedagogical and technical usability issues according to [1,2]. Broadly speaking, it is used to confirm the issues raised by the software agents and to better understand the student's behavior.

The questionnaire is composed by 36 questions and aims at providing both technical and pedagogical usability information, such as the content granularity, the quality of the content and if it is easy to use, the quality of the layout, the appropriateness of the tool during activities, the interaction of the user, the kind of pedagogical activity, and others.

2.4 Rules and Validation Process

A collection of usability rules were created taking into account pedagogical and technical aspects. It was created based on three aspects: (i) rules presented in the literature; (ii) rules made by experts; and (iii) rules inferred by the system.

This work has a knowledge base which contains 72 usability issues. Some examples of them are: the environment does not use jargon abbreviation or unknown expressions; the colors of links are consistent with the web conventions; users like the activities on the environment, and so on. The usability issues were made based on [1,2]. The Table 2 shows some usability issues and their description. It is important to say that the numbers for each usability issue presented in Table 2 represents a usability axiom. The mapping is based on the aforementioned works.

Table 2. Usability Category - Usability Issue

Usability Category	Usability Issue
1 - Pedagogical Usability	1, 2, 4-12, 14, 19, 20, 21, 25-29, 56-72
2 - Technical Usability	3, 13, 15-18, 22-24, 30-47, 67
3 - Technical and Pedagogical Usability	48-55

For each usability issue in usability category 1, the system has a question related to it. This question are available on the questionnaire. These questions are answered by the users. In addition, for each usability issue present in usability

categories 2 and 3, the system maps the usability issue to a set of rules described with SPARQL¹. Therefore, there are two validation methods: questionnaire, described in Subsection 2.3 and SPARQL rules performed by the agents described in Subsection 2.2.

3 Case Study

This section describes the case study applied to evaluate the system. The case study was executed in a real collaborative learning environment, Moodle² and it is used as the learning environment at Federal University of Alagoas - Brazil.

The goal of the case study is to apply the system in order to evaluate the technical and pedagogical usability of a course. For this reason, the questions related to the case study are: (1) how adequate is the technical usability in the course/environment? and (2) how adequate is the pedagogical usability of the system? (3) Which aspects of technical and pedagogical usability are inadequate?

The questionnaire contains 36 questions of usability evaluation, each question related to pedagogical usability issue. Moreover, each question was inspected to verify its applicability with regards to the learning material developed for the case study. The questionnaire was available online to the users, and 15 users answered it. For each question was quantified students' agreement or disagreement about the specific aspects of pedagogical usability. The higher was the agreement with the question answered by the student, the more important this pedagogical usability issue was considered.

The validation of usability category 2 and 3 (see subsection 2) were made through the execution of rules by the validation agent. In order to do that, the domain ontology was populated with instances. In these categories, the issues were considered adequate or non-adequate, and this status was obtained through the presence or absence of the object instance or interface feature of the learning environment.

After that, the report generator compiles the outputs into reports. The current version of the system has graphical reports and check lists with regards the usability issues.

The next stage was the validation of usability category 2 and 3 through the rules and intelligent agents described above. The Table 3 shows the result of some usability issues and the status after execution of the rules.

The results obtained through questionnaire were added with the results obtained by execution agents, in order to generate the usability reports. Therefore, the results obtained by usability evaluation of the three categories are shown to system administrator through the check list (Table 3). In addition, this work checked which usability issues were considered most critical from the point of users view.

¹ SPARQL Query Language for RDF is W3C recommendation since January 2008 - <http://www.w3.org/TR/rdf-sparql-query/>

² More details: <http://moodle.com/>

Table 3. Result of Rules Validation

Usability Issue	Status
The feedback (warning/response given by the system) is immediate	Non-Adequate
There is a common form of presentation and content organization used in all environment	Adequate

4 Conclusion and Future Work

This paper presented a validation system to improve usability in Computer Support Collaborative Learning Environments. To accomplish it, the system uses usability rules according to the literature, rules made by experts and rules inferred by the system. These rules validated both technical and pedagogical aspects. The system works in a semi-automatic way. On the one hand a group of agents interact with the ontology and perform the rules to evaluate the environment usability. On the other hand, a questionnaire is submitted to the students in order to obtain their opinions about the usability. As a result, the system provides reports in order to help the administrator to improve the usability of the CSCL environment. As future works, the authors intend to: i) extract the relevant features of the CSCL environment and populate the domain ontology automatically, ii) create a guide to recommend good usability practices, iii) improve the usability rules and iv) evaluate with different CSCL environments.

Acknowledgements. We would like to thank UFAL, CAPES (scholarship BEX 4025-11-3), CNPq, Faperqs, UNISINOS and USP for their support in this research.

References

1. Nokelainen, P.: An empirical assessment of pedagogical usability criteria for digital learning material with elementary school students. *Educational Technology and Society* 9(2), 178–197 (2006)
2. Sharp, H., Rogers, Y., Preece, J.: *Interaction Design: Beyond Human-Computer Interaction*. Wiley, NJ (2007)
3. Wang, S.-L., Hwang, G.-J.: The role of collective efficacy, cognitive quality, and task cohesion in computer-supported collaborative learning (cscl). *Computers & Education* 58, 679–687 (2012)

Program Representation for Automatic Hint Generation for a Data-Driven Novice Programming Tutor

Wei Jin¹, Tiffany Barnes², John Stamper³, Michael John Eagle²,
Matthew W. Johnson², and Lorrie Lehmann²

¹Shaw University, Raleigh, NC, USA
weijin.sz@gmail.com

²University of North Carolina at Charlotte, NC, USA
tiffany.barnes@uncc.edu

³Carnegie Mellon University, Pittsburgh, PA, USA
john@stamper.org

Abstract. We describe a new technique to represent, classify, and use programs written by novices as a base for automatic hint generation for programming tutors. The proposed linkage graph representation is used to record and reuse student work as a domain model, and we use an overlay comparison to compare in-progress work with complete solutions in a twist on the classic approach to hint generation. Hint annotation is a time consuming component of developing intelligent tutoring systems. Our approach uses educational data mining and machine learning techniques to automate the creation of a domain model and hints from student problem-solving data. We evaluate the approach with a sample of partial and complete, novice programs and show that our algorithms can be used to generate hints over 80 percent of the time. This promising rate shows that the approach has potential to be a source for automatically generated hints for novice programmers.

Keywords: Intelligent tutoring systems, automatic hint generation, programming tutors, educational data mining and data clustering.

1 Introduction

Our goal is to create a data-driven intelligent tutor for computer programming using Markov decision processes (MDPs), created from past student data, to generate contextualized hints for students solving a specific problem. This approach has been applied in the logic domain, providing hints for 70-90% of problem-solving steps [Barnes2010a, Barnes2010b, Stamper2011].

To use the MDP approach, we must describe the student's current solution attempt "target" state that can be compared to existing prior attempts, which are potential hint "sources". Jin, et al. proposed linkage graphs to represent novice program states [Jin2011]. In this paper, we present detailed algorithms for automatic linkage graph extraction from programs and automatic hint generation, and our feasibility study to evaluate the approach.

2 Linkage Graphs to Represent Data Flow and Dependencies

A linkage graph for a program is a directed acyclic graph, as shown in Figure 1, where nodes represent program statements and directed edges indicate order dependencies. If statements I and J access the same variable x , and J is the first statement after I that accesses variable x , then J directly depends on I and we add an edge from node I to node J with label x . We call a single trace through the graph a *linkage*, which connects statements that modify the same variable. A *linkage graph* is the combined set of linkages. Representation for control statements are discussed in [Jin2011] and we do not implement this aspect of linkage graphs here. In this section we describe our representation and extraction for linkage graphs. We use a 2-dimensional matrix to represent a linkage graph; Table 1 (left) shows the matrix for the program in Figure 1. Variable v_0 shows up in statements 0, 9 and 10, represented as 1's in the corresponding rows, indicating that variable v_0 's linkage starts with statement 0 and consists of edges (0,9) and (9,10)..

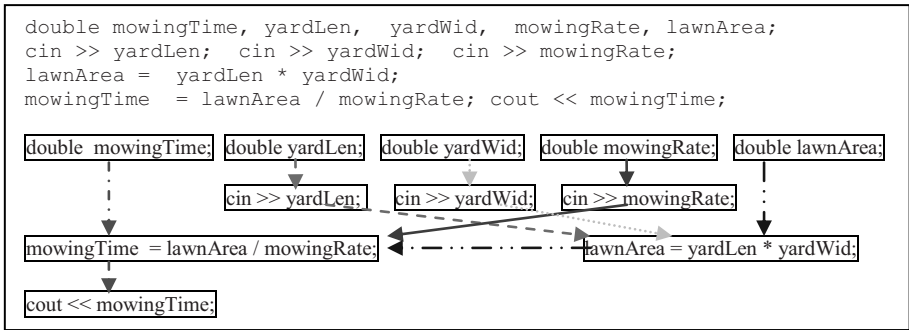


Fig. 1. The linkage graph for a program to calculate money earned for mowing grass. The colored directed edges identify variable dependency between nodes.

Table 1. Linkage Graph Matrices. The left is for the program in Fig. 1; the right is equivalent. (v_0 =mowingTime, v_1 =yardLen, v_2 =yardWidth, v_3 =mowingRate, and v_4 =lawnArea).

	v_0	v_1	v_2	v_3	v_4
0. double v_0 ;	1				
1. double v_1 ;		1			
2. double v_2 ;			1		
3. double v_3 ;				1	
4. double v_4 ;					1
5. cin >> v_1 ;		1			
6. cin >> v_2 ;			1		
7. cin >> v_3 ;				1	
8. $v_4 = v_1 * v_2$;		1	1		1
9. $v_0 = v_4 / v_3$;	1			1	1
10. cout << v_0 ;	1				

	v_0	v_1	v_2	v_3	v_4
0. double v_4 ;					1
1. double v_3 ;				1	
2. double v_2 ;			1		
3. double v_1 ;		1			
4. double v_0 ;	1				
5. cin >> v_3 ;				1	
6. cin >> v_2 ;			1		
7. cin >> v_1 ;		1			
8. $v_4 = v_1 * v_2$;		1	1		1
9. $v_0 = v_4 / v_3$;	1			1	1
10. cout << v_0 ;	1				

Table 1 shows matrices for two programs that differ only in order. Since there are no variable dependencies among statements 0-4 and among 5-7, they are equivalent.

We note that programs with the same output are not necessarily equivalent. For example, $a = b * c / d$ is not equivalent to $t = b * c; a = t / d$. Our goal is that equivalent programs should have the same linkage matrix representation. To accomplish this, we must determine the order of the variables (corresponding to columns of the matrix) and the order of the statements (corresponding to rows of the matrix). The statement order will be determined based on the variable order and the variable dependencies.

Instructor-Provided Specification File: An initial list of variables is taken from an instructor-provided variable specification file for the given programming problem, as shown in Table 2, or could alternatively be generated from the problem description using a bag-of-words approach. Each item specifies a program variable, and consists of three parts: (1) correct data types, (2) phrases that describe the item and may compose the variable name for that item, and (3) how the variable is assigned a value, with the keyword ‘input’ indicating user-entered values. In Table 2, a slash means “or” – either one of them may be present in the name.

Table 2. A Possible Variable Specification File for the Programming Problem in Figure 1

Name	Types	Variable Name Terms	Assignment
v_0	float, double	mowing time/hours	$v_1 * v_2 / v_3$
v_1	float, double	yard/lawn length	Input
v_2	float, double	yard/lawn width	Input
v_3	float, double	mowing rate/speed	Input
v_4	float, double	yard/lawn area	$v_1 * v_2$

Assigning Variables and Extending the Variable Specification: Meaningful variable names, such as *yardLength* or *yardLen*, are a common requirement in introductory programming courses. A preliminary analysis of novice programs shows that students choose variable names in this fashion. Second choice names are also common. This suggests that a simple list of all the variable names could be aggregated from all programs and compared to the instructor specification file. For those matching the specification, they are assigned the given variable names. If any remain, we can compute simple similarity and thesaurus lookups to determine if any match to the existing variables or one another. We can cluster the remaining variables and add representative variables to the variable specification.

Variable Normalization: In order to avoid the problem of having a program categorized as different simply because of varying names for variables, we normalize variable names. The variable specification file determines the variable order and normalized names. If a variable name is ambiguous, for example, *length* may refer to yard length or house length, we can use how the variable is used to determine its purpose. If programs are collected in an interactive environment, we could also ask the student which data item the variable refers to.

Statement Sorting: After variables are normalized, the statements will be sorted. Statement sorting consists of three steps. *Step 1 – Preprocessing.* We break a declaration statement for multiple variables into multiple declaration statements, with each declaring only one variable; we do the same for input and output statements. We also break a declaration with initialization into a declaration and an assignment.

Step 2 – Create statements sets according to variable dependencies. The first set consists of statements that do not depend on any other statement. The second set

consists of the statements that depend only on those from the first set. The third set consists of the statements that depend only on those from the first and second sets, and so on. For example, for the programs in Table 1, the first set consists of statements 0 – 4, the second set 5 – 7, the third set 8, the fourth set 9, and the last set 10.

Step 3 – Within each set, statements are sorted in the decreasing order of their variable signatures. Assume that there are n data items in the variable specification file. A statement’s variable signature is $s_0s_1\dots s_{n-1}$, where s_i is 1 if the normalized variable name v_i is in the statement and 0 otherwise. For example, the sorted version for the matrix in Table 1 (right) is the matrix shown in Table 1 (left).

Linkage Matrix Representation Uniqueness: The matrix columns are labeled and ordered by normalized variable names. The rows are labeled and ordered by sorted statements. Step 2 guarantees the equivalency of the new program to the original; sorting ensures that equivalent programs have the same matrix representation.

3 Hint Generation for Work-in-Progress Programs

The first step in hint generation for a programming problem is to collect a set of correct solutions from previous students. Then we build linkage graphs for these model solutions. They serve as the sources for hint generation. New solutions may be added to the set. We also build linkage graphs for intermediate states (e.g. program snapshots saved when the compile button is pressed), which are linked by directed arcs that indicate the order the program was written. Each complete program results in a sequence of states illustrating each step in development. These sequences are composed into a single large graph, with equivalent states (linkage graphs) mapped to one another. We then assign a reward value to each state (say 1 point for each linkage) and the correct solutions (say 100), and apply value iteration to create a Markov Decision Process [Barnes2010b]. The linkages act as state features for the states and the reward function computes the state value based its closeness to being complete.

When a student requests a hint, the tutor will build a linkage graph for the partial program. The tutor will find a linkage graph in the MDP that is closest in structure, or a ‘match’. When a student’s state is matched in the MDP, the MDP allows us to select the next best state by choosing the one with the highest value. We may generate a hint based on the next best state in the MDP or on the final complete solution if the student were to follow the path with the highest values at each step. Suppose that for the partial programs in Table 3, the complete linkage graph as the source for hint generation is Table 1 (left). A partial linkage graph matrix has the same underlying structure as the complete linkage graph: The statements and variables are in the same order as those in the complete graph. The numbers in the matrices (Table 3) represent the order of the statements in the partial programs. We can use missing items or items with wrong orders from the complete graph to generate hints. For example, $v_4 = v_1 * v_2$ is missing from Table 3 (left), so the hint might be “*Calculate $v_1 * v_2$ instead of v_1 / v_2* ”. In Table 3 (right), $cin >> v_1$ and $cin >> v_2$ are after $v_4 = v_1 * v_2$, so the hint might be “*cin >> v_1 and cin >> v_2 should be before $v_4 = v_1 * v_2$* ”. Note that when generating hints, we use student variable names (e.g. *yardLen*) instead of normalized names (e.g. v_0 and v_1).

No Matching State Found: Linkage Graph Transformation. If the work-in-progress solution takes a different approach from all existing correct solutions, we have to determine whether any of the existing complete solutions can be modified to

fit the current work-in-progress program. Table 4 shows how we expand the source linkage graph to match the target partial program by adding new rows (and columns) and splitting existing ones as needed. Once this transformation occurs, the new linkage graph can be compared to the partial program to generate hints. This allows us to provide hints right away with a provided expert solution.

Table 3. The Linkage Matrices for the Partial/Incorrect Programs

... cin >> v ₁ ; cin >> v ₂ ; cin >> v ₃ ; v ₄ = v ₁ / v ₂ ; // wrong expression						... v ₄ = v ₁ * v ₂ ; // wrong order cin >> v ₁ ; cin >> v ₂ ; cin >> v ₃ ;					
	v ₀	v ₁	v ₂	v ₃	v ₄		v ₀	v ₁	v ₂	v ₃	v ₄
0. double v ₀ ;						0. double v ₀ ;					
1. double v ₁ ;		1				1. double v ₁ ;		1			
2. double v ₂ ;			1			2. double v ₂ ;			1		
3. double v ₃ ;				1		3. double v ₃ ;				1	
4. double v ₄ ;					1	4. double v ₄ ;					1
5. cin >> v ₁ ;		2				5. cin >> v ₁ ;		3			
6. cin >> v ₂ ;			2			6. cin >> v ₂ ;			3		
7. cin >> v ₃ ;				2		7. cin >> v ₃ ;				2	
8. v ₄ = v ₁ * v ₂ ;						8. v ₄ = v ₁ * v ₂ ;		2	2		
9. v ₀ = v ₄ / v ₃ ;						9. v ₀ = v ₄ / v ₃ ;					
10. cout << v ₀ ;						10. cout << v ₀ ;					

Table 4. Transformed Linkage Graph Matrix to Match a Partial Program

Complete/Correct Program:		... v ₀ = (v ₁ * v ₂) / v ₃ ; cout << v ₀ ;					
Partial Program that Needs Hints:		... v ₄ = v ₁ * v ₂ ;					
		v ₀	v ₁	v ₂	v ₃	v ₄	//column v ₄ added
	0. double v ₀ ;	1					
	1. double v ₁ ;		1				
	2. double v ₂ ;			1			
	3. double v ₃ ;				1		
	3b. double v ₄ ;					1	// a row is added
	4. cin >> v ₁ ;		2				
	5. cin >> v ₂ ;			2			
	6. cin >> v ₃ ;				2		
7. v ₀ =v ₁ *v ₂ /v ₃ ;	7a. v ₄ = v ₁ * v ₂ ;		3	3		2	// a row broken into 2
	7b. v ₀ = v ₄ / v ₃ ;	2			3	3	rows
	8. cout << v ₀ ;	3					

4 Effectiveness of Linkage Graph for Hint Generation

We have implemented the algorithms described herein in the context of jFlex/CUP. To evaluate the effectiveness of using linkage graphs to generate hints, we analyzed student submissions for a lab from the Spring 2011 introductory programming course at UNC-Charlotte. The program is to calculate the pay for mowing the lawn around a house. There are 200 total submissions with 37 correct solutions.

We performed vertical and horizontal evaluations. The ‘vertical’ evaluation was applied to the set of correct submissions to generate hints for the first intermediate

version that can compile from its later complete counterpart. Since the same student wrote the partial and complete programs, we expected the hints to make sense. This baseline was to confirm that our ‘overlay’ hint generation approach would work. Among 16 randomly selected correct submissions, good hints were generated for 14 (87.5%) of them, with six correcting a mistake, four finishing one more step, four no hints due to program already complete. The only two cases, where the hints were not appropriate, occurred when variable names were reused for different purposes.

We applied a ‘horizontal’ evaluation to a sample of 15 randomly selected incorrect submissions. For each of incorrect solutions, we manually selected a similar correct solution, which was not necessarily the best match. We then ran our program to generate linkage graphs and generate hints based on their differences. We found that we could provide meaningful hints for 10 (66.6%) of the incorrect submissions. We believe this rate is promising, since we did not perform a best-match search. With a best-match search and full MDPs, we could leverage partial solutions on paths to correct solutions to provide intermediate states for hints. The remaining 5 programs fall into the following two categories: (1) Variable name reuse. (2) The current algorithm looks at each linkage separately. A hint is provided for the first missing statement along each individual linkage. For example, if a program didn’t convert the lawn area from square feet to square yards, the hints will most likely include “*double lawnSqYds*” A better hint should be the next statement along that linkage “*lawnSqYds = lawnSqFt / 9*”. This can be addressed by considering the relevant linkages together.

In both cases, implementing our proposed algorithms for detecting variable name reuse would bring the successful hint rates to over 86%. We believe this success rate indicates that our approach is likely to work.

Future Work will include implementing variable reuse detection, linkage graph transformation when a match cannot be found, and further automating the variable normalization process. Finally, we will also determine strategies for hint presentation, since a full list of ‘missing’ items may be intimidating for novices.

Acknowledgement. This work was partially supported by NSF grants IIS-0845997 and CCLI-0837505.

References

- Barnes, T., Stamper, J.: Automatic hint generation for logic proof tutoring using historical data. *Journal Educational Technology & Society, Special Issue on Intelligent Tutoring Systems* 13(1), 3–12 (2010)
- Barnes, T., Stamper, J.: Using Markov decision processes for student problem-solving visualization and automatic hint generation. In: *Handbook on Educational Data Mining*. CRC Press (2010)
- Jin, W., Lehmann, L., Johnson, M., Eagle, M., Mostafavi, B., Barnes, T., Stamper, J.: Towards Automatic Hint Generation for a Data-Driven Novice Programming Tutor. In: *Workshop on Knowledge Discovery in Educational Data, 17th ACM Conference on Knowledge Discovery and Data Mining* (2011)
- Stamper, J., Barnes, T., Croy, M.: Enhancing the automatic generation of hints with expert seeding. To appear in *Intl. Journal of AI in Education, Special Issue “Best of ITS”* (2011)

Exploring Quality of Constraints for Assessment in Problem Solving Environments

Jaime Galvez Cordero, Eduardo Guzman De Los Riscos,
and Ricardo Conejo Muñoz

Universidad de Malaga,
29071, Malaga, Spain
{jgalvez, guzman, conejo}@lcc.uma.es

Abstract. One of the approaches that has demonstrated by far its efficiency as a tutorial strategy in problem solving learning environments is the Constraint-Based Modeling (CBM). In existing works it has been combined with a data-driven technique for automatic assessment, the Item Response Theory (IRT). The result is a well-founded model for assessing students while solving problems. In this paper a novel technique for studying quality of constraints for this type of assessment is presented. It has been tested with two new systems, an independent component for assessment that implements CBM with IRT, which provides assessment to a new problem solving environment developed to assess the students' skills in decision-making in project investments. The results of testing our approach and the application of these two systems with undergraduate students are also discussed in this paper.

Keywords: Problem Solving Environments, Constraint-Based Modeling, Item Response Theory.

1 Introduction

Among the existing approaches that can be applied to modeling students in problem solving environments, Constraint-Based Modeling (CBM) has proved its effectiveness with a range of tutors and studies performed in the last years [1]. It is easier to be applied than other approaches, such as Model Tracing [2], since CBM does not require identifying all possible steps a student could take to reach a solution to a problem. On the contrary, only those constraints that any solution should not violate need to be identified.

CBM is an effective approach, whose power lies in the design of the constraints set. To build a new learning environment using authoring tools such as ASPIRE [3] is a very easy task, since no programming skills are needed. What is necessary to model constraints in an appropriate manner is to have a broad knowledge of the domain matter; the same happens in any other approach when a new learning environment is going to be developed. Nevertheless, even with human experts, constraints could not be reflecting properly a domain principle. In this sense, a constraint could actually represent a more specific principle or, otherwise, a more general one.

The work presented here is based on the model presented in [4, 5] which combines Item Response Theory (IRT) with CBM. IRT is a data-driven theory commonly used in testing environments for assessment. The IRT+CBM model generates probabilistic curves, called Constraint Characteristic Curve (CCC), which are inferred from a calibration process with prior data from students' performance.

Unfortunately, as mentioned before, constraints may not represent the domain model in the best possible way. Moreover, the calibration performed by the IRT+CBM model might not have enough evidence to infer the CCCs properly. In this paper we present a data-driven technique to determine quality of constraints, i.e., whether or not they are good enough to be used for assessment.

The content of the article is organized as follows: first, the work related to our research is mentioned. Then, we describe how IRT would help to determine quality of constraints. Next, we present a new assessment framework and a new problem solving environment we have used to carry out the experimentation. Section 5 describes our hypothesis, the experiment we designed, and our findings. Finally, conclusions and future research work are outlined in the section 6.

2 Related Work

The first methodology of interest to the work of this paper is the CBM, which is used to model the domain and student in problem solving environments with the goal of improving learning of a given subject. Its basis is the Olsson's theory of learning from performance errors [6], according to which incomplete or incorrect student's knowledge can be used within an intelligent tutoring system as guidance. Detection of this faulty knowledge is done by the main element of CBM: the constraint, which represents a principle that none of the possible solutions to a problem in this domain will violate.

The other technique employed here is the IRT conceived by Thurstone [7], a well-founded theory used in testing environments to measure certain traits, such as the student's knowledge. This theory is based on modeling the probability of answering a question/item correctly given a student's knowledge level by means of a function called Item Characteristic Curve (ICC) where the greater the student's knowledge level is, the higher the probability of answering correctly.

The main work related to the study conducted here is based on [4, 5], where a model combining CBM and IRT is proposed in order to provide CBM with a long-term student model. According to this work, constraints of CBM are equivalent to questions of a test and using IR assessment over constraints can improve the student model accuracy and, consequently, provide a better adaptation to the student learning process. The analogy made between these two methodologies is the basis that allowed us to apply techniques associated with the IRT into CBM to develop this work.

In literature there are works on CBM [8, 9] which explore whether or not groups of constraints, linked to more general concepts, would be more effective for learning than single constraints. However, our approach treats it from a different point of view since it is based on IRT.

3 Using IRT to Study Quality of Constraints for Assessment

The analogy that allows us to formulate the approach explained below is that constraints are equivalent to questions in the sense that both of them represent declarative knowledge units and both of them have two values as the result of the student performance: one positive and one negative. The positive value represents correct knowledge, which, in the case of CBM, corresponds to a satisfaction of a constraint and, in questions, to a correct response. The negative value would represent faulty knowledge, meaning that the constraint was violated or the response was wrong.

According to [4, 5], to apply IRT to constraints, a Constraint Characteristic Curve (CCC) is defined for every constraint in a calibration process with the evidence taken from the student's performance. As in IRT, it represents a probability distribution based on the knowledge: the broader the knowledge, the more probability of satisfying the constraint. Violations can be also modeled using the inverse of this function, which means that when the knowledge is broader, the probability of violation is lower. As a result of the calibration, the parameters representing the CCC are obtained.

Normally, the 3 parameters logistic function (3PL) is applied, producing the following three parameters: a represents discrimination which is a value proportional to the slope of the curve. The higher it is, the greater capacity to differentiate between the students' inferior and superior knowledge levels; b is the difficulty and it corresponds to the knowledge value for which the probability of satisfying the constraint is the same as that of violating it; the last parameter, c , is the guessing and it represents the probability that a student will satisfy the constraint even though he/she may not possess the knowledge required to do so.

The basis of our proposal is that, considering the parameters of a CCC, we could manage constraints as if they were items and, consequently, mechanisms applied over items to determine their quality are equally valid for constraints. Concretely, we propose to employ the Item Information Function (IIF) [10, 11], which is a technique used in adaptive testing in order to describe, select, and compare items and tests. Accordingly, we define the Constraint Information Function (CIF) that can be used to detect the most suitable constraints for assessment (see equation 1 based on [10]). In this way, assessment would be done over concepts representing more faithfully the reality, which would reduce misleading result of an inappropriate representation.

$$I_i(\theta) = \frac{2.89a_i^2(1-c_i)}{[c_i + e^{1.7a_i(\theta-b_i)}][1 + e^{-1.7a_i(\theta-b_i)}]^2} \quad (1)$$

The $I_i(\theta)$ represents how informative a constraint i is for a fixed value of the student's knowledge, θ . This knowledge ranges from $-\infty$ to ∞ , but in practice, only values from the interval $[-4, 4]$ or $[-3, 3]$ are normally considered because, out of this interval, the value of the CIF is very close to zero and hence it is negligible. Within this interval the function has a logistic bell shape with values close to zero in the extremes and a maximum in the value of $\theta=b_i$, which is the parameter corresponding to the difficulty of the constraint and the most representative for the CIF. Note that equation 1 assumes that CCC has been calibrated under the 3PL model.

To calculate the CIF of a particular constraint, given that the formula is the derivative respect to θ , we would apply equation 2 to get the total information, which would consider the whole range of student's knowledge.

$$I_i = \int_{-\infty}^{\infty} I_i(\theta) d\theta \quad (2)$$

We distinguish three particular cases where CIFs could help to explore the quality of constraints:

- a) The first case is related to the relevance of constraints. Some of the domain constraints are not always relevant to all the problems. They will have less evidence in comparison to others and, thereby, less information of the domain. The use of these constraints to assess students could produce an inaccurate assessment.
- b) Secondly, extremely high values of the information function in a constraint, in comparison to the others, could suggest that this constraint is grouping more than one domain principles. The recommendation here should be to consider splitting this constraint into several ones, each one modeling a more specific principle.
- c) The last case would be exactly the opposite of the second one: the value of the information function is extremely low. Two reasons could lead to this fact: first, the population is small and there is not enough evidence to calibrate the curves properly; and second, the constraint is too fine-grained and it should be merged with other constraint to model a more general principle. Finally, this CIF value could also suggest that the constraint is not a good indicator of the student's knowledge in the domain.

Regarding the distinction between good and bad constraints, it is clear that if the information is lower, it will be worse for assessment. Nevertheless, if we have to establish a limit or threshold to separate good constraints from bad ones, we still do not know if there is a common limit for different domains. In the experimentation section we give the threshold, obtained for our problem solving environment, as a reference point for further studies.

4 Tools Used in the Experiment

To perform the experiment we used three systems, each one for different purposes: the first one is Siette [12], a web-based authoring tool and testing environment where students can take tests on a subject matter, and where assessment with IRT is possible. The other two systems are presented in this paper for the first time and both are components of a bigger platform for teaching mathematics, DEDALO [13]. Following the philosophy of this framework, every component is independent and can communicate through Web Services with the rest of the platform components. These components are called Project Investments Problem Solving Environment (PIPSE) and CBM-Engine.

4.1 Project Investment Problem Solving Environment

PIPSE was developed to be used as part of a course of Project Management as a support tool. It is a problem solving environment focused on the study of the profitability of starting up a project given a series of variables associated with costs and benefits that it would generate. The system is a Web application implemented on .net through which students can apply several indexes, such as Net Present Value (NPV) or Internal Rate of Return (IRR) [14], to study the profitability of a project. Figure 1 shows the four main parts of PIPSE: A is a panel of actions related to the current session and to the student’s attempts; B contains the problem stem and buttons to hide / show it; C is the table with the student’s solutions which can be edited; and D contains the controls to add years or variables to the problem, with the solution variables and a workspace panel where all actions carried out by the student are represented, and new commands can be entered into a command line interpreter.

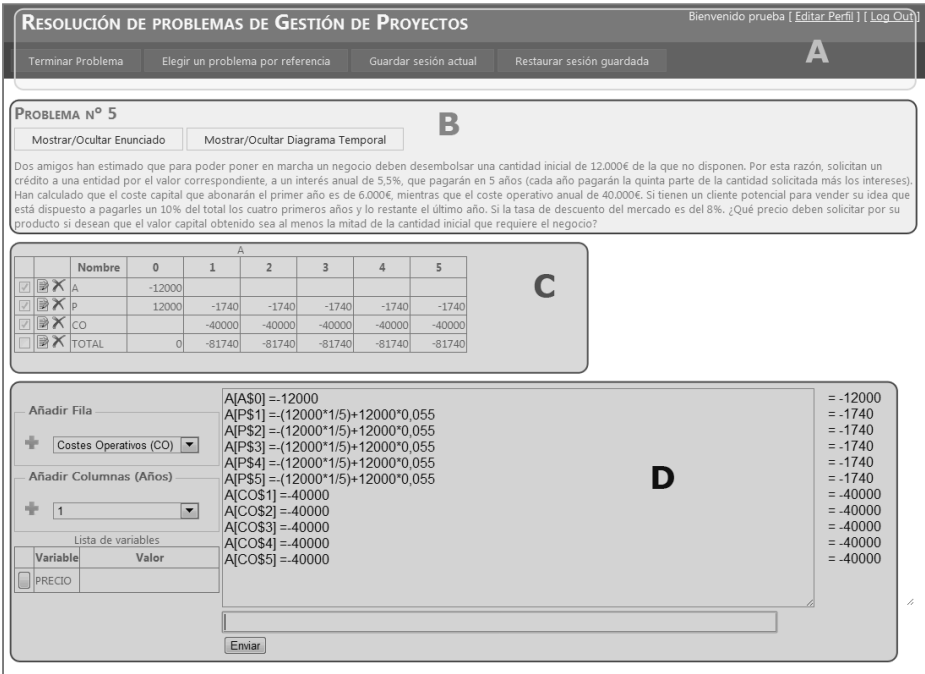


Fig. 1. Project Investment Problem Solving Environment

The system interface tries to reduce the cognitive overload [15], otherwise calculus inherent in this kind of problems would affect the student’s working memory. This is done by providing students with mechanisms similar to a datasheet, allowing them to use references to cells of a table to build formulas that will be automatically interpreted and calculated by the system. Those mechanisms make calculations unnecessary outside the interface and help students to focus on using their knowledge to solve the problem. Students should build a table with all the problem information and

provide other information, which, all together, would represent the solution to the problem. PIPSE is able to present information about the student performance errors obtained from the application of CBM to their solution. This characteristic makes the system not only an assessment tool, but also, suitable for learning purposes.

Information gathered from the student interaction with the system is used by it to generate different assessments. To accomplish this, the information is sent to different assessment subsystems, available through Web Services. Those subsystems are independent and they are not fixed, i.e., they can be dynamically replaced, added, or removed from the system. Although currently there are two different assessment subsystems implemented, each one associated with a different methodology, only one of them is of interest to this study: the one that implements the combination of CBM+IRT, which is explained in the following subsection.

4.2 CBM-Engine Assessment Component

The CBM-Engine is a SOA-based component following the same idea of [16] that implements CBM with IRT assessment. It has no interface but a set of services that can be used to apply the already-explained methodology in any external system/tutor. It is formed by a three-layered architecture comprising: a) a top level layer offering Web Services as interface with the external systems, b) an assessment layer where all inferences and application logic are carried out, and c) a persistence layer in charge of storing data structures common to any domain and those specific to each particular domain. New problem solving environments or tutors wanting to obtain assessment with this framework must be added to the system by using an authoring tool where constraints and data structures must be defined.

In the particular case of the PIPSE system, we are dealing with a well-defined domain where problems as well as tasks are well-defined [17]. The constraints and the specific data structures forming the domain model were added to the engine resulting in a set of 17 constraints, which can be categorized in three subsets: (a) correct definition of variables related to the problem; (b) manipulation of the data in the solution table; and (c) calculus and inference associated with the solution.

5 Experimentation

In this section we are going to describe the experiment we have conducted to validate our proposal. In this sense, the main hypothesis to be tested will be whether or not the IIF can be applied to constraints in the same way it is used in testing environments, to detect constraints not suitable for assessment.

As a secondary goal, the second part of our experimentation tries to study an important characteristic that any system should have in order to be used for assessment purposes: it should be able to provide a valid assessment of the student performance. To verify this with the PIPSE system presented in this study, we proceeded as it was done in [4, 5]. Following the same criteria, assessment produced with the system using the combination CBM+IRT should be similar to the one obtained by applying a

formal assessment of the same concepts involved in the system. Thus, the second part is focused on exploring whether or not the assessment provided by our new system, using a set of constraints valid for assessment, is equivalent to the one provided by a test where IRT would be applied to infer the student's knowledge.

5.1 Design and Implementation of the Experiment

In order to evaluate our methodology, we designed an experiment with students from the last year of the M.Sc. in Computer Science degree at the University of Malaga. A total of 24 students participated in the study that was performed in December 2011 and comprised of several stages. First, the students were instructed during several classes on the different indexes to solve the project investment problems. Next, they took a one-hour-long session where they were able to use the system to solve two problems seen previously in class; a week later, we performed a paper-based exam where two problems were proposed and a test was administered.

To test the experiment hypothesis, problems proposed in the exam did not cover the whole set of constraints; a characteristic we would use later in the analysis of constraints quality with the CIF. Regarding the test, it was designed, following the same premises as in [4, 5], in order to assess the same concepts involved in the problems. To achieve this, a question was written for each constraint, producing a total of 15 questions in the test. Two of the constraints were left out of the test since they were not associated with concepts, but with mathematical verifications.

Unlike the early work with this technique, the exam was made on paper with the aim of getting only the constraints violations and preventing students from receiving any type of feedback. With this omission of information about errors made in the solution, the learning factor associated with feedback was isolated and taken out of the experiment, which, according to IRT requirements, is important to generate a good calibration of constraints and to apply IRT mechanisms. Once all the students had finished the exam, the solutions they provided were then introduced into the problem solving environment and constraints were checked against them.

The experiment was used as an assessment item in the course, and all 24 students enrolled in the course participated in it. Additionally, the Siette test was also administered to the students. After all data had been gathered from students, we performed the analysis of constraints applying the approach explained before, filtering some of the constraints and leaving the rest to perform the assessment of students, which led us to the results described in the next section.

5.2 Results

The solution provided by every student was introduced into the PIPSE, which sent it to the CBM-Engine, recording all data and calibrating constraints. The calibration output, i.e., parameters representing the CCC, was analyzed by applying the information function to every constraint using the formula (1). As a result, we got an average value of 14.81 of the CIF and a standard deviation of 2.18 for the whole set of 17 constraints.

Before examining the results, we grouped the constraints into those that were not relevant during the problems taken in the exam and those that were. Looking at the results, the first supporting finding we made was that the group of relevant constraints, composed by 7 of them, had a greater mean of the CIF (16.29 versus 13.76). Although after a t-test we couldn't find significant difference in their means (p-value 0.68), we discovered that one of the constraints from this analysis had a strange value that was affecting the results by introducing noise. When we discarded it, the difference became significant (p-value 0.012).

Besides, we ordered the constraints according to their value in the CIF, finding that 5 out of 7 of the relevant constraints were at the top of the list. In this particular case, splitting the data with the threshold $\bar{x} + 0.5\sigma$, resulted in the division of the relevant constraints at the top of the list. This suggests that most of them could be detected using the CIF (conforming case *a*) of our proposal in section 3). Regarding the other two relevant constraints not found at the top, both of them were at the bottom with an order of -1.67 times the standard deviation, which was significant. This constraint with extremely low value was representing a principle of the domain that was implicit in other constraints and, therefore, it was not providing much information. The other constraint at the bottom of the list was not significantly different from the rest and experts in the domain didn't find any other constraint that could be merged with it. This probably is explained by a small population of students that didn't provide enough evidence to get a good calibration of the constraint. In any case, irregularities of both of these constraints were detected with our approach (conforming case *c*) of our proposal).

Additionally, during the analysis we found a constraint with an outstanding value of the information function over the remaining ones. It had a 20.07, which is an order of 2.4 times the standard deviation. Since we had not deliberately designed this constraint to be different from the others, by examining it to see what the cause of this exaggerated value would be, we realized it was due to grouping several concepts together, which led to students' faulty knowledge being more pronounced here. It means that we were able to detect a constraint which could be split into others representing more fine-grained principles of the domain (see case *b*) of our proposal in section 3).

The filtered set of constraints was used then in the assessment framework to provide a score for every student. This assessment was compared with the one obtained in the Siette test using a paired t-test at 95% confidence. As result of the t-test we got a p-value of 0.8155. This clearly suggests that in the case of pairs of scores belonging to a student, there is no significant difference between them. Furthermore, we performed a correlation analysis between both scores, obtaining a correlation coefficient of 0.06. This is a very small value that we think could be explained by two factors: a) the number of data from students / constraints is not big enough; or b) questions of the test were not correctly designed to evaluate the same concepts.

6 Conclusions

In this paper, a new approach, called Constraint Information Function (CIF), to study quality of constraints in CBM tutors has been introduced. This methodology is based

on the analogy discovered between questions and CBM constraints [4, 5], according to which, constraints are used as if they were questions in a test and, consequently, mechanisms of IRT can be applied to constraints. In this way, the IIF, normally used to study quality of questions in test development, has been proposed to determine whether or not constraints are representing the domain correctly and if they can be used for assessing students appropriately. This approach would help to generate a more accurate assessment, leading to a more precise student model and a better adaptation. In addition, our approach could contribute to the constraint elicitation process, since it could help to detect constraints that should be split or grouped, and even to reformulate or discard them.

As part of the study, the CBM with IRT assessment has been implemented in a new SOA-based assessment framework called CBM-Engine. This system is able to perform the same assessment procedure combining both techniques, with the advantage of being independent of the learning system. What is more, it can be used by any external learning environment as long as it is registered in the system and its domain model is incorporated into the specific domain data structures.

Besides, a new problem solving environment focused on the domain of project investment analysis has been presented. It has been designed to provide different assessments from independent subsystems, each one using different assessment mechanisms. For the study presented in this article, only the methodology provided by the CBM-Engine is of relevance. This problem solving environment can be used not only as an assessment tool, but also as a tutoring system since it is able to take the feedback produced by the CBM and present it to the students. However, this scaffolding mechanism goes beyond the scope of this paper.

In the experiments conducted, we used the problem solving environment working with the new assessment framework. Students' data were used by the framework to produce first a calibration of constraints and then an assessment. Between the two phases, the Information Function was successfully applied to detect those constraints which were not suitable to be used for assessment. The assessment performed after filtering the non-suitable set of constraints was compared to the assessment of a test covering the same concepts involved in the constraints. Statistical analysis suggests that our model could diagnose in the same way as an IRT-based test does. Nevertheless, no much correlation was found between the test and the problem solving scores, probably because the data used in the experiment was much reduced.

When we look at CBM with IRT as a problem solving environment assessment mechanism, the results are promising and a range of possibilities is opened with this synergy. Nevertheless it has a drawback that should be taken under consideration: so far, results have been found only in systems without a big population using it. Therefore, further work is being done to explore efficiency of this technique for bigger systems. Further work should be also done to explore if the process of the approach presented here, which was made entirely manual, could be automated within the CBM-engine; if some common threshold to distinguish good constraints from bad ones can be found in different systems; and whether there exist any automatic mechanism to determine it. Our current work is focused on these lines and exploring other utilities of IRT mechanisms that can be applied to CBM tutors.

Acknowledgements. This work has been co-financed by the Andalusian Regional Ministry of Science, Innovation and Enterprise (P07-TIC-03243 and P09-TIC-5105).

References

1. Mitrovic, A., Martin, B., Suraweera, P.: Intelligent Tutors for All: The Constraint-Based Approach. *IEEE Intelligent Systems* 22, 38–45 (2007)
2. Mitrovic, A., Koedinger, K.R., Martin, B.: A comparative analysis of cognitive tutoring and constraint-based modeling. In: *User Modeling*, pp. 313–322 (2003)
3. Mitrović, A., Suraweera, P., Martin, B., Zakharov, K., Milik, N., Holland, J.: Authoring Constraint-Based Tutors in ASPIRE. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006*. LNCS, vol. 4053, pp. 41–50. Springer, Heidelberg (2006)
4. Gálvez, J., Guzmán, E., Conejo, R., Millán, E.: Student Knowledge Diagnosis Using Item Response Theory and Constraint-Based Modeling. In: *The 14th International Conference on Artificial Intelligence in Education*, vol. 200, pp. 291–298 (2009)
5. Gálvez, J., Guzmán, E., Conejo, R.: Data-Driven Student Knowledge Assessment through Ill-Defined Procedural Tasks. In: Meseguer, P., Mandow, L., Gasca, R.M. (eds.) *CAEPIA 2009*. LNCS(LNAI), vol. 5988, pp. 233–241. Springer, Heidelberg (2010)
6. Ohlsson, S.: Constraint-based student modeling. In: *Student Modeling: the Key to Individualized Knowledge-based Instruction*, pp. 167–189 (1994)
7. Thurstone, L.L.: A method of scaling psychological and educational tests. *Journal of Educational Psychology* 16, 433–451 (1925)
8. Martin, B., Mitrović, A.: Using Learning Curves to Mine Student Models. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) *UM 2005*. LNCS (LNAI), vol. 3538, pp. 79–88. Springer, Heidelberg (2005)
9. Martin, B., Mitrović, A.: The Effect of Adapting Feedback Generality in ITS. In: Wade, V.P., Ashman, H., Smyth, B. (eds.) *AH 2006*. LNCS, vol. 4018, pp. 192–202. Springer, Heidelberg (2006)
10. Birnbaum, A.: Some latent trait models and their use in inferring an examinee’s ability. In: *Statistical Theories of Mental Test Scores*. Addison-Wesley, Reading (1968)
11. Hambleton, R.K., Swaminathan, H., Rogers, H.J.: *Fundamentals of Item Response Theory*. Sage Publications, Inc., Thousand Oaks (1991)
12. Guzmán, E., Conejo, R., Pérez-de-la-Cruz, J.L.: Improving Student Performance using Self-Assessment Tests. *IEEE Intelligent Systems* 22, 46–52 (2007)
13. DEDALO project, Assessment and Learning of Mathematics (January 23, 2012), <http://dedalo.lcc.uma.es>
14. Khan, M.Y.: *Theory & Problems in Financial Management*. McGraw Hill Higher Education, Boston (1993)
15. Sweller, J., van Merriënboer, J., Pass, F.: Cognitive architecture and instructional design. *Educational Psychology Review* 10(3), 251–296 (1998)
16. Gálvez, J., Guzmán, E., Conejo, R.: A SOA-Based Framework for Constructing Problem Solving Environments. In: *ICALT 2008*, pp. 126–127 (2008)
17. Mitrovic, A., Weerasinghe, A.: Revisiting ill-definedness and the consequences for ITSs. In: *14th International Conference on Artificial Intelligence in Education*, pp. 375–382 (2009)

Can Soft Computing Techniques Enhance the Error Diagnosis Accuracy for Intelligent Tutors?

Nguyen-Thanh Le and Niels Pinkwart

Clausthal University of Technology Germany
{nguyen-thinh.le, niels.pinkwart}@tu-clausthal.de

Abstract. Problems for which multiple solution strategies are possible can be challenging for intelligent tutors. These kinds of problems are often the norm in exploratory learning environments which allow students to develop solutions in a creative manner without many restrictions imposed by the problem solving interface. How can intelligent tutors determine a student's intention in order to give appropriate feedback for problems with multiple, quite different solutions? This paper focuses on improving the diagnosis capabilities of constraint-based intelligent tutors with respect to supporting problems with multiple possible solution strategies. An evaluation study showed that by applying a soft-computing technique (a probabilistic approach for constraint satisfaction problems), the diagnostic accuracy of constraint-based intelligent tutors can be improved.

Keywords: Soft computing, constraint satisfaction problems, error diagnosis, intelligent tutoring systems.

1 Introduction

Intelligent tutors which are able to deal with problems that have multiple solution variants usually have to face the challenge of diagnosing the student's intention, i.e., determining which solution strategy the student is pursuing as she is trying to solve a given problem. Diagnosing the solution strategy intended by a student is important, because only if this is done correctly, an accurate error diagnosis can be conducted and, in consequence, appropriate feedback on the student's solution can be given.

There are two main and established approaches for building intelligent tutors: model-tracing [1] and constraint-based modeling [2]. While a model-tracing system is able to diagnose the student's intention by monitoring and relating the student's problem solving steps to the correct solution paths captured in the cognitive model [3], constraint-based tutors do usually not contain sufficient information to decide on the most plausible hypothesis about the student's intention underlying a student's solution. Constraint-based error diagnosis can be conceived as a constraint satisfaction problem. If a student solution is correct, then all constraints will be satisfied. If an erroneous student solution is evaluated, an inconsistency between the erroneous student solution and the constraint system occurs, i.e., one or more constraints will be violated. In this case, the problem of error diagnosis is considered over-constrained.

The goal of constraint-based error diagnosis is not to search for a correct solution, but rather to identify the constraint violations which lead to the inconsistency between an erroneous solution and the constraint system. The result of this constraint-based error diagnosis – the constraints that are violated and those that are not – is a good starting point for giving feedback to students if there is only one main solution for a problem (possibly with slight variations that the constraint system can accommodate). Yet, it is of less use if different solution strategies (i.e., different constraint sets) for a problem are possible.

To build constraint-based intelligent tutors which *are* able to handle problems with multiple solution strategies, the approach presented in this paper adopts a soft computing technique for solving constraint satisfaction problems: a probabilistic framework. For that purpose, each constraint is associated with a constraint weight which represents heuristic information indicating the importance of the constraint. As we will argue in this paper, these weights can be used to hypothesize the student's intention underlying his solution.

In the next section, we review some typical soft computing techniques for solving constraint satisfaction problems and argue why we choose the probabilistic approach. Then, we briefly describe a weighted constraint-based model which can be used to build intelligent constraint-based tutors for problems with multiple solution strategies. Next, we show an evaluation study which confirms that by applying soft computing techniques, the diagnostic accuracy for constraint-based intelligent tutors can be enhanced as compared to traditional constraint-based modeling approaches. Finally, we summarize the benefits of the weighted constraint-based model and propose some future work.

2 Soft Computing for Constraint Satisfaction Problems

To deal with the issue of over-constrained satisfaction problems, some researchers have attempted to distinguish the level of importance between constraints. Here, hard constraints represent conditions which must always hold, and soft constraints represent preferences which should be satisfied when possible. Several techniques have been devised to express soft constraints and to allow their violation. The most popular approaches include fuzzy constraint satisfaction problems (CSPs) [4], cost-minimizing CSPs [5], partial CSPs [6], and probabilistic CSPs [7].

A partial CSP framework attempts to soften a constraint satisfaction problem by changing the domain of variables/constraints or a constraint system in several ways: by 1) enlarging the domain of a variable, 2) enlarging the domain of a constraint, 3) removing variables of a constraint, or 4) removing a constraint from the constraint system. This approach is not appropriate for enhancing the error diagnosis capability of a constraint-based intelligent tutor due to the following reason. To choose the most plausible solution strategy, we need to consider all possible evidence (based on solution components), whereas a partial CSP framework attempts to eliminate constraints which can be violated by a student solution and thus, evidence supporting the process

of hypothesizing the student's intention during error diagnosis is also eliminated. As a consequence, the diagnosis capability of a constraint-based intelligent tutor would be degraded.

While a partial CSP framework requires satisfying a partial set of constraints, the fuzzy CSP and the cost-minimizing CSP approaches allow all constraints to be satisfied by defining a preference ranking of the possible instantiations according to some criteria depending on the constraints. The solution of a fuzzy/cost-minimizing constraint satisfaction problem is the instantiation which meets the highest satisfaction degree. The fuzzy CSP framework associates a level of preference with each instantiation of variables in each constraint and searches a solution by maximizing the satisfaction degree of the least preferred constraint. On the contrary, in a cost-minimizing CSP framework instantiations are assigned with a cost and the goal is to find a solution which minimizes the total sum of costs of the chosen instantiation for each constraint. These approaches are very appropriate for problem situations where preference levels for certain instantiations of the constraint variables are available. They are, however, not well suited for improving the capability of constraint-based error diagnosis. The problem of error diagnosis in a constraint-based tutor is a situation where it is almost impossible to specify instantiations of constraint variables in advance because the amount of constraints required to model domain knowledge is relatively high and the space of possible instantiations is large.

A probabilistic CSP framework, finally, contains a set of constraints. Each of these constraints is intended to represent a piece of knowledge. It is associated with a probability of relevance. That is, some constraints can be specified as relevant to the problem with complete certainty, and for some others it can be specified that they may or may not be relevant to this problem. It is usually assumed that the probabilities of two different constraints are independent from each other. A solution of the probabilistic constraint satisfaction problem is an instantiation of all variables that has a maximal probability. A probabilistic CSP framework can be used to model situations where each constraint can be specified with a certain probability. Since such a situation is applicable to constraint-based intelligent tutors, Le and Pinkwart proposed to adopt the probabilistic approach for enhancing the diagnosis capability of traditional constraint-based tutors [8].

In the approach pursued here, a probability associated with each constraint indicates a measure of the importance of a constraint and is being referred to as a constraint *weight*. Applying the probabilistic CSP approach, the automated evaluation of student solutions resembles the assessment of written examinations by a human tutor: not only the quantity of the "correct" statements made by the student is important for the final mark, but also the importance of the contained statements.

In our approach, one goal for using constraint weights (in addition to coming to a more realistic estimation about solution quality by considering different importance degrees for different constraints) is to choose the most plausible hypothesis about the solution strategy pursued by a student.

3 Weighted Constraint-Based Models

As presented in more detail in [8], a weighted constraint-based model (WCBM) consists of a semantic table, a set of weighted constraints, and transformation rules. The WCBM model assumes that a problem can be solved by applying different alternative solution strategies, and each of them can be implemented in different variations. The semantic table is used to model alternative solution strategies and to represent generalized components for each solution strategy. Constraints are used to check the semantic correctness of the student solution with respect to the requirements specified in the semantic table and to examine general well-formedness conditions for a solution. Transformation rules serve to extend the coverage of a solution space (for instance, they allow for including general rules such as math equations, e.g., $X(Y+Z) = XY+XZ$, into the diagnosis process). The process of diagnosing errors in a student solution performed by a WCBM tutor consists of two interwoven tasks (hypotheses generation and hypotheses evaluation) which take place on two levels (strategy and solution variant level). First, on the strategy level, the system generates hypotheses about the student's intention by iteratively matching the student solution against the solution strategies that are specified in the semantic table. Then, once a solution strategy has been matched, the process initiates hypotheses about the student's solution variant by matching components of the student solution against corresponding components of the selected solution strategy. Next, hypotheses generated on the solution variant level are evaluated, and the most plausible variant of the student solution (within a strategy) is chosen. In this process, hypotheses are evaluated with respect to their plausibility by multiplying the weights of constraints which are violated by that hypothesis according the following formula:

$$\text{Plausibility}_{\text{Prod}}(\text{H}) = \prod_{i=1}^N W_i, \text{ where } W_i \text{ is the weight of a violated constraint} \quad (1)$$

On the strategy level, the hypothesis with the highest plausibility score (note that important constraints have weight values close to 0, while less important ones have weights close to 1) corresponds to the solution strategy which the student has most likely intended to pursue in his solution. This hypothesis is selected, and diagnostic information is derived from constraint violations resulting from the plausibility computation of the selected hypothesis.

4 Evaluation

In [8] it has been shown that an intelligent tutor built based on the weighted constraint-based model is better than a corresponding intelligent tutor that is built based on a classical constraint-based modeling approach with respect to evaluating *intention analysis*. The intention analysis of a tutor is the capability to hypothesize the solution strategy underlying the student solutions correctly. In this paper, we intend to go the next step and compare the *diagnostic validity* of a weighted constraint-based tutor with a corresponding traditional constraint-based tutor. Evaluating the diagnostic

validity means determining whether the diagnostic result is acceptable with respect to a gold standard. The diagnostic validity partially depends on the capability of intention analysis, because if the intention of the student is hypothesized wrongly, this makes it more difficult to detect errors with a high validity.

4.1 Design

To compare the diagnostic validity of the weighted constraint-based model with the classical constraint-based modeling approach, we used two versions of INCOM, a tutoring system for logic programming. The first one applies the weighted constraint-based model (INCOM-WCBM). A modified version of INCOM (INCOM-CBM) corresponds to a classical constraint-based tutor and uses constraints without weight values. Classical constraint-based tutors have no “standard” way of dealing with multiple solution strategies that each come with different constraint sets. To realistically compare INCOM-WCMB and INCOM-CBM, such a feature for plausibility of hypotheses about the solution strategy intended by the student had to be added to INCOM-CBM. We did this by summing up the number of constraint violations caused by each hypothesis:

$$\text{Plausibility}_{\text{Add}}(\text{H}) = |\text{C}| \quad (2)$$

C is the set of all constraint violations caused by each hypothesis H. This approach seems quite natural and straightforward in a situation where constraints do not have weights but can just either be violated or not – it models the idea that if a student solution violates X constraints for the constraint system corresponding to solution strategy A and Y(>X) constraints for the constraint system corresponding to solution strategy B, then the student has most likely pursued strategy A. That is, the plausibility of a hypothesis is associated to the number of corresponding violated constraints.

We collected exercises and solutions from past written examinations for computer science (specifically, a course in logic programming and AI) and input them into the two systems under comparison (INCOM-WCBM and INCOM-CBM). In total, we collected 221 student solutions, where the solutions have been collected based on the following criteria: 1) any piece of code which satisfies minimal requirements of interpreting it as a Prolog program is considered a solution, 2) syntax errors in the solutions are ignored (because during the written examination session students did not have access to a computer), 3) both correct and incorrect solutions are taken into account. Following are short versions of the seven tasks we selected:

1. Access to specific elements within an embedded list;
2. Querying a data base and applying a linear transformation to the result;
3. Modification of all elements of a list subject to a case distinction;
4. Creation of an n-best list from a data base;
5. Computing the sum of all integer elements of a list;
6. Counting the number of elements in an embedded list;
7. Finding the element of an embedded list which has the maximum value for a certain component.

To define a gold standard, we invited a human expert in logic programming to inspect all errors (i.e., diagnosis results) provided by the INCOM-WCBM system after analyzing 221 student solutions, either confirming or rejecting it. In addition, the human tutor had the possibility to add general comments which are not specific to the presented errors, for example, if he thought that crucial errors have been missed (due to high resource requirements for this human expert tasks, we did not involve multiple graders).

Once the gold standard was specified, we are able to determine the set of *gold standard errors* (which should have been identified by the system) and *gold standard not-errors* (which should not have been identified by the system). The sets *retrieved errors* and *not-retrieved errors* are the results of the systems diagnoses.

4.2 Results

To measure the diagnostic validity of an intelligent tutor, we use the metrics Recall and Precision. With respect to Table 1, Precision and Recall are defined as follows [9]:

$$Recall = \frac{A}{A + C}; Precision = \frac{A}{A + B}$$

Table 1. Categories for Precision and Recall

	Gold standard Errors	Gold standard Not-errors
Retrieved errors	A	B
Not retrieved errors	C	

Under these definitions, a high precision means that the model is based on fairly reliable constraints which have a low risk of producing false alarms, i.e., the developer was careful to avoid particularly risky constraints. A high recall, on the other hand, means that the diagnosis has a good coverage, i.e., it considers a sufficiently rich set of relevant constraints.

Table 2 summarizes the results of system diagnoses of INCOM-WCBM and INCOM-CBM with respect to diagnostic validity. From this table, we can notice three aspects. First, the precision of INCOM-WCBM is high (0.953), as is the recall (0.97). The latter indicates that the set of weighted constraints covers possible errors in the domain of logic programming sufficiently. As such, one can state that the diagnostic validity of WCBM-INCOM is high: one can expect this system to give appropriate feedback to students also in the situation of tasks that have multiple possible solution strategies (as is the case for most of the tasks we considered). Would these good results also have been possible without the constraint weights? Table 2 gives an answer to this: The second claim we can make is that the precision of INCOM-WCBM is remarkably higher than the one of INCOM-CBM (0.459). This can be attributed to the weight values associated to each constraint in the INCOM-WCBM, because constraint weights are used to determine the student’s intention and to control the error diagnosis process.

Table 2. Evaluation of the diagnostic validity

Task	INCOM-WCBM		INCOM-CBM	
	Precision	Recall	Precision	Recall
1	0.9	0.93	0.466	0.724
2	0.941	1	0.666	0.875
3	1	1	1	1
4	0.907	0.929	0.488	0.909
5	0.991	0.983	0.208	0.297
6	0.961	0.961	0.198	0.359
7	0.974	0.987	0.19	0.185
Avg.	0.953 (sd.=0.04)	0.97 (sd.=0.03)	0.459 (sd.=0.3)	0.621 (sd.=0.33)

Third, we notice that while the precision of INCOM-WCBM seems to be stable across the seven tasks, the precision of INCOM-CBM tends to decrease from task 4 on. This can be explained by the fact that the complexity of tasks 4-7 is higher than the one of tasks 1-4. In addition, we can see that both INCOM-WCBM and INCOM-CBM reach their maximum precision value at Task 3. Yet, this has to be interpreted in the light of the fact that only four student solutions were available for this task, and all of them contained very few errors.

We next want to illustrate the difference of diagnostic validity between INCOM-WCBM and INCOM-CBM using an erroneous example student solution for Task 6:

```
countz(N,L):- L=[], N is 0.
countz(N,L):- L=[Head|Rest], countz(N1, Rest), N is N1+1.
```

Task 6 can, among others, be solved by applying either a naive recursive strategy or a tail recursive strategy. Applying the weighted constraint-based model, INCOM-WCBM produced two hypotheses on the strategy level. The first hypothesis (H1) is that the student has implemented the naive recursive strategy and the student solution has violated three constraints, i.e., the solution has three errors (Table 3).

Table 3. Hypothesis 1 of INCOM-WCBM: The naive recursive strategy

ID	Weight	Feedback
s7c	0.8	At the position N , a number is expected. countz(N ,L):- L=[], N is 0.
p5c	0.8	The variable Head in the clause body is not used. Is it superfluous, or did you forget a subgoal to use it, or should it be an anonymous variable? countz(N,L):- L=[Head Rest], countz(N1, Rest), N is N1+1.
s5b	0.1	The arithmetic subgoal is superfluous. countz(N,L):- L=[], N is 0 .

Since the system does not allow the subgoal “N is 0” as a value assignment, the first constraint violation indicates that the variable **N** needs to be instantiated with a number and the third constraint violation shows that the arithmetic subgoal is not required. The second constraint violation shows that a variable **Head** is present but not used. Applying formula (1), the plausibility of this hypothesis is $0.8 * 0.8 * 0.1 = 0.064$.

The second hypothesis (H2) is that the student has implemented the tail recursive strategy. Following this hypothesis, the student solution violated constraints as listed in Table 4. These constraint violations occurred because the student tried to match the student solution with components of the tail recursive strategy which requires the following clauses: 1) the main clause which calls the accumulative predicate, 2) the base case of the accumulative predicate, and 3) a recursive clause which accumulates a value using an accumulative variable. Since the student solution could not be matched well to the tail recursive strategy (it violated five constraints, each with weight 0.01), the plausibility of this hypothesis is 0.01^5 . As such, H2 is less plausible than hypothesis H1. As a result, INCOM-WCBM decided that the student has most likely pursued the naive recursion strategy.

Table 4. Hypothesis 2 of INCOM-WCBM: The tail recursive strategy

ID	Weight	Feedback
s7g1	0.01	If you want to implement a non-recursive clause, at least one clause must have been specified as non-recursive.
s7g	0.01	A base case is missing.
s7h	0.01	A recursive case is required. Or did you forget a subgoal in a clause body?
s7i	0.01	countz/2: this predicate definition has more base cases than required. countz(N,L):- L=[], N is 0.
s7j	0.01	countz/2: this predicate definition has more recursive cases than required. countz(N,L):- L=[Head Rest], countz(N1, Rest), N is N1+1.

INCOM-CBM also generated two hypotheses. The first hypothesis (H1A) is that the student has implemented the naive recursive strategy. This hypothesis caused, in addition to three constraint violations in Table 3, two others, which address the arithmetic argument in the second clause. (`countz(N,L) :- L=[Head|Rest], countz(N1, Rest), N is N1+1`):

1. At the position **N1**, a constant number is required.
2. At the position **1**, a variable is required.

These additional constraint violations resulted from the fact that INCOM-CBM was not able to choose the most plausible hypothesis generated on the solution variant level. By matching the arithmetic term $N1+1$ of the student solution against the semantic table, two hypotheses have been generated: $H1_1 = \{map(N1, N1'); map(1,1)\}$ and $H1_2 = \{map(N1, 1); map(1,N1')\}$, where $N1'+1$ is a corresponding arithmetic

term specified in the semantic table. INCOM-CBM chose the second hypothesis on the solution variant level and forwarded this to the strategy level. Applying formula (2), the plausibility for hypothesis H1A is the number of violated constraints, i.e., the plausibility is 5.

The second hypothesis (H2A) INCOM-CBM has generated is that the student has implemented the tail recursive strategy. Diagnosing errors following this hypothesis, INCOM-CBM produced the same five constraint violations as in Table 4. That is, the plausibility of this hypothesis is also five and equal to the plausibility of hypothesis H1A. Thus, INCOM-CBM was not able to decide on the most plausible hypothesis about the student's solution variant, because two hypotheses of INCOM-CBM (naive recursive and tail recursive) have the same plausibility score (each hypothesis produces five constraint violations). Therefore, the system could not decide which strategy was most likely pursued by the student. Many of the cases where INCOM-WCBM had a higher diagnostic validity than INCOM-CBM can be explained in a similar fashion: the weights outperformed the simple counting of constraint violations.

4.3 Possible Limitations

Overall, our results indicate that adding constraint weights can improve the error diagnosis of constraint-based intelligent tutors. Yet, our results concerning the diagnostic validity of INCOM-WCBM might be a bit optimistic, because our method of determining the gold standard was based on actual system's diagnosis results. This might have created a bias toward these error interpretations. Other comparable tutor systems for programming, e.g. PROUST [10], APROPOS2 [11], and Hong's Prolog tutor [12], which also provide problems with multiple solution variants, defined the gold standard by hand analysis. That is, a human expert analyzed each student solution and detected errors independent from the system's diagnostic result. However, this way of defining a gold standard by hand analysis is not well-suited for constraint-based tutors due to two reasons. First, the human expert has to know the large set of constraints (the current implementation of INCOM includes 147 constraints [13]) which represent error types, and relate every error detected in a program to a corresponding constraint. This is a very laborious undertaking for a human expert. Second, a constraint can be relevant to different components of the same solution many times. If a human expert has to assign a detected error to one of the existing constraints, she would have to iterate through the list of constraints as many times as the system does. This is a bothersome and error prone task. Hence, we specified the gold standard in a way that provides a balance between human and system orientation.

Another possible limitation is that, in our study, we compared INCOM-WCBM to a classical constraint based tutor which made use of a reasonable but quite straightforward method for determining student strategies. Adding more advanced features (i.e., more sophisticated methods for guessing solution strategies) could probably have increased the diagnostic validity of our "control condition" INCOM-CBM. Yet, it remains to be shown if, with additional features but without weights, the diagnostic validity could have reached the level that can be reached with weighted constraints as available in INCOM-WCBM.

5 Conclusion

In this paper we have presented a weighted constraint-based model for intelligent tutors. We also demonstrated an evaluation study which compared the diagnostic validity of a tutor applying the weighted constraint-based model and a classical constraint-based tutor for the same domain in logic programming. The evaluation study showed that the precision of error diagnosis provided by the weighted constraint-based tutor (0.953) is remarkably higher than the one of the classical constraint-based tutor (0.459). From this result and the evaluation study in [8], we can conclude that the error diagnosis capability of constraint-based tutors can be improved if constraints are enriched with weight values which represent the importance of a constraint. In the future, we plan to test the applicability of the weighted constraint-based model in other domains.

References

1. Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R.: Cognitive tutors: Lessons learned. *Journal of the Learning Sciences* 4, 167–207 (1995)
2. Ohlsson, S.: Constraint-based Student Modeling. In: Greer, J.E., McCalla, G.I. (eds.) *Student Modelling: The Key to Individualized Knowledge-based Instruction*, pp. 167–189. Springer, Berlin (1994)
3. Anderson, J.R., Betts, S., Ferris, J.L., Fincham, J.M.: Neural imaging to track mental states while using an intelligent tutoring system. *Proceedings of the National Academy of Science* 107, 7018–7023 (2010)
4. Dubois, D., Fargier, H., Prade, H.: Possibility theory in constraint satisfaction problems: Handling priority, preference and uncertainty. *Applied Intelligence* 6, 287–309 (1996)
5. Schiex, T., Cedex, C.T., Fargier, H., Verfaillie, G.: Valued constraint satisfaction problems: Hard and easy problems. In: *Joint Conference in AI (1995)*
6. Freuder, E., Wallace, R.: Partial constraint satisfaction. *Artificial Intelligence* 58, 21–70 (1992)
7. Fargier, H., Lang, J.: Uncertainty in Constraint Satisfaction Problems: A Probabilistic Approach. In: Moral, S., Kruse, R., Clarke, E. (eds.) *ECSQARU 1993*. LNCS, vol. 747, pp. 97–104. Springer, Heidelberg (1993)
8. Le, N.-T., Pinkwart, N.: Adding Weights to Constraints in Intelligent Tutoring Systems: Does It Improve the Error Diagnosis? In: Kloos, C.D., Gillet, D., Crespo García, R.M., Wild, F., Wolpers, M. (eds.) *EC-TEL 2011*. LNCS, vol. 6964, pp. 233–247. Springer, Heidelberg (2011)
9. Rijsbergen, C.J.V.: *Information retrieval*, 2nd edn. Butterworths, London (1979)
10. Johnson, W.L.: Understanding and debugging novice programs. *Artificial Intelligence* 42(1), 51–97 (1990)
11. Looi, C.-K.: Automatic debugging of Prolog programs in a Prolog intelligent tutoring system. *Instructional Science* 20, 215–263 (1991)
12. Hong, J.: Guided programming and automated error analysis in an intelligent Prolog tutor. *International Journal of Human-Computer Studies* 61(4), 505–534 (2004)
13. Le, N.-T.: Using weighted constraints to build a tutoring system for logic programming. PhD Thesis. University of Hamburg (2011)

Identification and Classification of the Most Important Moments from Students' Collaborative Discourses

Costin-Gabriel Chiru¹ and Stefan Trausan-Matu^{1,2}

¹ "Politehnica" University of Bucharest, Department of Computer Science and Engineering,
313 Splaiul Independetei, Bucharest, Romania

² Research Institute for Artificial Intelligence of the Romanian Academy,
13 Calea 13 Septembrie, Bucharest, Romania
{costin.chiru, stefan.trausan}@cs.pub.ro

Abstract. In this paper we present a method that combines the cognitive and socio-cultural paradigms for automatically identifying the most important moments (the so-called pivotal moments) from a Computer Supported Collaborative Learning chat. The existing applications do not identify these moments and we propose a flexible visual method for filling this gap. Since these moments may have different roles in a discourse, we also propose a classification of the identified types of important moments from chat conversations.

Keywords: CSCL, Pivotal Moments, Discourse Analysis, Polyphony Theory.

1 Introduction

This paper proposes a method and a visualization tool for analyzing Computer Supported Collaborative Learning (CSCL) instant messenger (chats), with the main purpose of identifying and classifying the most important moments from such chats. The identification of these moments is extremely important for both the learners and the tutors since it provides hints about the areas where specific topics are debated and it is able to capture the connections that exist between different topics or the strength of a topic compared with the strength of other topics. For students this information helps learning in the phase of searching for answers to different problems since it can serve to build a better retrieval system of relevant texts, because it is possible to identify the areas from the chat where specific concepts are debated and therefore the retrieval system could index and retrieve only parts of that chat instead of the whole chat. More than that, the information provided by the identification of the most important moments could also suggest what solution has been chosen for solving a specific problem if multiple such solutions have been identified in the chat. For tutors, this information is helpful in providing an overview of the understanding students have on the topics debated in the chat since it reflects how well they understood the notions related to a given topic and also shows how they relate different topics. In the same time, it can provide information about differential positions relating the debated topics and whether these positions are finally reaching a consensus or not [7].

The method presented here combines the cognitive and socio-cultural paradigms in the analysis of CSCL chats under the concept of voice from the Polyphonic Theory [7, 8] and the WordNet (<http://wordnet.princeton.edu>) linguistic database. It uses Natural Language Processing (NLP) techniques [2] (for example, building lexical chains starting from the given text and a linguistic database) and the ideas related to identifying polyphonic threads presented in [8].

We have used the implemented system for the analysis of CSCL chats consisting of 4-8 participants debating about which is the best tool for collaborative learning. In the preparation of these chats, the students have been divided by the tutors in groups and each student from the group has been provided with learning materials about a specific topic from the Human-Computer Interaction (HCI) domain (chat, blog, forum, wiki). The students were supposed to study their specific topic and to defend it in a “confrontation” with the other students supporting their own topics. The desired outcome of these chats was the understanding in further detail of all the considered topics by all the participants. This outcome could only be reached if each participant would share its own knowledge with the others and, through debate, would be able to compare and relate to each other the considered topics.

In the next section we will state the theoretical ideas that represent the basis of this system. The paper continues with the presentation of the application and of the method used for identifying the important moments from a chat. As a consequence of the multiple things that may be observed in a chat after encountering such an important moment, we propose a classification of the identified important moments that is described in section 4. The paper concludes with our observations regarding the proposed method and classification.

2 Theoretical Ideas

In the discourse analysis field, two major directions can be identified: the cognitive paradigm, considered in NLP - “focusing on the knowledge in individuals’ minds” [7] - and the socio-cultural paradigm [9] - “stating that learning is achieved socially” [7].

One of the applications of the cognitive paradigm in Artificial Intelligence was to support learning, leading to the development of Intelligent Tutoring Systems that were trying to teach students by transferring knowledge from a tutor (human or computer) to them. Unfortunately, these systems did not acquire the expected results and, as a consequence, other theories for learning were searched for. In this context, the socio-cultural paradigm (stating that the knowledge is socially constructed) was considered suitable: Mikhail Bakhtin [1] introduced a new perspective in which dialogue was seen as a central concept and this idea was applied in learning: “discourse should be a central issue in a theory about learning” [7].

Bakhtin started from the polyphony model of the musical domain and extended it to discourse, considering that “the voices of others become woven into what we say, write and think” [3]. Therefore, the knowledge is acquired from the discussion with the other participating voices by interweaving the ideas expressed by each of the voices: “rather than speaking about ‘acquisition of knowledge,’ many people prefer to

view learning as becoming a participant in a certain discourse” [6]. Discourse is seen as a tool for enhancing learning, the things that are learnt reflecting the ideas of the contributing voices.

Bakhtin has also introduced another very important theory – the polyphonic character of some texts – stating that the voices that are present in those texts are influencing each other, which leads to inter-animation of the ideas presented by these voices.

The notion of voice, that is central in the work of Bakhtin, represents not the physical, vocal expression of a participant but rather a distinct position taken by one or more of the participants that is discussed in the conversation and that influences the subsequent evolution of the conversation.

Current approaches that implemented Bakhtin’s theory ([5], [8], for example) considered only two perspectives on the notion of voice: a voice might be represented by either an utterance from a conversation or by a participant. None of these approaches had the purpose of identifying the important moments from a conversation. Moreover, it is much more difficult to consider these approaches in narrative texts since it is very difficult to detect when different participant interfere and what an utterance means.

3 The Application

In this paper we propose an implementation that considers, from the perspective of Bakhtin’s dialogism, that a voice is a position, an idea expressed by those participants. Since each word is a potential voice by this definition, we needed to be able to identify the influence each word has on the subsequent conversation – the echoes (repetitions or more complex forms) of that particular word in the given chat. Therefore, we have used the lexical chains that could be built starting from that word and the WordNet database [4] to capture its echoes in the form of the repetitions of that word. The most important such lexical chains could become the voices of the discourse, but the problem of identifying which voices are of greater interest has been left to the user, since it depends on what that particular user is actually looking for.

Besides the concept repetitions, we have also investigated the repetition of the form of the words (paronymy) as another source of unity-difference and inter-animation in the discourse (and therefore another element that can be considered when investigating the voices – in fact, another type of voice). This type of repetition was also needed as a way to counter-balance the spelling errors that were present in the analyzed discourses.

Since we wanted to give the user the possibility to choose what he/she wants to see, we provided three independent options (chains of exact repetitions, chains of conceptually-related words and paronyms chains) and let the user choose one or more of them, therefore making the application more flexible.

The application offers a couple of different views of the chat. The implicit visualization consists in showing the content of the chat that is analyzed and the vocabulary (the important, non stop-words [2] in the chat). If one of the concepts from the vocabulary is selected, the occurrences of that concept in the text are highlighted using the blue color. If the options to use the synonyms/lexical chains and/or the paronyms

from the right area are checked, then the semantically related words are also highlighted using the yellow color while the paronyms are highlighted using the green color. An example of highlighted text is given for the concept of “chat” in the conversation presented in Fig. 1.

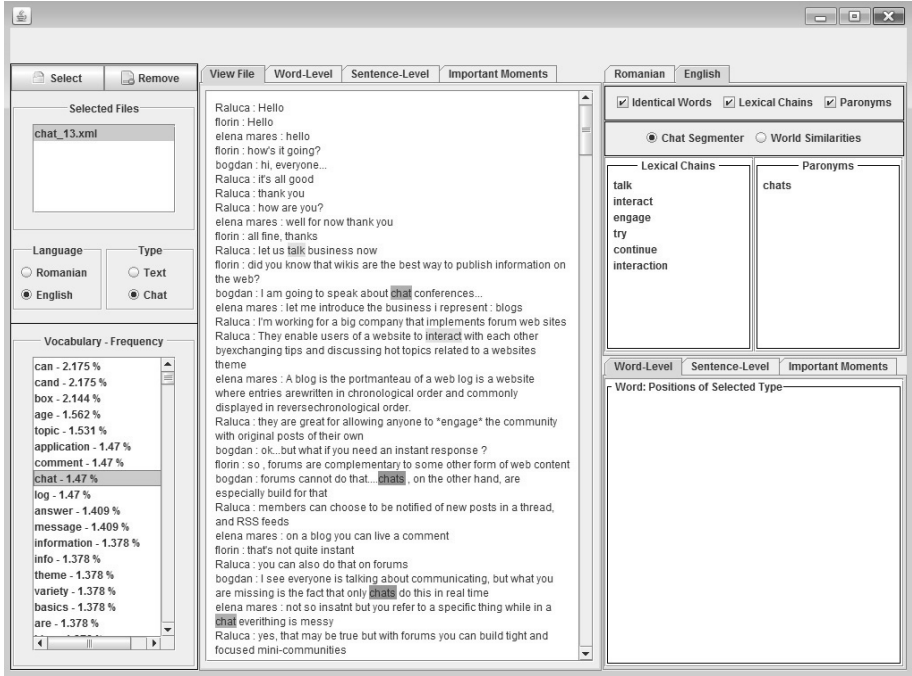


Fig. 1. Visualization of the “chat” concept (blue stands for identical words, yellow for words from the lexical chains of the chosen concept and green for paronyms of the chosen concept)

Another way to visualize the chat is represented by a graph describing the distribution and frequency of the concepts selected by the user. This graph allows the user to choose one or more different voices from the given chat and to visualize their flow in the text (see Fig. 2 for an example). Therefore, from now on we will be calling this graph as “voices visualization”.

In this graph we have represented the text using a number of points in the available area that depends on the display resolution. Each voice is represented by a thread of a different color. The points from each voice are placed in the position corresponding to the occurrence of that concept in the given text (the logic of the representation is an array of such nodes being similar to the one of writing the words using a text editor – such as Microsoft Word for example). We have also connected the nodes corresponding to a concept in order to visualize easier the flow of the voice they represent. This method of visualization is not independent to the previous one (the implicit visualization): one can observe the context of an occurrence of a concept considered important by clicking on that occurrence. In this case, the user is redirected to the implicit

visualization to see the text of the chat, where the occurrence considered important is highlighted with the same color as the voice from this visualization (now the whole sentence is highlighted and not individual words as in the previous visualization).

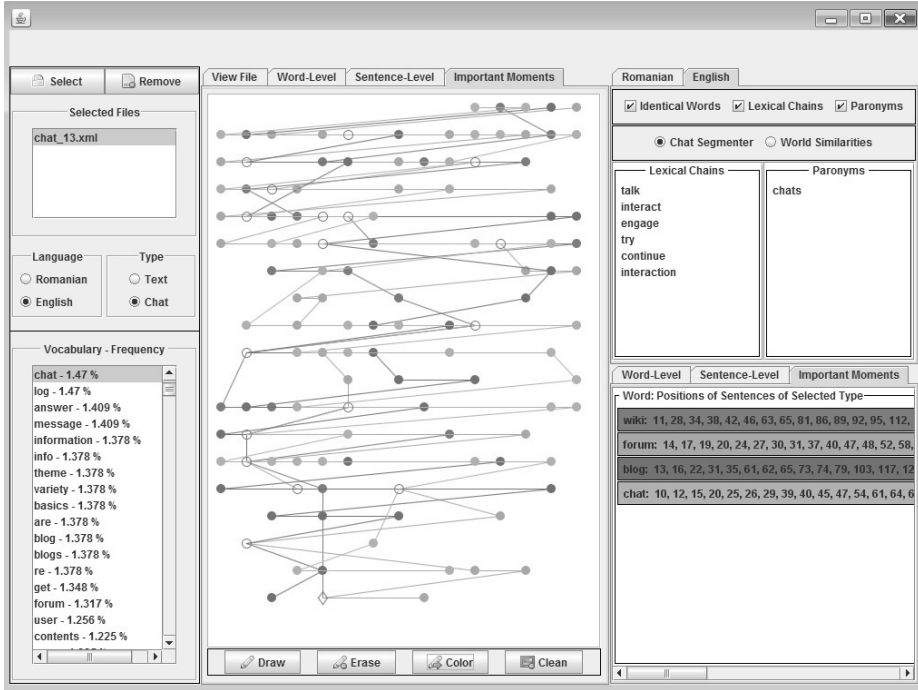


Fig. 2. The visualization of voices and of their inter-animation. \diamond stands for a singular moment, while with \circ we represented the threads of meeting points.

4 The Identification and Classification of the Most Important Moments from a Chat

Starting from the voices that are considered important by a particular user, we have investigated the areas from the chat where these voices might inter-animate [7, 8] (influence each other or co-participate to the utterance meaning) and considered these to be the most important moments from a chat. In order to influence each other (to inter-animate), the voices have to be close enough. Therefore, we considered that two or more voices inter-animate if they are found in the same unit of text, which for us means the same utterance.

As a consequence of identifying the areas where the voices inter-animate each other, the application is able to identify the important moments from a discourse – moments where something happens after the inter-animation of different voices: all the voices die out (cannot be observed in that discourse from that moment on); only a part of them die, the rest being further present in the discourse; the voices continue to be present in different areas of the discourse; one voice substitutes the other, etc.

Considering the observed types of interactions that are possible between the voices, we propose a classification of the important moments from a discourse in 5 different classes: pivotal moments, moments of convergence, singular moments, moments of divergence, and meeting points.

The different types of important moments from the discourse are represented in the voices visualization graph using 4 different symbols: a triangle for the pivotal moments, a square for the convergence moments, a diamond for the singular moments and an empty circle for the divergence moments. We considered that there is no need to introduce another symbol for the meeting points, since they can be interpreted as multiple divergence moments. An example of the important moments' visualization is offered in Fig. 2, where one can see a couple of meeting points and a singular moment close to the end of the file.

4.1 Pivotal Moments

In our opinion, pivotal moments are the most important type since they represent the switch from one voice to another (from one concept to another). The pivotal moment is identified when two voices are present in the same utterance, and one of the voices seizes its presence in the discourse, while the other one just starts it.

To exemplify this type of moments, we have considered the voices of “information” and “problem” in the graph from Fig. 3. As it can be seen, until that moment (represented by a triangle) the voice of “information”, with a local distribution in the middle of the conversation, has been present in the discourse, while from that moment on, this voice disappears and it is replaced by a new voice represented by “problem”.

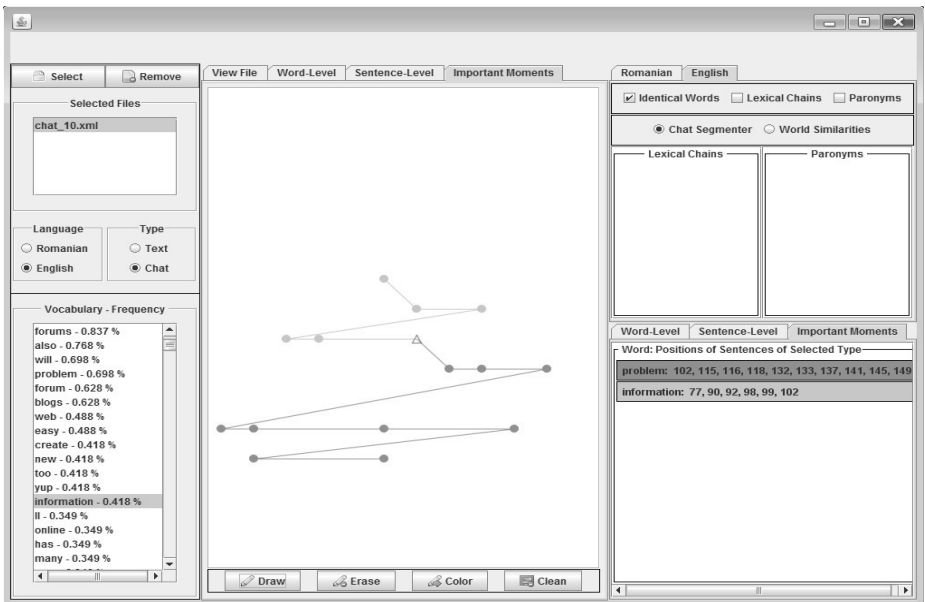


Fig. 3. The visualization of a pivotal moment. Δ represents the pivotal moment.

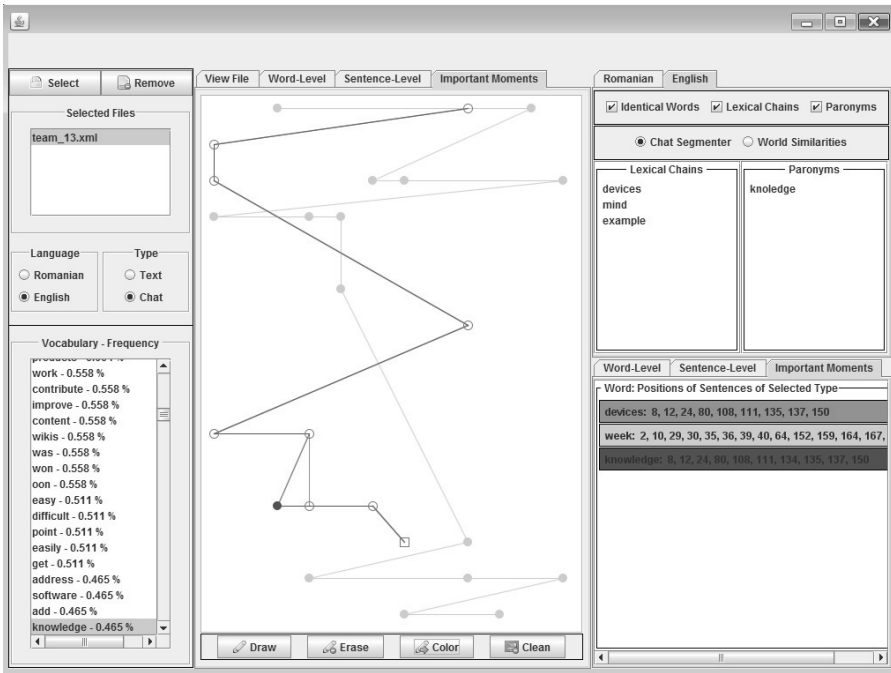


Fig. 4. The visualization of some meeting points and of a convergence moment. □ stands for a convergence moment, while with ○ we represented the threads of meeting points.

4.2 Convergence Moments

This type of moments is present when two or more voices meet and after that all of them die out – disappear from the text. This type of moments (represented by a square) may have the meaning of resolving all (or at least a part of) the dissonances that appear in discourse [7], of unifying the voices. It is like a conclusion regarding the voices that get to the convergence moment. This is why most of the time such moments are present towards the end of discourses. One such example is given in Fig. 4, where in a single utterance one can see the last occurrences of the voices “devices” and “knowledge”.

4.3 Singular Moments

This kind of moments – represented by diamonds in the graph – can be defined as the situation when two or more voices meet each other and all of them die out but one. The result is that only one from many voices continues to be present in the discourse, fact that made us calling it this way. The meaning of singular moments is the existence of a divergence between multiple voices, that meet to confront each other in a point of the discourse and one of the voices – the most important one, “the loudest” – dissolves the others, so that from that moment on the discussion focuses only on it.

An example can be seen in Fig. 5 where such a singular moment is found at the “confrontation” between the voices “topic” and “post”. After a couple of times where the two voices meet, the final “confrontation” is won by the “topic” voice which continues to be present in the chat, while the other voice (“post”) seizes its presence.

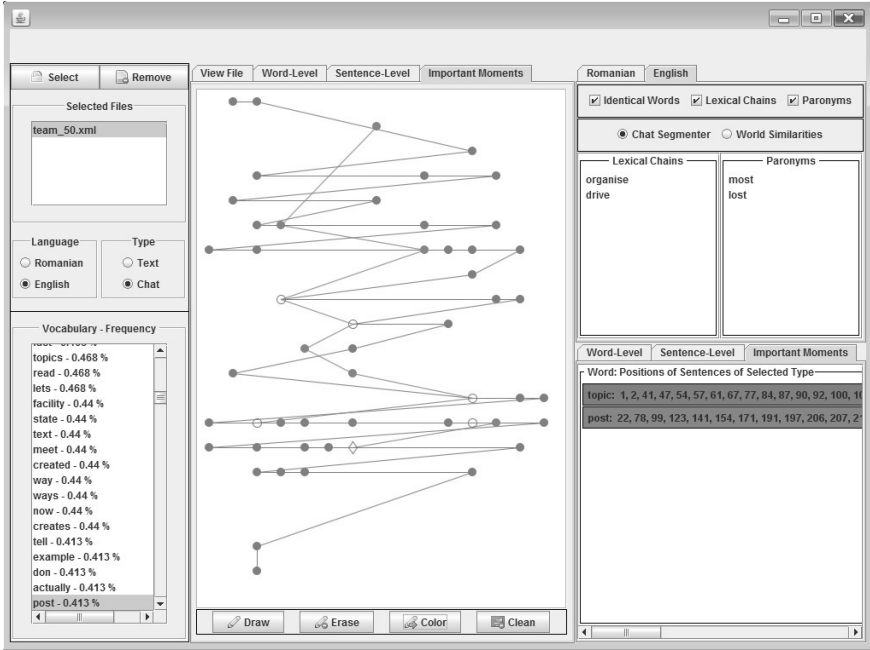


Fig. 5. The visualization of a singular moment. ◊ stands for a singular moment, while with ○ we represented the threads of meeting points.

4.4 Divergence Moments

The divergence moment is defined as the moment when two or more voices meet and after that they continue to be present in other utterances from the discourse. It is like a fight between multiple voices – with all the voices strong enough so that not to be assimilated by the other voices – and after this fight every voice continues its own flow in the discourse. An example is shown in Fig. 4 where the voices of “week” and “knowledge” meet at the beginning of the chat and after that they are present in different areas of the chat without interacting with each other.

4.5 Meeting Points

The last type of moments identified in a discourse is in fact a chain of important moments, unlike the previous situations where only one point from the discourse was identified. This kind of moments – the meeting points – can be observed when two or multiple voices are constantly debating during the discourse. They meet in several

points and they continue to be present and to interact with each other or with other voices, usually their flows being parallel.

These moments could be considered as multiple divergence points of the same voices, but in fact that is a misjudgment because there is a totally different situation. In the case of divergence moments there are two or more voices that interact once and after that each of them flows in different areas of discourse without interacting again – it is like a fight that has not been resolved, each of the participant voices continuing its own flow. In the case of the meeting points, there are multiple voices that fit very well together, be it because they are semantically related but the link between them has not been considered – either because the user did not select the lexical chains button, or because the used lexical database has flaws (missing links) that did not allow the reconstruction of the connection between the voices – or because they are discourse related: constructions such as collocations, syntagms or idioms. Therefore, it is extremely important to make the difference between the divergence and meeting points. On the provided graph, the difference between them can be identified considering the number of empty circles from the interaction of some considered voices: if there is only one such circle, we have a divergence moment; if multiple such points are present, then we have some meeting points.

An example that falls in the first category of meeting points is provided in Fig. 2, where the voices of “wiki”, “chat”, “blog” and “forum” (that are related from the HCI point of view, but are unrelated in the WordNet database) meet many times. An example from the second category is presented in Fig. 6 where some meeting points generated by two collocational words are present: “quality” and “control”.

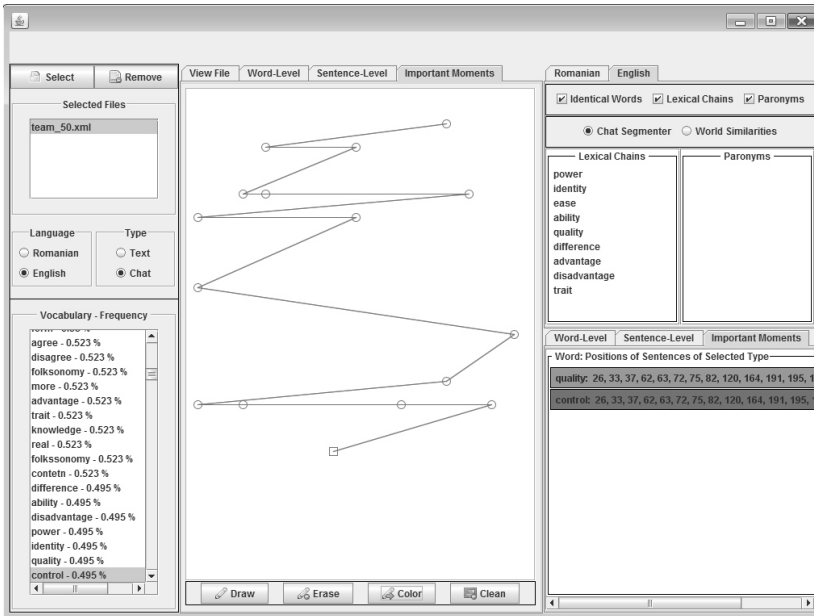


Fig. 6. The visualization of some meeting points. □ stands for a convergence moment, while with ○ we represented the threads of meeting points.

5 Conclusions

In this paper we have proposed a classification of the most important moments of a discourse and a visual method and implemented system for their identification. The resulted application is domain independent (since it is based on a general purpose database), language independent (as long as there is a means to extract the voices - the threads of ideas - from the discourse) and it is flexible, letting the user decide what voices he/she wants to analyze from the given discourse.

This method could be used for identifying the most important moments from a discourse, which could also give information about the areas where specific topics are debated in a chat, about the collocations, syntagms and idioms that are encountered in that chat or about the identification of missing links in the used lexical database. Other tasks in which this application could be used would be the identification of how “strong” different voices are (from the point of view of the chat participants), how focused these voices are, what are their types: local (artifacts) or global, which are the voices that can (or cannot) be used in the same area of text, the identification of the topic drifts (the areas where the debate was off-topic) or the disambiguation of the polysemous words by considering the context provided by the voices that are found in the vicinity of the polysemous word.

Acknowledgement. This research was supported by project No.264207, ERRIC-Empowering Romanian Research on Intelligent Information Technologies/FP7-REGPOT-2010-1.

References

1. Bakhtin, M.M.: *Problems of Dostoevsky's Poetics*, Ardis (1973)
2. Jurafsky, D., Martin, J.H.: *Speech and Language Processing: An Introduction to NLP, Computational Linguistics and Speech Recognition*, 2nd edn. Prentice-Hall (2009)
3. Koschmann, T.: *Toward a Dialogic Theory of Learning: Bakhtin's Contribution to Learning in Settings of Collaboration*. In: *Proceedings of the CSCL 1999 Conf.*, pp. 308–313 (1999)
4. Morris, J., Hirst, G.: *Lexical Cohesion, the Thesaurus, and the Structure of Text*. *Computational Linguistics* 17(1), 211–232 (1991)
5. Rebedea, T., Dascălu, M., Trausan-Matu, S., Banica, D., Gartner, A., Chiru, C., Mihaila, D.: *Overview and Preliminary Results of Using PolyCAFe for Collaboration Analysis and Feedback Generation*. In: *Proceedings of ECTEL 2010*, pp. 420–425. Springer (2010)
6. Sfard, A.: *On reform movement and the limits of mathematical discourse*. *Mathematical Thinking and Learning* 2(3), 157–189 (2000)
7. Trausan-Matu, S.: *The Polyphonic Model of Hybrid and Collaborative Learning*. In: Wang, Fong, Kwan (eds.) *Handbook of Research on Hybrid Learning Models: Advanced Tools, Technologies, and Applications*, pp. 466–486. Information Sc. Publishing, Hershey (2010)
8. Trausan-Matu, S., Rebedea, T.: *A Polyphonic Model and System for Inter-animation Analysis in Chat Conversations with Multiple Participants*. In: Gelbukh, A. (ed.) *CICLING 2010*. LNCS, vol. 6008, pp. 354–363. Springer, Heidelberg (2010)
9. Vygotsky, L.: *Mind in society*. Harvard University Press, Cambridge (1978)

When Less Is More: Focused Pruning of Knowledge Bases to Improve Recognition of Student Conversation

Mark Floryan, Toby Dragon, and Beverly Park Woolf

Department of Computer Science, University of Massachusetts, Amherst
140 Governors Dr. Amherst, MA USA
{mfloryan, dragon, bev}@cs.umass.edu

Abstract. Expert knowledge bases are effective tools for providing a domain model from which intelligent, individualized support can be offered. This is even true for noisy data such as that gathered from activities involving ill-defined domains and collaboration. We attempt to automatically detect the subject of free-text collaborative input by matching students' messages to an expert knowledge base. In particular, we describe experiments that analyze the effect of pruning a knowledge base to the nodes most relevant to current students' tasks on the algorithm's ability to identify the content of student chat. We discover a tradeoff. By constraining a knowledge base to its most relevant nodes, the algorithm detects student chat topics with more confidence, at the expense of overall accuracy. We suggest this trade-off be manipulated to best fit the intended use of the matching scheme in an intelligent tutor.

Keywords: knowledge base, ill-defined domains, collaboration.

1 Cognitive Support for Collaborative Inquiry Systems

While great strides have been made in the categorization and improvement of Intelligent Tutoring Systems (ITS) that support work in ill-defined domains [1] inherent challenges exist in working within these loosely structured spaces. One challenge is to identify the current focus of student work. The introduction of collaboration among students can create an even greater chance that students will become sidetracked, but also provides novel opportunities to automatically recognize the content focus of students. Our current research utilizes an expert knowledge base to detect student focus and identify opportunity for intervention. This paper presents a specific attempt to understand how pruning of an expert knowledge base can affect the content recognition of student discussion within a collaborative inquiry learning system.

For the remainder of this paper, we describe the related research (Section 2) and how our current research builds upon it (Section 3). We present the approach and methodology of the study (Section 4). Finally, we conclude by discussing the results and recognizing the tradeoff between *confidence* and *overall accuracy* that is observed after pruning the knowledge base (Section 5).

2 Related Work

Some previous work has focused on utilizing expert knowledge bases to detect patterns in student actions. Rahati and Kabanza describe a system that detects when student's constrained interactions are useful for learning [4]. Chen and Mostow constructed a model of predictable student responses [5] within a reading tutor and are able to detect on task behavior, but not offer dynamic feedback. These previous efforts are based on an assumption that in a constrained system, user actions can be predicted. These attempts succeed specifically because of the constrained nature of student interactions.

We can see potential for our theory of a confidence / overall accuracy tradeoff when changing the size of the knowledge base when considering [6]. This project took the opposite approach, increasing the size of their knowledge base using an online resource. Using this larger knowledge base to recognize student solutions, they report an increase in recall (number of recognized solutions) along with a decrease in precision (a measure of confidence in the solutions). We also look to recent work outside of the ITS community, in the field of machine-learning classification [7] that demonstrates the power of harnessing implicit expert knowledge encoded in the dataset to make informed decisions about pruning.

Finally, when considering recognition of textual input in order to support students' learning, we must also consider the offerings provided by the field of natural language understanding. Several researchers offer contributions in this manner [8, 9], yet they approach a different problem, and offer a different solution. The focus of this work is to mine large datasets for valuable information, placing emphasis on the sorting and filtering of data. Our work uses a smaller custom-built knowledge base to provide the set of items from which to identify helpful information for student use.

3 Rashi: An Inquiry Learning System with Collaborative Features

The following experiments were conducted using data collected from Rashi, a collaborative inquiry-learning system that provides the tools and environments necessary for students to consider authentic, real-world problems [2]. Students engage in inquiry learning by collecting data, (question / answer interface, interactive images, etc.) and formulating hypotheses, providing an introduction to methods commonly used by professionals. Although the framework is domain-independent, the students participating in these studies focused on challenges involving human biology, where their task is to diagnose ill patients.

Rashi provides several forms of collaborative features to support students. These features allow students to view and monitor the work of a peer, offer critiques of specific discussable objects, and receive feedback regarding discussions of interest. These features have been shown to prompt an increase in hypothesis creation, data collection, and recognizing connections between data and hypotheses [10]. Rashi also provides a chat facility that enables students to have unconstrained discussions with members of their group (Figure 1). The focus of the following studies is to detect the content of student chat, in order to provide personalized feedback. We attempt to do so by utilizing our system's expert knowledge base.

Our expert knowledge base (EKB) provides both the enumeration of the individual subjects we seek to identify, as well as the semantics necessary to provide support after identification [2]. The EKB is a directed, acyclic graph of domain concepts connected with supporting and refuting relationships (Figure 2).

Fig. 1 (left) and 2. (right): Students chat with group members to discuss the patient’s illness (left). These messages are matched against nodes from the knowledge base (right).

Rashi also has an established, text-matching algorithm that matches chat message content to the knowledge base. Previous effort demonstrated an average success rate of 70% in matching messages to content [3]. However, the confidence in any given judgment could be quite low (below 60%). Thus, we were motivated to experiment with methods of increasing confidence.

4 Research Design

We observed that chat tends to focus on the relevant aspects of the case. Thus, we analyzed the change in matching efficacy after pruning the knowledge base’s least relevant nodes. We defined relevance as the connectedness of particular hypothesis:

$$Relevance(H) = |inEdges(H)| + |outEdges(H)|$$

We were able to prune data by using a Boolean value that defines the node as case-specific or not. The algorithm was executed using the full knowledge base, and three successive levels of pruning. Each of these conditions was repeated over the messages from two cases (anemia and hyperthyroidism case). The conditions were:

- **All:** Full Knowledge Base
- **Min Hypo Relevance > 2:** The minimum hypothesis relevance must be greater than or equal to 2 to be included in the search.
- **Min Hypo Relevance > 5:** The minimum hypothesis relevance must be greater than or equal to 5 to be included in the search.
- **Min Hypo Relevance > 5 + Restricted Data:** Same as above, with the additional condition that only case-specific data nodes are included.

For each condition, the algorithm outputs the chat, and which node (if recognized) is the subject of that message. If no match is found, then the algorithm assumes the message not related to domain content and outputs “No Match”.

A human judge examined the algorithm’s output and placed each line of output into one of four categories: correctly matched (+); correctly ignored / not matched (+); incorrectly matched (-); or not matched / ignored even though an appropriate match existed (-). Once completed, we analyzed the results to determine how the algorithm was affected by the pruning of the knowledge base. We considered two statistics.

Match Confidence: A measure of how likely the average match given is correct.

$$Confidence = \text{Correct Matches} / [\text{Correct Matches} + \text{Incorrect Matches}]$$

Overall Accuracy: A measure of the total efficacy of the algorithm.

$$Accuracy = [\text{Correct Matches} + \text{Correct Non-Matches}] / \text{Total Messages}$$

Our data spanned multiple dialogues produced by students of varying age (middle-school - college), and after varying amounts of work time (45 min - 2 hrs). To eliminate bias, the human judge worked without knowledge of which match belonged to which condition.

5 Results and Conclusion

Figure 3 shows the raw data results for the two cases we considered.

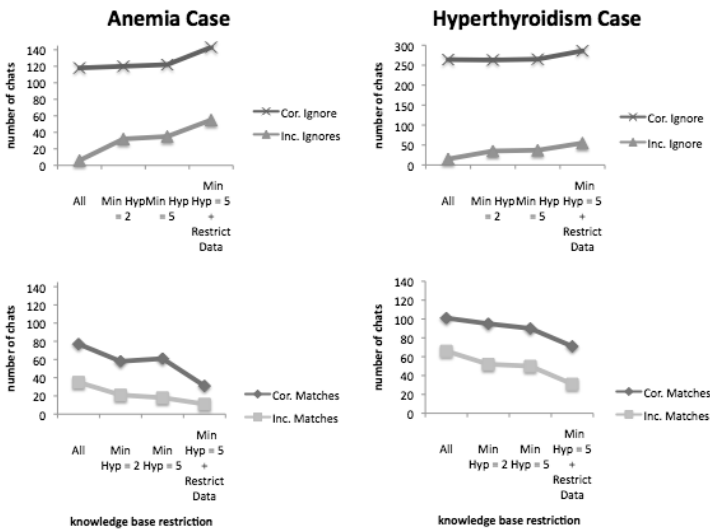


Fig. 3. Raw data results for both cases

We see that as we restrict the knowledge base, the number of matched chats decreases, while unmatched chats increases. The algorithm, over both cases, achieved overall accuracies between 72 and 82 percent. In addition, the confidence of matches ranged from 60 to 77 percent.

Figures 3 and 4 show the relationship between the accuracy of the algorithm, and the match confidence when restricting the knowledge base. We see that the overall accuracy tends to decrease, while the percentage of correct matches tends to increase. In addition, Figures 3 and 4 show that as we prune the knowledge base, we cannot recognize as many total individual pieces of dialogue, which is expected.

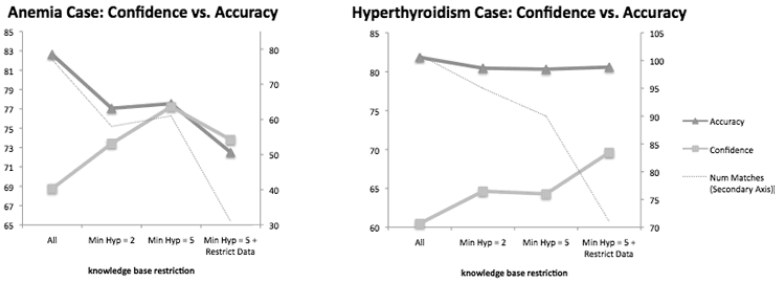


Fig. 4 and 5. Confidence vs. Accuracy for the each case

Because our human judge’s responses produce a distribution over four categories (see Section 4 for the judge’s response options), we used chi-squared tests to ensure the distributions from conditions are independent. Table 1 shows the results of these tests. The dependent condition is labeled on the y-axis, while the independent condition is on the x-axis. Thus, we see that the majority of our pruning levels produce significant changes in algorithmic behavior.

Table 1. Chi-square statistics, significant changes in algorithmic behavior were found

Anemia Case	Min Hyp = 2	Min Hyp = 5	Min = 5 + Data
All	*1.75E-26	*1.03E-32	*4.39E-97
Min Hyp = 2		0.826	*2.48E-08
Min Hyp = 5			*4.07E-07

Hyperthyroidism Case	Min Hyp = 2	Min Hyp = 5	Min = 5 + Data
All	*1.38E-06	*3.89E-08	*2.79E-29
Min Hyp = 2		0.925	*3.66E-06
Min Hyp = 5			*7.71E-05

In conclusion, we find that simple keyword matching to an expert knowledge base holds serious potential for identifying the content of student conversation within noisy environments. We find that the breadth of a knowledge base has a direct effect on the quality of subject recognition. If nodes are pruned to the most relevant, then subject recognition can be done with a significant increase in confidence, at the cost of the breadth of student input that can be identified.

We believe that this tradeoff is a useful observation for designers of Intelligent Tutors who utilize expert knowledge bases. When offering support utilizing expert

knowledge base matching, we show that the knowledge base can be intelligently pruned according to whether increased confidence or overall accuracy is preferable.

Acknowledgements. This research was funded by an award from the NSF 0632769, IIS CSE, Effective Collaborative Role-playing Environments, (PI) Beverly Woolf, with Merle Bruno and Daniel Suthers. Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the views of the funding agencies.

References

1. Lynch, C., Ashley, K.D., Pinkwart, N., Alevan, V.: Concepts, Structures, and Goals: Redefining Ill-Definedness. *International Journal of AI in Education; Special Issue on Ill-Defined Domains* 19(3), 253–266 (2009)
2. Dragon, T., Park Woolf, B., Marshall, D., Murray, T.: Coaching Within a Domain Independent Inquiry Environment. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006. LNCS*, vol. 4053, pp. 144–153. Springer, Heidelberg (2006)
3. Dragon, T., Floryan, M., Park Woolf, B., Murray, T.: Recognizing Dialogue Content in Student Collaborative Conversation. In: Alevan, V., Kay, J., Mostow, J. (eds.) *ITS 2010. LNCS*, vol. 6095, pp. 113–122. Springer, Heidelberg (2010)
4. Rahati, A., Kabanza, F.: Persuasive Dialogues in an Intelligent Tutoring System for Medical Diagnosis. In: Alevan, V., Kay, J., Mostow, J. (eds.) *ITS 2010. LNCS*, vol. 6095, pp. 51–61. Springer, Heidelberg (2010)
5. Chen, W., Mostow, J., Aist, G.: Exploiting Predictable Response Training to Improve Automatic Recognition of Children’s Spoken Responses. In: Alevan, V., Kay, J., Mostow, J. (eds.) *ITS 2010. LNCS*, vol. 6094, pp. 65–74. Springer, Heidelberg (2010)
6. Kazi, H., Haddawy, P., Suebnukarn, S.: Expanding the Space of Plausible Solutions in a Medical Tutoring System for Problem-Based Learning. *International Journal of Artificial Intelligence in Education* 19(3), 309–334 (2009)
7. Mahmood, A.M., Kuppa, M.R.: A novel pruning approach using expert knowledge for data-specific pruning. *Eng. Comput. (Lond.)* 28(1), 21–30 (2012)
8. Ravi, S., Kim, J., Shaw, E.: Mining On-line Discussions: Assessing, Technical Quality for Student Scaffolding and Classifying Messages for Participation Profiling. In: *Educational Data Mining Workshop. Conference on Artificial Intelligence in Education, Marina del Rey, CA, USA*, pp. 70–79 (July 2007)
9. Bernhard, D., Gurevych, I.: Answering learners’ questions by retrieving question paraphrases from social Q&A sites. In: *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications for the Association for Computational Linguistics, Columbus, Ohio*, pp. 44–52 (2008)
10. Dragon, T., Park Woolf, B., Murray, T.: Intelligent Coaching for Collaboration in Ill-Defined Domains. In: *Conference of Artificial Intelligence in Education, Brighton, England*, pp. 740–742 (2009)

Coordinating Multi-dimensional Support in Collaborative Conversational Agents

David Adamson and Carolyn Penstein Rosé

Language Technologies Institute
Carnegie Mellon University
dadamson@cs.cmu.edu, cprose@cs.cmu.edu

Abstract. The field of computer supported collaborative learning has evolved an ontology of types of support for group learning. In recent years, conversational agents have been used successfully to realize forms of dynamic micro and macro level script based support for group learning. However, using existing architectures for managing the coordination of these agent-based behaviors (which can vary widely in scope, timing, and constraints), infelicitous “collision” of behaviors have been observed. In this paper, we introduce a new architecture that facilitates the development, coordination, and co-performance of multiple agent-based support behaviors.

Keywords: collaborative learning, intelligent agents, multi-party conversational agents, conversational scripting, dynamic support.

1 Introduction

This paper describes a new architecture for intelligent support of collaborative learning, motivated by recent work in dynamic scripting. A *script* in CSCL is a method for structuring collaboration [1]. A script can provide structure at a macro-level, or it can scaffold a participant’s contributions at a micro-level. Such scripts can be implemented statically, providing the same support in all cases, or dynamically, responding to the students and their context to deliver an appropriate level of support at opportune times.

The *Basilica* agent architecture [7] pioneered dynamic collaborative support alongside traditional static macro- and micro-scripts. Agents were defined as a collection of modular components, any of which could influence the agents’ user-facing behavior. Despite Basilica’s design innovations, it left plenty of room for improvement in the realm of authoring and coordinating agent behavior.

The contribution of this paper is an illustration of the design space of multi-dimensional support for collaborative learning as enabled through *Bazaar*, a successor architecture to Basilica, designed to simplify the coordination of multiple dynamic supportive behaviors. In Section 3, we describe Bazaar in detail. In particular, Section 3.2 describes a feature of this architecture that allows for the graceful resolution of conflict between proposed system actions. Finally, Section 4 showcases a number of agents that were developed with this architecture, and locates them within the space of multi-dimensional support.

2 Collaborative Scripting and Support

A script can describe any of a wide range of features of collaborative activities, including task, timing, roles, and the patterns of interaction between the participants. A number of models have been proposed to aid the design and analysis of collaborative scripts [5] [6] [10]. Scripts can be classified as either macro-scripts or micro-scripts [2]. Macro-scripts are pedagogical models that describe coarse-grained features of a collaborative setting, such as the sequence and structure of an activity. Micro-scripts, in contrast, are models of dialogue and argumentation that are embedded in the environment, and are intended to be adopted and progressively internalized by the participants. Examples of macro-scripts include the classic Jigsaw activity, as well as specialized scripts like ArgueGraph and ConceptGrid [5]. Micro-scripting can be implemented by offering prompts or hints to the user to guide their contributions [9], which may depend on the current phase of the macro-script.

Early approaches to scripting have been static, offering the same script or supports for every group in every context. Such non-adaptive approaches can lead to over scripting [1], or to the interference between different types of scripts [11]. A more dynamic approach that triggered micro-scripted supports or the appropriate phases of macro-scripts in response to the automatic analysis of participant activity would be preferable. Such analysis can occur at a macro-discourse level, following the state of the activity as a whole, or it can be based on isolated user events. Such dynamic awareness might allow minimal scripting to be used to greater effect, with greater hopes of the users internalizing the support's intended interaction patterns. Further, the benefits of fading the support over time [9] could be more fully realized, as the timing and degree of such fading could be dynamically tuned to the group's level of internalization. The collaborative tutoring agents described by Kumar [7] were among the first to implement dynamic scripting in a CACL environment, and were quite successful at increasing both learning and the quality of collaborative behavior in groups.

Table 1. Sample of Agent Self-Collision

Student	1:03	I think it has to do with the flow through the membrane.
Tutor	1:05	That's interesting, Student - can you say more about permeability?
Tutor	1:06	Let's move on to the next problem.
Student	1:09	What about my answer? :-)

2.1 Coordinated Multi-dimensional Support

Participants in a collaborative session aren't just completing the assigned task. They're involved in numerous simultaneous processes including social bonding, idea formation, argumentation, and time management. To allow for rich, holistic interactive support, a tutor must be able to express several differently-scoped behaviors concurrently - it can be considered to be working through several

overlapping macro- and micro-scripts at once. However, the tutor has to remain effective while doing so. As illustrated in Table 1, a tutor managing several scripts at once can “step on its own toes”. When multiple responses from the tutor interfere with, or interrupt each other, the students’ belief in the tutor’s competence can be shattered. Although several approaches have been described to address some of these concerns [7], it remains an actively-pursued grail [8].

3 The Bazaar Architecture

The *Bazaar* architecture builds upon Basilica [7], a modular framework for designing multi-party collaborative agents. Both are event-driven systems where independent components receive and respond to user-, environment-, and system-generated actions, and present the unified output of these components to the user. We adapt the Basilica architecture to accommodate competing sources of agent behavior, to streamline agent development. Both architectures are able to interact with the same varied set of collaborative environments, which include text chatrooms and shared whiteboards, as well as more novel environments like the virtual world of SecondLife.

3.1 Events and Components

A Bazaar *Event*, like its Basilica counterpart, is an object representing something interesting that has happened in the world of the agent. An Event might represent an incoming student message or a user entering a chat room. Events can also result from the analysis of other events, or changes in system state. Events such as these are used to launch phases of macro-scripts, or to dynamically initiate suitable support behavior. A *Component* is a modular representation of related behavior and state-knowledge, and often corresponds to a single method of scripting or support. Basilica components were arranged in an agent-specific graph of relationships, frequently defining a custom event for each inter-component connection. This led to an undesirable degree of coupling, especially among components that sought to mediate or suppress the behavior of their neighbors. Bazaar replaces the web of components with a two-step event flow, dividing component responsibility between *Preprocessor* and *Reactor* interfaces. When a new event is received by the system, all Preprocessors registered for the event’s type are given the opportunity to respond to it, either generating new events (perhaps to indicate a shift in the conversation’s focus) or modifying the original (like adding a conceptual annotation to a user message). All preprocessed events are subsequently delivered to the Reactors registered for the resulting event types. Reactors have the opportunity to respond to events (and thus dynamically enact sub-scripts or supports) by proposing actions to the *Output Coordinator*. Figure 1 illustrates a typical Bazaar configuration.

3.2 Output Coordinator: Prioritizing Proposed Actions

Proposals for agent actions are queued in the Output Coordinator with a time-window of relevance and an initial priority value assigned by the originating

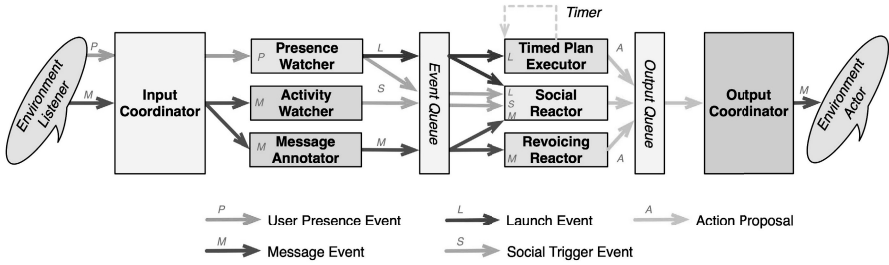


Fig. 1. A sample Bazaar configuration

Reactor. The Output Coordinator will periodically re-evaluate the priority of each remaining proposal, rejecting those that are no longer relevant and accepting and enacting the one with the highest priority.

As a solution to the multi-source management problem described in Section 2.1, we employ a generalization of the “concurrent mode” approach described by Lison [8]. A previously-accepted agent action can leave a lingering presence in the Output Coordinator, a *Proposal Source*, which can re-prioritize (or entirely suppress) incoming proposals until its influence expires. Each action proposal is constructed with a timeout-window after which it is no longer relevant - if a queued proposal has not been accepted when its timeout expires, it’s removed from the queue. When a message is accepted or rejected, a callback-method is invoked, allowing the originating Component to update its state accordingly.

4 Case Studies in Multi-dimensional Support

The tutors in the following case studies highlight the capabilities of the Bazaar architecture, notably the coordination of multiple sources of behavior and support. Table 2 illustrates their dimensions (macro or micro, static or dynamic, as discussed in Section 2) along which each system offered scripting or support. The first two were developed in-house, and have been used in recent studies. The third is one of a set of conversational agents developed by small teams of undergraduate students as part of a two-week CSCL workshop at IIIT Delhi.

Dynamic Feedback: Revoicing in Chemistry and Biology. How does tutor revoicing affect the quality of student explanations? In a college chemistry lesson on intermolecular forces, we deployed an agent that matched student input

Table 2. Dimensions of Support in Bazaar Agents

Support	Revoicing	CycleTalk	Devil & Guardian
Static Macro	X	X	
Dynamic Macro	X		X
Dynamic Micro	X	X	X

against a list of target concepts, and offered the matched concept as a rephrasing of their contribution. In addition, the tutor followed a macro-script to deliver the problem sets and background material that framed the discussions, and also employed dynamic macro-level social strategies as first implemented by Kumar [7]. The revoicing and social behaviors operated in tandem - higher-priority revoicing responses softly blocked any social prompts that were triggered until several seconds after the revoicing move had completed. The macro-script's timing was similarly softened - where previous Basilica tutors would drop everything and interrupt themselves for a macro-level timeout, in this tutor a prompt for the next macro-phase would be delayed long enough for the current move-sequence to play out. The same arrangement of behavioral components has since been re-deployed in a high-school biology domain [3]. Only the lesson's macro-script and the targeted-concept list for the revoicing behavior had to be modified. This study, showed a significant effect from the dynamic revoicing behavior on the quality of student discussion and explanation.

Multiple Agent Scripts: CycleTalk. How can we manipulate the self-efficacy of group members? This Bazaar tutor employed two chat-room user presences to present both an authoritative and non-authoritative face to the human users. The "Doctor Bob" presence delivered the macro-scripted lesson content, while dynamic social prompts and additional scripted questions were posed to a targeted student in each group by "Jimmy", portrayed as a clueless student. Results from this study indicate that this sort of targeting may be detrimental to students groups with low-self-efficacy [4].

Dynamic Macro-Scripting: Devil and Guardian. Will a balanced debate lead to greater mutual understanding? "Devil and Guardian" employed a topic-classification model to classify the recent history of a conversation (as a rolling window over past participant turns) by topic and by "side" (i.e., a Gun Control discussion dominated by Conservatives), and used this classification to select and insert talking-points and images on the current topic, supporting the under-represented opposing side. In addition, the rate of per-user contributions was monitored, dynamically triggering events to encourage participation by the less vocal user. This agent was has not yet been used in a study.

5 Conclusions and Future Work

Bazaar is a powerful tool for facilitating research in collaborative learning. Its flexibility and simplicity mean it can be used to very rapidly develop platforms for investigating a wide range of important questions within the design space of dynamic support for collaborative learning. We have developed a number of such research platforms, and actively employ them in our learning studies. As we continue to do so, we expect to discover ways in which the Bazaar architecture can be extended and refined. We look forward to sharing Bazaar with other researchers exploring dynamic supports for collaboration, and to continue to improve the architecture and make it accessible to this target audience.

Acknowledgments. This work was funded by NSF Grants DUE-1022958, EEC-0935145 and SBE-0836012.

References

- [1] Dillenbourg, P.: Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In: *Three Worlds of CSCL Can We Support CSCL*, pp. 61–91 (2002)
- [2] Dillenbourg, P., Hong, F.: The mechanics of CSCL macro scripts. *The International Journal of Computer-Supported Collaborative Learning* 3(1), 5–23 (2008)
- [3] Dyke, G., Howley, I., Adamson, D., Rosé, C.P.: Towards Academically Productive Talk Supported by Conversational Agents. In: *Intelligent Tutoring Systems* (in press, 2012)
- [4] Howley, I., Adamson, D., Dyke, G., Mayfield, E., Beuth, J., Rosé, C.P.: Group Composition and Intelligent Dialogue Tutors for Impacting Students Academic Self-Efficacy. *Intelligent Tutoring Systems* (in press, 2012)
- [5] Kobbe, L., Weinberger, A., Dillenbourg, P., Harrer, A., Hämäläinen, R., Häkkinen, P., Fischer, F.: Specifying computer-supported collaboration scripts. *The International Journal of Computer-Supported Collaborative Learning* 2(2-3), 211–224 (2007)
- [6] Kollar, I., Fischer, F., Hesse, F.W.: Collaborative scripts - a conceptual analysis. *Educational Psychology Review* 18(2), 159–185 (2006)
- [7] Kumar, R., Rosé, C.P.: Architecture for Building Conversational Agents that Support Collaborative Learning. *IEEE Transactions on Learning Technologies* 4(1), 1 (2011)
- [8] Lison, P.: Multi-Policy Dialogue Management. In: *Proceedings of the SIGDIAL 2011 Conference*, pp. 294–300. Association for Computational Linguistics (2011)
- [9] Wecker, C., Fischer, F.: Fading scripts in computer-supported collaborative learning: The role of distributed monitoring. In: *Proceedings of the 8th International Conference*, pp. 764–772 (2007)
- [10] Weinberger, A., Fischer, F.: A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education* 46(1), 71–95 (2006)
- [11] Weinberger, A., Stegmann, K., Fischer, F., Mandl, H.: Scripting argumentative knowledge construction in computer-supported learning environments. *Environments* 6(6), 191–211 (2007)

Textual Complexity and Discourse Structure in Computer-Supported Collaborative Learning

Stefan Trausan-Matu¹, Mihai Dascalu¹, and Philippe Dessus²

¹ University Politehnica of Bucharest, 313 Splaiul Independetei, Bucharest, Romania
² Grenoble University, 1251, Av. Centrale, BP 47, F-38040 Grenoble CEDEX 9, France
{stefan.trausan,mihai.dascalu}@cs.pub.ro,
philippe.dessus@upmf-grenoble.fr

Abstract. Computer-Supported Collaborative Learning (CSCL) technologies play an increasing role simultaneously with the appearance of the Social Web. The polyphonic analysis method based on Bakhtin's dialogical model reflects the multi-voiced nature of a CSCL conversation and the related learning processes. We propose the extension of the model and the previous applications of the polyphonic method to both collaborative CSCL chats and individual metacognitive essays performed by the same learners. The model allows a tight correlation between collaboration and textual complexity, all integrated in an implemented system, which uses Natural Language Processing techniques.

Keywords: Computer-Supported Collaborative Learning, metacognition, polyphonic model, dialogism, knowledge building, textual complexity, NLP.

1 Introduction

In recent years, Computer-Supported Collaborative Learning (CSCL) grew as an alternate solution to Intelligent Tutoring Systems (ITS) in supporting learning with computers. One of the explanations is the huge spreading of collaborative tools on the web, empowering social knowledge building: discussion forums, instant messenger (chat), social networks, and wikis. The transition from ITS to CSCL may be seen as a change of focus from learning as knowledge acquisition to learning as discourse building [1] or, from a higher abstraction level, from a cognitive to a socio-cultural paradigm. A theoretical basis for CSCL is Bakhtin's dialogism, multi-voicedness and polyphony [2, 3, 4]. We further consider that these concepts are present not only in any CSCL dialogical text (e.g., forum posts or chat utterances), but also in texts written by students, in manuals read by them and even in their inner thinking and they can be used for analyzing complex assignments [4].

We propose a model and a system based on the polyphony idea, which considers both the semantic content (at the individual level related to an expert standard, like in ITSs) and the social dimension (at a collaborative level, in CSCL) by analyzing the relationships between texts in a corpus (of the considered domain), texts collaboratively written by students in CSCL chat sessions and their individual metacognitive essays written afterwards, commenting their collaborative activity. To

achieve this aim we used Natural Language Processing (NLP) techniques enabling the computation of both *distances* between voices and the overall *complexity* of threads.

2 Theoretical Considerations

Let us consider students engaged in a distance learning situation (e.g., through an Internet-based platform). Typically, their main goal is to build knowledge through two lines of activities [3], *individual* (read texts, write out notes, essays, summaries from course material) and *collective* (discussions about the course material), which can both be supported either by a teacher or computer-based feedback. All the stakeholders (the computer included) performing these activities ‘say something’ in natural language, in other words, emit ‘utterances’ [5] that may become ‘voices’ populating the distance learning platform, responding to each other. The way a student can, upon a given question (from herself or others), gather information from multiple textual sources (either from course material or chat utterances) in order to compose her own piece of text (mainly, summaries or syntheses) might be viewed as “contexts” in which they try to handle the polyphony of voices.

This framework allows us assume that each utterance can be analyzed by some NLP or Social Network Analysis (SNA) techniques, thus leading to the production of (semi-) automated support of learners’ activities [6]. The achievement of the aim of supporting learning with computers should start from a model of how people learn. The development of any model usually begins with deciding the main ingredients to be considered as essential. The core model of ITSs was influenced by Knowledge-Based Systems, taking knowledge as major ingredient. The ITS model is centered on a knowledge base of the target domain, which may be seen as a model of what should be learned. Learners are modelled by the knowledge they acquired, either correct, usually a subset of the domain knowledge base, or erroneous, to be corrected (sometimes also described in knowledge bases). Some other types of knowledge about the particular learner may be considered, as her cognitive profile, emotional state, goals or other motivational facts.

We keep the ITS idea that students’ knowledge should be compared with a ‘gold standard’: experts’ knowledge. However, for comparing students’ performance (content of chat utterances and written essays) with the desired one (content of a corpus of reference texts), we are using NLP techniques like *Tf-Idf* or Latent Semantic Analysis (LSA) [7]. We consider that a deficiency of the ITS model is its relation to the transfer of knowledge model of learning, that learning is in a very important degree also socially built [1, 3]. Therefore, in addition to keeping an ITS-type semantic based content analysis, a CSCL-like analysis is also needed, because dialog, conversation, and multi-voiced discourse in natural language have major roles: “rather than speaking about ‘acquisition of knowledge,’ many people prefer to view learning as becoming a participant in a certain discourse” [1].

We further assume that dialogism, multi-vocality and polyphony [2] are in any text, conversation and even thinking. The ‘glue’ of all these is the idea of voice in a generalized sense: as a word, a phrase, an utterance (written or thought), a discussion thread, a lexical chain, or even a whole text (‘utterance’ may be used for words with ‘echoes’, phrases and texts, as Bakhtin mentioned [5]). In our view, an utterance may become a voice if it has an impact by its emission to the subsequent utterances.

3 The Implemented Model

We implemented, using NLP tools, an evaluation model of learners' utterances derived from Bakhtin's dialogic, polyphonic model. The entire analysis process is centered on the utterance graph automatically built from the discourse and is customized for two different types of assessed text: multi-participant chat conversations, on one hand, and essays (texts in general), on the other. Utterances may be considered pieces of text whose boundaries are represented by the change of speech subject [5] and are the central unit of analysis of the discourse in our approach. Whereas in chat conversation we adopt Dong's [8] perspective of separating utterances based on turn-taking events between speakers, in texts, in general, utterances are embedded within sentences that convey relevant information, units that can be separately and independently analyzed in the first phase of the evaluation.

We start the processing with a typical NLP pipe (spell-checking, elimination of stop-words, stemming, part-of-speech tagging and lexicalized dependency parsing [9]). We seek a shallow perspective over each utterance seen individually and we provide them a quantitative mark by merging the concept of entropy from information theory with the *Tf-Idf* measure [9]. The combination of *disorder and emphasis on diversity of concepts* induced by the entropy of stems after stop words elimination, with *summing up statistical importance* of stems given a training corpus, provides a good surface indicator of the information withheld in each utterance (Eq. 1):

$$quant(u) = \left(- \sum_i p(stem_i) \log(p(stem_i)) \right) \left(\sum_i (1 + |stem_i \in u|) \left(\frac{|D|}{|stem_i \in D|} \right) \right) \quad (1)$$

where: $p(stem_i)$ expresses the probability of a stem to occur in a given utterance; $|stem_i \in u|$ denotes the number of occurrences of each stem within the utterance; $|D|$ and $|stem_i \in D|$ are related to the training corpus used also with LSA that comprises a multitude of documents closely related to the topics at hand and a general set of documents for common words. In this context, entropy is used rather as an inhibitor, where low quality or spam utterances have a lower score.

The key-step is using the Directed Acyclic Graph (DAG) of utterances reflecting the sequential ordering. Our aim is to determine the semantic cohesion between two utterances by means of similarity and degree of inter-connection. Similarity between utterances can be expressed by combining *repetitions* of stems and *Jaccard similarity* as measures of lexical cohesion, with semantic similarity computed by means of LSA. Therefore, Eq. 2 covers the general approach of measuring *cohesion*:

$$coh(u, v) = |repetitions| \times \frac{|stems \text{ in common } u, v|}{|stems \text{ in } u \text{ or } v|} \times \cos(\text{vector}(u), \text{vector}(v))$$

$$\text{vector}(u) = \sum_i (1 + |word_i \in u|) \times \left(\frac{|D|}{|word_i \in D|} \right) \times U_k[word_i] \quad (2)$$

where $U_k[word_i]$ is the vector of $word_i$ in the U_k matrix obtained after SVD decomposition and projection over k most meaningful dimensions are performed. As a result, for a given conversation the DAG in Figure 1 is obtained automatically.

The next step in our analysis consists in determining the importance of each utterance within the discourse and two additional dimensions, besides quantitative

evaluation, are considered: a *qualitative* one centered on relevance, impact and coherence, and a *social perspective*, seen as an augmentation factor.

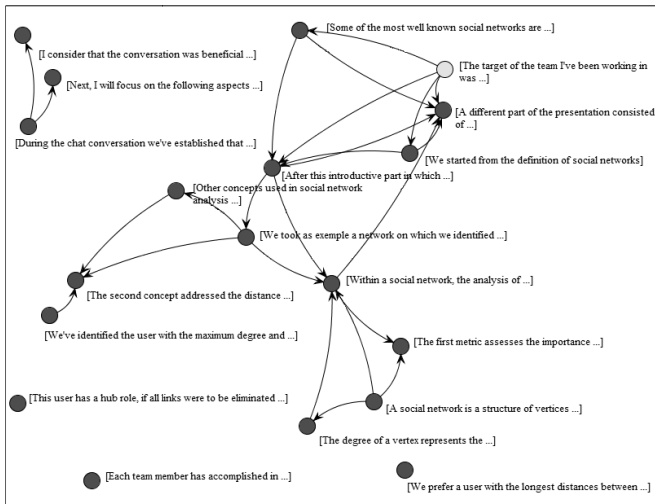


Fig. 1. Example of an utterance graph build upon a student's metacognition (texts are in Romanian)

Relevance is determined with regards to the entire discourse (practically the semantic coherence function applied between a specific utterance and the whole document) and to the vector space by means of cosine similarity between the utterance vector and the mean vector of the LSA vector space. *Completeness* is an optional factor and expresses the semantic similarity between a set of topics (manually defined by the tutor or automatically determined by the system) and the given utterance.

Thread cohesion and *future impact* express the impact of previous and of future utterances that are inter-connected to the current utterance in the utterance graph. These factors are obtained by summing up semantic cohesion values exceeding a threshold; after multiple experiments, the best empirical value of this threshold turned out to be the average minus standard deviation of all the edges of the utterance graph.

From the dialogic perspective, the *current utterance* is seen as a sum of overlapping voices materialized as concepts that are highlighted through information retrieval techniques. *Future impact* encapsulates all future echoes of the current utterance based on the coherence function that expresses both the voice, in the sense of concept repetition, and attenuation, simulated through semantic similarity. Meanwhile, *thread cohesion* acts as a memory function by referring to information previously stated and provides overall discourse coherence as a local function perpetuated through the entire discourse.

From a completely different point of view, the *social perspective* consists in applying social network analysis specific algorithms for estimating an utterances importance in the utterance graph. These metrics include degree, closeness centrality, distance centrality, eigenvector centrality, betweenness centrality and an adjusted version of the well-known Page Rank algorithm [10]. By combining all previous factors, the qualitative factor of each utterance can be expressed as follows (Eq. 3):

$$\begin{aligned} \text{relevance}(u) = & \cos(\text{vector}(u), \text{vector}(\text{doc})) + \cos(\text{vector}(u), \text{LSA vector mean}) \\ & + \cos(\text{vector}(u), \text{vector}(\text{topics})) \end{aligned}$$

$$\text{social}(u) = \prod_{\text{SNA factor } f} (1 + \log(f(u))) \quad (3)$$

$$\text{qualitative}(u) = \left(\sum_{\substack{i=1..m \\ v_i \rightarrow u}} \text{coh}(v_i, u) + 1 + \sum_{\substack{k=1..n \\ u \rightarrow v_k}} \text{coh}(u, v_k) \right) \times \text{relevance}(u) \times \text{social}(u)$$

Regarding the social factor, a normalization induced by the logarithm function provided a smoothing of results. The factor 1 in the coherence values sum expresses internal strength in a discussion thread and was induced by the cosine similarity measure applied between utterance u and itself. By combining the quantitative mark with the qualitative score, the overall rating of each utterance is obtained (Eq. 4):

$$\begin{aligned} \text{overall}(u) = & \left(\sum_{\substack{i=1..m \\ v_i \rightarrow u}} \text{coh}(v_i, u) \text{quant}(v_i) + \text{quant}(u) + \sum_{\substack{k=1..n \\ u \rightarrow v_k}} \text{coh}(u, v_k) \text{quant}(v_k) \right) \\ & \times \text{relevance}(u) \times \text{social}(u) \end{aligned} \quad (4)$$

Eq. 4 clearly comprises all factors required for thoroughly evaluating an utterance: its local and individual formula, its importance within all discourse threads measured through semantic cohesion with previous and future inter-connected utterances, its relevance expressed in terms of semantic similarity with the entire document, topics of discussion and the LSA learning space, but also social networks analysis applied on the utterance graph in order to integrate centrality features in our approach.

After having all previous assessments completed, *textual complexity* can be evaluated and gains the focus of the entire analysis. Due to the fact that textual complexity cannot be determined by enforcing a single factor of evaluation, we propose a multitude of factors, categorized in a multilayered pyramid, from the simplest to the more complex ones, that combined provide relevant information to the tutor regarding the actual “hardness” of a text [11]. The first and simplest factors are at a *surface level* and include readability metrics, utterance entropy at stem level and proxies extracted from Page’s [12] automatic essay grading technique. Slotnick’s six factors [13] of fluency, spelling, diction, sentence structure, punctuation and paragraph development are the main factors we implemented in our system.

At the *syntax level*, structural complexity is estimated from the parsing tree in terms of max depth and of max width [14]. Moreover, entropy applied on parts of speech and the actual number of specific parts of speech (mostly pronouns, verbs and nouns) provide additional information at this level. *Semantics* is addressed through topics that are determined by combining *Tf-Idf* with cosine similarity between the utterance vector and that of the entire documents. The textual complexity at this level is expressed as a weighted mean of the difficulty of each topic, estimated in computations as the number of syllables of each word. The last level of *pragmatics* and *discourse* addresses textual complexity as cohesion determined upon social networks analysis metrics applied at macroscopic level. Discourse markers, co-references, rhetorical schemas and argumentation structures are also considered, but are not included in current work.

By considering the disparate facets of textual complexity and by proposing possible automatic methods of evaluation, the resulted measurement vectors provide tutors valuable information regarding the hardness of presented texts.

4 Conclusions and Future Research Directions

Borrowing from Bakhtin's dialogism and polyphony theories, we devised a framework that takes into account several dimensions of learners' activities in CSCL. Reading course materials, understanding them, discussing about them produce utterances seen as polyphonic voices interacting to each other. Our model automatically assesses these utterances at multiple levels (cognitive, metacognitive, social), and accounts for learner's comprehension of textual materials.

Acknowledgement. This research was partially supported by project 264207, ERRIC-Empowering Romanian Research on Intelligent Information Technologies/FP7-REGPOT-2010-1.

References

1. Sfard, A.: On reform movement and the limits of mathematical discourse. *Mathematical Thinking and Learning* 2(3), 157–189 (2000)
2. Bakhtin, M.M.: *Problems of Dostoevsky's poetics*. University of Minnesota Press, Minneapolis (1993)
3. Stahl, G.: *Group cognition*. MIT Press, Cambridge (2006)
4. Trausan-Matu, S., Stahl, G., Sarmiento, J.: Supporting polyphonic collaborative learning. *E-service Journal* 6(1), 58–74 (2007)
5. Bakhtin, M.M.: *Speech genres and other late essays*. University of Texas, Austin (1986)
6. Dessus, P., Trausan-Matu, S.: Implementing Bakhtin's dialogism theory with NLP techniques in distance learning environments. In: Trausan-Matu, S., Dessus, P. (eds.) *Proc. 2nd Workshop on Natural Language Processing in Support of Learning: Metrics, Feedback and Connectivity (NLPsL 2010)*, pp. 11–20. Matrix Rom, Bucharest (2010)
7. Landauer, T.K., Dumais, S.T.: A solution to Plato's problem: the Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychol. Rev.* 104(2), 211–240 (1997)
8. Dong, A.: *The language of design: Theory and computation*. Springer, New York (2009)
9. Jurafsky, D., Martin, J.H.: *An introduction to natural language processing. Computational linguistics, and speech recognition*. Pearson Prentice Hall, London (2009)
10. Nguyen, Q.H., Hong, S.-H.: Comparison of centrality-based planarisation for 2.5D graph drawing. NICTA technical report, Sidney (2006)
11. Dascalu, M., Trausan-Matu, S., Dessus, P.: Utterances assessment in chat conversations. *Research in Computing Science* 46, 323–334 (2010)
12. Page, E.: The imminence of grading essays by computer. *Phi Delta Kappan* 47, 238–243 (1966)
13. Wresch, W.: The imminence of grading essays by computer—25 years later. *Computers and Composition* 10(2), 45–58 (1993)
14. Gervasi, V., Ambriola, V.: Quantitative assessment of textual complexity. In: Merlini Barbaresi, L. (ed.) *Complexity in Language and Text*, pp. 197–228. Plus, Pisa (2002)

Using Information Extraction to Generate Trigger Questions for Academic Writing Support

Ming Liu and Rafael A. Calvo

University of Sydney, Sydney NSW 2006, Australia

Abstract. Automated question generation approaches have been proposed to support reading comprehension. However, these approaches are not suitable for supporting writing activities. We present a novel approach to generate different forms of trigger questions (directive and facilitative) aimed at supporting deep learning. Useful semantic information from Wikipedia articles is extracted and linked to the key phrases in a students' literature review, particularly focusing on extracting information containing 3 types of relations (Kind of, Similar-to and Different-to) by using syntactic pattern matching rules. We collected literature reviews from 23 Engineering research students, and evaluated the quality of 306 computer generated questions and 115 generic questions. Facilitative questions are more useful when it comes to deep learning about the topic, while directive questions are clearer and useful for improving the composition.

Keywords: Information Extraction, Question Generation, Academic Writing Support.

1 Introduction

The purpose of academic writing is to document new knowledge, generally including a review of what is currently known about a given topic [1]. This is the particular focus of a literature review genre, a common activity in advanced undergraduate and postgraduate courses, and necessary for all research students. Afolabi [2] identified some of the most common problems that students have when writing a literature review including *not being sufficiently critical, lacking synthesis and not discriminating between relevant and irrelevant materials*. Helping students with these issues is difficult and time consuming, a significant problem in research methods courses. Automated and semi-automated feedback approaches are being developed to ease the burden.

One common form of feedback is questioning the writer about issues in the composition. This is considered an effective method for promoting critical thinking, yet not much is known about how human instructors generate questions or what type of questions are most effective. In order to find out how the human supervisors generate such specific trigger question, we conducted a large study [3] on an Engineering Research method course and analyzed 125 trigger questions generated by 25 human supervisors for supporting their research students' literature review writing.

In that study, we identified important concept types such as *Research Field*, *System*, *Technology* and *Technical Term*, which the questions generated from. The aim of the current study is to automatically generate two types of questions (Directive and Facilitative) from these important concept types. Q1 and Q2 in Example 1 were computer generated to ask the student writer to critically analyze the difference between the *Technology concept* Principal Component Analysis (PCA) and factor analysis in relation to the writing while Q1 In Example 2 to critically compare PCA with other types of true eigenvector-based multivariate analyses. Q2 in Example 2 triggers reflection on the limitations of the PCA.

Example 1

Q1: Have you discussed the differences between PCA and factor analysis in relation to your project? If not, please consider doing so. (Directive)

Q2: What do you think of the differences between PCA and factor analysis in relation to your project? (Facilitative)

Example 2

Q1: Have you compared the advantages and disadvantages of PCA to other types of the true eigenvector-based multivariate analyses in relation to your project? If not, please consider doing so. (Directive)

Q2: One limitation of principal component analysis is that the results of PCA depend on the scaling of the variables. How do you address these issues in your project? (Facilitative)

Another intention of this research is to explore how useful the directive and facilitative strategies shown in the examples above are. Black and William [4] defined directive feedback as that which tells the student what needs to be fixed or revised while facilitative feedback provides comments and suggestions to help guide students in their own revision and conceptualization. Ellis [5] found that teachers chose directive and facilitative strategies at different times to accomplish different purposes. For example, facilitative feedback may help students improve overall essay organization and coherence while directive feedback may help them to address spelling and grammatical errors or improve sentence structures. However, it is still unknown the impact of these two strategies on our question templates. In this study, we evaluated both directive and facilitative questions generated by the system.

The remainder of the paper is organized as follows: section 2 provides a brief review of the literature focusing on question generation and information extraction. Section 3 describes the linguistic patterns developed. Section 4 briefly describes the question generation process while section 5 details the evaluation and results.

2 Related Work

One of the first automatic QG systems proposed for supporting novices to learn English was AUTOQUEST [6]. This approach is based on simple pattern matching rules to transform the declarative sentence into a question. For example, the pattern S1 (cause) + so that (conjunction) + S2 (effect) can be used to generate why question.

E.g. sentence: Jamie had an afternoon nap **so that** he wouldn't fall asleep at the concert later. Question: Why did Jamie have an afternoon nap? Other systems that support reading and language learning include Kunichika et al. [7] who proposed a question generation method based on both syntactic and semantic information (Space, Time and Agent) so that it can generate more question types (Where, When and Who). More recently, Mostow and Chen [8] proposed an approach to generate questions based on a situation model. It can generate what, how and why questions. E.g. what did <character> <verb>? why/how did <character> <verb> <complement>? Although these approaches are useful for reading comprehension task, it is not suitable for writing support since it is not useful to ask an author questions about what they just wrote, especially then expecting the answer to be that contained in the same document. Our solution is to extract knowledge from Wikipedia that discuss concepts described in the student's writing.

Typically, information extraction can be used to identifying name entities (e.g. authors and books), and relationships between name entities (e.g. authorX write bookY). Most of work focused on supervised methods which identified the name entities and extract their relations [9]. However, these approaches required a manually annotated corpus, which is very time-consuming and laborious. Semi-supervised and unsupervised approaches depend on seeds patterns or examples of specific types of relations, which is learned by using regular expressions [10]. The comparative expressions in English are divisive and complex. Frizeman et al. [11] concentrated on extracting the comparative relation between two drugs based on a shared attribute in the medical domain.

In this study, the information extraction task focuses on extracting other entities that have comparative (similar-to and different-to) and hierarchical relations (kind-of) with a key concept in the student's composition.

3 Linguistic Patterns Generation

Our training set contains frequent comparative patterns identified by Frizeman [11] and 52 sentences (one of the three relations), extracted from 20 Wikipedia articles (one for each composition). After observing common linguistic patterns from our training set, we developed 26 Tregex rules including 5 for kind-of relation, 10 for different-to and 11 for similar-to. The reasons for using Tregex [12] are that it can identify target syntactic elements (e.g. main verbs or subject complement) which matched predefined cue phrases. If they are matched, we extract matched noun phrases (NP) as entities. For example, the sentence A extends B is identified as a *kind-of* type by detecting its main verb which matches the cue phrase '*extend*'. Then, the matched A (NP) in the Subject as Entity1 and B (NP) in the Object as Entity2 can be extracted.

3.1 Interpreting Kind-of Patterns

For *kind-of* sentences the frequent linguistic pattern is denoted as *the subject complement in the form of a possessive case*. Table 1 illustrates the frequent pattern, where the noun phrase (NP) possessed by the Entity2 (NP) in the Subject Complement matches the cue phrase, such as *kind*. Entity1 is the matched NP in the

Subject while Entity2 is the processor of a possessive. These linguistic patterns indicate necessary linguistic units. {BE} means some form of *be*, such as *is*, *am* and *belongs to* while slash indicates disjunction. {Prep} means the preposition word, such as *to* and *of*. From the example in Table 2, we extracted *feature extraction* as Entity1 and *dimensionality reduction* as Entity2 with *kind-of* relation.

Table 1. The frequent pattern in kind-of relation type sentence

Name	The subject complement in the form of a Possessive Case.
Frequent Pattern	Entity1 {BE} kind/family/form/generalization/class/example/extension {Prep} Entity2
Example	In pattern recognition and in image processing, feature extraction is a special form of dimensionality reduction.
Tregex Rule	S < NP=entity1 < (VP << belong is are was << (NP <<- family form generalization example type kind group \$+ (PP << of < (NP <<, NP=entity2a) < NP=entity2b)))
Other patterns	1. S < NP=entity1 < (VP << /generalize extend.?!/ & << (PP < (IN < to) < NP=entity2a) < NP=entity2b & << /generalize extend.?!/)" E.g. Entity1 extends Entity2. 2. NP << (/form type.?!/ . (of > (IN \$+ NP=entity2a))) \$+ (VP [< (NP <<, NP=entity1) < NP=entity1]) E.g. Feature extraction is a special form of dimensionality reduction. 3. S < NP=entity1 < (VP << (NP < (NP << /extension successor simplest.?!/ < (PP < (IN < of) < NP=entity2a))) E.g. SVMs can be interpreted as an extension of the perceptron.

Table 2. The frequent pattern in different relation type sentence

Name	Difference between Entity1 and Entity2
Pattern	The difference/differences/contrast/contrast between Entity1 and Entity2 {BE} that/in the ways clause.
Examples	An important difference between remote procedure calls and local calls is that remote calls can fail because of unpredictable network problems.
Tregex Rule	NP << /difference contrast.?!/ \$+ (PP < (IN < between \$+ (NP < (CC < and \$- /NP NN.?!/=entity2 \$+ /NP NN?!/=entity1))))
Other Patterns	1. VP << (/JJ/ < different dissimilar unrelated \$++ (PP < (IN TO < from to with) [< (NP <<, NP=entity2a) < NP=entity2b])) \$-- NP=entity1 E.g. The false positive rate is different from the familywise error rate. 2. NP=entity1 \$+ (VP << (ADJ ADVP << better easier accurate lower faster \$++ (PP < (IN < than) [< (NP <<, NP=entity2a) < NP=entity2b]))) E.g. SOAP can be considerably slower than CORBA. 3. VP << (/VB.?!/ < /differ.?!/ \$++ (PP < (IN < from) [< (NP <<, NP=entity2a) < NP=entity2b])) \$-- NP=entity1 e.g. E.g. The channel encoding on NTSC-J differs slightly from NTSC-M.

3.2 Interpreting Different-to Patterns

For *different-to* sentences the frequent linguistic pattern in Table 2 is denoted as *difference between Entity1 and Entity2*. The frequent pattern shows the NP, which

precedes a preposition phrase containing *between NP and NP*, matches a possible cue phrase, such as *difference*. The compared two NP as Entities are separated by a conjunction, *and*. The pattern could appear in either Subject or Object of the sentence.

3.3 Interpreting Similar-to Relation

For *similar-to* sentences, the frequent pattern is *the subject complement in the form of {Adjective Phrase} + {Prep} + {NP}* shown in Table 3. The {Adjective Phrase} matches a possible cue phrase, such as *similar*. Entity1 is NP in the Subject while Entity2 is the NP immediately after {Prep}.

Table 3. The frequent pattern in similar relation type sentence

Name	Similar to
Pattern	Entity1 {BE} similar/analogous/equivalent {Prep} Entity2
Examples	As noted earlier, PLSA is similar to LDA.
Tregex Rule	VP << (/JJ/ < parallellanalogousequivalentlcorrespondentlcomparable \$++ (PP < (IN TO < tolwith) [< (NP <<, NP=entity2a) < NP=entity2b])) \$-- NP=entity1
Other Rules and Examples	<ol style="list-style-type: none"> 1. NP << similaritylsimilarities \$+ (PP < (IN < betweenlof \$+ (NP < (CC < and \$- /NP NN?/=entity2a \$+ /NP NN?/=entity1)))) E.g. The similarities of NTSC-M and NTSC-N can be seen... 2. PP < (IN < Likellike) [< (NP <<, NP=entity2a) < NP=entity2b] \$++ NP=entity1 E.g. Like PAL, a SECAM receiver needs a delay line. 3. NP=entity1 \$+ (VP << (ADJP < (/VB?! < relatedllinked \$+ (PP [< (NP <<, NP=entity2a) < NP=entity2b])))) E.g. PCA is closely related to factor analysis.

4 Automatic Question Generation Process

In this section we provide an overview of the multi-stage question generation system. In stage 1, key phrases are extracted by using Lingo algorithm [13]. In stage 2, Wikipedia articles are retrieved by querying these key phrases through Java Wikipedia Library [14]. Each key phrase is then classified based on the content of the retrieved article by using a rule-based approach.

In stage 3, once the key phrase is classified as a valid concept (Research Field, Technology, System or Term) we extract knowledge from the retrieved Wikipedia article. Our previous approach [15] focused on extracting sentences, which have one of following five relations with the concept: *Is-a* (Definition of the Concept) *Has-Limitation* (Drawback of the Concept), *Has-Strength* (Advantage of the Concept), *Apply-to* (Application of the Concept) or *Include-technology* (Methods used in the Concept). In this study, we focus on extracting noun phrase (entities), which have one of three relations (Similar-to, Different-to and Kind-of). Each piece of information extracted can be expressed as a triple denoted as *relation-type (Concept, Sentence/Noun Phrase)*.

In the final stage, each question is generated based on 12 predefined question generation rules. Each rule contains a triple and one question template. In the example above, in stage 1, the PCA is extracted as a key phrase from a student's document. In stage 2, the Principal Component Analysis Wikipedia article is retrieved and the key phrase is classified as a Technology concept. In stage 3, from that Wikipedia article, we extract knowledge in term of triples, such as *Different-to(PCA, factor analysis)*. In stage 4, the questions in Example 1 are generated by matching *Different-to(PCA, factor analysis)* while the Q1 and Q2 in Example 2 are generated by respectively matching the *Kind-of(PCA, The true eigenvector-based multivariate analyses)* and *Has-Limitation(PCA, PCA is sensitive to the relative scaling of the original variables)*.

Table 4. Generic trigger questions chosen from educational learning materials for writing review

Have I critically analysed the literature I use? (Do I follow through a set of concepts and questions, comparing items to each other in the ways they deal with them?)
Could the problem have been approached more effectively from another perspective?
What is your project's contribution to the research field you are working on?
Have I critically analysed the literature I use? (Instead of just listing and summarizing items, do I assess them, discussing strengths and weaknesses?)
Are my literature reviews relevant, appropriate and useful?

5 Evaluation

5.1 Participants and Procedure

The participants consisted of 23 research students at the Faculty of Engineering at The University of Sydney. All participants signed an informed consent form approved by the Human Research Ethics Committee and given a movie voucher as a reward. As part of their degrees participants wrote the 23 literature review papers used in this study. From these literature reviews, 306 questions were generated by the automatic question generation system described in section 3. We also obtained five generic trigger questions from a literature review writing tutorial [16] (see Table 4).

For the evaluation, each question was rated on five quality measures: *QM1: This question is grammatically correct. QM2: This question is clear and not ambiguous. QM3: This question is relevant to my project. QM4: This question helped me to develop a better understanding of important concepts (e.g. research field, technologies, methods, algorithms, models and etc) related to my project. QM5: This question helped me to think more critically or to discriminate between relevant and irrelevant materials when writing the literature review.* The first three quality measures were derived from the Question Generation Shared Task Evaluation Challenge (QGSTE) [17] QM 1, 2 and 3 are about whether a question is understandable. QM 4 assess a question's usefulness for learning important concepts, while QM 5 assess whether the question's usefulness for improving the literature review document.

5.2 Relation Extraction Performance

In the key phrase classification stage, the computer system correctly classified 67 unique key phrases as a concept type (Research Field, System, Technology and Method) in the 23 literature review papers. This is excluding 32 duplicate key phrases (which occurs when, for example, the same key phrase was extracted from two papers) and 13 generic key phrases (algorithm, measurement, etc) as not likely to produce valuable questions in an engineering research project. Here, we report the relation extraction result based on the 197 sentences containing one of three relations (kind-of, similar and different) extracted from the Wikipedia articles linking to the 67 key phrases. Table 5 shows the rule-based relation extractor has reached high precision, but low recall. Anaphors are a major reason since they cause false negatives. The target sentence (*The task is very similar to that of information extraction (IE).*) contains the anaphor (the task), which refers to *Relation Extraction task*. In this sentence, it fails to extract the *similar-to* relation between *Information Extraction* and *Relation Extraction*. Another reason for the lower recall is that some implicit patterns indicate relations. For example, the pattern modifier (noun/adjective) + key phrase (e.g. *User-based collaborative filtering, Cloud Database*) often indicates kind of relation type. In this noun phrase, *user-based collaborative filtering* is a kind of *collaborative filtering*.

Table 5. Relation Extraction Performance

Relation Type	F-score	Precision	Recall
Kind-of	0.630	0.944	0.472
Similar	0.726	0.963	0.586
Different	0.800	0.944	0.694
Average	0.719	0.950	0.584

5.3 Evaluation of Computer Generated and Generic question

Each participant rated the quality of each question (115 generic and 306 computer generated) on a five-point Likert point scale along five quality measures (QM). Higher scores reflect stronger agreement with the quality measure statements; the midpoint, 3, reflects a neutral stance. The average results are displayed in Table 6. The scores indicate that generic questions were perceived to be clearer (QM2) and more grammatically correct (QM1) and relevant (QM3) than computer generated questions. However, ANOVA results showed that these differences were not significant for QM1 ($F(1,419) = 3.911, p > 0.05$), nor for QM2 ($F(1,419) = 0.007, p > 0.05$) and QM3 ($F(1,419) = 0.088, P > 0.05$).

QM4 and QM5 assess the perceived pedagogical usefulness of the questions. The computer generated slightly outscores generic questions in both. ANOVA results indicated that these differences were not statistically significant: $F(1,419) = 8.37, p > 0.05$ for QM4, and $F(1,419) = 0.003, p > 0.05$ for QM5.

However, after filtering out 24 computer generated questions from the 13 generic key phrases described in section 5.2, we found that the remaining 282 computer

generated questions significantly outperformed generic questions in QM3 ($F(1,396)=4.350$, $p < 0.05$). This indicates that the computer generated questions are more useful than generic question in terms of learning important concepts.

Table 6. Evaluation of computer generated and generic questions

Question Producer Quality Measure	Computer	Computer(after filtering some questions)	Generic
QM1:Correctness	4.173	4.177	4.382
QM2:Clarity	3.921	3.968	3.931
QM3:Relevancy	3.578	3.699	3.704
QM4:Useful for learning concepts	3.431	3.543	3.313
QM5:Useful to improve document	3.297	3.390	3.296

5.4 Evaluation of Questions Generated from Different Relation Types

Each question is generated from a triple with one of the relation types described in section 4. Here, we analyzed the 282 computer generated questions from the correctly classified key phrases. Table 7 displays the average score for the questions generated from each relation type under a certain quality measure. ANOVA results showed no significant difference among these relation types in QM1 ($F(7,274) = 1.579$, $p > 0.05$) and QM2 ($F(7,274) = 1.187$, $p > 0.05$). Has-strength, Has-limitation, Has-different and Include-Technology types got relatively higher scores than Apply-to, Has-definition, Kind-of and Similar-to across QM3, QM4 and QM5. A series of ANOVA and Fisher Least Significant Difference (LSD) test results show that Has-strength, Has-limitation, Has-different and Include-Technology significantly outperformed Apply-to and Similar-to in QM 3 and QM4. The Has-strength, Has-limitation, Has-different and Include-Technology question types had more pedagogical value than others.

Table 7. Evaluation of computer generated questions from different relation types

	Apply-to	Definiton	Diff	Include-tech	Kind-of	Limit	Similar	Strength
QM1	3.818	4.189	4.389	4.400	3.972	4.444	4.186	4.500
QM2	3.758	3.989	4.056	4.333	3.694	3.889	4.070	4.500
QM3	3.091	3.874	3.917	4.200	3.639	3.722	3.372	4.333
QM4	2.879	3.379	3.806	4.007	3.556	4.278	3.651	3.833
QM5	2.818	3.411	3.722	3.733	3.278	3.556	3.326	4.000

5.5 Directive versus Facilitative Questions

Facilitative and directive feedback strategies were used in our question templates design. The 282 computer generated questions included 143 facilitative questions and 139 directive questions. Table 8 shows the average scores for the two types of

questions for each quality measure. Facilitative feedback got higher scores than directive in QM3 and QM4 while directive type outscored facilitative type in QM1, QM2 and QM5. This result implied that directive questions are clearer than facilitative questions while facilitative questions are more useful to trigger reflection on important concepts. However, ANOVA results show that their differences are not significant across all quality measures.

Table 8. Evaluation of Two Feedback Types

Quality Measure	Facilitative Type	Directive Type
QM1	4.091	4.266
QM2	3.951	3.986
QM3	3.783	3.612
QM4	3.552	3.532
QM5	3.392	3.888

6 Conclusion and Future Work

Within the numerous types of learning activities for which we would like to provide automated feedback, writing is one of the hardest, amongst other reasons because there is not always a right or wrong answer. Feedback that helps writers reconsider their work is most useful. This is particularly true when it is produced with knowledge not contained in the actual document, like the one an experienced instructor has. In this paper we introduced a novel approach for generating questions by extracting semantic information from Wikipedia, particularly focusing on extracting three types of relations (Similar-to, Different-to and Kind-of). The study shows that computer generated questions are better than generic questions because there are more specific to the content and useful to promote deep thinking. Facilitative questions are more useful when it comes to deep learning (QM3: Relevancy and QM4: Learning Concept) while directive questions are clearer (QM2) and useful for improving writing. However, the automated question generation system is a multi stage pipe line processing. Error could happen at any stage and propagate to following stages so that it would impact the overall quality of generated questions. In the future work, we are looking into a generic question ranking function to improve the system performance.

Acknowledgments. This project was partially supported by a University of Sydney TIES grant, an Australian Research Council Discovery Project grant (DP0986873) and Google Research Award for measuring the impact of feedback on the writing process.

References

1. Graswell, G.: Writing for academic success: a postgraduate guide. SAGE Publications (2008)
2. Afolabi, M.: The review of related literature in research. International Journal of Information and Library Research 4(1), 59–66 (1992)

3. Liu, M., Calvo, R.A.: Question Taxonomy and Implications for Automatic Question Generation. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS, vol. 6738, pp. 504–506. Springer, Heidelberg (2011)
4. Black, P., William, D.: Assessment and classroom learning. *Assessment in Education: Principles, Policy & Practice* 5(1), 7–74 (1998)
5. Ellis, R.: A typology of written corrective feedback types. *ELT Journal* 63(2), 97–107 (2009)
6. Wolfe, J.H.: Automatic question generation from text - an aid to independent study. *SIGCUE Outlook* 10(SI), 104–112 (1976)
7. Kunichika, H., Katayama, T., Hirashima, T., Takeuchi, A.: Automated Question Generation Methods for Intelligent English Learning Systems and its Evaluation. In: International Conference on Computers in Education, pp. 1117–1124 (2002)
8. Mostow, J., Chen, W.: Generating instruction automatically for the reading strategy of self-questioning. In: International Conference on Artificial Intelligence in Education, pp. 465–472. IOS Press, Amsterdam (2009)
9. Soderland, S.: Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning* 34(1-3), 233–272 (1999)
10. Agichtein, E., Gravano, L.: Snowball: Extracting Relations from Large Plain-Text Collections. In: The Fifth ACM International Conference on Digital Libraries (2000)
11. Fiszman, M., Demner-Fushman, D., Lang, F.M., Goetz, P., Rindfleisch, T.C.: Interpreting comparative constructions in biomedical text. In: The Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, pp. 137–144. ACL, Prague (2007)
12. Levy, R., Andrew, G.: Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In: The Fifth International Conference on Language Resources and Evaluation, Genoa, Italy (2006)
13. Osinski, S., Stefanowski, J., Weiss, D.: Lingo: Search results clustering algorithm based on singular value decomposition. In: The International Conference on Intelligent Information Systems, Zakopane, Poland (2004)
14. Zesch, T., Müller, C., Gurevych, I.: Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In: The Sixth International Conference on Language Resources and Evaluation, Marrakech, Morocco (2008)
15. Liu, M., Calvo, R.A., Aditomo, A., Pizzato, L.A.: Using Wikipedia and Conceptual Graph Structures to Generate Questions for Academic Writing Support. *IEEE Transactions on Learning Technologies* (accepted)
16. Taylor, D.: The Literature Review: A Few Tips On Conducting It, <http://www.writing.utoronto.ca/advice/specific-types-of-writing/literature-review>
17. Rus, V., Wyse, B., Piwek, P., Lintean, M., Stoyanchev, S., Moldovan, C.: Overview of the first question generation shared task evaluation challenge. In: The Third Workshop on Question Generation, Pittsburgh, pp. 45–57 (2010)

Learning to Tutor Like a Tutor: Ranking Questions in Context^{*}

Lee Becker¹, Martha Palmer¹, Sarel van Vuuren¹, and Wayne Ward^{1,2}

¹ Center for Computational Language and Education Research
University of Colorado Boulder
Boulder, Colorado, USA 80309
lee.becker@colorado.edu

² Boulder Language Technologies, Boulder, CO 80301

Abstract. Asking questions in a context relevant manner is a critical behavior for intelligent tutoring systems; however even within a single pedagogy there may be numerous valid strategies. This paper explores the use of supervised ranking models to rank candidate questions in the context of tutorial dialogues. By training models on individual and aggregate judgments from experienced tutors, we learn to reproduce individual and average preferences in questioning. Analysis of our models' performance across different tutors highlights differences in individual teaching preferences and illustrates the impact of surface form, semantic and pragmatic features for modeling variations in tutoring styles. This work has implications for dialogue system design and provides a natural starting point towards creating tunable and customizable tutorial dialogue interactions.

Keywords: Question Ranking, Question Selection, Tutorial Dialogue, Dialogue Acts, Dialogue Modeling, Intelligent Tutoring Systems.

1 Introduction

Much of the benefit of conversational one-on-one tutoring comes from the tutor's ability to tailor his or her line of questioning to the needs of the student [10]. Despite the numerous improvements in making dialogue with intelligent tutoring systems (ITS) more robust and responsive, there is still a performance gap between human tutoring and ITS. Closing this gap will require even more human-like dialogue behavior that can adapt tutoring strategies to differences in student abilities and learning styles. For tutoring, much of this behavior hinges on knowing how to ask questions that encourage student uptake and promote understanding of the material. We approach the task of question asking as a process of ranking candidate questions for a given dialogue context. By pairing questions with judgments collected from experienced human tutors, we can train ranking models capable of reproducing individual preferences in question asking.

^{*} This work was supported by grants from the NSF (DRL-0733322, DRL-0733323), the IES (R3053070434) and the DARPA GALE program (HR0011-06-C-0022).

Analysis of model parameters shows the importance of semantically and pragmatically informed dialogue act features in accounting for variation in individual tutoring styles and preferences.

Connections to Prior Work: Learning tutorial dialogue policies from corpora is a growing area of research in NLP and ITS. Existing work has made use of hidden Markov models [4] and reinforcement learning [9,8] to discover tutoring strategies optimized to maximize learning gains; however, much of this work assumes there is only one correct behavior, and the additional complexity required to model individual tutoring styles would require much more data. This work adopts an approach similar to Ai and Litman [1] who utilize ranking to predict human judgments of simulated dialogue quality.

There is also an abundance of previous work in categorizing dialogue acts and questions for tutoring [12,11,21]. Corpora tagged with dialogue and tutoring acts have been used to explore the correlation between tutoring moves and learning [13,17] as well as specific behaviors such as when to ask “why” questions [22], provide hints [24], or insert discourse markers [16]. To our knowledge there has been no previous work in ranking questions for a dialogue context, nor has there been analysis into the role of dialogue act features for learning differences in tutoring style between experienced tutors.

2 Corpus

2.1 Tutorial Setting and Transcripts

Our investigations are grounded within the context of the My Science Tutor (MyST) [25], a conversational virtual tutor designed to improve science learning and understanding for students in grades 3-5 (ages 7-11). Students using MyST investigate and discuss science through natural spoken dialogues and multimedia interactions with a virtual tutor named Marni. The MyST dialogue design and tutoring style is based on a pedagogy called Questioning the Author (QtA) [2] which places an emphasis on eliciting student speech via open-ended questions. MyST curriculum is based on the Full Option Science System (FOSS, <http://www.fossweb.com>) an inquiry-centered curriculum that has been widely deployed in the United States.

For these experiments, we use MyST transcripts collected in a Wizard-of-Oz (WoZ) condition. During a WoZ session, human tutors (wizards) were responsible for accepting, overriding, and/or authoring system actions. Wizards were also responsible for managing which of the learning goals was currently in focus. Students talked to MyST via microphone, while MyST communicates using Text-to-Speech (TTS) in the WoZ setting. A typical MyST session revolves around a single FOSS lesson and lasts approximately 15 minutes. To obtain a dialogue transcript, tutor moves are taken directly from the system logfile, while student speech is manually transcribed from audio. In total we make use of transcripts from 122 WoZ dialogues conducted by 14 different tutors, which cover ten units on magnetism and electricity and two on measurement and standards.

2.2 Dialogue Annotation

To enable extraction of features representative of the underlying actions and intentions of the dialogue, we have annotated our transcripts and questions with the Dialogue Schema Unifying Speech and Semantics (DISCUSS) [3], a multidimensional dialogue move taxonomy that represents an utterance as a tuple composed of three dimensions: *Dialogue Act* (22 tags), *Rhetorical Form* (22 tags), and *Predicate Type* (19 tags). This scheme draws from past work in task-oriented dialogue acts [7,11], tutorial act taxonomies [21,23,6,5], discourse relations [18] and question taxonomies [12,20]. To motivate our use of the DISCUSS representation for question ranking, we give a brief primer on its dimensions below.

Dialogue Act (DA). The DA represents an utterance’s conversational action with moves such as *Ask*, *Answer*, *Assert*, etc. . .). DISCUSS supplements DA acts commonly found in other taxonomies with two acts common to QtA instruction: *Mark* and *Revoice*. A *Mark* act highlights key words from the student’s speech while a *Revoice* summarizes or refines student language to clarify a concept.

Rhetorical Form (RF). Although the DA is useful for identifying the speaker’s intent, it gives no indication of how the speaker is advancing the conversation. Consider the questions “What is the battery doing?” and “Which one is the battery?”. Both have *Ask* as a DA label, however they elicit two very different kinds of responses. The former, elicits a description (RF=*Describe*) while the latter elicits identification of an object (RF=*Identify*).

Predicate Type (PT). The PT aims to summarize the semantic relationships between the entities and keywords in an utterance. For questions this drives towards the kind of content the tutor is eliciting whether it is a *Procedure*, *Function*, *Causal Relation*, *Observation* or some other PT.

Annotation: All transcripts used in this experiment have been annotated with DISCUSS labels at the dialogue turn level. A reliability study using 15% of the transcripts was conducted to assess DISCUSS inter-annotator agreement. This consisted of 18 doubly annotated transcripts comprised of 828 dialogue utterances. Because DISCUSS permits multiple tuples per instance, we bound reliability with two metrics: exact agreement and partial agreement. For exact agreement, each annotators’ set of labels must match exactly to receive credit. Partial agreement is defined as the number of intersecting labels divided by the total number of unique labels. Table 2 lists these metrics broken down by DISCUSS dimension. While the DA and RF agreement are relatively high, the PT agreement reflects the difficulty and open-ended nature of the task.

2.3 Question Authoring

Though a question generation system would provide the most systematic control in varying questions, we instead use manually authored questions to avoid confounding our findings with issues of question grammaticality and well-formedness. To maintain consistency, we used a single author trained in MyST-oriented QtA

Table 1. Example dialogue context snippet and a collection of candidate questions. The DISCUSS (DA=dialogue act, RF=rhetorical form, PT=predicate type) labels illustrate how the questions vary in intent and meaning.

Candidate Question	DISCUSS (DA/RF/PT)
Q1 Roll over the switch and then in your own words, tell me again what a complete or closed circuit is all about.	Direct/Task/Visual Ask/Describe/Configuration
Q2 How is this circuit setup? Is it open or closed?	Ask/Select/Configuration
Q3 To summarize, a closed circuit allows the electricity to flow and the motor to spin. Now in this circuit, we have a new component. The switch. What is the switch all about?	Assert/Recap/Proposition Direct/Task/Visual Ask/Describe/Function
Q4 You said something about the motor spinning in a complete circuit. Tell me more about that.	Revoice/None/None Ask/Elaborate/CausalRel'n

and taught him to vary questions lexically, syntactically, and semantically. While the author was free to write any questions he thought appropriate for the context, our guidelines emphasized authoring by making permutations corresponding to changes in DISCUSS, wording, directness, and learning-goal content. To minimize the risk of rater bias, we advised our author to avoid using positive feedback expressions such as “Good job!”. Table 1 shows an example context along with a set of candidate questions and their corresponding DISCUSS representations.

The author was presented with information comparable to the dialogue contexts available to the human WoZ and computer MyST tutors such as the dialogue history, learning goals, and visuals, and was asked to author 5 candidate questions per context. Question authoring contexts were manually selected for scenarios that require a follow-up question. We also extracted the original question provided by the tutor, and filtered out those that did not contain questions related to the lesson content. Our corpus has 205 question authoring contexts comprised of 1025 manually authored questions and 131 questions extracted from the original transcript yielding 1156 questions in total.

2.4 Ratings Collection

To rate questions, we enlisted the help of four experienced tutors who had previously served as project tutors and wizards. The raters were presented with the same information used for question authoring. The interface included the entire dialogue history preceding the question decision point and a list of up to 6 candidate questions (5 manually authored, 1 taken from the original transcript if applicable). Because rating individual questions in isolation can lead to inconsistent scoring, we instead asked raters to simultaneously score all candidate

Table 2. Inter-annotator agreement for DISCUSS annotation (DA=Dialogue Act, RF=Rhetorical Form, PT=Predicate Type)

Reliability Metric	DA	RF	PT
Exact Agreement	0.80	0.66	0.56
Partial Agreement	0.89	0.77	0.68

Table 3. Rater-rater rank agreement (Kendall's- τ). The bottom row is the self-agreement for contexts rated by the same rater in two separate trials.

	Rater A	Rater B	Rater C	Rater D
Rater A	-	0.2590	0.1418	0.0075
Rater B	0.2590	-	0.1217	0.2370
Rater C	0.1418	0.1217	-	0.0540
Rater D	0.0075	0.2370	0.0540	-
mean	0.1361	0.2059	0.1058	0.0995
self	0.4802	0.4022	0.2327	0.3531

questions. While we did not define any specific criteria for rating, we instructed the raters to score questions as if they were conducting the tutoring and to consider which ones they felt most appropriate for this particular point in the dialogue. Scores were collected using an ordinal 10-point scale ranging from 1 (lowest/worst) to 10 (highest/best). Each set of questions was rated by at least three tutors with raters never scoring questions from sessions they had tutored themselves. In total we collected ratings for 1156 questions representing a total of 205 question contexts distributed across 30 transcripts.

Rater Agreement: Because these judgments are subjective, a key challenge in this work centers on understanding to what degree the tutors agree with one another. Since our goal is to rank questions and not score questions, we convert each tutors' scores for a given context into a rank-ordered list. To get a sense of inter-rater agreement for ranking, we use Kendall's- τ rank correlation coefficient [15], a non-parametric statistic that measures the agreement between two orderings of the same set of items. We compute τ between all pairs of raters across all question rating contexts. The mean value for all pairs of raters and contexts is $\tau = 0.1478$, a breakdown of rater-rater τ is shown in table 3. Though Kendall's- τ can vary from -1 to 1, its value is highly task dependent, and it is typically lower when the range of possible choices is narrow, as it is in this task. We can obtain the odds of pairwise agreement using the formula $(1 + \tau)/(1 - \tau)$, which shows that for $\tau = 0.1478$ our raters are 1.34 times more likely to agree on the relative ordering of two questions. While inter-rater agreement is fairly modest, we also see variation based on dependent on the pairs of tutors. This suggests that despite their common training and experience, the raters key in on different criteria when scoring. To get a sense of the tutor's internal agreement in rating, we had each tutor rerate a batch of 60 question sets. Kendall's- τ self-agreement values are listed in the bottom row of table 3. In contrast with the rater-rater agreement, self-agreement is much more consistent, giving further evidence for a difference in rating criteria. Together these inter-rater and self-agreement help bound expected system performance in ranking.

3 Automatic Question Ranking

We approach the problem of question selection as a supervised machine learning ranking task. The gold-standard rank-orderings used for training and evaluation are derived from the rater scores. When modeling individual raters, an individual rater’s scores are converted directly into rankings. To average the scores from all raters for a general model, we combine the scores from all raters by tabulating pairwise wins for all pairs of questions $q_i, q_j, (i \neq j)$ within a given dialogue context C . If $rating(q_i) > rating(q_j)$, question q_i receives a win. We sum wins across all raters for a given set of questions. The question with the most wins has rank one. Questions with an equal number of wins are considered tied, and are given the average ranking of their ordinal positions.

Using this rank-ordering we then train a pairwise classifier to determine if one question has a better rank than another. For each question q_i within a context C , we construct a vector of features ϕ_i . For a pair of questions q_i and q_j , we then create a new vector using the difference of features: $\Phi(q_i, q_j, C) = \phi_i - \phi_j$. For training, if $rank(q_i) < rank(q_j)$, the classification is positive otherwise it is negative. To account for the possibility of ties, and to make the difference measure appear symmetric, we train both combinations (q_i, q_j) and (q_j, q_i) . During decoding, we run the trained classifier on all pairs and tabulate wins using the approach described above.

Because we are interested in understanding how well our features can account for individual tutoring preferences, the machine learning algorithms employed in this study are limited to those with interpretable feature weights. Specifically, we use a Maximum Entropy classifier [19]. In previous experiments we observed similar performance between Maximum Entropy models and ranking optimized Support Vector Machines [14], consequently we do not sacrifice model performance for this interpretability. To explore the impact of different dialogue features and semantic forms in system performance we build several models by incrementally adding classes of DISCUSS-based features. To assess our models’ ability to replicate question ranking behavior we train and evaluate in the following conditions:

Training / Evaluation	Individual Rankings	Combined Rankings
Individual Rankings	X	
Combined Rankings	X	X

Features. When designing features for this task, we wanted to capture the factors that may play a role in the tutor’s decision making process during question selection. Scorers may consider factors such as the question’s wording, lesson relevance, and contextual relevance, consequently our feature space consists of four categories: surface form features, lexical similarity features, DISCUSS features, and Context probability features. We model learning goal completion as a conditional probability of a DISCUSS act given task progress . Table 4 lists the features used to create ranking models.

Table 4. Model features by category. While most features are real-valued, WH-Word, POS-tag, and DISCUSS features are vectorized as a bag-of-features with 0/1 values.

Feature Class	Features
Surface Form Features	Question Length, Part-of-Speech tags, WH-Words
Lexical Similarity Features	Word and POS Uni/Bigram Overlap between: * Question-Previous Student Turn * Question-Last Tutor Question * Question-Current Learning Goal * Question-max(Other Learning Goal)
DISCUSS Features	Dialogue Act (DA), Rhetorical Form (RF) Predicate Type (PT) RF/PT-matches previous turn
Context Probability Features	$p(DA, RF, PT_{question} DA, RF, PT_{student\ turn})$ $p(DA, RF_{question} DA, RF_{student\ turn})$ $p(PT_{question} PT_{student\ turn})$ $p(DA, RF, PT_{question} \% elements\ filled)$

Evaluation. To evaluate our models’ agreement with the tutor rankings, we employ the same mean Kendall’s- τ statistic used for assessing inter-rater reliability with the coefficients averaged over all sets of questions. Gold-standard rankings for each context are computed using the approaches described above. For model comparison we apply the Wilcoxon-signed rank test to test whether the distribution of taus (i.e. per dialogue context agreement coefficients) between models is statistically significant. We train and evaluate our models using 10-fold cross validation (3 transcripts/fold, ≈ 7 dialogue contexts/transcript, ≈ 6 questions/context). The exact number of contexts depends on the evaluation condition with raters A, B, C, and D each with 148, 155, 151, and 161 contexts respectively. Folds are partitioned by FOSS unit, to ensure the test set comes from an unseen lesson.

4 Results and Discussion

Table 5 lists the Kendall’s- τ rank order agreement for models trained on individual tutors as well as the combined model. Applying the Wilcoxon-signed rank test to the distribution of Kendall’s- τ values (i.e. per dialogue context agreement coefficients) shows a statistically significant improvement ($p < 0.01$, $148 \leq n \leq 161$) between the baseline and top-performing models for all raters. This suggests that features extracted from DISCUSS provides additional information not available in the surface form and lexical similarity features. However, performance is not strictly tied to the number of DISCUSS-based features. Unlike the models trained on average rankings, models trained to replicate an individual rater’s rankings may require only a subset of the total features. For example, the best model for Rater C used only the dialogue act and baseline features, whereas Rater D showed improvement when adding the more complex

Table 5. System Mean Kendall’s τ rank-order agreement scores by model and rater. Model training and evaluation is conducted per rater, or in the case of **All**, a combination of the four raters. The **General Model** row shows agreement between output from a system trained on the combination of raters (the best model in the ‘All’ column) and the gold standard rankings from individual raters. Presence or absence of features is denoted with a ‘+’ or ‘-’. The **Baseline** features consist of the Surface Form and Lexical Similarity features.

Model	Rater A	Rater B	Rater C	Rater D	All
Context+DISCUSS+POS-	0.3374	0.1482	0.1203	0.1433	0.1910
DISCUSS+POS-	0.3324	0.1558	0.1213	0.1272	0.1789
DISCUSS+	0.3056	0.1319	0.1240	0.1072	0.1628
DA+RF+PT+	0.3092	0.1281	0.1236	0.1177	0.1466
DA+RF+	0.2881	0.1363	0.1243	0.1057	0.1303
DA+	0.3022	0.1503	0.1396	0.0903	0.1201
Baseline	0.2783	0.1160	0.0995	0.0797	0.1051
Random Baseline	0.0000	0.0000	0.0000	0.0000	0.0000
General Model	0.2121	0.1451	0.0924	0.0948	0.1910

Table 6. Distribution of the 20 most influential features by coarse category

	Baseline	DA	RF	PT
Rater A	0.163	0.312	0.245	0.281
Rater B	0.557	0.123	0.134	0.187
Rater C	0.275	0.200	0.195	0.330
Rater D	0.581	0.114	0.101	0.204
All	0.374	0.151	0.139	0.336

Table 7. Cosine similarity between rater model feature weights

	Rater A	Rater B	Rater C	Rater D
Rater A	1.000	0.526	0.167	0.163
Rater B	0.526	1.000	0.106	0.250
Rater C	0.167	0.106	1.000	0.184
Rater D	0.163	0.250	0.184	1.000
mean	0.464	0.470	0.364	0.399

contextual features. These differences in performance roughly outline what level of linguistic detail underlies a rater’s preference for one question over another.

Comparing the results from table 5 with the inter and intra-rater agreement values from table 3, we see that we are best able to replicate the rankings for raters who have the highest self-agreement, which suggests that data collection should be improved to limit variation in judgment. One potential way to improve rater reliability would be to back away from having raters simultaneously scoring all questions and instead present them with paired comparisons.

Feature Analysis: To get a qualitative perspective of our models, we asked our lead tutor to give a brief description of each rater’s tutoring style. She offered: ‘Rater A focuses more on the student than the lesson.’, ‘Rater B focuses on the lesson objectives.’, ‘Rater C tries to get the student to relate to what they see or do.’, and ‘Rater D likes to add more to the lesson than what was done in class.’. Looking at the models in light of these comments, we see the feature

weights reflect these differences in tutoring philosophies. Rater A’s model was the only one to give a negative weight to the *Assert* DA feature, which may stem from a desire to elicit speech instead of lecture. Rater B’s emphasis on learning goals manifests itself with larger weights for the lexical overlap features than the other rater models. Rater C’s emphasis on visuals results in PT features weighted towards *Observation* over *Function* or *Process*. Rater D’s desire to create a new experience yields a DA *Metastatement* weight that is twice that found in the other raters’ models. Additionally, rater D had the heaviest weight for the contextual probability features.

Looking at the feature category distributions for the 20 (15%) most influential features (those with the largest weight magnitudes), we observe wide variance in distribution (table 6) from rater to rater. However, differences in feature category distributions does not fully account for differences in rater model behavior. Probing further, we compute cosine similarities between the model weight vectors for each rater’s model (table 7). The similarities in this table mirror inter-rater agreement found in table 3, which gives further evidence that our models rank in a manner like the tutors on which they were trained.

Error Analysis: Cross-referencing rater feedback with analyses of contexts with low system-rater agreement helped to identify three categories of errors: 1) question authoring errors, 2) DISCUSS annotation errors and 3) model deficiency errors. Example question authoring mistakes include referencing an interactive visual when a static one was on-screen, and writing questions which were too wordy or used incorrect terminology. Unlike the raters, our models were unable to key in on these mistakes. While better quality control would help to reduce many of these errors, for future work in fully-automatic question generation, language model or vocabulary features may help to give a better account of a question’s surface form. In instances with incorrect DISCUSS annotation, we found that the correct label would have likely yielded better classification accuracy and consequently ranking agreement. Although we model student learning goal completion as part of the DISCUSS context probability features, a large proportion of errors coincided with rater comments about student understanding and misconceptions. This suggests that additional features that capture student correctness could benefit system performance.

5 Conclusions and Future Work

We have presented an approach for ranking candidate follow-up questions in the context of a tutorial dialogue. Furthermore, we have shown that adding features extracted from a rich, linguistically-motivated tutorial act representation to baseline lexical and surface form features, enables statistical machine learning of question ranking behavior in agreement with experienced human tutors. This framework provides a straightforward means for collecting and using human judgment to customize tutorial dialogue behavior, and this methodology shows that a variety of tutoring styles can be captured by having tutors evaluate other tutors’ sessions. Looking forward, we plan on further refining our

system by adding more detailed accounts of student misconceptions. Lastly, we feel this work is a natural starting point to explore the use of fully-automatic question generation within an intelligent tutoring system.

References

1. Ai, H., Litman, D.: Assessing dialog system user simulation evaluation measures using human judges. In: Proc. of ACL 2008: HLT, Columbus, Ohio, USA (June 2008)
2. Beck, I.L., McKeown, M.G., Worthy, J., Sandora, C.A., Kucan, L.: Questioning the author: A year-long classroom implementation to engage students with text. *The Elementary School Journal* 96(4), 387–416 (1996)
3. Becker, L., Ward, W., van Vuuren, S., Palmer, M.: DISCUSS: A dialogue move taxonomy layered over semantic representations. In: Proc. IWCS 2011, Oxford, England, January 12-14 (2011)
4. Boyer, K., Ha, E., Wallis, M., Phillips, R., Vouk, M., Lester, J.: Discovering tutorial dialogue strategies with hidden markov models. In: Proc. of AIED 2009, Brighton, U.K, pp. 141–148 (2009)
5. Boyer, K., Lahti, W., Phillips, R., Wallis, M.D., Vouk, M.A., Lester, J.C.: An empirically derived question taxonomy for task-oriented tutorial dialogue. In: Proc. of the 2nd WS on Question Generation, Brighton, U.K., pp. 9–16 (2009)
6. Buckley, M., Wolska, M.: A classification of dialogue actions in tutorial dialogue. In: Proc. of COLING 2008, pp. 73–80. ACL (2008)
7. Bunt, H.C.: The DIT++ taxonomy for functional dialogue markup. In: Proc. EDAML 2009 (2009)
8. Chi, M., Jordan, P., VanLehn, K., Hall, M.: Reinforcement learning-based feature selection for developing pedagogically effective tutorial dialogue tactics. In: Proc. EDM 2008, pp. 258–265 (2008)
9. Chi, M., VanLehn, K., Litman, D.: Do Micro-Level Tutorial Decisions Matter: Applying Reinforcement Learning to Induce Pedagogical Tutorial Tactics. In: Alevan, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6094, pp. 224–234. Springer, Heidelberg (2010)
10. Chi, M., Siler, S., Jeong, H., Yamauchi, T., Hausman, R.: Learning from tutoring. *Cognitive Science* 25, 471–533 (2001)
11. Core, M.G., Allen, J.: Coding dialogs with the DAMSL annotation scheme. In: AAAI Fall Symposium, pp. 28–35 (1997)
12. Graesser, A., Person, N.: Question asking during tutoring. *American Educational Research Journal* 31, 104–137 (1994)
13. Jackson, G., Person, N., Graesser, A.: Adaptive tutorial dialogue in autotutor. In: Proc. of WS on Dialog-based Intelligent Tutoring Systems (2004)
14. Joachims, T.: Making large-scale svm learning practical. In: *Advances in Kernel Methods - Support Vector Learning*. MIT Press (1999)
15. Kendall, M.: A new measure of rank correlation. *Biometrika* 30(1-2), 81–89 (1938)
16. Kim, J., Glass, M., Freedman, R., Evens, M.: Learning the use of discourse markers in tutorial dialogue learning the use of discourse markers in tutorial dialogue. In: Proc. of the Cognitive Sciences Society (2000)
17. Litman, D., Forbes-Riley, K.: Correlations between dialogue acts and learning in spoken tutoring dialogue. *Natural Language Engineering* 12(2), 161–176 (2006)

18. Mann, W., Thompson, S.: Rhetorical structure theory: Description and construction of text structures. In: Proc. of the 3rd Int'l WS on Text Generation (1986)
19. McCallum, A.K.: MALLET: A Machine Learning for Language Toolkit (2002), <http://mallet.cs.umass.edu>
20. Nielsen, R.D., Buckingham, J., Knoll, G., Marsh, B., Palen, L.: A taxonomy of questions for question generation. In: Proc. of the WS on the Question Generation Shared Task and Evaluation Challenge (September 2008)
21. Pilkington, R.: Analysing educational discourse: The discount scheme. Tech. Rep. 99/2, Computer Based Learning Unit, University of Leeds (1999)
22. Rose, C., Bhembe, D., Siler, S., Srivastava, R., VanLehn, K.: The role of why questions in effective human tutoring. In: Proc. of AIED 2003 (2003)
23. Tsovaltzi, D., Karagjosova, E.: A view on dialogue move taxonomies for tutorial dialogues. In: Proc. of SIGDIAL 2004, pp. 35–38. ACL (2004)
24. Tsovaltzi, D., Matheson, C.: Formalising hinting in tutorial dialogues. In: EDILOG: 6th WS on the Semantics and Pragmatics of Dialogue, pp. 185–192 (2001)
25. Ward, W., Cole, R., Bolaños, D., Buchenroth-Martin, C., Svirsky, E., van Vuuren, S., Weston, T., Zheng, J., Becker, L.: My Science Tutor: A conversational multimedia virtual tutor for elementary school science. ACM Transactions on Speech and Language Processing (TSLP) 7(4) (August 2011)

Analysis of a Simple Model of Problem Solving Times^{*}

Petr Jarušek and Radek Pelánek

Faculty of Informatics, Masaryk University Brno

Abstract. Our aim is to improve problem selection and recommendation in intelligent tutoring systems by modeling students problem solving times. We describe a simple model which assumes a linear relationship between latent problem solving ability and a logarithm of time to solve a problem. We show that this model is related to models from two different areas: the item response theory and collaborative filtering. Each of these areas provides inspiration for parameter estimation procedure and for possible extensions. The model is already applied in a widely used “Problem solving tutor”; using the data collected by this system we evaluate the model and analyse its parameter values.

Keywords: Problem solving, intelligent tutoring systems, item response theory, collaborative filtering.

1 Introduction

Problem solving is an important part of education in general and of intelligent tutoring systems in particular. To use problem solving activities efficiently, it is important to estimate well their difficulty – easy problems are boring, difficult problems are frustrating (this observation is elaborated by the flow concept [4]).

In intelligent tutoring systems [1,12] problem selection is often done with respect to knowledge concepts – matching students mastery of concepts with concepts required to solve a problem. In some domains, however, there are many problems which are based on the same knowledge concepts, but differ significantly in their difficulty. In this work we focus on these types of domains, specifically on logic puzzles and introductory programming – these problems require little background knowledge, do not have easily identifiable skills, and yet span wide range of difficulty [6].

In this work we focus on predicting students problem solving times based on the data about previous problem solving attempts. To attain clear focus, we consider both students and problems as “black boxes”, i.e., the only information that we use are the problem solving times. For practical application it may be useful to combine this approach with other data about students and problems (e.g., from knowledge tracing models [3]). Nevertheless, even the basic “black box” approach

^{*} This work is supported by GA ČR grant no. P202/10/0334.

is applicable and has an important advantage of being simple and cheap (e.g., compared to knowledge tracing models which require significant expertise).

We describe a model which assumes a linear relation between a problem solving ability and a logarithm of time to solve a problem (i.e., exponential relation between ability and time). We provide connections of the model to two different areas – item response theory and collaborative filtering. Item response theory [2,5] is used mainly in computerized adaptive testing to predict a probability of a correct answer and thus to select a suitable test item. Collaborative filtering [8] is used in recommender systems to predict user ratings of items (e.g., books) and recommend items to buy. Using inspiration from these areas we describe different variants of the model and different methods for parameter estimation.

The model is currently used in a “Problem solving tutor” – a web-based system which recommends students problems of suitable difficulty. The tutor contains more than 20 types of problems from areas of programming, math, and logic puzzles. The system is used in several schools and contains data about more than 5 000 users and 220 000 solved problems. Using this extensive data we evaluate the model and its different variants.

The evaluation shows several interesting results. The data support the basic model assumption of linear relation between ability and a logarithm of time to solve a problem. For predicting future times even a simple baseline predictor provides reasonable results; the model provides only slight improvement in predictions. Nevertheless, it brings several advantages. The model is group invariant and gives a better ordering of problems with respect to difficulty. It also brings additional insight – we can determine not just average difficulty of problems, but also their discrimination and randomness. With an extension of the model we can even determine similarity of individual problems (using just the problem solving times). All these parameters are useful for automatic problem selection in intelligent tutoring systems.

2 Modeling Problem Solving Times

We describe the setting, the basic models and we elaborate on its relation to the item response theory and collaborative filtering.

2.1 The Setting and Simple Models

We assume that we have a set of students S , a set of problems P , and data about problem solving times: t_{sp} is a logarithm of time it took student $s \in S$ to solve a problem $p \in P$ (i.e, t is a matrix with missing values). In this work we do not consider any other information about students and problems except for the problem solving times. We study models for predicting future problem solving times based on the available data. These predictions are denoted \hat{t}_{sp} .

As noted above, we work with a logarithm of time instead of the untransformed time itself. There are several good reasons to do so. At first, problem solving times have a natural “multiplicative” (not “additive”) nature, e.g., if

Alice is a slightly better problem solver than Bob, then we expect her times to be 0.8 of Bob's times (not 20 second smaller than Bob's times). At second, previous research on response times in item response theory successfully used the assumption of log-normal distribution of response times [9,11], analysis of our data also suggests that problem solving times are log-normally distributed. At third, the use of a logarithm of time has both theoretical advantages (e.g., applicability of simple linear models) and pragmatic advantages (e.g., reduction of effect of outliers).

Given our setting, the simplest way to predict problem solving times is to use mean time, i.e., $\hat{t}_{sp} = m_p$, where m_p is the mean of $t_{s'p}$ over students s' who solved the problem p . A straightforward way to improve and personalize this prediction is to take into account the performance of individual students. This leads to the "baseline" model $\hat{t}_{sp} = m_p - \delta_s$, where δ_s is a "mean performance of student s with respect to other solvers", i.e., $\delta_s = (\sum m_p - t_{sp})/n_s$, where n_s is the number of problems solved by the student.

Our basic model, on which we will further elaborate, is an extension of this baseline model. It is a linear model, which combines problem difficulty for average solver (b_p), problem discrimination (a_p) and student's ability (θ_s), i.e., $\hat{t}_{sp} = b_p + a_p\theta_s$. In the following we describe two different ways how to derive and further develop this basic idea.

2.2 Model Inspired by Item Response Theory

The item response theory (IRT) deals with test items with discrete set of answers and models the probability of a correct answer. There has been research on modeling response times in the context of IRT (see e.g., [9]), but in this research time is used only as an additional information (the main focus being on the correctness of response), not on the time itself.

The basic models of IRT assume that probability of correct response depends on one latent ability θ . The most often used model is the three parameter logistic model $P_{a,b,c,\theta} = c + (1 - c)e^{a(\theta - b)} / (1 + e^{a(\theta - b)})$. This model has three parameters (see Fig. 1): b is the basic difficulty of an item, a is the discrimination factor (slope of the curve, how well the item discriminates based on ability), and c is the pseudo-guessing parameter (lower limit of the curve, probability that even a student with very low ability will guess the correct answer).

In our setting, we similarly assume that a problem solving performance depends on one latent problem solving ability θ . We are interested in a "problem response function" $f(\theta)$, which for a given ability θ gives an estimate of a time to solve a problem. More specifically, the function gives a probabilistic density of times.

As a specific model we use the simplest "natural" choice: a normal distribution with the mean linearly dependent on the ability and with constant variance (remember that we work with a logarithm of time, i.e., this model assumes that the untransformed time to solve a problem is exponentially dependent on ability). The model thus has 3 problem parameters with the following intuitive meaning

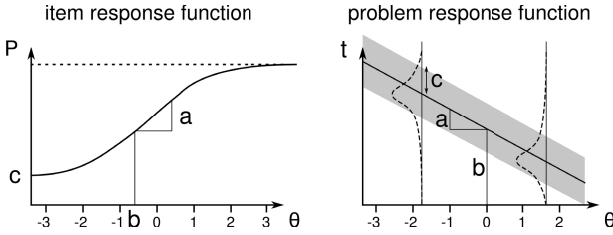


Fig. 1. An intuitive illustration of an item response function and a problem response function. Dashed lines illustrate distributions for certain abilities; solid line denotes the expected problem solving time, grey area depicts the area into which most attempts should fall. Note that we are dealing with a logarithm of time.

(we intentionally use notation analogical to IRT): discrimination factor a_p , basic difficulty of the problem b_p , and randomness factor c_p .

The problem response function, i.e., the probability that a student s with ability θ_s will solve a problem p at (a logarithm of) time t , is thus given by a normal distribution with a mean $b_p + a_p\theta$ and a variance c^2 : $f_{a_p, b_p, c_p, \theta_s}(t) = \mathcal{N}(b_p + a_p\theta_s, c_p)(t)$.

The predicted time for a student s and a problem p is the expected value of $f_{a_p, b_p, c_p, \theta_s}$, i.e., $\hat{t}_{sp} = b_p + a_p\theta_s$. The model and intuition behind its parameters are illustrated in Fig. 1. Discrimination factor a_p describes the slope of the function, i.e., it specifies how the problem distinguishes between students with different ability. Basic difficulty describes expected solving time of a student with average ability. The randomness factor describes variance in solving times for a particular ability.

Note that the presented model is not yet identified as it suffers from the “indeterminacy of the scale” issue in the same way as the basic IRT model. This is solved by normalization – we require that the mean of all θ_s is 0 and the mean of all a_p is -1.

Since we do not know either parameters of problems, or abilities of students, we need to estimate them from available data. Similarly to the procedures used in IRT [5], this estimation can be performed by iterative maximum likelihood estimation. We iteratively update problem (student) parameters assuming that student (problem) parameters are known. Maximum likelihood estimation for parameter values leads to ordinary least squares regression. Maximum likelihood estimation for students abilities gives the following update rule [7] (weighted sum of “local” ability estimates across all problems solved by a student): $\theta_s =$

$$\sum_p \frac{a_p^2}{c_p^2} \frac{t_{sp} - b_p}{a_p} / \sum_p \frac{a_p^2}{c_p^2}.$$

2.3 Models Inspired by Collaborative Filtering

Collaborative filtering is a method used in recommender systems, e.g., systems for recommending movies (Netflix) or books (Amazon). The goal in these cases is to predict future user ratings based on past ratings. Instead of predicting ratings,

we predict problem solving times, but otherwise our situation is analogical (in both cases the input is a large sparse matrix).

There are two basic methods for collaborative filtering: neighbourhood based (memory based) and matrix factorization (model based). The main principle of matrix factorization methods is based on singular value decomposition (SVD) – a linear algebra theorem which states that any matrix A can be decomposed as $A = UDV^T$, where D is a diagonal matrix and U, V are orthonormal matrices. Using this decomposition it is possible to find an approximation of A by using only first few rows in the product.

This theorem can be directly used only for complete matrices. In collaborative filtering, however, there are typically many missing values. This can be overcome by imputing data (e.g., substituting means in place of missing values), but such approach has many disadvantages (e.g., imprecision, computational demands). It is preferable to construct directly an approximation of the form: $\hat{r}_{ij} = \mathbf{p}_i^T \cdot \mathbf{q}_j$, where \hat{r}_{ij} is the predicted rating and \mathbf{p}_i and \mathbf{q}_j are feature vectors of length k . This model is typically further extended to include the baseline prediction for a given item [8]. This leads (using our notation) to the following model: $\hat{t}_{sp} = b_p + \mathbf{a}_p^T \cdot \boldsymbol{\theta}_s$, where \mathbf{a}_p and $\boldsymbol{\theta}_s$ are vectors of length k which specify problem-feature and user-feature interactions. The parameters of the model are typically estimated using stochastic gradient descent with the goal to minimize sum of square errors [8].

Note that for $k = 1$ the resulting model has the same structure as the model inspired by IRT. Both the item response theory model and our analogical model of problem solving times can also be extended to incorporate multidimensional ability [10].

Collaborative filtering has to deal with parameter changes during time (e.g., user book preferences evolve) [8]. Similarly, in our setting it is sensible to incorporate learning into the model – students problem solving ability should improve as they solve more problems. A natural extension of the model is the following: $\hat{t}_{sp} = b_p + a_p(\theta_s + \Delta_s \cdot f(k))$, where k is the order of the problem in problem solving sequence; f is a monotone function, and Δ_s is a student's learning rate.

2.4 Group Invariance

The mean predictor and the simple baseline predictor are misleading if the subgroup of students which solved a particular problem is not representative of the whole population. An important feature of our approach is that the models are “group invariant” (similarly to IRT models [5]), i.e., problem (student) parameters do not depend on a subgroup of students which solved the problem (problems solved by a student).

Let us describe this important feature on a specific example. When we have a set of problems, then typically the harder problems are solved only by students with above average ability. If we use a mean problem solving time as a predictor of problem difficulty, than we underestimate the difficulty of these harder problems. Our model takes abilities of solvers into account and thus the obtained problem parameters are independent of the group of solvers.

3 Application and Evaluation

Now we briefly introduce a “Problem solving tutor”, which uses the described approach to make predictions and recommendations to students. Data collected by this system are used for evaluation of described models.

3.1 Problem Solving Tutor

The described approach is currently used in a “Problem solving tutor” – a free web-based tutoring system for practicing problem solving, which is available at `tutor.fi.muni.cz`. At the moment the system focuses solely on the “outer loop” of intelligent tutoring [12], i.e., recommending problem instances of the right difficulty.

The system contains more than 20 types of problems, particularly computer science problems (e.g., binary numbers, robot programming, turtle graphics, introductory C and Python programming), math problems (e.g., describing functions, matching expressions), and logic puzzles (e.g., Sokoban, Nurikabe). All problems are “pure” problem solving problems with clearly defined correct solution – problem solving time is the single measure of students performance, there are no “quality of solution” measures (i.e., no hints during solutions or acceptance of partial solutions).

The system was launched in March 2011, it is already used by more than 20 schools and has more than 5 000 registered users (mainly university and high school students) who have spent more than 8 000 hours solving more than 220 000 problems. The collected data are used for the below described evaluation. The number of solved problems is distributed unevenly among different problem types, in the evaluation we use only problems for which we have sufficient data.

3.2 Analysis of Parameter Values

We begin our evaluation by analysis of the basic model with one ability. The parameter values were estimated as described in Section 2. We have described two ways to derive our basic model and estimate parameters: model inspired by item response theory with parameters estimated by alternating maximum likelihood estimation and the SVD inspired model with parameters estimated by stochastic gradient descent (the specific algorithm parameters were used as in [8]). Our results show that these two ways to estimate the parameters lead to nearly the same results. Thus here we report only on the computed parameters (student abilities θ , problem parameters a, b, c) of the IRT inspired model.

Student abilities should be normally distributed in a population, and the results show that the estimated abilities θ are really approximately normally distributed (see Fig. 2.). The variance of the distribution depends on the problem type – for educational problems we have larger variance of abilities than for logic puzzles.

Generally the data suggest that the basic assumptions on which the model is based are suitable, e.g., for particular problems the relation between the

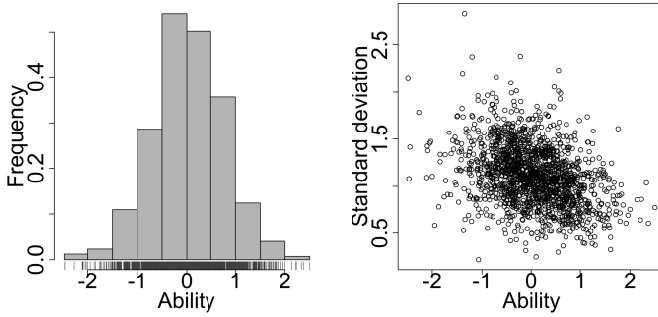


Fig. 2. Left: Distribution of abilities for the Robotanist problem. Right: Ability versus variation in the student performance for the Robotanist problem.

estimated ability θ and a logarithm of time is really linear as the model assumes. Nevertheless, one result shows that some of the model assumptions are too simple. Fig. 2. shows a relation between an estimated ability and a variation in student performance (the standard deviation of ability estimates for individual problem instances). There is a slight negative correlation, i.e., students with lower ability have larger variance, whereas the model assumes a constant variance. Thus the model can be extended by another parameter to describe this decrease of variance with increasing ability.

Fig. 3. shows scatter plots for problem parameters a, b, c . There is a correlation between the basic problem difficulty and its discrimination – more difficult problems are more discriminating. The randomness parameter (which corresponds to variance of problem solving times) is nearly independent of the basic problem difficulty (there is a positive correlation, but only small). Note that this result indirectly supports the application of logarithmic transformation of times. If we had used untransformed times or some different transformation, there would be much stronger dependence.

Although there are some correlations among the parameters, generally the parameters are rather independent, i.e., each of them provides a useful information about the problem difficulty. For example, in intelligent tutoring system, it may be suitable to filter out problems with large randomness or low discrimination.

3.3 Evaluation of Predictions

Evaluation of model predictions was done by repeated random subsample cross-validation. We performed 20 repetitions, each with 90% of data as a training set and the remaining 10% of data as a test set. Table 1. compares the results using the root mean square error metric. We have also evaluated other metrics like the Pearson and Spearman correlation coefficients and mean absolute error. The relative results are very similar.

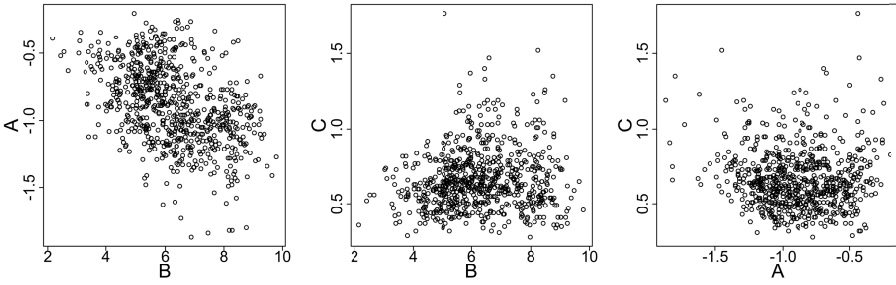


Fig. 3. Relations between parameters a, b, c of the model

Table 1. Quality of predictions for different models and problems measured by root mean square error metric. Baseline model, IRT-model, and SVD-model are models described respectively in Sections 2.1, 2.2, 2.3. All models assume a single latent ability.

Problem type	Mean time	Baseline model	IRT-model	SVD-model
Binary numbers	1.1717	0.9941	0.9856	0.9860
Graphs and functions	1.2868	1.0477	1.0395	1.0419
Nurikabe	0.9021	0.7111	0.7191	0.7175
Robotanist	1.3137	1.2056	1.1944	1.1963
Rush hour	0.8937	0.8072	0.7948	0.7975
Slither Link	1.0252	0.7873	0.7766	0.7760
Sokoban	1.1491	0.8965	0.8876	0.8893
Tents	1.0238	0.9355	0.9423	0.9434
Tilt maze	1.0044	0.8665	0.8620	0.8656

The results show that all models provide improvement over the use of a mean time as a predictor. Most of the improvement in prediction is captured by the baseline model; models with more parameters bring a slight, but not very important improvement. As mentioned above, IRT-based and SVD-based parameter estimations lead to nearly the same parameter values and thus the predictions are also nearly the same.

So far we have evaluated absolute predictions of problem solving time. In practical applications it may be more important to focus on relative predictions, i.e., on ordering of individual problem instances, so that students can progress from easy problems to difficult ones. Here the group invariance issue (described in Section 2.4) becomes important. The baseline model leads to same ordering of problems as the mean time, i.e., it is not group invariant, whereas other described models are group invariant. An analysis of data shows that the ordering based on our models is better than the ordering based on mean time (to make this comparison we ordered problems into sequences P_1, \dots, P_n and counted how many times did some student solve problem P_i faster than problem P_j for $i > j$).

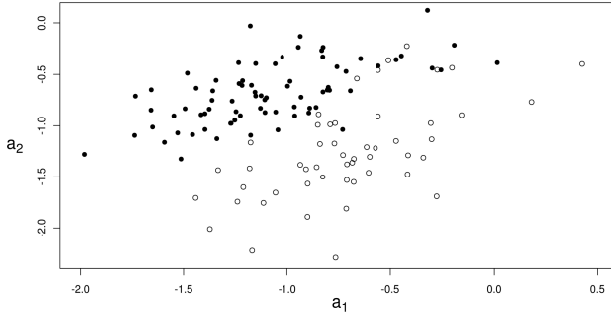


Fig. 4. Determination of problem similarity by the extended model with two abilities. Graph axes are the problem discrimination parameters a_1 , a_2 . White dots are Sokoban problems, black dots are Slither Link problems.

3.4 Extended Models

Finally, we provide a brief evaluation of extended models described in Section 2.3 – a model with learning and a model which assumes multiple abilities (i.e., model corresponding to SVD technique with several features). Parameters for these models were estimated using the stochastic gradient descent method in a similar way as for the basic model.

On our current data these models do not improve predictions due to the overfitting – we get improved fit over the training set, but worse fit on the test set. Nevertheless, even with the current data these models can give us some interesting insight.

The model with learning is of the following form: $\hat{t}_{sp} = b_p + a_p(\theta_s + \Delta_s \cdot f(k))$, where k is the order of the problem in a problem sequence, f is a monotone function, and Δ_s is a learning rate. Our analysis confirms an intuitive expectation that f should be sublinear (learning is faster at the beginning and then slows down); a use of square root leads to a good fit. Results also show that for our problems the learning rate Δ_s is weakly positively correlated with ability θ_s (i.e., better students improve faster).

We also evaluated a model with two abilities: $\hat{t}_{sp} = b_p + a_{1p}\theta_{1s} + a_{2p}\theta_{2s}$. Although the model does not improve predictions on our current data due to the overfitting, we can at least evaluate whether the automatically learnt concepts (abilities) are sensible. To do so we performed the following experiment: we mix data for two types of logic puzzles, let the algorithm learn the concepts, and then check, how well are the puzzles separated. Fig. 4 shows results for two particular problems. As we can see, the two problem types are separated quite well by the automatically learnt concepts. This extended model can thus be used for automatic determination of similarity between problems within a given set of problems. This can be useful for problem recommendation in intelligent tutoring systems. If a student solved a particular problem slowly, we can give her a similar problem, but easier problem, if a student solved problem quickly, we can give her a problem utilizing different concept.

4 Conclusions

We describe a model of students problem solving times, which assumes a linear relationship between a problem solving ability and a logarithm of time. We derive the model details and parameter estimation procedures from two different areas: the item response theory and collaborative filtering. The model is already applied in an online “Problem solving tutor” to recommend problems of suitable difficulty. This system is already widely used (more than 220 000 problems solved), the collected data were used for evaluation of the model. The results show that the model brings only slight improvement compared to the baseline predictor, but also that the model provides interesting information about problems (including determination of problem similarity based only on problem solving times). This information can be useful for problem selection and recommendation in intelligent tutoring systems.

References

1. Anderson, J., Boyle, C., Reiser, B.: Intelligent tutoring systems. *Science* 228(4698), 456–462 (1985)
2. Baker, F.: The basics of item response theory. University of Wisconsin (2001)
3. Corbett, A., Anderson, J.: Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction* 4(4), 253–278 (1994)
4. Csikszentmihalyi, M.: *Flow: The psychology of optimal experience*. HarperPerennial, New York (1991)
5. De Ayala, R.: *The theory and practice of item response theory*. The Guilford Press (2008)
6. Jarušek, P., Pelánek, R.: What determines difficulty of transport puzzles? In: *Proc. of Florida Artificial Intelligence Research Society Conference*, pp. 428–433. AAAI Press (2011)
7. Jarušek, P., Pelánek, R.: Modeling and Predicting Students Problem Solving Times. In: Bieliková, M., Friedrich, G., Gottlob, G., Katzenbeisser, S., Turán, G. (eds.) *SOFSEM 2012. LNCS*, vol. 7147, pp. 637–648. Springer, Heidelberg (2012)
8. Koren, Y., Bell, R.: Advances in collaborative filtering. In: *Recommender Systems Handbook*, pp. 145–186 (2011)
9. Van der Linden, W.: A lognormal model for response times on test items. *Journal of Educational and Behavioral Statistics* 31(2), 181 (2006)
10. Mulder, J., Linden, W.: Multidimensional adaptive testing with kullback–leibler information item selection. *Elements of Adaptive Testing*, 77–101 (2010)
11. Van Der Linden, W.: Conceptual issues in response-time modeling. *Journal of Educational Measurement* 46(3), 247–272 (2009)
12. Vanlehn, K.: The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education* 16(3), 227–265 (2006)

Modelling and Optimizing the Process of Learning Mathematics

Tanja Käser¹, Alberto Giovanni Busetto^{1,2},
Gian-Marco Baschera¹, Juliane Kohn⁴, Karin Kucian³,
Michael von Aster^{3,4,5}, and Markus Gross¹

¹ Department of Computer Science, ETH Zurich, Zurich, Switzerland

² Competence Center for Systems Physiology and Metabolic Diseases,
Zurich, Switzerland

³ MR-Center, University Children's Hospital, Zurich, Switzerland

⁴ Department of Psychology, University of Potsdam, Potsdam, Germany

⁵ Department of Child and Adolescent Psychiatry,
German Red Cross Hospitals Westend, Berlin, Germany

Abstract. This paper introduces a computer-based training program for enhancing numerical cognition aimed at children with developmental dyscalculia. Through modelling cognitive processes and controlling the level of their stimulation, the system optimizes the learning process. Domain knowledge is represented with a dynamic Bayesian network on which the mechanism of automatic control operates. Accumulated knowledge is estimated to select informative tasks and to evaluate student actions. This adaptive training environment equally improves success and motivation. Large-scale experimental data quantifies substantial improvement and validates the advantages of the optimized training.

Keywords: learning, control theory, optimization, dynamic Bayesian network, dyscalculia.

1 Introduction

Computer-assisted learning is gaining importance in children's education. Intelligent tutoring systems are successfully employed in different fields of education, particularly to overcome learning disabilities [1]. The application of computers extends conventional learning therapy. This study presents a computer-based training program for enhancing numerical cognition, aimed at children with developmental dyscalculia (DD) or difficulties in learning mathematics. It entertains the idea that the learning process can be optimized through modelling cognitive development and control.

Motivation. DD is a specific learning disability affecting the acquisition of arithmetic skills. Genetic, neurobiological, and epidemiological evidence indicates that DD is a brain-based disorder with a prevalence of 3-6% [2]. Challenges are subject-dependent and hence individualization is needed to achieve substantial improvements. Computer-based approaches enable the design of adaptable

training, by estimating abilities and by providing intensive training in a stimulating environment. The learner gains self efficacy and success, in turn leading to increased motivation.

Related Work. Previous studies evaluated computer-based trainings for number processing and calculation, documenting promising results [3,4,5]. Available trainings are designed specifically for children with DD, yet provide limited user adaptation. In the domain of mathematics, intelligent tutoring systems focus on specific aspects of the domain [6,7,8]. A plethora of advanced control approaches aimed at optimization of complex mechanisms exists in the literature [9]. As in this study, controllers can be based upon explicit models obtained through intervention-driven identification [10]. Related predictive models aimed at treating learning disabilities have been introduced for spelling learning [1,11].

Contribution. We model the cognitive processes of mathematical development. Recent neuropsychological findings are incorporated into a predictive dynamic Bayesian network. We introduce automatic control aimed at optimizing learning. This model predictive control enables a significant level of cognitive stimulation which is user- and context-adaptive. Results from two large user-studies quantify and validate the improvements induced by training.

2 Training Environment

Current neuropsychological models postulate the existence of task-specific representational modules located in different areas of the brain. The functions of these modules are relevant to both adult cognitive number processing and calculation [12]. Dehaene's triple-code model [13] presumes three representational modules (verbal, symbolic, and analogue magnitude) related to number processing. These modules develop hierarchically over time [14] and the overlap of the number representations increases with growing mathematical understanding [17]. The development of numerical abilities follows a subject-dependent speed which is influenced by the development of other cognitive as well as domain general abilities and biographical aspects [14]. Hence, when teaching mathematics, a substantial degree of individualization may not only be beneficial, but even necessary. The introduced computer-based training addresses these challenges by

1. structuring the curriculum on the basis of natural development of mathematical understanding (hierarchical development of number processing).
2. introducing a highly specific design for numerical stimuli enhancing the different representations and facilitating understanding. The different number representations and their interrelationships form the basis of number understanding and are often perturbed in dyscalculic children [14].
3. training operations and procedures with numbers. Dyscalculic children tend to have difficulties in acquiring simple arithmetic procedures and show a deficit in fact retrieval [15,16].

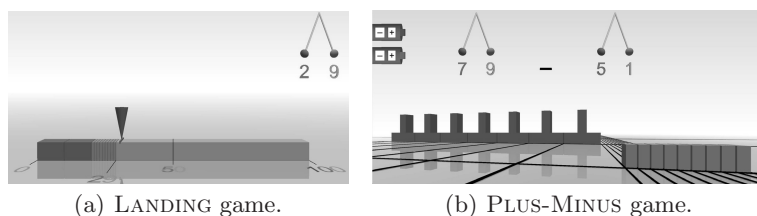


Fig. 1. In the LANDING game, the position of the displayed number (29) needs to be indicated on the number line. In the PLUS-MINUS game, the task displayed needs to be modeled with the blocks of tens and ones.

4. providing a fully adaptive learning environment. Student model and controlling algorithm optimize the learning process by providing an ideal level of cognitive stimulation.

Structure of the Training Program. The training is composed of multiple games in a hierarchical structure. Games are structured according to number ranges and further grouped into two areas. The first area focuses on “number representations and understanding”. It trains the transcoding between alternative representations and introduces the three principles of number understanding: cardinality, ordinality, and relativity. Games in this area are structured according to current neuropsychological models [13,14]. The first area is exemplified by the LANDING game (Fig. 1(a)). The second area is that of “cognitive operations and procedures with numbers”, which aims at training concepts and automation of arithmetical operations. This is illustrated by the PLUS-MINUS game (Fig. 1(b)). Games are divided into main games requiring different abilities and support games training specific ones, serving as basic prerequisites. Difficulty estimation and hierarchy result from the development of mathematical abilities.

Design of the Numerical Stimuli. Properties of numbers are encoded with auditory and visual cues such as color, form, and topology. The digits of a number are attached to the branches of a graph and represented with different colors according to the positions in the place-value system (see left of Fig. 2). Numbers are illustrated as a composition of blocks with different colors, i.e., as an assembly of one, ten and hundred blocks. Blocks are linearly arranged from left to right or directly integrated in the number line (Fig. 2 right). Showing all stimuli simultaneously in each game of the training program reinforces links between different number representations and improves number understanding.

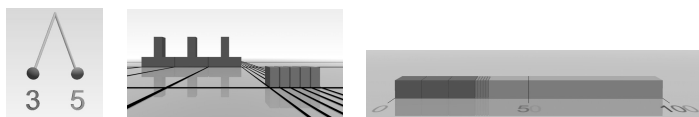


Fig. 2. Design of numerical stimuli for the number 35

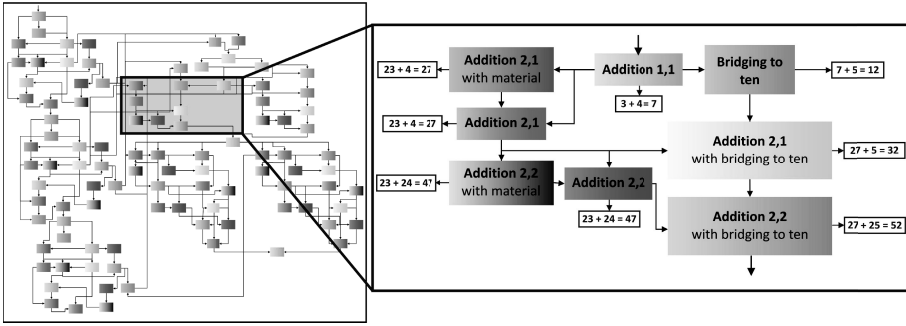


Fig. 3. Skill net containing 100 skills (left), zoom of addition skills from 0-100 (right)

3 Selection of Actions

A fundamental component is the pedagogical module: the subsystem making the teaching decisions. It selects the skills for training and determines the actions. The mechanisms adaptively assess user inputs and dynamically optimize decisions [9]. The learner state is estimated and internally represented by the student model. An attached bug library enables recognition of error patterns.

3.1 Student Model

The mathematical knowledge of the learner is modelled using a dynamic Bayesian network [18]. The network consists of a directed acyclic graphical model representing different mathematical skills and their dependencies. This representation is ideal for modelling mathematical knowledge as the learning domain exhibits a distinctively hierarchical structure. The resulting student model contains 100 different skills (Fig. 3). The structure of the net was designed using experts' advice and incorporates domain knowledge [13,14,15,16]. Two skills s_A and s_B have a (directed) connection, if mastering skill s_A is a prerequisite for skill s_B . The belief of a skill s_{Ai} (probability that skill is in the learnt state) is conditioned over its parents π_i :

$$p(s_{A1}, \dots, s_{An}) = \prod_i p_{s_{Ai}} \text{ where } p_{s_{Ai}} := p(s_{Ai}|\pi_i) \tag{1}$$

As the skills cannot be directly observed, the system infers them by posing tasks and evaluating user actions. Such observations (E) indicate the presence of a skill probabilistically. The posteriors $p_{s_{Ai}|E_k}$ of the net are updated after each solved task k using the sum-product algorithm (libDAI [19]). Initially, the probabilities are initialized to 0.5 (principle of indifference). The dynamic Bayesian net has a memory of 5, i.e. posteriors are calculated over the last five time steps.

3.2 Controller

The selection of actions is rule-based and non-linear. Rather than following a specified sequence to the goal, learning paths are adapted individually. This

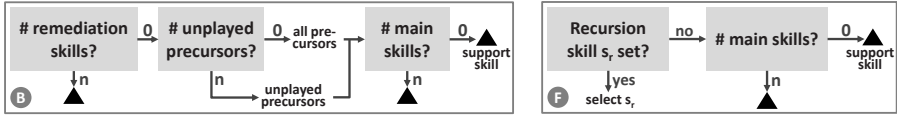


Fig. 4. Decision trees for 'Go Back' (left) and 'Go Forward' (right) options. At the end nodes (triangles), the candidate skill with lowest posterior probability ('Go Back' option)/with posterior probability closest to 0.5 ('Go Forward' option) is selected.

increases the set of possible actions (due to multiple precursors and successors). The controller selects one of the following options based on the current state:

1. **Stay:** Continue the training of the current skill;
2. **Go back:** Train a precursor skill;
3. **Go forward:** Train a successor skill;

The decision is based on the posterior probabilities delivered by the student model. After each solved task, the controller fetches the posterior probability $p_{s|E}(t)$ of the skill s being trained at time t . Then, $p_{s|E}(t)$ is compared against a lower and an upper threshold, denoted by $p_s^l(t)$ and $p_s^u(t)$. The resulting interval defines the optimal training level: if the probability lies between the thresholds, 'Stay' is selected. In contrast, 'Go Back' and 'Go forward' are selected when $p_{s|E}(t) < p_s^l(t)$ and when $p_{s|E}(t) > p_s^u(t)$. Thresholds are not fixed: they converge with more played samples (n_c):

$$p_s^l(t) = p_s^{l0}(t) \cdot l_c^{n_c} \quad \text{and} \quad p_s^u(t) = p_s^{u0}(t) \cdot u_c^{n_c} \tag{2}$$

Initial values of the upper ($p_s^{l0}(t)$) and lower ($p_s^{u0}(t)$) thresholds as well as the change rates (l_c, u_c) are heuristically determined. The convergence of the thresholds ensures a sufficiently large number of solved tasks per skill and prevents training the same skill for too long without passing it.

When 'Stay' is selected, a new appropriate task is built. Otherwise, a precursor (or successor) skill is selected by fetching all precursor (successor) skills of the current skill and feeding them into a decision tree. Figure 4 shows the simplified decision trees for 'Go Back' and 'Go Forward'. The nodes of the trees encode selection rules. If errors matching patterns of the bug library are detected, the relevant remediation skill is trained. If a user fails to master skill s_A and goes back to s_B , s_A is set as a recursion skill. After passing s_B , the controller will return to s_A . To consolidate less sophisticated skills and increase variability, selective recalls are used.

This control design exhibits the following advantages:

1. *Adaptability:* the network path targets the needs of the individual user (Fig. 5).
2. *Memory modelling:* forgetting and knowledge gaps are addressed by going back.
3. *Locality:* the controller acts upon current nodes and neighbours, avoiding unreliable estimates of far nodes.
4. *Generality:* the controller is student model-independent: it can be used on arbitrary discrete structures.



Fig. 5. Skill sequences of three children in addition. Colours are consistent with Fig. 3. User 2 and 3 passed all skills in the range, while user 1 did not pass this range within the training period. The length of the rectangles indicates the number of samples.

4 Methods and Results

Quality of controller and student model have been measured through external effectiveness tests. Experimental data consist of input logs of two on-going large-scale studies (Germany and Switzerland). The studies are conducted using a cross-over design, i.e. participants are divided into a group starting the training immediately and a waiting group. The groups are mapped according to age (2.-5. grade of elementary school), intelligence and gender. All participants visit normal public schools and are German-speaking. They exhibit difficulties in learning mathematics indicated by a below-average performance in arithmetic (addition T-score: 35.4 [SD 7.1], subtraction T-score 35.4 [SD 7.9]) [22]. Participants trained for a period of 6 weeks with a frequency of 5 times per week, during sessions of 20 minutes. Due to technical challenges, a subset of 33 logfiles were completely and correctly recorded. On average, each user completed 29.84 (SD 2.87, min 24, max 36.96) sessions. The total number of solved tasks is 1562 (SD 281.53, min 1011, max 2179), while the number of solved tasks per session corresponds to 52.37 (SD 7.9, min 37.8, max 68.1).

4.1 Logfile Analyses

The analyses of the input data show that the participants improved over time. They provide evidence that the introduced control mechanism significantly speeds up the learning process and that it rapidly adapts to the individual user.

Key skills. To facilitate the analysis of the log files, the concept of 'key skills' is introduced. Key skills are defined in terms of subject-dependent difficulty, they are the hardest skills for the user to pass. More formally,

Definition 1. A skill s_A is a **key skill** for a user U , that is $s_A \in \mathcal{K}_U$, if the user went back to a precursor skill s_B at least once before passing s_A .

From this follows that the set of key skills \mathcal{K}_U may be different for each user U (and it typically is). In the sequence in Fig. 5, user 2 has no key skills, while user 3 has one key skill (coloured in green) and user 1 has several key skills.

Adaptability of controller. During the study, all participants started the training at the lowest (easiest) skill of the net. The adaptation time $[t_0, t_{\mathcal{K}_U}]$ is defined as the period between the start t_0 of the training and the first time the

user hits one of his key skills $t_{\mathcal{K}_U}$. On average, the participants reached their $t_{\mathcal{K}_U}$ after solving 144.3 tasks (SD 113.2, min 10, max 459). The number of complete sessions played up to this point is 1.95 (SD 1.63, min 0.08, max 6.48). These results show, that the model rapidly adjusts to the state of knowledge of the user.

Improvement analysis. To quantify improvement, the learning rate over \mathcal{K}_U is measured from all available samples (both if the participant mastered them during training or not). The improvement over time $I([t_{\mathcal{K}_U}, t_{\text{end}}])$ is computed using a non-linear mixed effect model (NLME) [20] employing one group per user and key skill:

$$y_i \sim \text{Binomial}(1, p_i) \text{ with } p_i = \frac{1}{1 + e^{-(b_0 + b_1 \cdot x_i + u_i)}} \text{ and } u_i \sim \mathcal{N}(0, \sigma^2) \quad (3)$$

where u_i denotes the noise term, x_i the normalized sample indices ($x_i \in [0, 1]$) and y_i the sample correctness. The resulting model (Fig. 6) exhibits an estimated mean improvement of 22.6% (95% confidence interval = [0.21 0.24]).

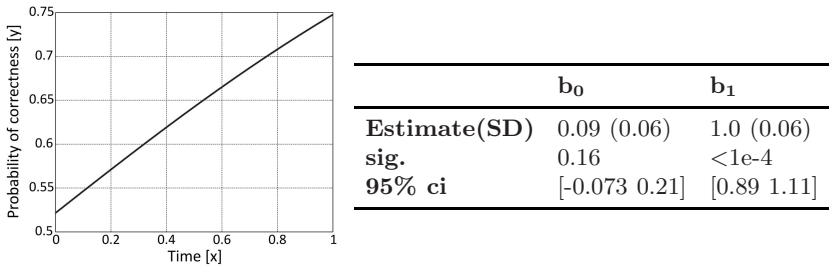


Fig. 6. The percentage of correctly solved tasks (of key skills) increases over the training period by 22.6% (left side). Exact coefficients of NLME along with standard deviation (in brackets) are plotted by respective significance (sig.) and confidence intervals (ci).

Further analysis demonstrates that the possibility to go back to easier (played or unplayed) skills yields a substantially beneficial effect. The user not only immediately starts reducing the rate of mistakes, but also learns faster. The log files recorded 533 individual cases of going back. All cases in which users play a certain skill (samples x_b), go back to one or several easier skills, and finally pass them to come back to the current skill (samples x_a) are incorporated. Per each case k the correct rate over time $c_{a,k}$ ($c_{b,k}$) is estimated separately for x_a and x_b . Fitting is performed via logistic regression using bootstrap aggregation [21] with resampling ($B = 200$). The direct improvement d_k is the difference between the initial correct rate $c_{a,k}$ (at $x_a = 0$) and the achieved correct rate $c_{b,k}$ (at $x_b = 1$). The improvement in learning rate r_k is the difference in learning rate over $c_{a,k}$ and $c_{b,k}$. The distributions over \bar{d} (mean over d_k) and \bar{r} (mean over r_k) are well approximated by a normal distribution (Fig. 7). Both measurements are positive on average and a two-sided t-test indicates their statistically significant difference from zero (Tab. 1).

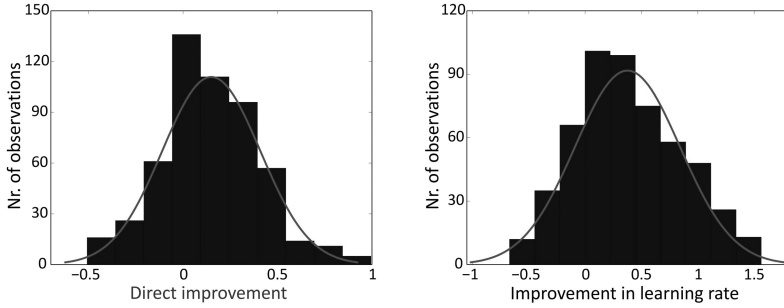


Fig. 7. Distributions over direct improvement \bar{d} and improvement in learning rate \bar{r}

Table 1. Statistics for the improvement after going back: Mean improvement μ , significance of mean (sig.), standard deviation (SD), and confidence intervals (ci)

	Mean μ	sig.	99% ci of μ	SD σ	99% ci of σ
\bar{d}	0.1494	<1e-6	[0.1204 0.1784]	0.2593	[0.2403 0.2814]
\bar{r}	0.3758	<1e-6	[0.3236 0.4280]	0.4662	[0.4319 0.5059]

4.2 Training Effects

Training effects were measured using external paper-pencil and computer tests. The **HRT** [22] is a paper-pencil test. Children are provided with a list of addition (subtraction) tasks ordered by difficulty. The goal is to solve as many tasks as possible within a time frame of 2 minutes. The **AC** (arithmetic test) exists in a paper-pencil and a computer-based version. Children solve addition (and subtraction) tasks ordered by difficulty. Tasks are presented serially in a time frame of 10 minutes.

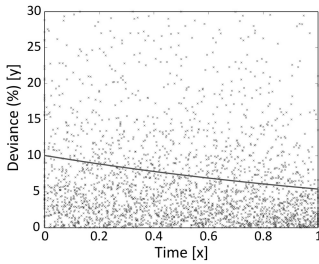
Analyses are done by comparing the effects of the training period (T_c) with those of the waiting period (W_c). First results stem from 33 subjects (26 females, 7 males) in the training condition and 32 subjects (23 females, 9 males) in the waiting condition. The training induced a significant improvement in subtraction (HRT and AC), while no improvement was found after the waiting period (Tab. 2). Pre-tests showed no significant difference between the groups.

The improvement is supported by additional evidence: the percentage of training time children spent with subtraction tasks. In fact, 62% (73% if considering key skills only) of arithmetical tasks consist of subtractions. The focus on subtraction and the significant improvement coming with it is scientifically interesting as performance in subtraction is considered the main indicator for numerical understanding [12]. Consistently with this, improved number line representation is directly measurable from the input data. Over time, children achieved greater accuracy when giving the position of a number on a number line (Fig. 8). The analysis of the accuracy is performed using a NLME model:

$$y_i \sim \text{Poisson}(\lambda_i) \text{ with } \lambda_i = e^{b_0 + b_1 \cdot x_i + u_i} \text{ and } u_i \sim \mathcal{N}(0, \sigma^2) \tag{4}$$

Table 2. Comparison of test improvement between training and waiting condition. The last column shows the results of a t-test on the improvements assuming same variance and different variances, respectively.

	Cond.	Pre-Score(SD)	Post-Score(SD)	sig.	Comparison
HRT	T_c	12.9 (5.38)	16.7 (5.3)	1.5e-8	2.6e-5 (2.9e-5)
	W_c	14.84 (6.47)	15.06 (5.87)	0.72	
AC	T_c	50.53 (27.25)	60.63 (26.3)	4.5e-4	1.9e-3 (2.0e-3)
	W_c	55.18 (25.24)	52.9 (27.74)	0.42	



	b_0	b_1
Estimate(SD)	2.3 (0.07)	-0.63 (0.02)
sig.	<1e-4	<1e-4
95% ci	[2.17 2.44]	[-0.67 -0.58]

Fig. 8. Landing accuracy in the range 0-100 increases over time (left). Exact coefficients of NLME along with standard deviation (in brackets) are plotted by respective significance (sig.) and confidence intervals (ci).

where u_i denotes the noise term, x_i the normalized sample indices ($x_i \in [0, 1]$) and y_i the deviance. Fitting is performed using one group per user.

5 Conclusion

This study introduces a model of the cognitive processes of mathematical development based on current neuropsychological findings. Experimental results demonstrate that domain knowledge is well represented by dynamic Bayesian networks. The predictive model enables the optimization of the learning process through controlled cognitive stimulation. Regression analysis highlights sustained improvement; in particular, the possibility to go back significantly (and rapidly) reduces the error rate and yields an overall increased learning rate. Results are validated by large-scale input data analysis as well as external measures of effectiveness. The student model has the potential to be further refined by incorporating available experimental data.

Acknowledgments. We thank B. Solenthaler for helpful suggestions. The work was funded by the CTI-grant 11006.1 and the BMBF-grant 01GJ1011.

References

1. Baschera, G.-M., Gross, M.: Poisson-Based Inference for Perturbation Models in Adaptive Spelling Training. Int. J. of Artificial Intelligence in Education 20 (2010)

2. Shalev, R., von Aster, M.G.: Identification, classification, and prevalence of developmental dyscalculia. In: *Enc. of Language and Literacy Development*, pp. 1–9 (2008)
3. Wilson, A.J., Dehaene, S., Pinel, P., Revkin, S.K., Cohen, L., Cohen, D.: Principles underlying the design of "The Number Race", an adaptive computer game for remediation of dyscalculia. *Behavioral and Brain Functions* 2(19) (2006)
4. Butterworth, B., Varma, S., Laurillard, D.: *Dyscalculia: From Brain to Education*. *Science* 332, 1049 (2011)
5. Kucian, K., Grond, U., Rotzer, S., Henzi, B., Schönmann, C., Plangger, F., Gälli, M., Martin, E., von Aster, M.: Mental Number Line Training in Children with Developmental Dyscalculia. *NeuroImage* 57(3), 782–795 (2011)
6. Koedinger, K.R., Anderson, J.R., Hadley, W.H., Mark, M.A.: Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education* 8(1), 30–43 (1997)
7. Mislevy, R.J., Almond, R.G., Yan, D., Steinberg, L.S.: Bayes nets in educational assessment: Where do the numbers come from? In: *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pp. 437–446 (1999)
8. Rau, M.A., Alevan, V., Rummel, N.: Intelligent tutoring systems with multiple representations and self-explanation prompts support learning of fractions. In: *Proc. of the 14th Int. Conf. on Artificial Intelligence in Education*, pp. 441–448 (2009)
9. García, C.E., Prett, D.M., Morari, M.: Model predictive control: theory and practice. *Automatica* 25(3), 335–348 (1989)
10. Busetto, A.G., Buhmann, J.M.: Structure Identification by Optimized Interventions. In: *Journal of Machine Learning Research, Proc. of the 12th Int. Conf. on Artificial Intelligence and Statistics*, pp. 57–64 (2009)
11. Baschera, G.-M., Busetto, A.G., Klingler, S., Buhmann, J.M., Gross, M.: Modeling Engagement Dynamics in Spelling Learning. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011. LNCS*, vol. 6738, pp. 31–38. Springer, Heidelberg (2011)
12. Dehaene, S.: *The Number Sense: How the Mind Creates Mathematics*. Oxford University Press (2011)
13. Dehaene, S.: Varieties of numerical abilities. *Cognition* 44, 1–42 (1992)
14. von Aster, M.G., Shalev, R.: Number development and developmental dyscalculia. *Developmental Medicine and Child Neurology* 49, 868–873 (2007)
15. Ostad, S.A.: Developmental differences in addition strategies: A comparison of mathematically disabled and mathematically normal children. *British Journal of Education Psychology* 67, 345–357 (1997)
16. Ostad, S.A.: Developmental progression of subtraction strategies: A comparison of mathematically normal and mathematically disabled children. *European Journal of Special Needs Education* 14, 21–36 (1999)
17. Kucian, K., Kaufmann, L.: A developmental model of number representation. *Behavioral and Brain Sciences* 32, 313 (2009)
18. Friedmann, N., Murphy, K., Russell, S.: Learning the Structure of Dynamic Probabilistic Networks. In: *Uncertainty in AI* (1998)
19. Mooij, J.M.: libDAI: A free & open source C++ library for Discrete Approximate Inference in graphical models. *J. of Machine Learning Research* 11, 2169–2173 (2010)
20. Pinheiro, J.C., Bates, D.M.: Approximations to the Log-likelihood function in Nonlinear Mixed-Effects Models. *Journal of Computational and Graphical Statistics* 4(1), 12–35 (1995)
21. Breiman, L.: Bagging predictors. *Machine Learning* 24(2), 123–140 (1996)
22. Haffner, J., Baro, K., Parzer, P., Resch, F.: *Heidelberger Rechentest (HRT): Erfassung mathematischer Basiskompetenzen im Grundschulalter*. Hogrefe Verlag (2005)

The Student Skill Model

Yutao Wang and Neil T. Heffernan

Worcester Polytechnic Institute
Department of Computer Science
yutaowang@wpi.edu, nth@wpi.edu

Abstract. One of the most popular methods for modeling students' knowledge is Corbett and Anderson's[1] Bayesian Knowledge Tracing (KT) model. The original Knowledge Tracing model does not allow for individualization. Recently, Pardos and Heffernan [4] showed that more information about students' prior knowledge can help build a better fitting model and provide a more accurate prediction of student data. Our goal was to further explore the individualization of student parameters in order to allow the Bayesian network to keep track of each of the four parameters per student: prior knowledge, guess, slip and learning. We proposed a new Bayesian network model called the Student Skill model (SS), and evaluated it in comparison with the traditional knowledge tracing model in both simulated and realword experiments. The new model predicts student responses better than the standard knowledge tracing model when the number of students and the number of skills are large.

Keywords: Knowledge Tracing, Individualization, Bayesian Networks, Data Mining, Prediction, Intelligent Tutoring Systems.

1 Introduction

One of the most popular methods for modeling students' knowledge is Corbett and Anderson's[1] Bayesian Knowledge Tracing model. The original Knowledge Tracing model does not allow for individualization. Several researchers have tried to show the power of individualization. Corbett and Andersen presented a method to individualize students' parameters with a two phase process and reported mixed results[2]. Recently, Pardos and Heffernan [4] showed that by a single process Bayesian network model: the prior per student model, more information about students' prior knowledge can help better fit model and provide more accurate prediction of student data. The result is inspiring; however, the author only looked into the students' prior knowledge and didn't extend the individualization to the other aspects of student knowledge, such as guess rate or learning rate. Pardos and Heffernan [5] also tried a method where they trained all four parameters per student in a pre-process, then took those values and put them into a per skill model to learn how the user parameters interacted with the skill. This method requires a two phase data process, which is complicated to use in real-world.

Our goal was to further explore the individualization of student parameters in order to allow the Bayesian network to keep track of all our parameters per student as well

as skill specific parameters simultaneously. We proposed a new Bayesian network model called the Student Skill model (SS), and evaluated it in comparison to the traditional Knowledge Tracing model (KT) in both simulation and real data experiments. The new model predicts student responses better than standard knowledge tracing model when the number of students and the number of skills are large.

2 The Student Skill Model

The Knowledge Tracing model assumes that all students have the same probability of knowing a particular skill at their first opportunity, or guess/slip in one skill, or learning a particular skill even though students seem likely differ in these aspects. Our goal was to add individualization into the original Knowledge Tracing model.

The new model we proposed in this paper is called the Student Skill model. It can learn four student parameters and four skill parameters simultaneously in a single phase process. The model is shown in Fig.1.

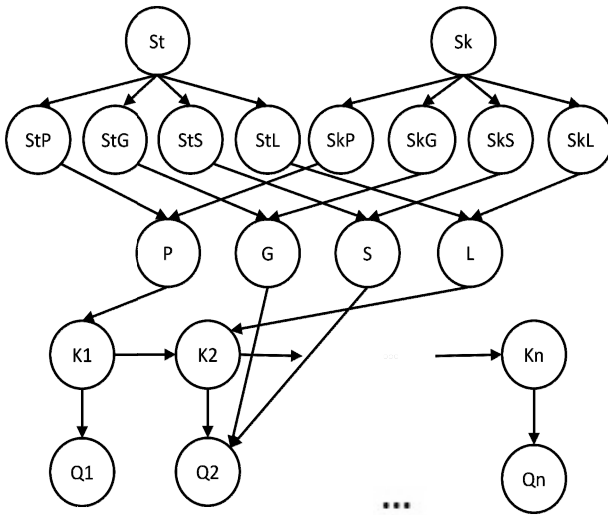


Fig. 1. The Student Skill model

The lowest two levels of this model are the same as the original Knowledge Tracing model (nodes K1~Kn and Q1~Qn in Fig.1). The Student Skill model adds upper levels to represent the student and skill information and their interaction. We used two multinomial nodes to represent the identity of each student (node St in Fig.1) and each skill (node Sk in Fig.1). Instead of pointing the student identity and the skill identity nodes directly to the knowledge nodes, which would result in a huge number of parameters, we added a level of nodes to represent the four student parameters (node StP, StG, StS and StL in Fig.1) and the four skill parameters (node SkP, SkG, SkS and SkL in Fig.1). Those parameter nodes are binary nodes that represent the high/low level of the corresponding parameters. For example, if the StP node is 1 for a student,

then the student has high level of prior knowledge, and if the StP node is 0 for a student, means the student has low level of prior knowledge. The next level uses conditional probability tables to combines the influence of the student parameters and the skill parameters and generates the four standard Knowledge Tracing parameters (node P, G, S and L in Fig.1) to be used in the lowest two levels.

The number of parameters in this model for n students and m skills can be computed as: $4n + 4m + 16$, while the number of parameters in the Knowledge Tracing model is: $4m$. The cost of individualization is the additional $4n + 16$ parameters.

3 Model Evaluation

The model is evaluated in both simulated and real data experiments. In our experiments, we used the Bayes Net Toolbox for Matlab developed by Murphy [3] to implement the Bayesian network student models and the Expectation Maximization (EM) algorithm to fit the model parameters to the dataset. We choose initial parameters for each skill in Knowledge Tracing as follows: initial knowledge = 0.5, learning = 0.1, guess = 0.1, slip = 0.1.

3.1 Simulation Experiments

Methodology.

To evaluate the ability of the Student Skill model to function properly, in this experiment, we generated data from the Student Skill model and compared the prediction accuracy with the Knowledge Tracing model. The data records generated in the simulation represent student performances, with 1 representing correct and 0 representing incorrect. To simulate the random noise in the real data, we randomly flipped over 1% of the student performance data.

To split the training and testing data set, for each student, we randomly selected half of the skills data and put them into a training set. The remaining data went to the testing set. Both the Knowledge Tracing model and Student Skill model were trained and tested on the same dataset. A sequence of performances of given students and skills were predicted by both of these models.

Results.

Prediction accuracy is the selected metric for evaluating the results. In one simulation, the number of skills was set at 30 while the number of students was changed from 5 to 100 to observe the influence the number of student had on SS and KT respectively. Similarly, in another simulation, the number of students was set to be 30 while the number of skills was changed.

We observed that, in situations with a small number of students as well as those with a small number of skills, the Knowledge Tracing model outperformed the Student Skill model. However, when the number of students and the number of skills were increased, the performance of the Student Skill model improved and eventually exceeded the Knowledge Tracing model. The reason for this trend could be the fact that the Student Skill model contains more parameters than the Knowledge Tracing model, and with fewer data points, the model behaves less reliably.

We also compared the Student Skill model and the Knowledge Tracing model under different student parameter variance. The number of students and the number of skills were both set to 40, and the number of data points per student per skill was set to 10. The student variance was controlled by the real parameters used to generate simulated data. When the student variance was 0, all students shared the same parameters. We observed that the Student Skill model performs worse when there is no variance in student parameters and when the students are highly variant, the Student Skill model outperformed the Knowledge Tracing model.

3.2 Real Data Experiments

One of the dangers of relying on simulation experiments is that the dataset may not reflect real-world conditions. Without evaluation using real data, the success of the new model during simulation could simply be caused by the data being generated from this model. To further evaluate the Student Skill model, we applied it to real datasets and again compared its performance with the Knowledge Tracing model.

Dataset.

The data used in the analysis presented here came from the ASSISTments platform, a freely available web-based tutoring system for 4th through 10th grade mathematics. We randomly pulled out the data of one hundred 12-14 year old 8th grade students and fifty skills from September 2010 to September 2011 school year. There are 53,450 total problem logs in the dataset.

Methodology.

The dataset was randomly split into four bins by student and skill in order to perform a four-fold cross-validation of the predictions and increase the reliability of the results. For each student, we made a list of the skills the student had seen and split that list randomly into four bins, placing all data for that student and that skill into the respective bin. There were four rounds of training and testing, during each round a different bin served as the test set, and the data from the remaining three bins served as the training set. Again, both the Knowledge Tracing model and the Student Skill model were trained and tested on the same dataset. A sequence of performances of the given students and skills were predicted by both of these models.

Results.

The accuracy of the prediction was evaluated in terms of the Root Mean Squared Error (RMSE). A lower value means higher accuracy. The cross-validation results are shown in Table 1.

Table 1. RMSE results of KT vs SS

<i>Fold ID</i>	<i>SS</i>	<i>KT</i>	<i>P value</i>	<i>Student Level P value</i>
Fold1	0.4017	0.4055	0.0432	0.0404
Fold2	0.4194	0.4385	0.0459	0.0365
Fold3	0.4144	0.4348	0.0477	0.0451
Fold4	0.4441	0.4538	0.0420	0.0406
average	0.4199	0.4331	-----	-----

To test the reliability of the four folds experiment, we did a paired T test for each fold as well as the result of all the folds. The P value that compares the final RMSE of the SS model and the KT model of the four folds is 0.0439. The P value for each individual fold is shown in the fourth column. Our experiment shows that the difference between SS and KT is statistically significant, and the average RMSE shows that SS is more accurate than KT under our experimental conditions. We also did reliability analysis by computing RMSE for each student to account for the non-independence of actions within each student's dataset, and then compared each pair of models using a two tailed paired t-test. The Student Level P values are reported in the last column. All the results are statistically reliable.

4 Discussion and Future Work

In this paper, we built a new Bayesian network model for modeling individual student parameters called the Student Skill model and compared it with the knowledge tracing model in both simulation and real data experiments.

In our experiments, we found that the Student Skill model is not always better than the Knowledge Tracing model. Under simulated conditions, we found that the new model is generally more accurate when the amount of students and skills are large. We are interested in other features that can indicate which model works better under what situations, in the hope that these two models can be combined in order to utilize both models' advantages.

5 Contribution

Several researchers have tried to show the power of individualization. Corbett and Andersen's presented a method to individualize students' parameters with a two phase process: first run Knowledge Tracing on all the students and then run a separate regression to learn a set of slip, guess, learning and prior parameters per students. Pardos and Heffernan [4] explored the individualized student prior, but did not learn all of the student parameters and skill parameters in one single model. We presented the SS model, which is elegant in accounting for individual differences (of learning rate, prior knowledge and guess and slip rates). Our simulation showed that we could reliably fit such a model. The simulation showed plausible results, such as that the SS model is better if more variation per student.

Our contribution is in presenting a model that allows us to use EM to learn parameters individualized to each student, while at the same time learn parameters for each skill. We presented simulation and real data experiments that showed this method can provide meaningful results. Knowledge Tracing is a special case of this model and can be derived by fixing the student parameters of the Student Skill model to the same values. In a practical sense, researchers need to figure out when the SS model can start to be used, as our simulation showed that SS is better than KT when 1) the number of skills a student has learned is high, and 2) the number of students is high.

Acknowledgements. This research was supported by the National Science foundation via grant “Graduates in K-12 Education” (GK-12) Fellowship, award number DGE0742503 and Neil Heffernan’s CAREER grant. We also acknowledge the many additional funders of ASSISTments Platform found here: <http://www.webcitation.org/5ym157Yfr>

All of the opinions expressed in this paper are those solely of the authors and not those of our funding organizations.

References

1. Corbett, A., Anderson, J.: Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction* 4, 253–278 (1995)
2. Corbett, A., Bhatnagar, A.: Student Modeling in the ACT Programming Tutor: Adjusting a Procedural Learning Model with Declarative Knowledge. In: *User Modeling: Proceedings of the 6th International Conference*, pp. 243–254 (1997)
3. Murphy, K.P.: The Bayes Net Toolbox for Matlab. In: *Computing Science and Statistics: Proceedings of Interface*, vol. 33 (2001)
4. Pardos, Z.A., Heffernan, N.T.: Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In: De Bra, P., Kobsa, A., Chin, D. (eds.) *UMAP 2010*. LNCS, vol. 6075, pp. 255–266. Springer, Heidelberg (2010)
5. Pardos, Z.A., Heffernan, N.T.: Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. To appear in the *Journal of Machine Learning Research W & CP* (in press)

Clustered Knowledge Tracing*

Zachary A. Pardos, Shubhendu Trivedi, Neil T. Heffernan, and Gábor N. Sárközy

Department of Computer Science, Worcester Polytechnic Institute, United States
{zpardos, s_trivedi, nth, gsarkozy}@cs.wpi.edu

Abstract. By learning a more distributed representation of the input space, clustering can be a powerful source of information for boosting the performance of predictive models. While such semi-supervised methods based on clustering have been applied to increase the accuracy of predictions of external tests, they have not yet been applied to improve within-tutor prediction of student responses. We use a widely adopted model for student prediction called knowledge tracing as our predictor and demonstrate how clustering students can improve model accuracy. The intuition behind this application of clustering is that different groups of students can be better fit with separate models. High performing students, for example, might be better modeled with a higher knowledge tracing learning rate parameter than lower performing students. We use a bagging method that exploits clusterings at different values for K in order to capture a variety of different categorizations of students. The method then combines the predictions of each cluster in order to produce a more accurate result than without clustering.

Keywords: Bayesian Knowledge Tracing, Clustering, Bagging.

1 Introduction

A recent work that involved clustering of the knowledge tracing (KT) space was that by Ritter *et al.* [1]. Their work focused on clustering the parameter space of KT [2] and essentially showed that the information compression offered by clustering was enough to significantly reduce the parameter space without compromising the performance of the system. Ritter *et al.* also mention this as their motivation. It thus cannot be considered an extension to KT per se, but it raises important questions about the nature of the parameter space. Trivedi *et al.* [3] used clustering to make better out-of-tutor predictions and didn't deal with knowledge tracing at all. They clustered students based on features of tutor usage and then used those features to fit a model to predict performance on a test that students are given at the end of the school year. In our case, we cluster students based on some tutor usage features and then use these distinct clusters to train KT on them. We use a technique by Trivedi *et al.* [3] that exploits the information handed down by varying the granularity of the clustering to learn a more distributed representation.

* A longer version of this paper is available online at:
<http://web.cs.wpi.edu/~gsarkozy/CikkeK/57.pdf>

2 Clustered Knowledge Tracing

For each student we have a number of features that measure his/her interaction with the tutor. Students could be clustered on the basis of these features and once the groups have been found the item sequences for these groups of students could be used for training KT separately. Below we briefly review the clustering algorithms and the bootstrapping method used.

2.1 Clustering Algorithms Used and Strategy for Bootstrapping

In our experiments we clustered students based on the features on tutor usage based on two algorithms: k-means and spectral clustering [4]. The basic k-means algorithm finds groupings in the data by randomly initializing a set of K cluster centroids and then iteratively minimizing a distortion function and updating these K cluster centroids and the points assigned to them. This is done till a point is reached such that sum of the distances of all the points with their assigned cluster centroids is as low as possible. Clustering methods such as k-means estimate explicit models of the data (specifically spherical gaussians) and fail spectacularly when the data is organized in very irregular and complex shaped clusters. Spectral clustering on the other hand works quite differently. It represents the data as an undirected graph and analyses the spectrum of the graph laplacian obtained from the pairwise similarities of the data-points. This view is useful as it does not estimate any explicit model of the data and instead works by unfolding the data manifold to form meaningful clusters. Usually spectral clustering is a far more “accurate” clustering method as compared to k-means except in cases where the data indeed confirms to the model that the k-means estimates. This leads to another interesting question – Which of the two works better in our scenario? This question is more interesting than just the comparison of two algorithms. If the per-user-per-skill KT parameters are arranged in approximately spherical clusters then the k-means algorithm might do better and vice versa. Note that this should happen even though we are clustering tutor usage features and not the per-user-per-skill KT parameters themselves. This is because student groupings in the feature space should correspond to the groupings found in the KT parameter space unless the features collected are irrelevant. An exploration of this correspondence could be used to collect or engineer better features. These features should also be more useful for out-of-tutor predictions as well.

Using the methodology due to Trivedi *et al.* [3] we use clustering for bagging predictors. Using the features from tutor usage we initially employ clustering to find K student groups. Corresponding to each group identified we train KT models separately, thus getting K different models (Trivedi *et al.* call each such model trained on one cluster a “cluster model”). All of these models together will make one set of predictions on the test data (all of the cluster models together for a given K are called a “prediction model” PM_K). This process is schematically described in Fig. 1. The number of clusters K is then varied and the above process is repeated iteratively from $K - 1$ to 1 ($K = 1$ corresponds to KT trained on the entire dataset, this should serve as

the baseline KT). By this process we get a set of K different predictions. These predictions are then averaged to get a single final prediction.

3 Empirical Validation

In this section we present results of experiments to evaluate the performance of “Clustered Knowledge Tracing” as described above and compare it with the baseline. Both k-means and spectral clustering are used. Specifically we used the classical k-means with random initialization and for spectral clustering we used self-tuned spectral clustering with a fully connected graph of data-points.

3.1 Dataset Description

The data comes from the 2010 KDD Cup competition on educational data mining. We used the Algebra 2005-2006 and the Bridge to Algebra 2006-2007 datasets. These represent two different Algebra tutoring systems which are part of the Cognitive Tutor family of tutors [5]. The number of students in the Algebra set was 575 with 813,661 total logged responses over 387 skills. There were 1,146 students in the Bridge to Algebra set with 3,656,871 total logged responses over 470 skills. These datasets included skill information for each response and no response was tagged with more than one skill. The Cognitive Tutor divides its online curriculum into units. Skills which appear in different units, even if they have the same name, are considered different skills. Within units there are many problems which students try to solve. Each problem consists of many sub questions called steps. Steps are the level at which the responses in this dataset were logged. Our training and test set is the same as defined by the competition organizers [6]. We stick to the competition’s train and test set format so that comparisons can be made between the error levels we find and the error levels of other published work with this dataset. The various tutor features that were used to cluster the students were: number of skills completed, total number of data-points, user prior, user learn rate, user guess, user slip, number of EM iterations, Log likelihood improvement, percent correct, average response time. In experiments, students were clustered using *all* these features and also only using the user tutor features (user prior, user learn rate, user guess, user slip). These user specific KT parameters were generated like in [6] by training a separate KT model per student based on all of that student’s data in the training set (across all skills).

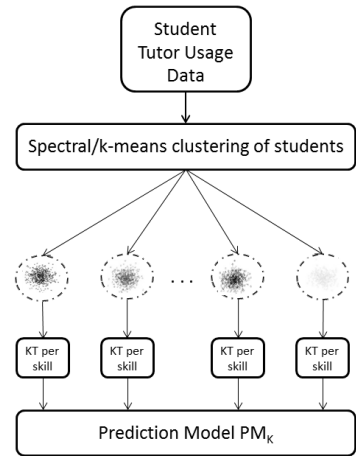


Fig. 1. Construction of a Prediction Model for a given K . In each case a new PM_K is obtained and thus a prediction on the test data.

3.2 Results of the Bagging Strategy to Knowledge Tracing

For both datasets we report results using *all* the features described above and also by only using the user features. The results while using all features are with both kmeans and spectral clustering, and while using the user features are only by kmeans. We report the results for both the individuals prediction models (i.e. the model obtained by training KT on each cluster for a given K i.e. PM_K) and the ensemble results (results obtained by averaging from PM_1 to PM_K). For results we report the RMSE defined per user. The justification to use the RMSE per user is that it equally weighs the benefit to each student without biasing it to students who have contributed more data points.

Initially we tried spectral clustering for the purpose of bootstrapping. This was motivated by the fact that spectral clustering is generally better than k-means clustering as discussed in section 2.1. Fig 2 shows the results for bagging using spectral clustering considering all the features on both the datasets. We see the declining trend in error when the results are ensemble and also notice that the individual prediction models don't do too well showing that clustering alone does not help but blending the predictions does. Fig 3 indicates that a similar result is repeated in the same scenario with k-means (all features) in the algebra dataset. Such a result is not observed in the bridge dataset however. In fact in the bridge dataset both the various PM_k and the ensemble results do worse than the baseline (which is PM_1 i.e. KT trained on the entire dataset). But in further experiments we see that we can do better even on the bridge dataset if we consider only the user features. For the algebra dataset the baseline (i.e. PM_1) RMSE is 0.32185, which represents standard KT with no clustering. The best result in the Algebra dataset for spectral (Fig 2) is obtained on averaging the first ten prediction models (0.31706). The best result for k-means (Fig 3) on this dataset is 0.31696, also after averaging the first ten prediction models. The result is surprising as kmeans seems to do better than spectral clustering in this case. Perhaps this might be explained by the intuition in section 2.1. The trend however is reversed in the Bridge to algebra data-set, however we still note that the ensemble using spectral clustering does better than the baseline for all the K 's considered in this dataset. Given that k-means appeared to do well in one dataset and also given its speed, the above procedure was repeated in both the datasets with k-means using only the user specific features. We also cluster to a much higher K and see that the error trend line only decreases as K is increased as is shown in Fig 4. Here again, for the Algebra dataset, PM_1 has an RMSE of 0.32185. The best prediction accuracy on averaging is attained at $K = 20$ where the RMSE is 0.3149. This accuracy is even better as was reported earlier considering both the clustering methods indicating that the user features are much richer for clustering the students. When only the user features are considered a similar error profile is also observed in the bridge to algebra dataset too (PM_1 RMSE = 0.28397 and RMSE of the average from PM_1 to PM_{30} is 0.28225). Except for the case when kmeans was run on the bridge to algebra set considering all the features, all the improvements are statistically significant over the baseline ($p < 0.05$). In another experiment in which all the above models are combined, the best accuracy that we obtain for the algebra dataset is 0.31506 and 0.2827 for the bridge to algebra dataset. Like we noted earlier, we report the RMSE per user. However even if we considered the RMSE on the leaderboard we get a statistically significant improvement over the baseline with PM_1 being 0.32408 and the best prediction being 0.32318.

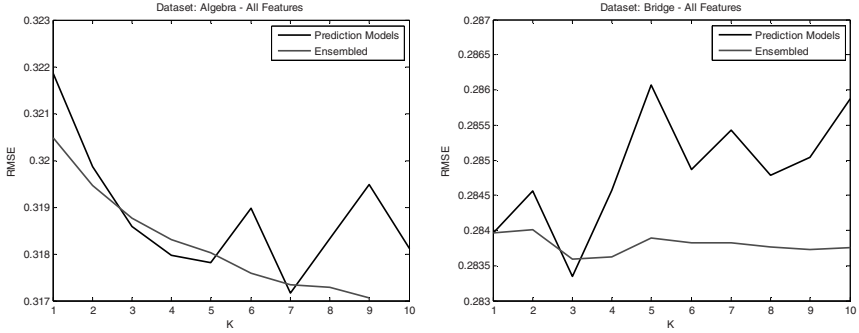


Fig. 2. Results on the Algebra (L) and the Bridge to Algebra (R) datasets with spectral clustering when all the features are considered. The red line shows the ensembled results after averaging from PM_1 to PM_K while the black one shows the results for each Prediction Model (PM_K).

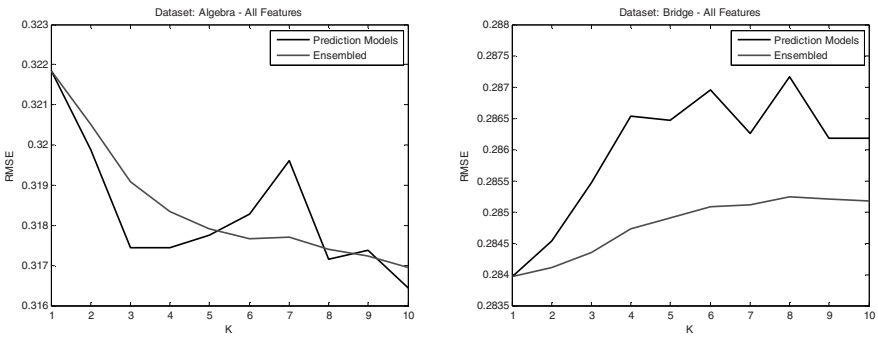


Fig. 3. Algebra (L) and the Bridge to Algebra (R) with k-means clust. considering all features

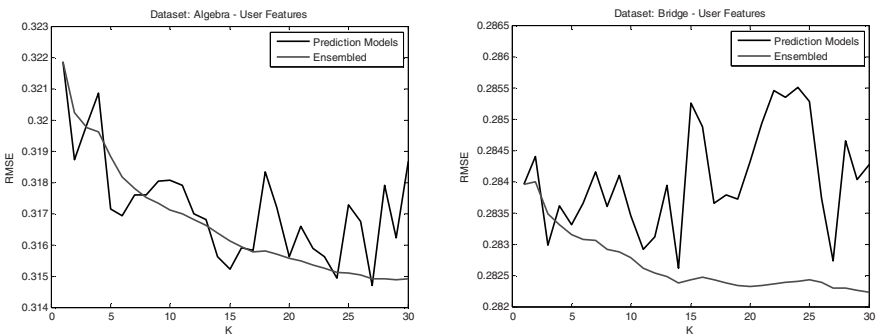


Fig. 4. Algebra (L) and the Bridge to Algebra (R) with k-means clust. considering user features

4 Discussion and Future Work

While various extensions to the base KT model have focused on adding new features to the base model, in this work we took a slightly different view. Instead of trying to model new parameters we try to learn a more distributed representation of the KT input space. We achieve this by using clustering for bootstrapping. In extensive validation we show that our strategy indeed works very well. We report an improvement in prediction accuracy in most cases. We also report that the user features are much richer for clustering than the features of interaction of a student with a tutor. We believe that this leads to an interesting research problem. Often, the interaction of students with a tutor is measured and recorded as features. These features should be such that if students were clustered on this feature space, the clustering should correspond to one on the KT parameter space. If it is not the case then it indicates that the task of feature generation in the tutor is noisy and could be improved in a more principled manner. An improvement in methodology here would be greatly useful in getting features that would be most helpful in making better out-of-tutor predictions. An interesting problem would be to consider a case study in which the various clusters are analyzed and an attempt is made to interpret them on the basis of the associated KT parameters. Such a study could be quite useful, especially in making some data driven inferences and pedagogy. Lastly, this exploration concerning the KT input space, especially concerning learning a more distributed representation could be quite useful even when used in conjunction with KT variants such as [6] that are known to be stronger predictors than the base KT.

References

1. Ritter, S., Harris, T., Nixon, T., Dickison, D., Murray, R., Towle, B.: Reducing the knowledge tracing space. In: In Proceedings of the International Conference on Educational Data Mining, Cordoba, Spain, pp. 151–160 (2009)
2. Corbett, A.T., Anderson, J.R.: Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User Adapted Interaction* 4, 253–278 (1995)
3. Trivedi, S., Pardos, Z.A., Heffernan, N.T.: Clustering Students to Generate an Ensemble to Improve Standard Test Score Predictions. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011*. LNCS, vol. 6738, pp. 377–384. Springer, Heidelberg (2011)
4. Luxburg, U.: A Tutorial on Spectral Clustering. In: *Statistics and Computing*, vol. 17(4). Kluwer Academic Publishers, Hingham (2007)
5. Koedinger, K.R., Corbett, A.T.: Cognitive tutors: Technology bringing learning science to the classroom. In: Sawyer, K. (ed.) *The Cambridge Handbook of the Learning Sciences*, pp. 61–78. Cambridge University Press, New York (2006)
6. Pardos, Z.A., Heffernan, N.T.: Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. To appear in *Journal of Machine Learning Research W & CP*

Preferred Features of Open Learner Models for University Students

Susan Bull

Electronic, Electrical and Computer Engineering, University of Birmingham, UK
s.bull@bham.ac.uk

Abstract. This paper describes features and purposes for opening the learner model to the learner. Building on previous studies of use of a range of open learner models, it considers the features that are preferred by university level, experienced open learner model users. Recommendations are presented to help guide open learner model designers in their choices of features to make available to learners, with reference to user control, privacy, navigation, visualisation content and detail, comparisons and releasing models to peers.

Keywords: Open learner models, learner preferences.

1 Introduction

Intelligent tutoring systems model the user's knowledge or strength of knowledge in a domain, and may also model an individual's difficulties and/or misconceptions or other learning-related attributes. Based on this model, inferred during the learner's use of the system (e.g. their answers to questions; problem-solving tasks; tasks or subtasks attempted or completed; hints used; time taken to complete a task; number of attempts required), the system is able to personalise the educational interaction appropriately according to the current needs of the user. This may result in a range of interventions or interaction types, such as: additional exercises or tasks; explanations; tutoring on new or problematic topics; prompting reflection on difficult concepts; suggestions for navigation, and so on.

The system will therefore usually provide some kind of tutoring, scaffolding or guidance, as suited to the individual user according to the current state of their learner model. However, intelligent tutoring systems are now increasingly identifying benefits of opening the learner model directly to the learner, for example: to promote awareness and reflection; to aid planning; to facilitate independent learning; to encourage collaboration; to encourage and help learners recognise and take greater responsibility for their learning (see [1]). Open learner models are also used independently of tutoring systems with a particular focus on promoting metacognitive activities and learner independence [2]; and may incorporate data from a range of sources [3,4].

This paper takes benefits such as the above as a starting point, and then focuses on the preferences of 230 experienced open learner model users, for features of an open

learner model. This leads to recommendations for open learner model designers, about features to include in their open learner models.

2 Open Learner Models

Open learner models have been used with various types of model, ranging from visualisations of simple weighted numerical models of knowledge level (e.g. [5]); to more complex models incorporating conceptual and/or hierarchical relationships (e.g. [6,7,8,9]; constraint-based models [10]; and Bayesian models [11]. The method by which the learner model is externalised to the user may not match the format or complexity of the underlying model [1]. For example, skill meters have indicated level of understanding represented in a simple weighted numerical model [5], and also in a constraint-based model [10]. A primary concern is that the model should be presented in a form that is *understandable by the user*. This is not as straightforward as simply showing the learner the representations in the underlying system's model, as these are not designed for human interpretation. In particular, it must be taken into account that learners are often still learning a subject, and so may not be able to easily interpret a learner model presentation with reference to their progress. For this reason, multiple views of the learner model have been made available (e.g. [9,12]).

As an example of learner model presentations, Figure 1 shows skill meters and a pre-formatted structured view of the learner model, both of which are available in the same environment, presenting information from the same learner model data [13]. This example is for a general open learner model; the screen shots from an Adaptive Learning Environments course. The skill meters (left) show current level of understanding of each topic (medium shading); existence of any misconceptions in a topic (dark shading); and general difficulties that are not inferred to be caused by specific misconceptions (light shading). Brief statements of misconceptions are revealed by clicking on the 'misconceptions' links. For example: "you may believe that an intelligent tutoring system does not have to 'understand' the learner model." This misconception is sometimes identified early in the course before students have fully understood what it means to have a *model* of knowledge to enable adaptive interaction (e.g. students think of system responses as a form of feedback, perhaps tied to specific questions or sets of questions). The misconception can be identified by selection of response options to a range of questions (in multiple choice format in this case), for example, options indicating that a student believes:

- a learner model is simply a record of the student's answers;
- an open learner model externalises the underlying form of the model (i.e. believing that the 'view' of the learner model available to them is the way the learner model is stored in the system);
- a learner model is simply the system's feedback (a misconception sometimes first arising when the notion of open learner models is introduced).

The structured view of the learner model (right) shows level of understanding by the colour of the nodes, also indicating the structure of the topics in the course (e.g. learner modelling techniques and open learner models are part of the learner model

topic; various aspects of individual differences feed into the individual differences topic which, in turn, relates to the learner model topic; and so on. Students may use whichever of these (or other) views of the learner model that they wish. (Previous work has demonstrated individual differences in preferences for learner model views, in several open learner model systems [12], hence our use of this approach here).

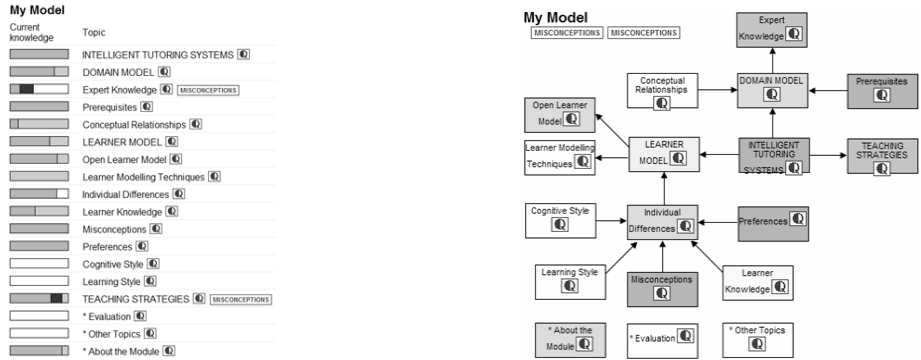
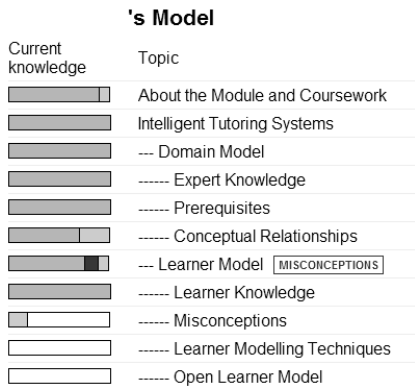
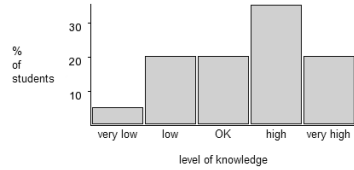


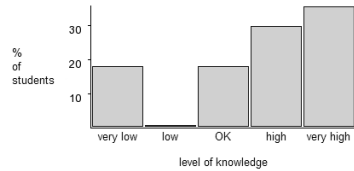
Fig. 1. Skill meters and structured learner model view [13]



Intelligent Tutoring Systems



Domain Model



Course: Computer Hardware & Digital Design

Topic	Users attempting exercises on topic	Users with good knowledge	Users with fair knowledge	Users with weak knowledge	Users with misconceptions
Number representation	34	23 (67%)	7 (20%)	4 (11%)	3 (8%)
Structural VHDL	32	29 (90%)	1 (3%)	2 (6%)	2 (6%)
Timing simulation in VHDL	33	25 (75%)	6 (18%)	2 (6%)	2 (6%)
VHDL processes	33	28 (84%)	3 (9%)	2 (6%)	0 (0%)
Pipelining	33	30 (90%)	2 (6%)	1 (3%)	0 (0%)
Digital systems testing	20	14 (70%)	4 (20%)	2 (10%)	0 (0%)

Fig. 2. Peer models: individual and group models [5,15]

In addition to the learner being able to view their learner model, it may also be accessible to other users. Examples have so far been developed primarily for teachers/instructors and peers (e.g. [9,14]), though other stakeholders in the education context can also be included [4]. Individual learner models may be presented to others, and/or aggregate or group models may be shown. In this paper we are concerned with university level open learner models, and focus the discussion on learner models open to the learner and peers. Figure 2 gives an excerpt from an example of how a student may view individual peer models where peers have given their permission for their model to be available to them (top left); and a group model for two topics (top right) [13]; and a numerical summary of group understanding (bottom) [15].

It is not only the visualisation of the model that is important: an open learner model may involve more than just the externalisation of its contents. The learner may also be able to interact with their model, and/or change it in some way. For example, the learner may have complete control over the model contents by being able to edit them [8]; some control by being able to offer evidence or additional information for the system to take into account [6]; or by enabling the learner and system to discuss and negotiate the model towards joint agreement on its contents, achieved, for example, through dialogue games [7], or chatbot [16]. This contrasts with the simplest definition of "open learner model", where the system presents the model data for student inspection as described above, but does not allow the learner any direct comment or input about the model data. Figure 3 shows an example of how a learner model may allow direct input from the learner, to correct the learner model (as might be useful, for example, following learning away from the computer environment) [8].

Edit Learner Model

Topic: Control of flow statements
Stage 2: Review System's Evidence

Misconception: "*control of flow statements are followed by a semicolon*"
 Currently: Your proposal:

Below are some of your responses which suggest you may hold this misconception.

Question	Your response
What should go in the gap indicated by the underscore? i.£ (£<5) __	a semicolon

If you still wish to edit the model, click 'continue' and **your proposed changes will be applied**. Otherwise click 'back' to review your changes.

Fig. 3. Editing the learner model

There are a variety of purposes for opening the learner model [1], including:

- Addressing the user’s right to view data about themselves;
- Aiding navigation directly from the open learner model;
- Raising awareness (of knowledge, progress, difficulties, etc.);
- Facilitating planning;
- Helping the learner to take greater control over their learning decisions;
- Promoting collaborative interaction.

This paper investigates student preferences for the presentation of the learner model data, and the purposes for which they would use an open learner model.

3 Previous Findings from Open Learner Model Use

Use of open learner models alongside university courses has become more widespread in recent years (e.g. [5,9,10,17,18]). Previous use of an independent open learner model over time has shown that many students may have misconceptions and, in most cases, they will view statements of their misconceptions [5]. It has also been found that learners will consult peer models and open their own learner model to peers to facilitate collaborative interactions [14]; and that viewing peer models can benefit learning [18]. Other studies investigating learning from open learner models found that students may become better at problem selection [10]; an that open learner models may be used to visualise work on long term group projects [19].

Previous work has also investigated user opinions of various features of open learner models, including access to the model using methods such as overview, zoom, filtering, and possibility to modify [20]; and the context of use of the model, such as whether it is assessed; point in the course at which it becomes available; method of introduction of the environment [21]. The following section investigates student preferences for features of an open learner model in greater detail, with a large number of students with experience of working with several independent open learner models. This is the first study on such a scale that considers user preferences for open learner model features in general (i.e. not with reference to a specific system). This allows future open learner model designers to take into account, the likely attitude of students towards various kinds of open learner model that are designed to support learning alongside university courses.

4 User Preferences for Open Learner Models

As stated above, there have been positive findings for university level use of open learner models. In this section we build on these results, introducing a survey into students' preferences for open learner model features.

4.1 Participants, Materials and Methods

Participants were 230 university students over six years, who completed a survey of their open learner model preferences, giving responses to statements on a three point scale (agree, neutral, disagree). The participants were in their final year of a 3 year BEng degree, in their third or fourth year of a 4 year MEng degree, or studying for a 1 year postgraduate (MSc) degree. All had previous direct experience of interacting with at least three open learner models, from: CALMSystem (a negotiated learner model with a simple model view) [16], MusicaLM (an inspectable learner model with multiple simple views) [22]; AniMis (an inspectable learner model using animations of understanding) [23]; Flexi-OLM (an editable learner model having simple and structured views) [8], t-OLM (an editable learner model that can be released to peers, having simple and structured views) [13]; OLMlets (a simple learner model that can be released to peers) [5], UMPTEEN (a simple learner model that can be released to

peers) [15]; with at least one of these used throughout a term as a learning support alongside one or more lecture courses. (The latter four open learner models have each been available to support longer term learning.) Each of the above is an ‘independent open learner model’ [2] (i.e. the interaction is focused around the learner model with no system tutoring – it is the students’ own responsibility to determine their learning choices). All participants had a good understanding of educational technology and adaptive learning environments, as they were taking courses in these areas as part of their degree at the time of the study, and the survey questions were distributed during one of the teaching sessions. The results, therefore, are based on the views of knowledgeable and experienced open learner model users.

4.2 Results

Figure 4 shows that there is a preference for using an inspectable learner model, with about 70% of students selecting this option. However, when the neutral responses are also taken into account, the difference between preference for inspectable and the more interactive (editable and negotiated) learner models, diminishes¹. (Inspectable - where the user has no control over the model data; editable - where the learner has complete control over the model; negotiated - where there is joint control). In each case there is a minority of respondents stating that they would *not* use that type of open learner model. In accordance with the above, there is also a lower level of preference for contributing to the learner model to help improve its accuracy.

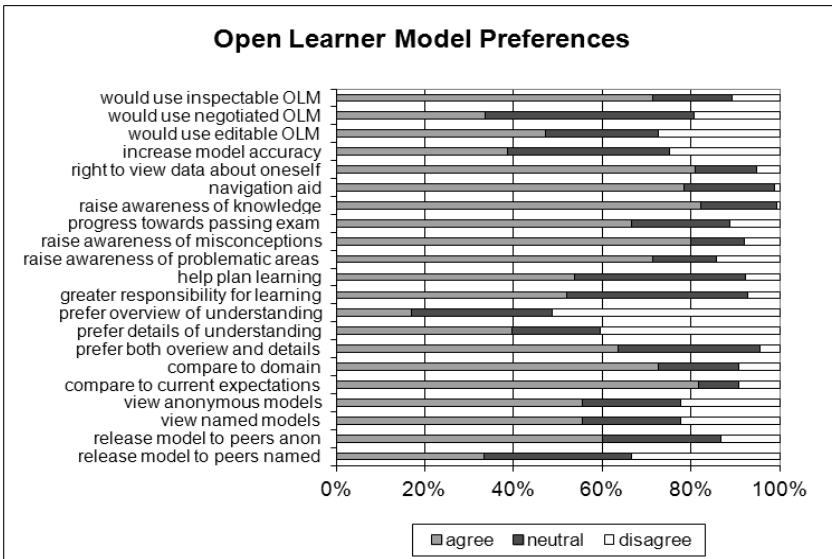


Fig. 4. User preferences for open learner models

¹ Note that a student may have more than one preference.

There is a strong belief that it is the user's right to be able to view the contents of their learner model (just over 80%), with very few participants disagreeing with this. A little under 80% consider an open learner model useful as an aid to navigation; and a little over 80% as a means of raising their awareness of their own knowledge state, with even higher appreciation of the use of an open learner model to help them recognise their misconceptions. Around three quarters of respondents indicated that an open learner model could help them realise any areas of general difficulty (not related to specific misconceptions), and around two thirds would like to use it to help them judge their progress towards passing an examination or other forthcoming assessment. Only around half wished to use an OLM for metacognitive activities such as planning their learning, and taking responsibility for their own learning.

Most have a preference for the availability of both overview and detailed presentations of the learner model (nearly two thirds); though some would prefer one option over the other. Most students would prefer to be able to compare their current understanding against the domain (over 70%); and even more (just over 80%) want to be able to compare their current knowledge to the current expectations for a course.

Over half of students would view named and anonymous learner models that had been released to them; 60% would be willing to release their own learner model anonymously to others; and over one third would release their learner model in identifiable form (i.e. with their name).

4.3 Discussion and Recommendations

As indicated above, open learner models are increasingly being used in university education, with some positive findings for improvements to the learning process (e.g. [10]), data on use of an open learner model [5], important model features [20], and sharing learner models [14,19]. Our findings build on such results – the starting point is the positive outcomes of previous research in the field (i.e. we take it as given that, at least in some circumstances, open learner models can be beneficial for learners).

The participants were experienced not only in open learner model use, but also had theoretical knowledge of open learner models and intelligent tutoring systems, as an academic subject. This can be viewed either as an advantage, in that the participants fully understand what is, for many users, an unusual learning application; but could also be viewed as a limitation because this experience may make the results less generalisable. Either way, this study has introduced some new information: it is on a larger scale than previous investigations of student preferences for open learner model features, and considers a range of features that are not typically combined in a single system, the context in which most open learner model studies are undertaken. Minimally, therefore, this study provides information about features of open learner models that students may prefer to use once they are accustomed to an open learner model environment, and understand its purpose. (Previous work has considered contexts in which a specific open learner model is most likely to be taken up [21].)

The study focused on independent open learner models. Findings are likely to be generally applicable to open learner models in intelligent tutoring systems (questionnaire items were general, and participants were familiar with intelligent

tutoring systems); but a similar broad study with open learner models in larger systems could confirm or provide additional results.

The fact that the majority of students were in favour of using inspectable learner models, coupled with previous data on actual use of inspectable learner models [2,5], is a strong indication that they can facilitate learning or improve the learning experience in some way. We may not yet know quite how these benefits are perceived, but learners clearly believe there to be some benefit to their learning. Our first recommendation is therefore:

- *Provide an inspectable learner model where possible and appropriate, to support independent learning in courses.*

Preferences for learner models where the learner has greater direct influence of the model contents, were considered less crucial, with around 40% stating that they would use such an environment. In line with this, students were not highly concerned about improving the accuracy of the learner model with their own direct contributions. Therefore our second recommendation is:

- *Where development resources are available, consider an optional mechanism for students to contribute information directly to their learner model. This may be to give them full control or partial influence (which may include a requirement for system verification of changes).*

As the right to view data about oneself was considered to be so important by students, this could contribute to the initial introduction of an open learner model as a learning resource:

- *When introducing an open learner model in a course, explain the privacy issues in addition to the learning benefits.*

An open learner model to support navigation was considered very useful:

- *When introducing an open learner model in a course, explain the benefits to users, of being able to access materials and/or exercises from within the learner model (i.e. that they can use the learner model for guidance).*

There was strong appreciation of each of the progress-related items: awareness of knowledge, progress towards passing an exam, awareness of misconceptions and awareness of general difficulties. Thus:

- *Where modelled in a system, provide learners with information about both positive aspects of their learning, and more problematic areas.*

The more metacognitive aspects, planning and responsibility for learning, were considered important by only around half the students. This is of concern, since one of the primary aims of open learner models is to promote metacognition [24]. From this data we do not know whether learners benefitted in this way from an open learner model, but did not consider it important (recall that participants had experience of real use of at least one open learner model during a term); or whether they did not engage in any additional metacognitive activity. Further research is required on this issue.

When it comes to overviews versus detailed views of learner model data, most students would prefer to have both, though some have a preference for one approach. Previous work has allowed learner selection of simpler versus more complex views [8], and our survey results are in line with this. Therefore we recommend:

- *Provide learners with the choice of whether and when to view their learner model in overview or detailed form, if there is no specific pedagogical reason to offer one over the other.*

With reference to comparison of the model to other information, students were particularly keen to compare their progress against the *current* expectations for their course and, to a slightly lesser extent, to the overall expert or domain knowledge:

- *If possible, provide comparisons for students, firstly so that they can identify their overall progress (not only their knowledge); but also to allow them to gauge their progress with reference to what is expected of them at the stage of the course they have reached.*

In contrast, while still a majority (with many of the other students remaining neutral), participants were less concerned with comparisons to peers' knowledge. Nevertheless, responses still indicate that it would be useful to offer this:

- *Allow students to release their learner models to each other, if they wish. (Experience has shown that it is useful to explain the benefits – e.g. prompting collaboration – for students to try this.)*

Furthermore, sufficient numbers of users would be prepared to release their models named or anonymously, for this feature to be a realistic benefit for those who wish to use it. (Indeed, other research has demonstrated this in practice [14]).

5 Summary

This paper has presented some of the key issues described in the open learner model literature as a starting point for identifying features of open learner models that students would most like to use. Participants were experienced users of open learner models, with additional theoretical knowledge of open learner modelling. A survey of 230 users enabled the identification of features considered important by students for open learner models in general, with reference to: user control over the model data, privacy, navigation, model content and detail, comparisons, and releasing models to each other. This enabled the recommendation of features to include in open learner models at university level, according to those who had experienced alternatives.

References

1. Bull, S., Kay, J.: Student Models that Invite the Learner In: The SMILI Open Learner Modelling Framework. *Int. J. of Artificial Intelligence in Education* 17(2), 89-120 (2007)

2. Bull, S., Mabbott, A., Gardner, P., Jackson, T., Lancaster, M., Quigley, S. & Childs, P.A.: Supporting Interaction Preferences and Recognition of Misconceptions with Independent Open Learner Models, in W. Neijdl, J. Kay, P. Pu & E. Herder (eds), *Adaptive Hypermedia and Adaptive Web-Based Systems*, Springer-Verlag, Berlin Heidelberg, 62-72, (2008)
3. Mazzola, L., Mazza, R.: GVIS: A Facility for Adaptively Mashing Up and Presenting Open Learner Models, in M. Wolpers, P.A. Kirschner, M. Scheffel, S. Lindstaedt & V. Dimitrova (eds), *EC-TEL 2010*, Springer-Verlag, Berlin Heidelberg, 554-559, (2010)
4. Reimann, P., Bull, S., Halb, W. & Johnson, M.: Design of a Computer-Assisted Assessment System for Classroom Formative Assessment, CAF11, IEEE (2011)
5. Bull, S. Jackson, T. & Lancaster, M. Students' Interest in their Misconceptions in First Year Electrical Circuits and Mathematics Courses, *International Journal of Electrical Engineering Education* 47(3), 307-318, (2010)
6. Kay, J.: Learner Know Thyself: Student Models to Give Learner Control and Responsibility. In: Halim, Z., Ottomann, T., Razak, Z. (eds.), *ICCE, AACE*, 17-24 (1997)
7. Dimitrova, V.: StyLE-OLM: Interactive Open Learner Modelling. *Int. J. of Artificial Intelligence in Education*. 13(1), 35-78 (2003)
8. Mabbott, A. & Bull, S.: Student Preferences for Editing, Persuading and Negotiating the Open Learner Model, in M. Ikeda, K. Ashley & T-W. Chan (eds), *Intelligent Tutoring Systems*, Springer-Verlag, Berlin Heidelberg, 481-490, (2006)
9. Perez-Marin, D., Pascual-Nieto, I.: Showing Automatically Generated Students' Conceptual Models to Students and Teachers, *Int. J. of Artificial Intelligence in Education* 20(1), 47-72 (2010)
10. Mitrovic, A., Martin, B. Evaluating the Effect of Open Student Models on Self-Assessment. *Int. J. of Artificial Intelligence in Education* 17(2), 121-144 (2007)
11. Zapata-Rivera, J.D., Greer, J.E. Interacting with Inspectable Bayesian Models. *Int. J. of Artificial Intelligence in Education* 14, 127-163 (2004)
12. Bull, S., Gakhal, I., Grundy, D., Johnson, M., Mabbott, A., Xu, J.: Preferences in Multiple View Open Learner Models, in M. Wolpers, P.A. Kirschner, M. Scheffel, S. Lindstaedt & V. Dimitrova (eds), *EC-TEL 2010*, Springer, Berlin Heidelberg, 476-481, (2010)
13. Ahmad, N., Bull, S.: Learner Trust in Learner Model Externalisations. *Artificial Intelligence in Education 2009*, IOS Press, Amsterdam (2009)
14. Bull, S., Britland, M.: Group Interaction Prompted by a Simple Assessed Open Learner Model that can be Optionally Released to Peers, in P. Brusilovsky, K. Papanikolaou & M. Grigoriadou (eds), *Proceedings of Workshop on Personalisation in E-Learning Environments at Individual and Group Level (PING), User Modeling* (2007)
15. Bull, S., Mabbott, A., Abu-Issa, A. UMPTEEN: Named and Anonymous Learner Model Access for Instructors and Peers, *Int. J. of AI in Education* 17(3), 227-253, (2007)
16. Kerly, A., Ellis, R., Bull, S.: CALMsystem: A Conversational Agent for Learner Modelling, *Knowledge-Based Systems* 21(3), 238-246, (2008)
17. Demmans Epp, C., McCalla, G.: ProTutor: Historic Open Learner Models for Pronunciation Tutoring. In G. Biswas, S. Bull., J. Kay & A. Mitrovic (eds.), *Artificial Intelligence in Education*, Springer-Verlag, Berlin Heidelberg, 441-443 (2011)

18. Hsiao, I-H, Bakalov, F., Brusilovsky, P., Koenig-Ries, B.: Open Social Student Modeling: Visualizing Student Models with Parallel Introspective Views, in J.A. Konstan, R. Conejo, J.L. Marzo & N. Oliver (eds), *User Modeling, Adaptation and Personalization*, Springer-Verlag, Berlin Heidelberg, 171-182 (2011)
19. Kay, J., Yacef, K., Reimann, P.: Visualisations for Team Learning: Small Teams Working on Long-Term Projects, In C. Chinn, G. Erkens & S. Puntambekar (eds.), *Minds, mind, and society CSCL*, International Society of the Learning Sciences, 351-353 (2007)
20. Bakalov, F., Koenig-Ries, B., Nauerz, A., Welsch, M.: Introspective Views: An Interface for Scrutinizing Semantic User Models. In P. De Bra, A. Kobsa & D. Chin (eds.), *User Modeling, Adaptation and Personalization*, Springer, Berlin Heidelberg, 219-230 (2010)
21. Bull, S.: Features of an Independent Open Learner Model Influencing Uptake by University Students, in De Bra, P., Kobsa, A. & Chin, D. (eds), *User Modeling, Adaptation and Personalization 2010*, Springer-Verlag, Berlin Heidelberg, 393-398, (2010)
22. Johnson, M., Bull, S.: Belief Exploration in a Multiple-Media Open Learner Model for Basic Harmony, in V. Dimitrova, R. Mizoguchi, B. du Boulay & A. Graesser (eds), *Artificial Intelligence in Education 2009*, IOS Press, Amsterdam, 299-306, (2009)
23. Johan, R., Bull, S.: Promoting Collaboration and Discussion of Misconceptions Using Open Learner Models, in A. Bader-Natal, E. Walker & C.P. Rose (eds), *Proceedings of Workshop on Opportunities for Intelligent and Adaptive Behavior in Collaborative Learning Systems, Intelligent Tutoring Systems*, 9-12, (2010)
24. Bull, S., Kay, J.: Open Learner Models as Drivers for Metacognitive Processes, R. Azevedo, V. Aleven (eds), *Int. Handbook Metacognition & Learning Technologies* (in pr.)

Do Your Eyes Give It Away? Using Eye Tracking Data to Understand Students' Attitudes towards Open Student Model Representations

Moffat Mathews, Antonija Mitrovic, Bin Lin, Jay Holland,
and Neville Churcher

Intelligent Computer Tutoring Group, University of Canterbury,
Christchurch, New Zealand

moffat.mathews@canterbury.ac.nz,
tanja.mitrovic@canterbury.ac.nz

Abstract. There is sufficient evidence to show that allowing students to see their own student model is an effective learning and metacognitive strategy. Different tutors have different representations of these open student models, all varying in complexity and detail. EER-Tutor has a number of open student model representations available to the student at any particular time. These include skill meters, kiviath graphs, tag clouds, concept hierarchies, concept lists, and treemaps. Finding out which representation best helps the student at their level of expertise is a difficult task. Do they really understand the representation they are looking at? This paper looks at a novel way of using eye gaze tracking data to see if such data provides us with any clues as to how students use these representations and if they understand them.

Keywords: open student modelling, eye tracking, gaze tracking, intelligent tutoring systems, metacognition.

1 Introduction

A student model is how an Intelligent Tutoring System (ITS) views a student, or more precisely, views their domain knowledge. ITSs use this model to make pedagogical decisions for each student. The student model is not visible to the student. However, it has been shown that opening up the student model to the student, so that they could view “what the system thinks of them” is conducive to learning. In fact, the Open Student Model (OSM) plays quite a large role in increasing their metacognitive skills, which in turn helps their long-term learning [2,11]. Opening up the student model means that ITS authors have to consider how to best visualise this data so that the student can understand and make use of it.

As research continues in this area, there are now several new visualisations of the OSM, each giving different details, at different levels, and using different representations. Skill meters [2,11] have been used in a number of systems. Other types of OSMs include a tree structure [7,11], and concept graphs [5]. Most of these models are dynamic; others can be interactive; such as the negotiable student model

[15]. With so many proposed OSM representations, new questions now exist, such as, “Do students actually understand these representations?”, “Can we tell which representations they find easier to understand than others?”, or “Are certain representations better for certain populations? e.g. novices versus experts?”. If we, as ITS authors, could get the answers to these questions, we could 1) design better, more comprehensible representations, and 2) figure out which representations suit the particular student and guide them towards viewing that one. In this paper, we make an attempt at answering some of these questions for four of the representations contained in an ITS, namely EER-Tutor, in the hope that these methods could then be used to test other OSM representations.

Our method utilises eye (gaze) tracking in combination with test results to see if the student actually understood the model they were presented. Gaze tracking gives us an indication of where the student is looking and what they are paying attention to, while trying to understand the problem and the model. Gaze tracking has been used previously to find comparisons between novices and experts; e.g. during *Visual Flight Rules* flight [6], during laparoscopic surgery [8], working within a Learning Management System [14], while playing chess [3], and within collaborative environments [9]. It has been found to be a good indicator of the “Yes!” moments of delight while a student interacts with an ITS [13]. Gaze tracking data has also been used to supplement and change the underlying student model [4,10]. Bull, Cooke and Mabbott [1] found that students spend more attention on certain OSM representations for a reason, and that developers must take visual gaze attention into account when creating and presenting student models.

In this paper, we want to find out if gaze data gives us any information on how difficult a student finds and understands an OSM representation. For this, we had students viewing four different OSM representations and answering questions on each, while eye gaze data was recorded for each student. We looked at the scores of their answers and compared it to eye gaze data. We believe that if eye gaze data gives us information on how much difficulty a student is having with a particular model, we could, in time, incorporate eye gaze data to dynamically inform the pedagogical module of each student’s experience with a certain OSM representation. The score in that future case would be the knowledge score taken from the student’s model. The tutor can then intervene and present them with other options of OSM representations.

2 Design and Methodology

Seventeen participants took part in this study. They were all students who belonged to a second-year database course at the University of Canterbury. Each participant was given a NZ\$20 voucher on completion of the study. The ITS chosen for this study was EER-Tutor [12] and the study was conducted using a Tobii TX300¹ (300Hz) eyetracker. Each participant took part in the study separately.

¹ <http://www.tobii.com/en/eye-tracking-research/global/products/hardware/tobii-tx300-eye-tracker>

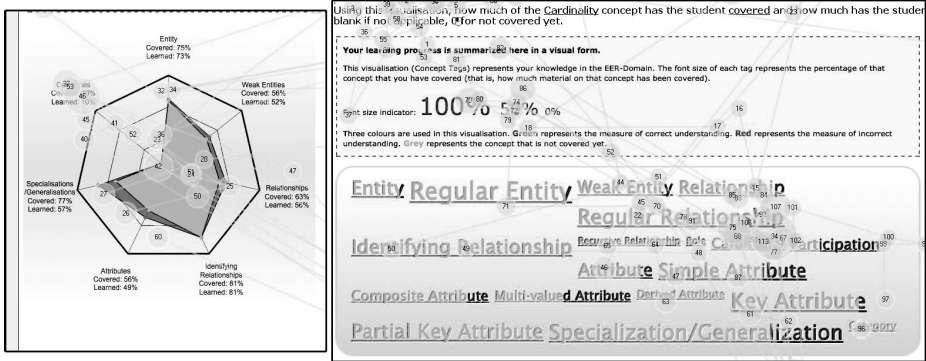


Fig. 1. Gazeplot for the kiviart and concept tag graphs

EER-Tutor is a standard tutor that students use in the lab sessions of this course; the lab sessions occur after the relevant lectures on the topic at hand. All participants had logged into EER-Tutor once during the first EER lab and completed the pretest. The version used in the course was similar in all respects to that used in the study, except that it only had one OSM representation: the skill meter.

Each participant took approximately an hour to complete the study. After the initial formalities of the study (information, consent form, etc.), each participant was asked to spend twenty minutes on the evaluation version of EER-Tutor. This version had all four OSM representations: kiviart chart; concept tags; concept hierarchy; and treemap. Participants were instructed to try solving problems, but to mainly focus on understanding each of the representations. During these twenty minutes, there were no restrictions put on the participants; they could solve as many or as few problems as they wished, as long as they focused on understanding the OSM representations.

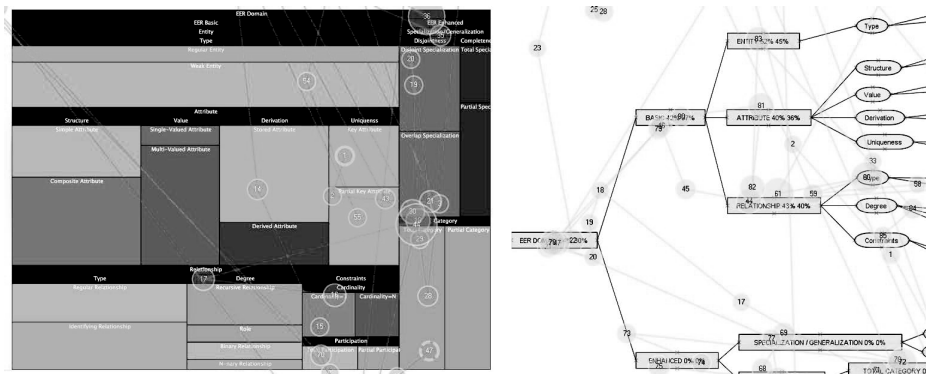


Fig. 2. Gazeplot for the treemap and the concept hierarchy

After the twenty minute session, the participants were automatically redirected to a web survey that we created, where they were asked questions about each of the OSM representations. To keep the eye gaze data clean and separated, all questions relating to a particular OSM representation were on the same page; the eyetracker generated new eye gaze plots for each page. Each page had a different representation of a pre-made model; all participants received the same pre-made OSMs. There were three questions directly related to each OSM and the understanding of the OSM in terms of the domain. As an example of a question, participants were asked to view an OSM and answer how much the student (represented by the OSM) had learnt and covered for a particular concept. Each question then could be given a score and participants were marked accordingly. At the end of the questionnaire there were two unmarked questions where participants could: 1) give general feedback and 2) rank their preference of the OSMs in the context of learning. Once a participant moved on from a page, they could not go back and change their answers. Participants were allowed to see their gaze data after completion of the evaluation study.

We have included cropped figures to show examples of the gazeplots from one question for one participant for each of the OSMs; Fig. 1 for the kiviak graph and the tag clouds, Fig. 2 for the concept hierarchy, and the treemap. Each node in the gaze data is a fixation. The longer the fixation, the bigger the node.

Our idea for this research was to find if eye gaze data added any value to figuring out how quickly and efficiently a student understood a particular OSM. If eye gaze data could be used in such a manner, then ITSs in future could track a student using one version of the OSM, figure out if they are having difficulty with it, and then intervene in some way, such as presenting them with a different OSM.

3 Results and Discussion

We defined a new variable called *OSM Efficiency*. The more efficient someone was at understanding a particular model, the higher their score would be. They would also be able to understand the OSM in less time with fewer fixations; experts take fewer fixations than novices to complete a task [6,9]. With this logic, we came up with our equation for OSM Efficiency, which is given in Equation 1.

$$OSM\ Efficiency = \frac{Score}{Time \times Number\ of\ fixations} \quad (1)$$

An expert marked the answers to the OSM questions according to the marking schema and came up with a score for each OSM. The time and number of fixations were extracted from the eyetracker.

We used repeated measures ANOVA and found a significant difference between the efficiencies in the OSM groups ($F(3, 42) = 43.567, p > .05$). To find out which groups were significantly different from each other, we conducted a Bonferroni post-hoc test. There is a significant difference between the kiviak graph and two other OSMs (tag cloud and treemap). There is no difference

between kiviatic graph and concept hierarchy. Similarly, there is a significant difference between concept hierarchy and two other OSMs (tag cloud and treemap). There is no difference between tag cloud and treemap.

This shows that participants were on average more efficient (with our definition of efficiency) using the kiviatic graph and the concept hierarchy, but had difficulties understanding and answering questions using the tag cloud and treemap representations.

However, how does this match up with participants' attitudes towards the OSM representations? In our questionnaire, we asked participants to rank the OSMs according to their preference in a learning context.

There was a statistically significant difference in the rankings of the OSMs ($\chi^2(3) = 17.118, p = 0.001$). Post-hoc analysis with Wilcoxon Signed-Rank Tests was conducted with a Bonferroni correction applied, resulting in a significance level set at $p < 0.0125$. Median (IQR) ranking levels were 1.0 (1.0 to 2.5) for kiviatic graph, 3.0 (2.0 to 4.0) for tag clouds, 2.0 (2.0 to 3.0) for concept hierarchy, and 4.0 (3.0 to 4.0) for treemap. There was a statistically significant difference in tag cloud vs. kiviatic graph rankings ($Z = -2.545, p = .011$), and in treemap vs. kiviatic graph rankings ($Z = -3.103, p = 0.002$).

The comments' question gave participants a chance to tell us about their experience with the OSMs. Many agreed that the Kiviatic Chart was best for an overall and quick indication of their levels but the other representations had their uses if more information was required. This led to the conclusion that the best OSMs depend on the context of the situation.

ITS designers are becoming more creative with their OSM designs. There has to be a method of testing between the various OSMs rather than just assuming that all OSMs are easy to understand. In this paper, we were able to compare four OSMs and found significant differences between them in terms of efficiency. This efficiency took into account the participant's score, their time for fixations, and their number of fixations. Future ITSs would gather the student's knowledge score (from the student model) instead of a questionnaire score to determine if the student is having difficulties. We compared this with subjective questionnaires that the participants had submitted rating their preference for each of the OSMs. There were significant commonalities between the efficiencies and the preferences. Furthermore, we manually analysed their comments and found that their attitude towards the OSMs were significantly similar to both the efficiencies and preferences. Following on from the background research and the participants' comments, we wonder if there would be a difference between different groups of students (say, novices versus experts). Novices might be interested in an easy to understand smaller OSM, while experts might want further detail and not be content with the smaller OSMs. We also found that eye tracking can play a large role in automatically understanding how the student is feeling towards each OSM. This could later on be harnessed with ITSs to present students with different OSMs when the ITS notices that they are struggling using their eye gaze data.

References

1. Bull, S., Cooke, N., Mabbott, A.: Visual Attention in Open Learner Model Presentations: An Eye-Tracking Investigation. In: Conati, C., McCoy, K., Paliouras, G. (eds.) UM 2007. LNCS (LNAI), vol. 4511, pp. 177–186. Springer, Heidelberg (2007)
2. Bull, S., Quigley, S., Mabbott, A.: Computer-based formative assessment to promote reflection and learner autonomy. *Engineering Education: Journal of the Higher Education Academy Engineering Subject Centre* 1(1), 8–18 (2006)
3. Charness, N., Reingold, E.M., Pomplun, M., Stampe, D.M.: The perceptual aspect of skilled performance in chess: evidence from eye movements. *Memory & Cognition* 29(8), 1146–1152 (2001)
4. Conati, C., Merten, C., Muldner, K., Ternes, D.: Exploring Eye Tracking to Increase Bandwidth in User Modeling. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 357–366. Springer, Heidelberg (2005)
5. Dimitrova, V.: Style-olm. *Artificial Intelligence in Education* 13(2), 35–78 (2003)
6. Kasarskis, P., Stehwien, J., Hickox, J., Aretz, A.: Wickens: Comparison of expert and novice scan behaviors during vfr flight. In: 11th Int. Symposium on Aviation Psychology (2001)
7. Kay, J.: Learner know thyself: Student models to give learner control and responsibility. In: Proc. of the Int. Conf. on Computers in Education, pp. 17–24. AACE, Charlottesville (1997)
8. Law, B., Atkins, M.S., Kirkpatrick, A.E., Lomax, A.J.: Eye gaze patterns differentiate novice and experts in a virtual laparoscopic surgery training environment. In: Proc. 2004 Symposium on Eye Tracking Research & Applications, ETRA 2004, pp. 41–48. ACM, New York (2004)
9. Liu, Y., Hsueh, P.Y., Lai, J., Sangin, M., Nussli, M.A., Dillenbourg, P.: Who is the expert? analyzing gaze data to predict expertise level in collaborative applications. In: IEEE Int. Conf. on Multimedia and Expo., pp. 898–901 (2009)
10. Merten, C., Conati, C.: Eye-tracking to model and adapt to user meta-cognition in intelligent learning environments. In: Proc. 11th Int. Conf. Intelligent User Interfaces, IUI 2006, pp. 39–46. ACM, New York (2006)
11. Mitrovic, A., Martin, B.: Evaluating the effect of open student models on self-assessment. *Artificial Intelligence in Education* 17(2), 121–144 (2007)
12. Mitrovic, A.: Fifteen years of constraint-based tutors: what we have achieved and where we are going. *User Modeling and User-Adapted Interaction*, 1–34 (2011)
13. Muldner, K., Burleson, W., VanLehn, K.: “Yes!”: Using Tutor and Sensor Data to Predict Moments of Delight during Instructional Activities. In: De Bra, P., Kobsa, A., Chin, D. (eds.) UMAP 2010. LNCS, vol. 6075, pp. 159–170. Springer, Heidelberg (2010)
14. Pretorius, M., van Biljon, J.: Learning management systems: Ict skills, usability and learnability. *Interactive Technology and Smart Education* 7(1), 30–43 (2010)
15. Thomson, D., Mitrovic, A.: Preliminary evaluation of a negotiable student model in a constraint-based its. *Research and Practice in Technology Enhanced Learning (RPTEL)* 5(1), 19–33 (2010)

Fuzzy Logic Representation for Student Modelling*

Case Study on Geometry

Gagan Goel¹, Sébastien Lallé², and Vanda Luengo²

¹ Electronics and Communication Engineering Department, National Institute of Technology
(NIT) Hamirpur, H.P. - 177005, India
gagan.nith@gmail.com

² Laboratoire Informatique de Grenoble (LIG METAH), Université Joseph Fourier,
110 av. de la Chimie, BP 53, 38041, Grenoble Cedex 9, France
{sebastien.lalle,vanda.luengo}@imag.fr

Abstract. Our aim is to develop a Fuzzy Logic based student model which removes the arbitrary specification of precise numbers and facilitates the modelling at a higher level of abstraction. Fuzzy Logic involves the use of natural language in the form of If-Then statements to demonstrate knowledge of domain experts and hence generates decisions and facilitates human reasoning based on imprecise information coming from the student-computer interaction. Our case study is in geometry. In this paper, we propose a fuzzy logic representation for student modelling and compare it with the Additive Factor Model (AFM) algorithm implemented on DataShop. Two rule-based fuzzy inference systems have been developed that ultimately predict the degree of error a student makes in the next attempt to the problem. Results indicate the rule-based systems achieve levels of accuracy matching that of the AFM algorithm.

Keywords: Student model, fuzzy inference system, rule-base.

1 Introduction

Student Model is one of the primary components of an Intelligent Tutoring System (ITS). Our objective here is to study one of the AI approaches (fuzzy logic) for the conception of these kinds of models. Our methodology emphasizes the collection of real-world data for evaluating and comparing the model. Building student models is a complex and intractable task, as seen in [1]. Students pose the real challenge to a tutoring system in the sense that it is very difficult to study their minds and hence extract information under different circumstances. Moreover, recent approaches to develop an effective student model have lacked in one way or the other. Specifically, we will consider the case of Additive Factor Model (AFM) algorithm, [2], which performs the knowledge diagnosis of the student by predicting the error rate. It has

* This work has been granted by the Rhône-Alpes Region in France.

been graphically shown on the learning curve diagrams on Datashop that the actual values of Error rate and the values predicted by AFM sometimes differ significantly. Later in the paper, we will compare the results obtained with the fuzzy inference systems with the AFM predicted values as well as with the actual error rate values coming directly from the student-computer interaction.

Fuzzy logic is an AI technique that involves the use of natural language in the form of If-Then rule paradigms which allows the modelling of complex systems using a higher level of abstraction. The main advantage of using fuzzy logic is that humans often reason in terms of vague concepts when dealing with situations in which they experience uncertainty, [3]. Hence we go for a technique that effectively maps the subjective concepts such as skilled, unskilled, average etc. (when talking about a student's skill level) into numerical values with the help of membership function curves.

2 Related Previous Work

In [4], the Brilliant Scholar Series 1 (BSS1) tutoring system has been designed based on fuzzy logic techniques. This way, it has improved the performance of the system by introducing intelligent features which can better manage the student's learning like monitoring the student's progress, trends in performance etc. A general fuzzy logic engine has been designed and implemented to support development of intelligent features for BSS1. Again in [4], it has been shown that a fuzzy logic based system offers the flexibility to manipulate the system as per the designer's need, for instance, by modelling the problem suitably, defining fuzzy variables and suitable membership functions for their fuzzy sets, and developing a comprehensive set of rules relating input and output variables.

3 The Proposed Student Model

We make use of the student-computer interaction data available on Datashop, described in [5], which is an online repository of data-sets coming from different Intelligent Tutoring Systems covering a wide variety of domains. Our approach involves the design of a student model based on Knowledge Tracing, [6], and fuzzy inference using If-Then statements for the development of the rule-base. Two rule-based systems have been designed, one for the diagnosis of student knowledge i.e. Knowledge Component (KC) diagnosis and other (using the first rule-base) for the prediction of a parameter for student performance i.e. Error Rate. Learning curves have been used that visually present measures of student performance. We have considered data of Geometry Cognitive Tutor 1996 with Geometry Area (1996-1997) as the Dataset accessed via DataShop which was tested on 59 students with an existing KC model ("Original"), [7].

3.1 Knowledge Component (KC) Diagnosis

The amount of learning that a student acquires in various concepts of geometry is an important measure for the student knowledge diagnosis, so a rule-base for the

prediction of knowledge component level has been designed. In geometry domain, a student progressing through various problems encounters a total of 15 KCs when Original KC model is considered. Here, we will only consider the diagnosis of Parallelogram Area KC (given the base and height, the student is able to find the area of a parallelogram). This consideration has been generalized for the diagnosis of remaining KCs.

Input-Output Consideration and Membership Function (MF) Curves. KC diagnosis is a 3-input 1-output rule-base. The inputs are the probability that student knows the KC, the Opportunity Count (OC) and the Outcome (correct or incorrect answer) at time t . The single output considered is the probability that the student knows the KC at time $t+1$. With ParallelogramArea KC as an output, we can infer about the student skill level or the gaps in his knowledge about some concepts. Diagnosis results of this KC from previous step serve as an input for the KC diagnosis for current step. This information considers the fact that the current level of the student knowledge about a particular concept also depends on his previous knowledge about that concept.

Membership Function curves represents linguistic levels (non-numeric variables such as skilled, unskilled, average etc.) that a fuzzy variable can take. We have considered Universe of Discourse for ParallelogramArea from 0 to 100 as it is in terms of percentage of the KC learnt. Fig. 1 shows that a total of 7 linguistic levels have been considered for ParallelogramArea. Here, for example, the linguistic level “Above Average” has a Triangular curve with its range from 45 to 95. Opportunity Count and Outcome for the KC have 3 and 2 linguistic levels; Low, Medium, High, and Incorrect, Correct respectively.

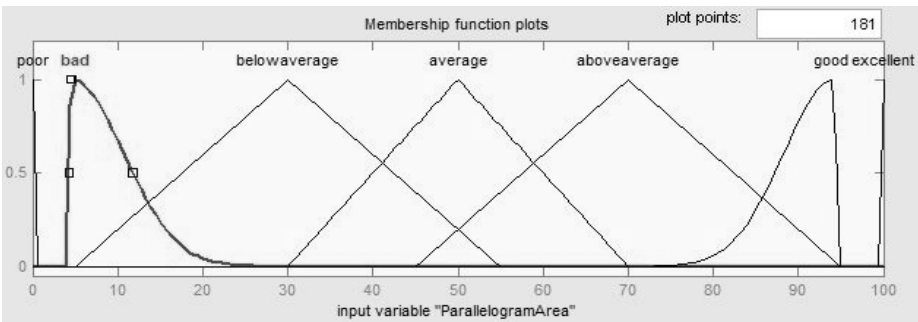


Fig. 1. Membership Function curves for ParallelogramArea

Rule-Base. For KC diagnosis, 24 If-Then rules are developed for the 3-input 1-output system. These rules allow inferring the value of the output.

Sample Rule. If ParallelogramArea is Average and OCParallelogramArea is Medium and OutcomeParallelogramArea is Correct, then ParallelogramArea is Above Average.

At Medium OC, if the student makes correct attempt with an Average level of ParallelogramArea (from previous step), then his knowledge about this KC will rise to Above Average. Following this procedure, remaining rules can also be interpreted.

3.2 Error Rate Prediction

We use the KC diagnosis for the prediction of error rate i.e. to predict about the probability that a student makes an error on a step. This will help us to compare our results with the actual values and also with the values predicted by AFM algorithm. The intuition of AFM is that the probability of a student getting a step correct depends on the response of the student on a step, the amount of knowledge that the student possesses, the difficulty level of the KC, the skill level of student, and the amount of learning gained for each practice opportunity.

Input-Output Consideration and Membership Function Curves. Considering the variables and intuition of AFM, an analogy is applied that results in a 3-input 1-output rule-base for error rate prediction. The output considered here is the ErrorRate and the inputs are KC (Knowledge Component), Student (Skill Level of Student), and KC-DifficultyLevel (Difficulty Level of KC). KC diagnosis rule-base considers outcome of ITS and OC as two of its inputs, so we take the inferred KC level (from first rule-base) as an input. This reduces the number of input variables for error rate prediction (as compared to the number of variables in AFM). 3 linguistic levels are taken both for Student and KC-DifficultyLevel inputs; Skilled, Average, Unskilled, and Easy, Medium, Hard respectively. For the ErrorRate output, 7 linguistic levels are considered; Very Low, Low, Below Medium, Medium, Above Medium, High, Very High.

Rule-Base. 31 If-Then rules are developed for the prediction of error rate for the 3-input 1-output fuzzy inference system. The rules are developed on the basis of learning curve plots. Two such curves for error rate and assistance score are taken. On seeing the Assistance Score learning curve plot, we find those OCs which correspond to Average KC. Then values for actual error rate are computed from its learning curve plot and its average is taken for all the OCs under consideration. This mean value is then mapped to a relevant level of error rate using its MF curves and the inferred level obtained this way is assigned as the output to this rule.

Sample rule. If KC is Average and Student is Unskilled and KC-DifficultyLevel is Hard, then ErrorRate is High.

4 Results

For Fig. 2, error rates (as computed with the fuzzy inference) for all students with every KC are recorded individually. Then, all the readings are grouped so that we get values of Actual error rate, predicted error rate by AFM, and the error rate as computed by the rule-based system. Observations are then plotted as a function of Opportunity Count present in DataShop traces. Figures 2-4 show a good accuracy of the prediction; the rule-based system RMSE (Root Mean Square Error) in general is rather close to that of AFM, on Fig. 2. In particular, Table 1 indicates for Figures 2-4 the correlation between the actual error rate and the AFM prediction on one hand, between the actual error rate and the rule-based system on the other hand. The correlation is again significantly good as compared to AFM. However, our results present over-fitting issues, as we have not yet used cross-validation (i.e. to use a part of the data to train and the rest to test the model). This point is discussed in more detail in the conclusion.

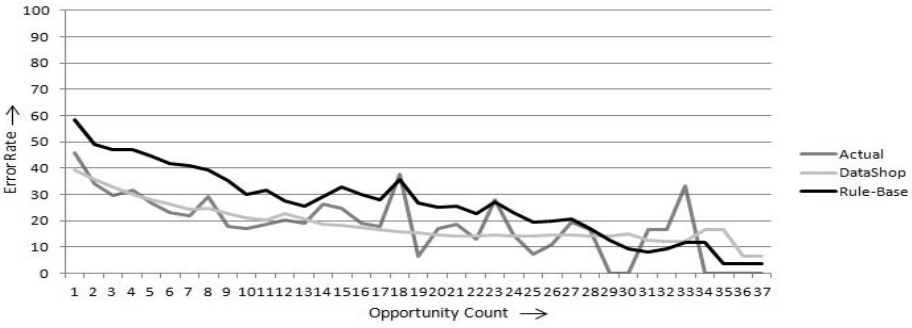


Fig. 2. General plot for all KCs and all students. Rule-Based System RMSE: 0.18635, AFM RMSE: 0.13447

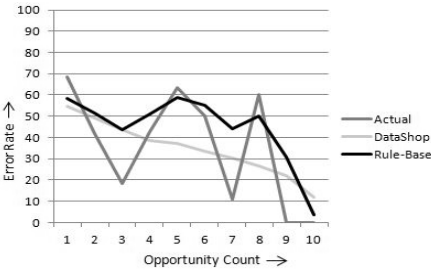


Fig. 3. Plot for Trapezoid-Base KC and all students

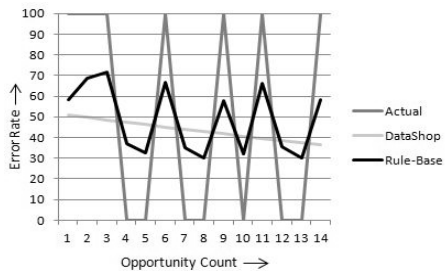


Fig. 4. Specific plot for Circle-Radius KC and Stu_0a8e3638e3c0deb4e5e49c72286

Table 1. Corresponding Pearson Correlation Coefficient (PCC) values for above plots

Fig. No	2	2	3	3	4	4
Pair of Curves	Actual-DataShop	Actual-Rule-base	Actual-DataShop	Actual-Rule-base	Actual-DataShop	Actual-Rule-base
PCC Value	0.662	0.785	0.616	0.817	0.234	0.966

5 Conclusions and Perspectives

With the fuzzy logic representation for student modelling, we have developed a student model that respects the process of knowledge tracing. This model can handle data at a higher level of abstraction and it also has the ability to deal with uncertainty. Moreover, fuzzy logic needs fewer parameters (in comparison to AFM) and this facilitates modelling with continuous variables (e.g. the membership function curves for the fuzzy variables). As the inputs and the rules are particularly comprehensible for humans due to the linguistic levels expressed in natural language, it is quite easy to adapt and refine the model (for instance by experts). In the past, knowledge tracing

has been implemented with Hidden-Markov model and Logistic regression. Here, the main contribution of this paper is the design of a cognitive student model based on Fuzzy Logic. The expressive power of fuzzy inference is comparable to full Bayesian inference, but it requires fewer parameters due to the continuous Membership Functions. Moreover, the structure of this fuzzy logic model is not dependent on our domain, so it can be reused by ITS designers for another work/domain/ITS as long as the KC Model is developed. Determination of the parameters (the thresholds of each membership function) may be done either by experts or by machine learning algorithms. As said previously, validating the proposed student model in a more formal way is a crucial perspective. Our results show some over-fitting and a lack of precision in the beginning, so constructing the model with machine learning techniques is also important, this would help us improve the accuracy of the model during the initial stages. As a perspective, methods like training the model with real data (bagging algorithms) may help to overcome this issue. We also plan to compare our model with other student models (in a specific domain) both from cognitive sciences and AI, consider for example [8].

References

1. Self, J.A.: Formal Approaches to Student Modelling. Technical Report No. 92. In: McCalla, G.L., Greer, J. (eds.) *Student Modelling: the Key to Individualized Knowledge-Based Instruction*, pp. 295–352. Springer, Berlin (1994)
2. Cen, H., Koedinger, K.R., Junker, B.: Comparing Two IRT Models for Conjunctive Skills. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) *ITS 2008. LNCS*, vol. 5091, pp. 796–798. Springer, Heidelberg (2008)
3. Zadeh, L.: The role of fuzzy logic in the management of uncertainty in expert systems. *Fuzzy Sets and Systems* 11, 199–227 (1983)
4. Warendorf, K., Jen, T.S.: Application of Fuzzy Logic Techniques in the BSS1 Tutoring System. *Journal of Artificial Intelligence in Education* 8(1), 113–142 (1997)
5. Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J.: A Data Repository for the EDM community: The PSLC DataShop. In: Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (eds.) *Handbook of Educational Data Mining*, pp. 43–56. CRC Press, Boca Raton (2010)
6. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction* 4, 253–278 (1995)
7. Cen, H., Koedinger, K., Junker, B.: Is Over Practice Necessary? - Improving Learning Efficiency with the Cognitive Tutor through Educational Data Mining. In: Luckin, R., Koedinger, K.R., Greer, J. (eds.) *The 13th International Conference on Artificial Intelligence in Education (AIED 2007)*, pp. 511–518 (2007)
8. Lallé, S., Luengo, V., Guin, N.: An Automatic Comparison Between Knowledge Diagnostic Techniques. In: Cerri, S.A., Clancey, B. (eds.) *ITS 2012. LNCS*, vol. 7315, pp. 623–624. Springer, Heidelberg (2012)

Content Learning Analysis Using the Moment-by-Moment Learning Detector

Sujith M. Gowda¹, Zachary A. Pardos², and Ryan S.J.D. Baker¹

¹ Department of Social Science and Policy Studies,
Worcester Polytechnic Institute, Worcester, MA USA

² Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA USA
{sujithmg, zpardos, rsbaker}@wpi.edu

Abstract. In recent years, it has become clear that educational data mining methods can play a positive role in refining the content of intelligent tutoring systems. In particular, efforts to determine which content is more and less effective at promoting learning can help improve tutoring systems by identifying ineffective content and cycling it out of the system. Analysis of the learning value of content can also help teachers and system designers create better content by taking notice of what has and has not worked in the past. Past work has looked solely at student response data in doing this type of analysis; we extend this work by instead utilizing the moment-by-moment learning model, P(J). This model uses parameters learned from Bayesian Knowledge Tracing as well as other features extracted from log data to compute the probability that a student learned a skill at a specific problem step. By averaging P(J) values for a particular item across students, and comparing items using statistical testing with post-hoc controls, we can investigate which items typically produce more and less learning. We use this analysis to evaluate items within twenty problem sets completed by students using the ASSISTments Platform, and show how item learning results can be obtained and interpreted from this analysis.

Keywords: Educational data mining, item sequencing, learning gains.

1 Introduction

The last several years have begun to see a shift in the sources of intelligent tutor content. As recently as five years ago, most intelligent tutor content was authored in programming development kits, and took considerable work to create – according to one estimate, it takes over 200 hours of a Ph.D.-level researcher’s time to create one hour of student-usable content [16]. However, the recent advent of tools for rapid problem authoring by non-programmers [cf. 1, 13] has begun to change this practice. In fact, some intelligent tutoring systems are being authored via crowd-sourcing methods, where a wide range of individuals can contribute problems and content. For example, in the ASSISTments Platform [10], many problems and associated tutoring for those problems are now authored by teachers.

The move toward a wider base of content developers presents both opportunities and challenges. A wider developer base enables new content to be created more quickly and more responsively than traditional approaches. However, assuring and maintaining quality is a greater challenge when content is being created by a wider range of individuals, many of whom do not have explicit training in creating intelligent tutoring systems. (Though this is an opportunity in itself, as some teachers may have innovative new ideas for problem content that are better than current approaches). Also, as community-authored content grows rapidly, it is not feasible for small research teams to continually vet new content.

Given rapidly expanding content of uncertain quality, one approach to assuring and maintaining quality is to use educational data mining to vet content. The data produced by students as they use a tutoring system can provide indicators of which problems are most effective. Work in this area can build off of prior approaches to determine which pedagogical strategies lead to better learning experiences for students. For example, Beck and colleagues [6] used learning decomposition methods to study the effectiveness of different learning strategies for different groups of students. Chi and VanLehn [7] used reinforcement learning to study this same issue.

The approach proposed in [6] was adopted by Feng et al. [11], who used learning decomposition to determine that problems had varying efficacy within the ASSISTments Platform. This approach used logistic regression to analyze the future performance associated with having received a specific problem. Similarly, Pardos and Heffernan [17] addressed this same issue with a model based on Bayesian Knowledge-Tracing. Pardos et al. showed that models based this framework could be modified to measure the learning probability of individual items within particular knowledge components (KCs). Pardos suggested that item learning effects can be measured so long as the order of the items within a KC is randomized per student. Given randomization of item order, the sets of items can be analyzed as a quasi-randomized controlled trial.

These approaches provide actionable information on which problems are most effective and least effective. However, they are somewhat limited in terms of their sensitivity. First, assessments of problem effectiveness are dependent on performance in immediately subsequent problems; if those problems are of varying difficulty, there may be substantial noise in estimations of learning effectiveness. In addition, correctness does not take into account all of the information about a student action; other aspects of student performance have also been shown to predict knowledge and learning [cf. 9].

To address this possible limitation and create a richer indicator of the differential learning associated with different problems, we adopt an alternate paradigm for measuring learning: the moment-by-moment learning model [4]. This model is designed to specifically assess the learning that occurs within a specific problem. Instead of assessing the current degree of latent knowledge, it assesses the degree of knowledge learned at a specific moment using a function of the aspects of the student's actions on that problem (such as speed of response and use of help features).

In this paper, we apply the moment-by-moment learning model to a group of problem sets from the ASSISTments Platform. We then conduct statistical analysis to

determine the degree to which different problems have different moment-by-moment learning across students, and study the problems associated with the largest and smallest degree of moment-by-moment learning in two data sets.

2 Data

The data used in this analysis comes from the ASSISTments Tutoring Systems [10], with data drawn from the 2009-2010 school year. The students were from 7th and 8th grade Algebra classes with ages 12-14. The 8,519 students in the data set were drawn from 108 schools, primarily in Massachusetts. Students used the software for one class day approximately every two weeks throughout the school year, completing a range of problem sets involving different mathematical skills. The system provided instructional assistance to troubled students by breaking the original problem into scaffolding steps or displaying hint messages on-screen, upon student request. The ASSISTments tutoring system allows teachers to control the ordering of the problems within a problem set, choosing between a pre-chosen order, or random order. In this paper, we analyze a subset of the data drawn from students using random order problems within a problem set, selecting only problems that are associated with at least one cognitive skill.

There were a total of 78,558 student actions, made by 3,169 students on 1,170 problems, for whom the problem order was set to random and each problem was associated with at least one skill. There were some problems that were associated with more than one skill. For these problems, we treated them as representing evidence for each skill equally and with full credit assignment to each skill (i.e. a problem with three skills was treated the same as three problems, one tied to each of the three skills). Within the data set, there were a total of 945 skill-problem sets, out of which we selected 20 skill-problem sets that had the highest number of student actions, giving a final data set with 20,760 student actions produced by 2,210 students on 80 problems.

3 Detecting Learning Using Moment-by-Moment Learning Model

In this section we describe the moment-by-moment learning model developed by Baker and colleagues [4]. This model estimates the probability that a student learned a skill at a specific problem step, termed $P(J)$. Recent results have argued in favor of this model's face validity; derivatives of this model can successfully predict students' final knowledge as assessed by Bayesian Knowledge Tracing [4], and can successfully predict students' preparation for future learning [5]. Bayesian Knowledge-Tracing (BKT) is a well-established approach for modeling student knowledge within an intelligent tutoring system [8]. BKT uses a four-parameter two-node dynamic Bayesian network to probabilistically assess the knowledge of a student for a specific skill. We use $P(J)$ values in this analysis to assess the amount that students typically learn from each problem within a randomly ordered problem set.

3.1 Development of P(J) Model

The P(J) model was developed using a two-step process, the same procedure used in [4]. First, training labels to detect moment-by-moment learning were generated for each problem step in a tutor data set. The labels were generated by applying Bayes' Rule to knowledge estimates from a traditional BKT model, in combination with the information about the correctness of the next two problem-solving actions of the student on items involving the same skill. Next, a set of predictor features was generated using past tutor data to form a training data set. The predictor feature set included 4 categories of features: 1) Action correctness, this category included features like is the action correct, incorrect or hint request, 2) Step interface type included feature that are based on type of interface widget involved, like is the problem multiple choice or just a single choice, 3) Response times, this categories included features that are derived from the amount of time taken to complete problem-solving steps, and 4) Problem solving history included features that characterize the student's problem-solving history in the tutor. These predictor features date back to the development of "gaming the system" detectors for Cognitive Tutors [3]. In addition to these features, skill difficulty related features were also included to increase the goodness of the model [12]. Linear regression was conducted within Rapidminer 4.6 [15] to develop models to predict P(J). This resulted in a set of numerical predictions of P(J), moment-by-moment learning, for each problem-solving step. The cross-validated correlation between the model and the original training labels was 0.449.

4 Overall Comparison of Problem Effectiveness

With the outputs of the P(J) detector, it is possible to assess the learning effectiveness of each problem in each skill-problem set. We do so by obtaining the set of values of P(J) for each problem, across students. We can then search for particularly poor problems and particularly effective problems. We analyze this in two ways. First, we conduct a one-way ANOVA to determine whether there are overall differences in the mean value of P(J) between problems in the same skill-problem set. Next, we attempt to determine if each skill-problem set has a single problem that is either better or worse than all other problems in the skill-problem set, an indicator that this problem is particularly effective or ineffective. It should be noted that the P(J) value is capturing the combined learning value of the problem and its tutoring (scaffolds and hints). Results are summarized in Table 1.

We found that 12 sets out of 20 skill-problem sets had statistically significant differences in learning between problems. Within these 12 skill-problem sets, we studied whether there was a best and worst problem, using post-hoc methods. The Levene test [14] was used to determine if the P(J) values for each problem in a skill-problem set had equal variance or not, to avoid violating the assumptions of the post-hoc analysis methods. Tukey's test was used when equal variance was assumed, and Tamhane's T2 test was used when equal variance was not assumed. Given the post-hoc differences between problems, a problem was labeled a best problem if it had positive mean difference with all the other problems and was significantly different from all the other problems in the skill-problem set. Similarly, a problem was labeled a worst

problem if it had negative mean difference with all the other problems and was significantly different from all the other problems in the skill-problem set. According to this test, 7 of the 12 problem sets had a single problem that was substantially better or worse than all other problems.

Table 1. ANOVA results of 20 skill-problem sets. ** = statistical significance of $p < 0.05$.

Skill-Problem Set	Total Actions	Best Problem	Worst Problem	F-test
ConversionOfFraction- DecimalsPercents	867	---	---	$F(1, 865) = 0.22$
CountingMethods	752	No	Yes	$F(2, 749) = 12.64^{**}$
Estimation	510	---	---	$F(2, 507) = 0.52$
FindingFractionsandRa- tio	849	Yes	Yes	$F(1, 847) = 8.28^{**}$
HistogramasTableOr- Graph	481	---	---	$F(2, 478) = 1.721$
LineOfBestFit	713	---	---	$F(3, 709) = 1.60$
Median	612	Yes	No	$F(2, 609) = 9.76^{**}$
MultiplicationandDivi- sionIntegers	850	No	No	$F(7, 842) = 28.16^{**}$
NumberLine	864	---	---	$F(1, 862) = 2.89$
PercentOf	1703	No	No	$F(7, 1695) = 84.85^{**}$
PickingEquationandEx- pressionFromChoices	535	---	---	$F(3, 531) = 0.54$
PointPlotting-1	868	Yes	Yes	$F(1, 866) = 8.59^{**}$
PointPlotting-2	520	---	---	$F(1, 518) = 0.99$
Proportion-1	1220	Yes	Yes	$F(1, 1218) = 8.59^{**}$
Proportion-2	2716	Yes	No	$F(4, 2711) = 41.47^{**}$
Proportion-3	1056	No	No	$F(4, 1051) = 33.06^{**}$
PythagoreanTheorem	2174	No	No	$F(12, 2161) = 6.65^{**}$
Range	810	No	Yes	$F(2, 807) = 16.44^{**}$
Transformation	878	No	No	$F(7, 870) = 5.30^{**}$
UnitConversionWithin-a- System	595	---	---	$F(1, 593) = 0.80$

4.1 Case Study of Individual Problems and Their Tutoring

Of the 20 skill-problem sets, there were seven problems that were significantly better or worse than all other problems in the same skill-problem set. These seven are shown in Table 2. Since the learning value of an item is a latent measurement, we have no ground truth to compare it to in order to verify that the best or worst items detected

were in fact the correct ones. Instead, we present the problems chosen as best and worst as dictated by the P(J) item learning detector and see if the results have face validity, that is if the detector looks like it measured what we intended for it to measure. Due to space limitations, we focus on two skill-problem sets, comparing a problem that is significantly different from all other problems with another problem from the set. We select the two skill-problem sets among the four possible options that have the largest difference in P(J) between the problems with the highest and lowest P(J). To facilitate discussion of differences, we compare the significantly different problems to the problem at the other end of the range. Within the PDF version of this document, the reader can inspect the problems, by clicking on any of the IDs in Table 2. The hyperlinks lead to a public preview of the items on the ASSISTments system.

Table 2. skill-problem sets with significant learning items


Problem set	Best item ID	Worst item ID	Mean difference between P(J) values
Proportion-2	15792	24642	0.0183
Range	27521	25796	0.0127
Counting Methods	24754	24752	0.0106
Median	1059	2239	0.0090
Proportion-1	15792	15844	0.0049
PointPlotting-1	12353	12354	0.0048
FindingFractionsandRatios	12375	12376	0.0038

4.2 Case study of Proportion-2's Best and Worst Problems

The skill-problem set Proportion-2 had the largest difference in P(J) between the best and worst problem among the four skill-problem sets with a significant best or worst problem, 0.018. In this skill-problem set, one problem had statistically significantly higher P(J) than all the other problems in the skill-problem set. The problems with the highest P(J) and lowest are shown in Figure 1.

Figure 1 shows the best problem (on the left) in the Proportion-2 skill problem set. This problem has a visual component (the figure of the triangle) and is multiple-choice. The choices contain possible fraction equalities and the student is asked to select the one that can be used to solve for X. The first hint shows the student that there is a small triangle within the larger one. The following three hints proceed to evaluate the three wrong choices and tell the student which part of the answer is wrong and why. The last hint shows the correct answer, explains why it is correct, and shows four other proportion equalities that would have also been correct. The total hint count in this problem is five. Due to space limitations, the figure only shows the first three hints. This highly effective problem has more than double as many hints as the comparison problem, and uses visuals and significantly more text to teach the concept of proportion. From this comparison, it is not immediately clear which of these differences is beneficial, but multiple hypotheses are now available for improving other problems in this skill-problem set. For problems with much lesser magnitude of P(J) difference, additional attributes of the problems and their help would likely need to be defined in order to tease out an explanation for the more subtle difference in learning.

Which proportion can be used to calculate x?




$A: \frac{x}{15} = \frac{3}{4}$ $C: \frac{x}{3} = \frac{19}{4}$
 $B: \frac{15}{4} = \frac{3}{x}$ $D: \frac{x}{19} = \frac{4}{3}$


Comment on this question

Here is a picture of the triangles separated. The red side of the big triangle corresponds to the red side of the little triangle and the green side of the large triangle corresponds to the green side of the small triangle.

Big Triangle



Small Triangle



Comment on this hint

Choice A looks OK because corresponding side are next to each other in the equation

$$\frac{x}{15} \rightarrow \frac{3}{4}$$

but 15 is not the length of the green side of the big triangle, so it cannot be the correct answer.

Comment on this hint

Choice B also has 15 in it, so it cannot be right either.

Comment on this hint

What is the value of x in this equation?

$$\frac{x}{12} = \frac{25}{50}$$

Comment on this question

To solve this problem, look for a relationship between the fractions, or within one fraction.

Comment on this hint

Look specifically at the relationship between 25 and 50. Then apply this same relationship to x and 12 to solve for x.

$$\frac{x}{12} = \frac{25}{50}$$

Comment on this hint

25 is half as big as 50.
X should also be half as big as 12.
X is 6.

Comment on this hint

Type your answer below:

Submit Answer

Fig. 1. Proportion 1: the (sig) best problem and its tutoring (left) and the worst problem (right)

The problem to the right in Figure 1 shows the worst P(J) problem, which asks the student to solve for X where X is part of a fraction equal to another fraction. The tutoring for the problem gives the student a first hint which suggests the student observe the relationship between the numerator and denominator of the fraction on the right side of the equation and apply this relationship to the fraction on the left side to determine X. The second hint explicitly tells the student the relationship between numerator and denominator which is that the numerator is half of the denominator of the fraction on the right side of the equation. The third hint is a bottom-out hint, and gives the student the answer.

4.3 Case study of Range, Best and Worst Problems

The skill-problem set Range had the second-largest difference in P(J) between the best and worst problem, 0.013. In this skill-problem set, one problem had statistically significantly lower P(J) than all the other problems in the skill-problem set.

Figure 2 shows the worst problem in this skill-problem set (on the right), which asks for the range of the points scored in the table. This problem contains three scaffolds that in turn prompt the student for the maximum and minimum scores observed, and then re-asks the original question. Each of the scaffolds contains two hints. The first hint suggests the student look at the table for the answer and the second hint shows another picture of the table with the relevant row highlighted. The total number of hint in this problem was six.

The problem to the left in Figure 2 shows the best problem, which also shows a two column table but asks which of four multiple choice statistics has the highest value. This problem has six scaffolds. The first prompts the student to count the

number of animals listed. The next four scaffolds teach the student how to compute the mean, median, mode, and range using the table in the problem. The last scaffold re-asks the original question. There are 20 hints in this problem.

The average life spans of some animals are shown in the chart below:

Animal Life Spans

Animal	Average Life Span (in years)
Bear	22
Chicken	7
Deer	12
Dog	11
Duck	10
Elephant	35
Fox	9
Horse	22
Hippopotamus	30
Wolf	11

Source: Farmer's Almanac, 2009.

Based on the information given in the chart, which of the following statistics yields the greatest numerical value?

Comment on this question

[Break this problem into steps](#)

Select one:

mean

median

mode

range

[Submit Answer](#)

The Patriots football team won the national championship in the 2003-2004 season. The table below shows the number of points scored by the Patriots in each of the team's games during the season.

Points Scored by Game

Game	Number of Points Scored
1st	0
2nd	31
3rd	23
4th	17
5th	38
6th	17
7th	19
8th	9
9th	30
10th	12
11th	23
12th	38
13th	12
14th	27
15th	21
16th	31

What is the range of the number of points scored?

Comment on this question

[Break this problem into steps](#)

Type your answer below (mathematical expressions):

[Submit Answer](#)

Fig. 2. Range: the best problem (left) and the (sig) worst (right)

Comparing the two problem's tutoring of the skill of range, there does not appear to be anything strikingly deficient about the significantly worst problem's tutoring. However, the most significant difference in the content between the worst and best problem is that the best problem contains three additional skills (mean, median, and mode) while the worst problem only contains range. A look at the Q-matrix for both problems revealed that the best problem was indeed tagged with four skills while the worst problem was tagged with only a single skill. Since $P(J)$ is computed based on the relative learning value of the problems in a set, it appears that $P(J)$ has detected a skill difference between problems. The tutoring of the problem that teaches and requires only the skill of range has little chance of providing the requisite knowledge to solve a problem that requires mean, median, mode, and range; however the four-skill problem has the tutoring to provide the requisite knowledge for the single-skill problem which would explain significant $P(J)$ difference between problems.

5 Discussion

We have shown how the moment of learning detector can be applied to evaluate the relative learning value of problems in a set and how statistical tests can be run to determine if there are problems which are significantly better or worse. We conducted a case study of problem pairs in two skill-problem sets which showed the most significant differences in $P(J)$ in order to investigate if differences could be plainly observed by viewing the problems and their tutoring approaches.

Several avenues exist for further research in the area of learning value analysis. Firstly, the method could be applied at the skill-problem set level to detect which problem sets pertaining to a common skill are providing the most learning value. This analysis would require a dataset where the order of problem sets, at least within a skill, were randomized per student. A second area for further study is a more stringent validity test. Face validity tests are subjective and fall far short of confirming that the claimed underlying construct is being accurately measured. A gold standard validity test would be a randomized controlled trial where individual problems were tested for learning gain with a pre/post-test design. The existence of a significantly higher or lower learning gain problem could be identified and compared to the findings of the P(J) learning value detector and other learning item analysis techniques.

Acknowledgements. We would also like to thank Lisa Rossi for valuable comments and suggestions. This research was supported by the National Science Foundation via the Pittsburgh Science of Learning Center, grant award #SBE-0836012, and by a “Graduates in K-12 Education” (GK-12) Fellowship, award number DGE0742503. We would like to thank the additional funders of the ASSISTments Platform found here: <http://www.webcitation.org/5ym157Yfr>.

References

1. Alevan, V., McLaren, B.M., Sewall, J., Koedinger, K.R.: The Cognitive Tutor Authoring Tools (CTAT): Preliminary Evaluation of Efficiency Gains. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 61–70. Springer, Heidelberg (2006)
2. Baker, R.S.J.d., Corbett, A.T., Alevan, V.: More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 406–415. Springer, Heidelberg (2008)
3. Baker, R.S.J.d., Corbett, A.T., Roll, I., Koedinger, K.R.: Developing a Generalizable Detector of When Students Game the System. *User Modeling and User-Adapted Interaction* 18(3), 287–314 (2008)
4. Baker, R.S.J.d., Goldstein, A.B., Heffernan, N.T.: Detecting Learning Moment-by-Moment. To appear in *International Journal of Artificial Intelligence in Education* (in press)
5. Baker, R.S.J.d., Gowda, S.M., Corbett, A.T.: Automatically Detecting a Student’s Preparation for Future Learning: Help Use is Key. In: *Proceedings of the 4th International Conference on Educational Data Mining*, pp. 179–188 (2011)
6. Beck, J.E., Mostow, J.: How Who Should Practice: Using Learning Decomposition to Evaluate the Efficacy of Different Types of Practice for Different Types of Students. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 353–362. Springer, Heidelberg (2008)
7. Chi, M., VanLehn, K., Litman, D.: Do Micro-Level Tutorial Decisions Matter: Applying Reinforcement Learning to Induce Pedagogical Tutorial Tactics. In: Alevan, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6094, pp. 224–234. Springer, Heidelberg (2010)
8. Corbett, A.T., Anderson, J.R.: Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction* 4, 253–278 (1995)

9. Feng, M., Heffernan, N.T., Koedinger, K.R.: Predicting State Test Scores Better with Intelligent Tutoring Systems: Developing Metrics to Measure Assistance Required. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 31–40. Springer, Heidelberg (2006)
10. Feng, M., Heffernan, N.T., Koedinger, K.R.: Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. *The Journal of User Modeling and User-Adapted Interaction* 19, 243–266 (2009)
11. Feng, M., Heffernan, N.T., Beck, J.: Using learning decomposition to analyze instructional effectiveness in the ASSISTment system. In: Graesser, A., Dimitrova, V., Mizoguchi, R. (eds.) Proceedings of the 14th International Conference on Artificial Intelligence in Education, Brighton, UK (2009)
12. Gowda, S.M., Rowe, J.P., Baker, R.S.J.d., Chi, M., Koedinger, K.R.: Improving Models of Slipping, Guessing, and Moment-by-Moment Learning with Estimates of Skill Difficulty. In: Proc. of the 4th International Conference on Educational Data Mining, pp. 199–208 (2011)
13. Razzaq, L., Patvarczki, J., Almeida, S.F., Vartak, M., Feng, M., Heffernan, N.T., Koedinger, K.R.: The ASSISTment builder: Supporting the Life-cycle of ITS Content Creation. *IEEE Transactions on Learning Technologies Special Issue on Real-World Applications of Intelligent Tutoring Systems* 2(2), 157–166 (2009)
14. Levene, H.: Robust tests for equality of variances. In: Olkin, I., Hotelling, H., et al. (eds.), pp. 278–292. Stanford University Press (1960)
15. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: YALE: Rapid Prototyping for Complex Data Mining Tasks. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006), pp. 935–994 (2006)
16. Murray, T.: Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education* 10, 98–129 (1999)
17. Pardos, Z., Heffernan, N.: Detecting the Learning Value of Items in a Randomized Problem Set. In: Dimitrova, Mizoguchi, du Boulay, Graesser (eds.) Proceedings of the 2009 Artificial Intelligence in Education Conference, pp. 499–506. IOS Press (2009)
18. Pardos, Z.A., Dailey, M., Heffernan, N.: Learning what works in ITS from non-traditional randomized controlled trial data. *International Journal of Artificial Intelligence in Education* (in press)

Towards Automatically Detecting Whether Student Learning Is Shallow

Ryan S.J.D. Baker¹, Sujith M. Gowda¹, Albert T. Corbett², and Jaclyn Ocumpaugh¹

¹ Department of Social Science and Policy Studies,
Worcester Polytechnic Institute, Worcester, MA USA
{rsbaker, sujithmg, jocumpaugh}@wpi.edu

² Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA USA
corbett@cmu.edu

Abstract. Recent research has extended student modeling to infer not just whether a student knows a skill or set of skills, but also whether the student has achieved robust learning – learning that leads the student to be able to transfer their knowledge and prepares them for future learning (PFL). However, a student may fail to have robust learning in two fashions: they may have no learning, or they may have shallow learning (learning that applies only to the current skill, and does not support transfer or PFL). Within this paper, we present an automated detector which is able to identify shallow learners, who are likely to need different intervention than students who have not yet learned at all. This detector is developed using a step regression approach, with data from college students learning introductory genetics from an intelligent tutoring system.

Keywords: robust learning, student modeling, educational data mining, intelligent tutoring system.

1 Introduction

Over the last two decades, student models have become effective at predicting which skills a student knows at a given time [cf. 16, 21, 22]. Recent research has gone beyond this to also assess the robustness of student learning [20] – whether students will be able to transfer their knowledge, whether they will be prepared for future learning, and whether they will retain their knowledge over the long-term. [8] presents a model that infers whether a student will perform well on a transfer test after using the tutor software – where the student must succeed at a related skill not taught in the tutor. Similarly, [9] presents a model that infers whether a student will be able to learn a new but related skill from an instructional text, after using the tutor. These types of models represent a step towards intelligent tutoring systems that can respond not just to whether a student has learned a skill, but to whether the student has achieved robust learning that will help them apply the knowledge broadly, in novel situations going forward.

However, while this work is a step towards modeling and remediation of robust learning, it is not sufficient to enable sophisticated differential intervention for shallow and robust learners. The robust learning detectors in [8, 9] only measure the extent to which a student has acquired robust learning. If the student has not acquired robust learning, these detectors cannot differentiate between a student who has acquired shallow knowledge (where the student knows the skills taught in the tutor, but cannot transfer those skills and is not prepared for future learning) and a student who has not learned at all. A student who has not learned at all may simply need more tutor practice [cf. 15], whereas a student who has shallow learning may need support in building from their procedural skill to deeper conceptual understanding. There are now interventions which have been shown to help students acquire robust learning [cf. 11, 13, 23, 24, 27], but not all students may need such interventions. A detector which can identify a student who has shallow learning, when combined with such interventions, may have the potential to enable richer intervention and better learner support than is currently possible.

As a step towards this vision, this paper presents a model designed to identify shallow learners, within a Cognitive Tutor for Genetics problem-solving [17]. This model is generated using a combination of feature engineering and step regression, and is cross-validated at the student level (e.g. repeatedly trained on one group of students and tested on other students). We report this detector's effectiveness at identifying shallow learners, and analyze its internal features, comparing them to features previously used to predict transfer and preparation for future learning (PFL).

2 Data Set

The data analyzed in this study come from 71 undergraduates using the Genetics Cognitive Tutor [17]. The Genetics Cognitive Tutor consists of 19 modules that support problem solving across a wide range of topics in genetics. Various subsets of the 19 modules have been piloted at 15 universities in North America. This study focuses on the data from a tutor module that employs a gene mapping technique called *three-factor cross*, in which students infer the order of three genes on a chromosome based on offspring phenotypes, as described in [5]. The data used in this analysis, first published in [8], were produced by students who were enrolled in genetics or introductory biology classes at Carnegie Mellon University.

These students used Cognitive Tutor-supported activities in two one-hour laboratory sessions, on successive days. In each session, students completed standard three-factor cross problems. During the first lab session, some students piloted cognitive-tutor activities designed to support deeper understanding; however, no differences were found between conditions for any robust learning measure, so in this analysis we collapse across the conditions and focus solely on student behavior and learning within the standard problem-solving activities.

The 71 students completed a total of 22,885 problem-solving actions across 10,966 problem steps in the tutor. Four paper-and-pencil post-tests followed the tutor activities [cf. 5]. Three tests were given immediately after tutor usage: a

straightforward problem-solving post-test, a transfer test, and a test of preparation for future learning. A retention test was administered one week later.

Within this paper we focus analysis on the immediate problem-solving post-test, and the transfer test of robust learning. The problem-solving post-test consisted of two problems, and had two test forms, counterbalanced with the pre-test. Each of the two problems on each test form consisted of 11 steps involving 7 of the 8 skills in the three-factor cross tutor lesson, with two skills applied twice in each problem and one skill applied three times. The transfer test included two problems intended to tap students' understanding of the underlying processes of three-factor cross. The first was a three-factor cross problem that could not be solved with the standard solution method and required students to improvise an alternative method. The second problem asked students to extend their reasoning to four genes. It provided a sequence of four genes on a chromosome and asked students to reason about the crossovers that must have occurred in different offspring groups.

Students demonstrated successful learning in this tutor, with an average pre-test performance of 0.31 (SD=0.19), and an average post-test performance of 0.81 (SD=0.18). Students were also successful on the transfer test, with an average score of 0.85 (SD=0.18). The correlation between the problem-solving post-test and the transfer test was 0.59, suggesting that, although problem-solving skill and transfer skill were related, transfer may be predicted by more than just simply skill at problem-solving within this domain.

3 Shalowness Detector

3.1 Label Generation

The first step towards developing a data-mined model to predict which students have shallow learning is to create an operational definition of shallow learning that can be used as a training label (e.g. a "ground truth" label of the construct being predicted) for our shallowness detector. We employed data from the post-test of problem-solving skill and the transfer test posttest to do this. We operationalized shallow learning as the difference between a student's problem-solving test score and their transfer test score. Better performance on the problem-solving test than the transfer test indicates the student has acquired basic problem-solving knowledge, but in a shallow fashion, without the deep understanding that enables the application of that knowledge in novel situations.

Given the approximately equal average performance on the two tests, we can take simple percent correct on each test to assess whether a student is a shallow learner or not (if the tests had radically different average performance, it might be better to use percentile rank on each test, or Z scores). As such, the present analysis treats students who achieve higher scores on the problem-solving post-test than on the transfer test as having shallow learning.

According to this operational definition, 24 of the 71 students in this study are labeled as shallow learners. Of the remaining 47 students, treated as not having

shallow learning, 17 had perfect scores on both the transfer test and post-test. No other students had the same score on the two tests. The other 30 students had higher scores on the transfer test than the post-test. Among the 24 students labeled as shallow learners, there was an average of a 0.14 point difference between performance on the two tests (standard deviation = 0.10), with an average score of 0.87 on the problem-solving post-test, and an average score of 0.73 on the transfer test.

3.2 Data Features

The next step in our process of developing a model that could automatically identify shallow learning was to identify properties of students' problem-solving actions in the Cognitive Tutor that may be hallmarks of shallow learning. Towards this end, we selected a set of action-level features based on a combination of theory and prior work to model and detect related constructs. In particular, prior research on detectors of transfer [8] and PFL [9] influenced our design of features. As in that work, we can infer which students had shallow learning, using the method discussed in the previous section; but we do not know exactly what actions are associated with the shallow learning in advance. Hence, we take features calculated at the level of actions, and aggregate them across actions. We do so using two kinds of computations: the proportion of time specific behaviors occurred, and average quantitative values across actions. The 24 features used in this analysis included two categories of basic features, and two categories of complex features.

The first category of basic features focused on overall response time and time spent processing tutor-provided assistance, including: (1) average response time, (2) the average unitized response time (in standard deviations above or below the mean for students on the current skill), (3) the proportion of actions that involved a fast response after the student received a bug message (bug messages indicate why the system thinks the student made an error), (4) the proportion of slow responses after a bug message, (5) the proportion of fast responses after requesting a hint, (6) the proportion of slow responses after requesting a hint, (7) the proportion of slow actions after receiving a hint and entering a correct answer [cf. 25], and (8) the proportion of fast actions after receiving a hint and entering a correct answer.

The second category of basic features focused on the content of a student action: (9) the proportion of correct answers, (10) the proportion of help requests, and (11) the proportion of answers that were incorrect and received bug messages.

The first category of complex features involved Bayesian Knowledge Tracing estimates of the student's knowledge of relevant skills and performance probabilities [16]: (12) the average probability the student knew the skill, (13) the average probability the student would give a correct answer according to the model, (14) fast actions on well-known skills, and (15) slow actions on well-known skills.

The second category of complex features focused on features derived from previous research on meta-cognition and disengagement: (16) help avoidance, the proportion of actions where the skill was not known and help was not sought [cf. 2], (17) the proportion of actions where the skill was known and help was not sought, (18) fast actions not involving gaming the system [using the detector from 6], (19)

slow actions not involving off-task behavior [using the detector from 3], (20) the average contextual probability that an error was due to slipping [cf. 4], (21) the average contextual probability of slip among actions with over 50% probability of being a slip (called “certainty of slip”) [cf. 5], (22) the average contextual probability that a correct response was a guess [cf. 4], (23) the “certainty of guess” (corresponding to certainty of slip), and (24) the average moment-by-moment learning [cf. 7].

Some of these features relied upon cut-offs; in these cases, an optimized cut-off was chosen using a procedure discussed in the next section.

3.3 Detector Development

We fit detectors of shallowness using step regression models. (Note that step regression is not the same as step-wise regression.) Step regression involves fitting a linear regression model to predict the labels of shallowness using the features of student behavior in the tutor, and then thresholding that model’s predictions with a pre-chosen cut-off, in this case 0.5. Within this statistical framework, all students for whom the linear regression predicted values of 0.5 or higher are assessed to have non-shallow learning, whereas all students for whom the linear regression predicted values below 0.5 are assessed to have shallow learning. The choice of 0.5 is an arbitrary standard convention (0.5 is halfway between 0 and 1); so long as the step cut-off is chosen prior to model fitting, equal performance can be achieved for any step cut-off (different step cut-offs are adjusted for by the constant term of the equation). Hence, this framework takes numerical predictions of shallowness and transforms them into a binary prediction of whether the student’s learning is shallow or not, which can be compared to the labels initially derived from the two tests.

These detectors of shallowness are assessed using 10-fold student-level cross-validation [18]. In 10-fold cross-validation, the data points are divided into ten groups (in this case divided by students), each of which serves successively as a test set. That is, for each of the ten groups, the other nine groups are used to produce a model, and then the tenth group is used to test that model. Hence, each model’s goodness is never tested on the same students it was trained on, but each model is tested on every student. Because this process does not exclude any data points (or students) from the modeling process, cross-validation is typically preferred to holding out a test set that is entirely excluded from model development. Cross-validated performance assesses the model’s predictive performance when applied to new data, an indicator of the model’s ability to generalize.

Two metrics were used as the assessment of goodness for each model: (1) A' (also called AUC, for “Area Under [the ROC] Curve”) [19], and (2) Cohen’s [14] Kappa, or κ . A' is the probability that if the detector is comparing two students, one labeled as having shallow learning and the other one not labeled as having shallow learning, it will correctly identify which clip is which. A' is mathematically equivalent to W , the Wilcoxon statistic [19]. A model with an A' of 0.5 performs at chance, and a model with an A' of 1.0 performs perfectly. In these analyses, A' was computed using the AUC (area under the curve) method. Cohen’s Kappa (κ) assesses whether the detector

is better than chance at identifying the correct action sequences as involving the category of interest. A Kappa of 0 indicates that the detector performs at chance, and a Kappa of 1 indicates that the detector performs perfectly. A' and Kappa both compensate for the possibility that successful classifications can occur by chance [cf. 10]. A' can be more sensitive to uncertainty in classification than Kappa, because Kappa looks only at the final label whereas A' looks at the classifier's degree of confidence in classifying an instance.

We fit two detectors. The first detector uses only the individual features discussed above in section 3.2. Some of the features, involving proportions of specific types of actions, depend on a threshold parameter (such as how many seconds differentiates a "long pause" from a "short pause"); these parameters were optimized by computing the single-feature step regression model for a range of potential thresholds (see [8] for more details) and selecting the threshold with the best A' value. The second detector also includes multiplicative interactions between the individual features. In order to reduce the potential for over-fitting (where a set of features does not generalize well to data from new students), we reduce the parameter space of both models prior to fitting full models. The individual feature model is limited to considering features for which a single-feature step regression model has a better value for the Akaike information criterion (AIC [1]) than the empty model, reducing the data space from 24 features to 11 features. The multiplicative interaction model only considers the 66 interactions of those 11 features, and furthermore discards features that fail the same Akaike test, resulting in a set of 35 multiplicative interaction features, plus 11 individual features, for a total data space of 46 features.

We used Forward Selection to find the best model for each one of the two feature sets. In Forward Selection, the best single-parameter model is chosen, and then the parameter that most improves the model is repeatedly added until no more parameters can be added which improve the model. In this case, the goodness criterion for model selection was cross-validated Kappa.

4 Results

The best-fitting models for each feature set are as follows:

Table 1. Step regression models with student-level cross-validated A' and Kappa (higher values of model coefficients correspond to non-shallow learners)

Model Type	Model	A'	Kappa
Multiplicative-Interactions	2221 * SlowResponseAfterBugMsg - 0.22 * AverageCertaintyOfSlip * AvgTime + 1.03	0.758	0.389
No-Interactions	34.74 * SlowResponseAfterBugMsg - 1232 * AvgTimeSD + 0.6726	0.767	0.346

As can be seen in Table 1, the multiplicative-interactions model achieves moderately better cross-validated Kappa than the no-interactions model, and slightly worse cross-validated A'. The model with multiplicative interactions achieved an acceptable cross-validated kappa of 0.389 (39% better than chance according to the baseline [cf. 14]). It is worth noting that kappa values typically achieved in data mining are usually lower than kappa values achieved in inter-rater reliability checks among human coders; the standards are different because the goals are different. The agreement between a data-mined model and a construct which is itself noisy will inherently be lower than human agreement on a tightly-defined construct. The A' value for the multiplicative-interactions model is 0.758, which indicates that the model can differentiate a student who performs better on the problem-solving test than the transfer test from a student who does not perform better on the problem-solving test than the transfer test, 75.8% of the time. This level of performance on the A' metric is typically considered to be sufficient to enable fail-soft intervention. This level of performance is significantly better than chance, $Z=-3.56$, $p<0.001$, using the test from [19].

The features that constitute the two models are similar, and both models are quite simple. In both models, the first feature is slow responses after bug messages. The positive coefficient for this feature indicates that students who pause when receiving bug messages are less likely to be shallow learners. Bug messages in this tutor lesson tell students about what order to complete steps in, and which information is necessary to draw valid conclusions. As such, reflective pauses upon receiving these messages may indicate a student trying to understand why certain information is necessary at specific steps in the reasoning process. It seems reasonable that these reflective pauses would be associated with more robust learning. This feature is also associated with a greater probability of transfer, in the same tutor lesson [8].

The second feature in both models involves average response time. This feature has a negative coefficient in both models, indicating that in general slow response times are associated with shallow learning. Shallow learners are not characterized by fast guesses (which may lead to no learning at all), but just the opposite – they seem rather to be struggling compared to students achieving robust learning. More specifically, average response speed relative to all students enters into the individual feature model. In the multiplicative-interactions model, response speed enters the model as an interaction with the average certainty of slip (the probability of slip among actions that are likely to be slips, an error despite knowing the skill). The average certainty of slip has been previously shown to predict final tutor knowledge, even after controlling for predictions from Bayesian Knowledge Tracing [5]; as such, it makes sense that this feature may be related to the depth of learning. While the more common interpretation of a slip is carelessness, an alternative interpretation is that a slip indicates imperfect acquisition of skill, where a student's skill knowledge works on some problems but not on others [cf. 4]. Such lack of transfer within even basic problem solving would be consistent with shallow learning.

5 Discussion and Conclusions

Within this paper, we have presented models that can distinguish with reasonable accuracy whether a student has acquired shallow learning, operationally defined as performing better on a test of the material learned in the tutor, than on a test of the ability to transfer that skill to related problems. These models are developed in the context of a Cognitive Tutor for Genetics, and cross-validated at the student level; exploring this model's generality to other learning domains and types of educational software is an important area of future work.

The better of these models can distinguish a shallow learner from a non-shallow learner 76% of the time, performing 39% better than chance. These models are based on three features of the student's interactions with the learning software, including two found in both models: the speed of student actions, and the speed of student responses after receiving bug messages. A third feature, probable slips during performance, is only found in the multiplicative-interactions model. As with the previous model of transfer [cf. 8], how students respond to evidence that they do not understand the skill (bug messages) appears to be particularly important for modeling shallow learning. This result is in line with theory that suggests a key role for meta-cognition in robust learning [20]; it also suggests that student responses to bug messages – not currently a key aspect of theoretical models of meta-cognition in intelligent tutoring systems [cf. 2] – deserves a more prominent place in future theoretical models.

Shallowness detectors have considerable potential usefulness for intelligent remediation. Students who have learned the exact skills taught in the tutor but who have not achieved robust learning are a group especially in need of remediation. Traditional student modeling methods are likely to fail to provide them any remediation, as they have learned the skills being taught by the tutor and can demonstrate that skill. A detector of shallow learning can identify these students and offer them remediation specific to their needs, helping a student to build on his or her procedural knowledge to achieve the conceptual understanding necessary for future use of their knowledge. Thus, we view this detector as a second step – building on the first step of transfer and PFL detectors – towards educational software that can predict and respond automatically to differences in the robustness of student learning, an important complement to ongoing research on designing educational software that promotes robust learning [cf. 11, 13, 23, 24, 26].

Acknowledgements. The authors thank award #DRL-0910188 from the National Science Foundation, “Empirical Research: Emerging Research: Robust and Efficient Learning: Modeling and Remediating Students’ Domain Knowledge”.

References

1. Akaike, H.: A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 19(6), 716–723 (1974)

2. Aleven, V., McLaren, B.M., Roll, I., Koedinger, K.R.: Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence and Education* 16, 101–128 (2006)
3. Baker, R.S.J.d.: Modeling and Understanding Students' Off-Task Behavior in Intelligent Tutoring Systems. In: *Proceedings of ACM CHI 2007: Computer-Human Interaction*, pp. 1059–1068 (2007)
4. Baker, R.S.J.d., Corbett, A.T., Aleven, V.: More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) *ITS 2008. LNCS*, vol. 5091, pp. 406–415. Springer, Heidelberg (2008)
5. Baker, R.S.J.d., Corbett, A.T., Gowda, S.M., Wagner, A.Z., MacLaren, B.A., Kauffman, L.R., Mitchell, A.P., Giguere, S.: Contextual Slip and Prediction of Student Performance after Use of an Intelligent Tutor. In: De Bra, P., Kobsa, A., Chin, D. (eds.) *UMAP 2010. LNCS*, vol. 6075, pp. 52–63. Springer, Heidelberg (2010)
6. Baker, R.S.J.d., Corbett, A.T., Roll, I., Koedinger, K.R.: Developing a generalizable system to detect when students game the system. *User Modeling and User-Adapted Interaction* 18(3), 287–314 (2008)
7. Baker, R.S.J.d., Goldstein, A.B., Heffernan, N.T.: Detecting the Moment of Learning. In: Aleven, V., Kay, J., Mostow, J. (eds.) *ITS 2010. LNCS*, vol. 6094, pp. 25–34. Springer, Heidelberg (2010)
8. Baker, R.S.J.d., Gowda, S.M., Corbett, A.T.: Towards Predicting Future Transfer of Learning. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011. LNCS*, vol. 6738, pp. 23–30. Springer, Heidelberg (2011)
9. Baker, R.S.J.d., Gowda, S., Corbett, A.T.: Automatically Detecting a Student's Preparation for Future Learning: Help Use is Key. In: *Proceedings of the 4th International Conference on Educational Data Mining*, pp. 179–188 (2011)
10. Ben-David, A.: About the Relationship between ROC Curves and Cohen's Kappa. *Engineering Applications of Artificial Intelligence* 21, 874–882 (2008)
11. Butcher, K.R.: How Diagram Interaction Supports Learning: Evidence from Think Alouds during Intelligent Tutoring. In: Goel, A.K., Jamnik, M., Narayanan, N.H. (eds.) *Diagrams 2010. LNCS*, vol. 6170, pp. 295–297. Springer, Heidelberg (2010)
12. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: *Proc. of the International Conference on Machine Learning*, pp. 161–168 (2006)
13. Chin, D.B., Dohmen, I.M., Cheng, B.H., Opezzo, M.A., Chase, C.C., Schwartz, D.L.: Preparing Students for Future Learning with Teachable Agents. *Educational Technology Research and Development* 58(6), 649–669 (2010)
14. Cohen, J.: A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20(1), 37–46 (1960)
15. Corbett, A.: Cognitive Computer Tutors: Solving the Two-Sigma Problem. In: Bauer, M., Gmytrasiewicz, P.J., Vassileva, J. (eds.) *UM 2001. LNCS (LNAI)*, vol. 2109, pp. 137–147. Springer, Heidelberg (2001)
16. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction* 4, 253–278 (1995)
17. Corbett, A.T., MacLaren, B., Kauffman, L., Wagner, A., Jones, E.A.: Cognitive Tutor for Genetics Problem Solving: Learning Gains and Student Modeling. *Journal of Educational Computing Research* 42(2), 219–239 (2010)
18. Efron, B., Gong, G.: A leisurely look at the bootstrap, the jackknife, and cross-validation. *American Statistician* 37, 36–48 (1983)

19. Hanley, J., McNeil, B.: The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology* 143, 29–36 (1982)
20. Koedinger, K.R., Corbett, A.T., Perfetti, C.: The Knowledge-Learning-Instruction (KLI) Framework: Toward Bridging the Science-Practice Chasm to Enhance Robust Student Learning. *Cognitive Science* (in press)
21. Martin, J., VanLehn, K.: Student assessment using Bayesian nets. *International Journal of Human-Computer Studies* 42, 575–591 (1995)
22. Pavlik, P.I., Cen, H., Koedinger, K.R.: Performance Factors Analysis – A New Alternative to Knowledge Tracing. In: *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, pp. 531–540 (2009)
23. Roll, I., Alevan, V., McLaren, B.M., Koedinger, K.R.: Can help seeking be tutored? Searching for the secret sauce of metacongitive tutoring. In: *Proceedings of the 13th International Conference on Artificial Intelligence in Education* (2007)
24. Salden, R.J.C.M., Alevan, V., Renkl, A.,Schwonke, R.: Worked Examples and Tutored Problem Solving: Redundant or Synergistic Forms of Support? In: *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, pp. 589–594 (2008)
25. Shih, B., Koedinger, K.R., Scheines, R.: A response time model for bottom-out hints as worked examples. In: *Proceedings of the 1st International Conference on Educational Data Mining*, pp. 117–126 (2008)
26. Tan, J., Biswas, G.: The Role of Feedback in Preparation for Future Learning: A Case Study in Learning by Teaching Environments. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006*. LNCS, vol. 4053, pp. 370–381. Springer, Heidelberg (2006)

Item to Skills Mapping: Deriving a Conjunctive Q-matrix from Data

Michel C. Desmarais, Behzad Beheshti, and Rhouma Naceur

Polytechnique Montréal

Abstract. Uncovering which skills are determining the success to questions and exercises is a fundamental task in ITS. This task is notoriously difficult because most exercise and question items involve multiple skills, and because skills modeling may involve subtle concepts and abilities. Means to derive this mapping from test results data are highly desirable. They would provide objective and reproducible evidence of item to skills mapping that can either help validate predefined skills models, or give guidance to define such models. However, the progress towards this end has been relatively elusive, in particular for a conjunctive skills model, where all required skills of an item must be mastered to obtain a success. We extend a technique based on Non-negative Matrix Factorization, that was previously shown successful for single skill items, to construct a conjunctive item to skills mapping from test data with multiple skills per item. Using simulated student test data, the technique is shown to yield reliable mapping for items involving one or two skills from a set of six skills.

Keywords: Student model, Skills modeling, Psychometrics, Q-matrix, matrix factorization, SVD, NMF.

1 Introduction

When an ITS personalizes the learning content presented to a student, it has to rely on some classification of this content with regards to skills, and on the student's skills assessment. Therefore, the question items and exercises involved in the assessment must be aligned with these skills. The mapping of items to skills plays a pivotal role in most if not all ITS.

A standard means to model this mapping is the Q-matrix [10,9]. It defines which skills are necessary to correctly answer an item. Take the Q-matrix in figure 1 (matrix \mathbf{Q} on the left) composed of 3 skills and 4 items. We find that item i_1 requires two skills, s_2 and s_3 , whereas item i_2 requires a single skill, s_3 , and so on.

Assuming now that a set of three examinees have mastered skills according to matrix \mathbf{S} of figure 1 (middle), and that all skills of an item are necessary to correctly answer this item, then we would expect a result that corresponds to matrix \mathbf{R} in figure 1 (right). This framework corresponds to a *conjunctive* Q-matrix: a line in Figure 1's Q-matrix indicates a conjunction of necessary skills

$$\mathbf{Q} = \begin{array}{c} \text{skills} \\ s_1 \ s_2 \ s_3 \\ \begin{array}{c} i_1 \\ i_2 \\ i_3 \\ i_4 \end{array} \end{array} \begin{array}{|ccc|} \hline 0 & 1 & 1 \\ \hline 0 & 0 & 1 \\ \hline 1 & 0 & 0 \\ \hline 1 & 0 & 1 \\ \hline \end{array} \quad \mathbf{S} = \begin{array}{c} \text{examinee} \\ e_1 \ e_2 \ e_3 \\ \begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} \end{array} \begin{array}{|ccc|} \hline 1 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \hline 0 & 1 & 1 \\ \hline \end{array} \quad \mathbf{R} = \begin{array}{c} \text{examinee} \\ e_1 \ e_2 \ e_3 \\ \begin{array}{c} i_1 \\ i_2 \\ i_3 \\ i_4 \end{array} \end{array} \begin{array}{|ccc|} \hline 0 & 0 & 1 \\ \hline 0 & 1 & 1 \\ \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$$

Fig. 1. Q-matrix and skills matrix examples

to succeed the corresponding item. The goal is to bring this framework to a linear system, allowing the application of standard linear algebra techniques.

Barnes [1] gives the following equation for inferring the expected examinee results as the product of the Q-matrix and the skills matrix (adapted from [1] for the transpose of \mathbf{R}):

$$\mathbf{R} = \neg(\mathbf{Q}(\neg\mathbf{S})) \quad (1)$$

where the operator \neg is the boolean negation, which is defined as a function that maps a value of 0 to 1 and any other value to 0. This equation will yield values of 0 in \mathbf{R} whenever an examinee is missing one or more skills for a given item, and yield 1 whenever all necessary skills are mastered by an examinee.

Applying the operator \neg on both side of equation (1) and normalizing matrix \mathbf{Q} to ensure the row sums are 1 yields:

$$\neg\mathbf{R} = \mathbf{Q}(\neg\mathbf{S}) \quad (2)$$

Equation (2) is a standard linear equation where the matrices \mathbf{R} and \mathbf{S} are negated. The task of inferring the Q-matrix from $\neg\mathbf{R}$ can therefore be seen as a matrix factorization: the matrix $\neg\mathbf{R}$ is the product of the two matrices, \mathbf{Q} and $\neg\mathbf{S}$.

2 Comparison with a One Skill Per Item Condition

The matrix factorization approach to inferring the Q-matrix from data has been explored by a few researchers [3,11], but for Q-matrices that involved only a single skill per item. They investigated the Non-negative Matrix Factorization (NMF) [8] technique and showed that it works very well for simulated data, but the technique's performance with real data was degraded. For highly separable skills like mathematics and French, its performance is quite good, assigning correctly the items belonging to each topic. But the technique is very weak at classifying items according to skills such as History and Biology, as measured by Trivia type of questions. These results suggest that expertise necessary to succeed Biology and History questions is not well separated into these two general topics. Presumably, we would find a stronger skill separation if we studied very specific skills, like the pieces of knowledge behind each question. This is in fact what tutors such as the Cognitive family of tutors and the ASSISTment system do, they rely on fine grain skills mapped to items [7,5]. For these low-level

skills, the conjunctive model, which requires that each skill is mastered for every item that require them, is in general the model used by widely known learning environments such as the Cognitive Tutors family.

The matrix factorization approach of the studies in [3,11] was based on the additive (compensatory) model of skills, where each skill increases the chances of success to an item. This corresponds to the following equation where the negation operator \neg is omitted:

$$\mathbf{R} = \mathbf{Q}\mathbf{S} \quad (3)$$

For the one skill per item condition, equations (1) and (3) are equivalent, but they give very different results for two or more skills per item. Following the skill structure example in figure 1, item i_4 would be failed by all examinees according to equation (1) whereas it would be (partly) succeeded according to equation (1), with values above 0 for all examinees on this item.

An obvious followup over the studies by [1,3,11] is to apply the NMF technique to equation (2), and to determine if NMF can successfully derive a conjunctive Q-matrix, where skills do not add up to increase the chances of success to an item, but instead are necessary conditions. This is the goal of the current investigation.

3 Non-negative Matrix Factorization

Non-negative matrix factorization (NMF) decomposes a matrix into two smaller matrices. It is used for dimensionality reduction, akin to Principal Component Analysis and Factor analysis. NMF decomposes a matrix of $n \times m$ positive numbers, \mathbf{V} , as the product of two matrices:

$$\mathbf{V} \approx \mathbf{W}\mathbf{H} \quad (4)$$

Clearly, the matrix \mathbf{W} corresponds to the Q-matrices of equations (2) and (3).

Whereas most other matrix factorization techniques impose constraints of orthogonality among factors, NMF imposes the constraint that the two matrices, \mathbf{W} and \mathbf{H} , be non-negative. This constraint makes the interpretation much more intuitive in the context of using this technique for building a Q-matrix. It implies that the skills (latent factors) are additive “causes” that contribute to the success of items, and that they can only increase the probability of success and not decrease it, which makes good sense for skill factors.

It is important to emphasize that there are many solutions to $\mathbf{V} = \mathbf{W}\mathbf{H}$. Different algorithms may lead to different solutions. Indeed, many NMF algorithms have been developed in the last decade and they can yield different solutions. We refer the reader to [2] for a more thorough and recent review of this technique which has gained strong adoption in many different fields.

The non-negative constraint and the additive property of the skills bring a specific interpretation of the Q-matrix. For example, if an item requires skills a and b with the same weight each, then each skill will contribute equally to the success of the item. This corresponds to the notion of a *compensatory* or *additive* model of skills as we mentioned earlier. The negation of matrix \mathbf{R} in

equation (2) brings a new interpretation of the Q-matrix where the conjunction of skills are considered necessary conditions to answer the corresponding item. This requires that the matrix \mathbf{S} be also negated, and it corresponds to \mathbf{H} in equation (4). However, in applying the negation operator, \neg , all values greater than 1 are replaced by 1, and that can be considered as a loss of information.

4 Simulated Data

To validate the approach, we rely on simulated data. Although it lacks the external validity of real data, it remains the most reliable means of obtaining test results data for which the underlying, latent skills structure is perfectly known. Any experiment with real data is faced with the issues that the expert-defined Q-matrix may not contain all determinant skills, may not have a perfect mapping, and that all skills may not combine conjunctively and with equal weight, making the interpretation of the results a complex and error prone task. Therefore, assessing the technique over simulated data is a necessary first step to establish the validity the approach under controlled conditions. Further studies with real data will be necessary, assuming the results of the existing study warrants such work.

The underlying model and methodology of the simulated data are explained in a previous paper [4] and we briefly review some details this methodology below.

A first step to obtain data of simulated examinee test results is to define a Q-matrix composed of j skills and k items. We chose to define a Q-matrix that spans all possible combinations of 6 skills with a maximum of two skills per item, and at least one skill per item. A total of 21 items spans this space of combinations. This matrix is shown in Figure 2(a). Items 1 to 15 are two-skills and items 16 to 21 are single-skill.

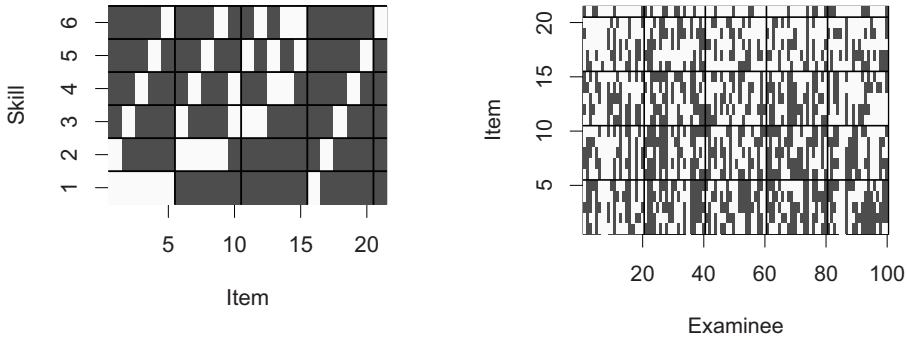
We do not assume that skills all have the same difficulty level, and therefore we assign various difficulty level to each skill. The difficulty is reflected by the probability of mastery. That difficulty will transfer to items that have this skill. The difficulty of the two-skills items will further increase by the fact that they require the conjunction of their skills. An item difficulty is therefore inherited by the difficulty of its underlying skills.

In addition to skills difficulty, examinees need to be assigned ability levels. The ability is reflected by the probability of mastering some skill. Therefore, the probability of mastery of a given skill by a given examinee is a function of examinee ability and skill difficulty levels.

Finally, two more parameters are used in the simulated data, namely the *slip* and *guess* factors. These factors are set as constant values across items. They are essentially noise factors and the greater they are, the more difficult is the task of inducing the Q-matrix from data.

Given the above framework, the process of generating simulated examinee data follows the following steps:

1. Assign a difficulty level to each skill.
2. Generate a random set of hypothetical examinee skills vectors based on the difficulty of each skill and the examinee's ability level. Skill difficulty and



(a) Q-matrix of 6 skills and for which 21 items are spanning the full set of 1 and 2 skill combinations. Items 16 to 21 require a single skill and all others require 2-skills. (b) Simulated data example of 100 examinees with parameters: slip: 0.1, guess: 0.2, skills difficulties: (0.17, 0.30, 0.43, 0.57, 0.70, 0.83).

Fig. 2. Q-matrix and an example of simulated data with this matrix. Light pixels represent 1's and dark (red) ones represent 0's.

examinee ability are each expressed as a random normal variable. The probability density function of their sum provides the probability of mastery of the skill for the corresponding examinee. The skill vector is a sampling in $\{0, 1\}$ based on each skill probability of mastery.

3. Generate simulated data based on equation (2) without taking into account the *slip* and *guess* parameters. This is referred to as the *ideal response pattern*.
4. Randomly change the values of the generated data based on the *slip* and *guess* parameters. For example, with values of 0.1 and 0.2 respectively, this will result in 10% of the succeeded items in the ideal response pattern to become failed, and 20% of the failed items to become succeeded.

The first two steps of this process are based on additive gaussian factors and follow a similar methodology to [3]. For brevity we do not report the full details but refer the reader to the R code available at www.professeurs.polymtl.ca/michel.desmarais/Papers/ITS2012/its2012.R.

A sample of the results matrix is given in figure 2(b). Examinee ability shows up as vertical patterns, whereas skills difficulty creates horizontal patterns. As expected, the mean success rate of the 2-skills items 1 to 15 is lower (0.51) than the single skill items 16 to 21 (0.64).

5 Simulation Methodology

The assessment of the NMF performance to infer a Q-matrix from simulated test data such as figure 2(b)'s is conducted by comparing the predefined Q-matrix, \mathbf{Q} ,

as shown in figure 2(a), with the \mathbf{W} matrix obtained in the NMF of equation (4).

As mentioned above, the negation operator is applied over the simulated test data and the NMF algorithm is carried over this data. We used the R NMF package [6] and the Brunet NMF algorithm.

We defined a specific method for the quantitative comparison of the matrix \mathbf{W} with \mathbf{Q} . First, the \mathbf{W} matrix contains numerical values on a continuous scale. To simplify the comparison with matrix \mathbf{Q} , which is composed of $\{0, 1\}$ values, we discretize the numerical values of \mathbf{W} by applying a clustering algorithm to each item in \mathbf{W} , forcing two clusters, one for 0's and one for 1's. For example, item 1 in the NMF inferred matrix of figure 4(a) (which we explain later) corresponds to a vector of six numerical values, say $\{1.6, 1.7, 0.0015, 0.0022, 0.0022, 0.0018\}$. This vector clearly cluster into the $\{1, 1, 0, 0, 0, 0\}$ vector of item 1 in figure 4(b). The K-means algorithm is used for the clustering process of each item and we use the `kmeans` routine provided in R (version 2.13.1).

Then, to determine which skill vector (column) of the \mathbf{W} matrix corresponds to the skill vector of the \mathbf{Q} matrix, a correlation matrix is computed and the highest correlation of each column vector \mathbf{W} is in turn matched with the corresponding unmatched column in \mathbf{Q} .

We will use visual representations of the raw and the “discretized” (clustered) \mathbf{W} matrix to provide an intuitive view of the results, as well as a quantitative measures of the fit corresponding to the average of the correlations between the matched skills vectors \mathbf{W} and \mathbf{Q} .

6 Results

In order for the mean and variance of the simulated data to reflect realistic values of test data, the skill difficulty and examinee ability parameters are adjusted such that the average success rate is close to 60%. Examinee ability is combined with the skill difficulty vectors to create a probability matrix of the same dimensions as \mathbf{S} , from which \mathbf{S} is obtained. Figure 3(a) displays a histogram of the 21 items success rate of the *ideal* response patterns for a sample of 2000 examinees, which is generated according to equation (1). Figure 3(b) shows the item success rates after the data is transformed by the application of *slip* and *guess* transformations. This transformation will generally decrease the spread of the distribution.

Figure 4(a) shows a heat map of the matrix \mathbf{W} inferred from an *ideal* response pattern of 200 simulated examinees. Skill difficulties were set at (0.17, 0.30, 0.43, 0.57, 0.70, 0.83) and examinee mean ability and standard deviation respectively at 0 and 0.5. The discretized version of figure 4(a)'s matrix is shown in figure 4(b) and it is identical to the underlying matrix \mathbf{Q} in figure 2(a).

Figure 4(c) and 4(d) shows the effect of adding *slip* and *guess* parameters of 0.2 for each. The mapping to the underlying matrix \mathbf{Q} degrades as expected, but remains relatively accurate.

Table 1 reports the results of the quantitative comparison between the \mathbf{Q} matrix and the \mathbf{W} matrix inferred as a function of different *slip* and *guess*

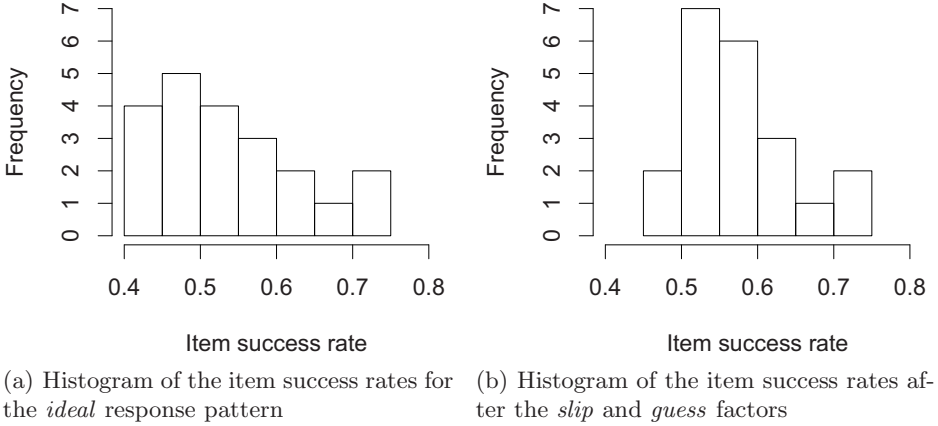


Fig. 3. Histogram of item success rates

parameters. These results are based on 10-fold simulations. The mean of the Pearson correlation coefficient ($\bar{\tau}$) between \mathbf{Q} and \mathbf{W} is reported for the discretized version of \mathbf{W} obtained with the clustering algorithm described in section 5. In addition, the error rate as computed by this formula is also provided:

$$Err = \frac{\sum_{ij} |w_{ij} - q_{ij}|}{2 \cdot \sum_{ij} |q_{ij}|} \tag{5}$$

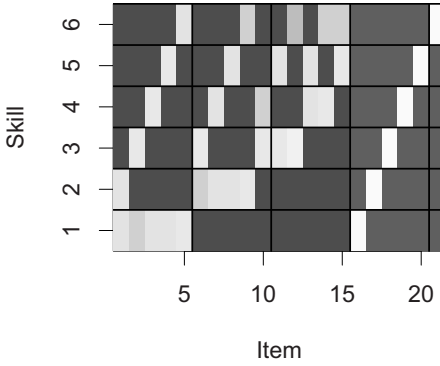
Where w_{ij} and q_{ij} are respectively the (i, j) cells of the matrices \mathbf{W} and \mathbf{Q} . The error rate will be 0 for a perfectly matched \mathbf{Q} and 1 when no cells match. A value of 0.5 indicates that half of the non-zero cells are correctly matched. For the matrix \mathbf{Q} , the error rate of a random assignment of the 36 skills is 69%.

The 0 *slip* and 0 *guess* condition (first line) correspond to figures 4(a) and 4(b), whereas the corresponding 0.2–0.2 condition (line 3) correspond to figures 4(c) and 4(d).

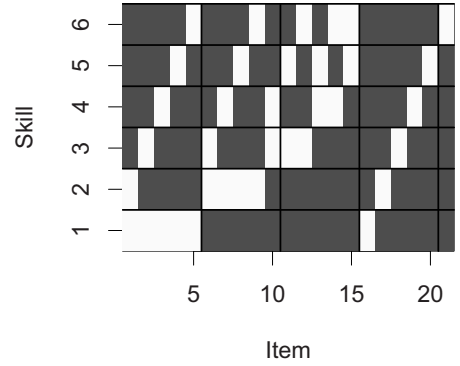
Up to the 0.2–0.2 *slip-guess* condition, the skill mapping stays relatively close to perfect. On average, approximately only 2 or 3 skills requirements are wrongly assigned out of the 36 skills requirements (7%) at the 0.2–0.2 condition. However,

Table 1. Quantitative comparison between original \mathbf{Q} matrix and NMF inferred matrices \mathbf{W} . Results are based on means and standard deviation over 10 simulation runs.

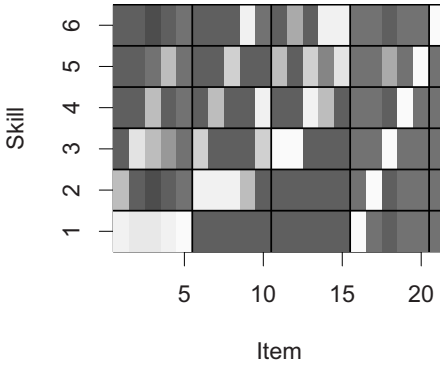
Slip	Guess	$\bar{\tau}$	sd($\bar{\tau}$)	Err	sd(Err)
0.00	0.00	1.00	0.00	0.00	0.00
0.20	0.10	0.97	0.03	0.02	0.02
0.20	0.20	0.90	0.06	0.07	0.04
0.20	0.30	0.63	0.08	0.26	0.06
0.20	0.40	0.49	0.07	0.36	0.06



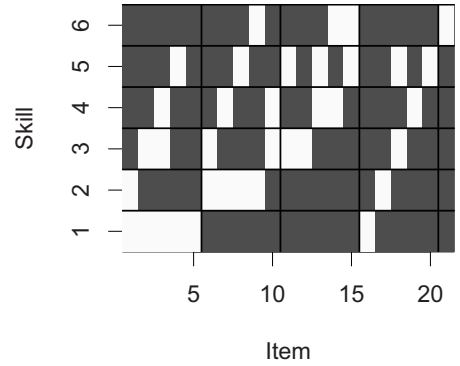
(a) Matrix \mathbf{W} without slip and guess factors ($r = 1$).



(b) Discretized \mathbf{W} without slip and guess factors ($r = 1$).



(c) Matrix \mathbf{W} with slip and guess factors of 0.2 ($r = 0.91$).



(d) Discretized \mathbf{W} for slip and guess of 0.2 ($r = 0.93$). Four out of 36 skill requirements are incorrectly mapped in this example.

Fig. 4. Visual representations of the original \mathbf{Q} matrix and NMF inferred matrices \mathbf{W} . The correlation reported (r) is computed by a comparison with the theoretical (real) matrix as explained in the text.

the error rate increases substantially at the 0.3–0.2 slip-guess condition, and at the 0.2–0.4 condition, the quality of the match degrades considerably, with an average of 13/36 wrong assignments (36%).

7 Conclusion

The proposed approach to infer a conjunctive \mathbf{Q} -matrix from simulated data with NMF is successful but, as we can expect, it degrades with the amount of *slips* and *guesses*. If the conjunctive \mathbf{Q} -matrix contains one or two items per

skill and the noise in the data remains below slip and guess factors of 0.2, the approach successfully derives the Q-matrix with very few mismatches of items to skills. However, once the data has slip and guess factors of 0.2 and 0.3, then the performance starts to degrade rapidly.

Of course, with a slip factor of 0.2 and a guess factor 0.3, about 25% of the values in the results become inconsistent with the Q-matrix. A substantial degradation is therefore not surprising. But in this experiment with simulated data, we have a number of advantages that are lost with real data: the number of skills is known in advance, no item has more than two conjunctive skills, skills are independent, and surely other factors will arise to make real data more complex. Therefore, we can expect that even if real data does not have a 50% rate of inconsistent results with the examinees' skills mastery profile, other factors might make the induction of the Q-matrix subject to errors of this scale.

Further studies with real and simulated data are clearly needed. For example, we would like to know what is the mapping accuracy degradation when an incorrect number of skills are modelled. And, naturally, a study with real data is necessary to establish if the approach is reliable in practice.

References

1. Barnes, T.: The Q-matrix method: Mining student response data for knowledge. In: AAAI 2005 Workshop on Educational Data Mining, technical report WS-05-02 (2005)
2. Berry, M.W., Browne, M., Langville, A.N., Paul Pauca, V., Plemmons, R.J.: Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis* 52(1), 155–173 (2007)
3. Desmarais, M.C.: Conditions for effectively deriving a q-matrix from data with non-negative matrix factorization. In: Conati, C., Ventura, S., Calders, T., Pechenizkiy, M. (eds.) 4th International Conference on Educational Data Mining, EDM 2011, Eindhoven, Netherlands, June 6-8, pp. 41–50 (2011)
4. Desmarais, M.C., Pelczar, I.: On the faithfulness of simulated student performance data. In: de Baker, R.S.J., Merceron, A., Pavlik Jr., P.I. (eds.) 3rd International Conference on Educational Data Mining EDM 2010, Pittsburgh, PA, USA, June 11-13, pp. 21–30 (2010)
5. Feng, M., Heffernan, N.T., Heffernan, C., Mani, M.: Using mixed-effects modeling to analyze different grain-sized skill models in an intelligent tutoring system. *IEEE Transactions on Learning Technologies* 2, 79–92 (2009)
6. Gaujoux, R., Seoighe, C.: A flexible R package for nonnegative matrix factorization. *BMC Bioinformatics* 11(367) (2010), <http://www.biomedcentral.com/1471--2105/11/367>
7. Koedinger, K.R., Anderson, J.R., Hadley, W.H., Mark, M.A.: Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education* 8, 30–43 (1997)
8. Lee, D.D., Sebastian Seung, H.: Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755), 788–791 (1999)
9. Tatsuoaka, K.K.: Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement* 20, 345–354 (1983)

10. Tatsuoka, K.K.: *Cognitive Assessment: An Introduction to the Rule Space Method*. Routledge Academic (2009)
11. Winters, T., Shelton, C.R., Payne, T.: Investigating generative factors of score matrices. In: Luckin, R., Koedinger, K.R., Greer, J.E. (eds.) *Artificial Intelligence in Education, Building Technology Rich Learning Contexts That Work*, Proceedings of the 13th International Conference on Artificial Intelligence in Education, AIED 2007, Los Angeles, California, USA, July 9-13. *Frontiers in Artificial Intelligence and Applications*, vol. 158, pp. 479–486. IOS Press (2007)

The Role of Sub-problems: Supporting Problem Solving in Narrative-Centered Learning Environments

Lucy R. Shores, Kristin F. Hoffmann, John L. Nietfeld, and James C. Lester

North Carolina State University, Raleigh, NC
{lrshores,klhoffma,jlnietfe,lester}@ncsu.edu

Abstract. Narrative-centered learning environments provide an excellent platform for both content-knowledge and problem-solving skill acquisition, as these experiences require students to apply learned material while solving real-world problems. Solving complex problems in an open-ended environment can be a challenging endeavor for elementary students given limitations in their cognitive skills. A promising potential solution is providing students with explicit *quests*, or proximal goals of a larger, more complex problem-solving activity. Quests have the potential to scaffold the process by breaking down the problem into cognitively manageable units, providing useful, frequent feedback, and maintaining motivation and the novelty of the experience. The aim of this research was to investigate the role of quests as a means for supporting situational interest and content-knowledge acquisition during interactions with a narrative-centered learning environment. Of the 299 5th grade students who interacted with CRYSTAL ISLAND, a narrative-centered learning environment for science, it was found that students who completed more quests exhibited significant increases in content learning and had higher levels of situational interest. These preliminary findings suggest potential educational and motivational advantages for integrating quest-like sub-problems into the design of narrative-centered learning environments.

Keywords: Narrative-Centered Learning Environments, Game-Based Learning, Problem Solving, Situational Interest.

1 Introduction

Leveraging affordances of technology for improving students' problem-solving skills is a long-term objective of the intelligent tutoring systems community. Kim and Hannafin [1] define problem solving as "situated, deliberate, learner-directed, activity-oriented efforts to seek divergent solutions to authentic problems through multiple interactions amongst problem solver, tools, and other resources." Students should be challenged with more open-ended problem-solving scenarios requiring domain knowledge, creativity, and high-level thinking skills [2], as it "affords them with opportunities to notice patterns, discover underlying causalities, and learn in ways that are seemingly more robust" [3]. Unfortunately, as students lacking sufficient problem-solving skills interact in such environments, they often suffer from cognitive overload [1,4] resulting in unfavorable learning outcomes [5,6].

However, narrative-centered learning environments—immersive spaces that engage users by juxtaposing domain knowledge and practical skill acquisition with narrative and game elements—may mitigate this overload by providing adequate scaffolding or constraints [1,3,7,8]. Kim and Hannafin [1] suggest helping students reduce problems into reasonable units in order to maintain focus and interest. This approach has been used in several technology-enhanced, inquiry-based learning environments for science [1,9,10]. Narrative-centered learning environments allow for such scaffolding by casting sub-tasks of the overarching problem as sub-plot events, benefiting students four-fold. First, flexible problem solving is promoted by charging students with unique problem-solving scenarios each requiring different content knowledge and actions, yet emphasizing the generality of the basic problem-solving model [11]. Second, creating smaller, more defined activities reduces the amount of relevant information to be synthesized by the student thereby freeing up working memory resources [4]. Third, since multiple quests can be completed during one session, students are provided with frequent, informational feedback to regularly prompt reflection on efficiency and strategy use, an important component of skill development [11]. Finally, by breaking the problem down into manageable units, students are able to efficiently complete tasks, a triumph associated with maintaining situational interest [12,13], which has been shown to influence cognitive performance [14] and facilitate deeper learning [15]. Thus, the primary aim of this study was to examine the relationship between providing students with manageable sub-problems and student game performance and situational interest.

2 Current Investigation

Fifth-grade students from 4 large public elementary schools in Raleigh, North Carolina interacted with CRYSTAL ISLAND, a narrative-centered learning environment for fifth-grade science education (Figure 1). The curriculum underlying the CRYSTAL ISLAND mystery narrative is derived from the state of North Carolina's standard course of study for landforms and map skills and is also intended to support learning strategies such as problem solving, critical thinking, and metacognitive skill development in an applied setting.

Students played the role of a student-selected protagonist who is one of several ship-wrecked passengers stranded on a cluster of volcanic, fictional islands trying to establish a village community. This overall goal is decomposed into three distinct sub-problems, or *quests* as they are referred to within the game environment, each with two levels, totaling seven distinct tasks—the overall problem plus six quests. The three quests are self-contained adventures that challenge students to use their domain expertise in order to complete game-like activities, and each focuses on landform identification, map navigation, and modeling, respectively, and are leveled based on difficulty. For example, level two of the modeling quest challenges the student to create a virtual model of the village by correctly arranging the island's huts on a 2-D space. The students are free to complete the quests in any order they please; however, students must successfully complete the first level of all quests before engaging in any of the second level quests. To aid their problem solving, students can seek counsel from map and landform experts who happen to be among the

ship-wrecked crew as well as the player's iPad-like device equipped with note-taking tools, a camera, a log to monitor quest completion and progress, a glossary of key landform and map skill terminology, and a problem-solving app that details the steps to the problem-solving method. To succeed, students must complete all seven quests.



Fig. 1. The CRYSTAL ISLAND narrative-centered learning environment

After cleaning the data for incomplete and outliers, a total of 293 (134 male, 159 female) cases were used for the investigation. Approximately 6% of the participants were American Indian or Alaska Native, 4% were Asian, 22% were African American, 12% were Hispanic or Latino, 54% were European American, and 7% identified themselves as other. *Content knowledge* was measured with a researcher-constructed, 19-item multiple-choice test that was based on the North Carolina Standard Course of Study curriculum and was designed to measure domain-related material integrated within the learning environment. Specifically, the test utilized fact-level and application-level questions targeting problem-solving skills, map skills, and landform knowledge. *Situational interest* was measured using the Perceived Interest Questionnaire (PIQ), a 10-item measure on a 5-point Likert scale, which has been shown to be internally reliable [16]. Students were also asked a series of open-ended, reflection questions to identify and better understand their favorite aspects of the game. In particular, one question, “What did you like best about playing CRYSTAL ISLAND,” was independently and reliably coded by two researchers ($r = .98$) and used for analysis. The experiment took place during three 60-minute sessions held on three consecutive days. Two weeks prior to data collection, students completed the content knowledge pre-test, and the post-test items were completed immediately following gameplay during the final session.

3 Results

In order to determine if content knowledge was affected as a result of interacting with the learning environment, a repeated measures analysis of variance was conducted comparing the pre- and post-content tests. Results showed a significant within-subjects effect ($F_{(1, 292)} = 25.79, p < .001, \eta^2 = .08$), indicating a significant mean

difference in content test scores between pre-test and post-test. Furthermore, significant correlations were found between content pre-test scores and total quests completed ($r = .40, p < .001$), content post-test scores and total quests completed ($r = .44, p < .001$), and situational interest and total quests completed ($r = .18, p < .001$).

To further investigate the relationship between quest completion and content knowledge acquisition, a hierarchical linear regression was conducted. Pre-content test scores (first block) and number of quests completed (second block) were used to predict post-content test scores. Both models were found to be significant (respectively, $F_{(1, 292)} = 440.35, p < .001$; $F_{(2, 291)} = 237.72, p < .001$; Table 3). The total number of quests completed was found to be a significant predictor in conjunction with pre-test scores with the entire model accounting for 61% of the variance. Interestingly, in order to determine whether the trend was simply due to high content knowledge students completing more quests, students were divided using a tertiary split on their pre-content-test scores, and the students in the lower third were isolated for analysis ($N = 83$). Both models in a similar hierarchical linear regression considering only those students also were found to be significant (respectively, $F_{(1, 82)} = 33.63, p < .001$; $F_{(2, 83)} = 21.465, p < .001$). Again, both prior knowledge ($t = 4.84, p < .001$) and total quests completed ($t = 2.63, p = .01$) were found to be significant predictors accounting for 35% of the variance within this population.

Table 1. Hierarchical linear regression predicting post content test scores

Predictor	Model 1			Model 2		
	B	SE	β	B	SE	β
Pre Content Test	.79**	.04	.77**	.73**	.04	.70**
Total Quests Completed				.42**	.11	.16**

Notes: ** - $p < .01$

Finally, analyses were performed to determine the effect of quest completion on student situational interest levels. Again, a hierarchical linear regression was conducted to predict situational interest with pre content test entered into the first block, and post content test and total quests completed entered into the second block. Only the second model was found to be significant ($F_{(3, 290)} = 3.62, p < .05$; Table 4) and was responsible for 4% of the variance. The results found only total quests completed to be a significant predictor of situational interest.

Table 2. Hierarchical linear regression predicting situational interest

Predictor	Model 1			Model 2		
	B	SE	β	B	SE	β
Pre Content Test	.02	.01	.08	.01	.02	.06
Post Content Test				-.02	.02	-.07
Total Quests Completed				.12**	.04	.19**

Notes: ** - $p < .01$

Furthermore, students' responses to the reflection questions were coded for mentions of the quests, which divided students into two groups, those who mentioned the quests as their favorite part ($N = 132$) and those who mentioned other aspects of the game (e.g., choosing a player; $N = 167$). An analysis of variance (ANOVA) found students who mentioned quests as their favorite part of the CRYSTAL ISLAND experience reported significantly higher ratings of situational interest than those who did not ($F_{(1, 298)} = 12.38, p < .001$). Students stating that they enjoyed completing the quests made comments such as, "*The quests were the best...They kept you active and seeing what's behind the corner...*" and "*My favorite part was the quests you had to do because they teach you, but they are very fun!*" Positive reflections from the students further endorse the motivational advantages for implementing quests.

4 Conclusions

The findings of the study suggest that quests could be effectively utilized to scaffold problem solving in narrative-centered learning environments. Completing more quests during gameplay significantly predicted performance on the content post-test and indicated higher levels of situational interest. Interestingly, quest completion is a better predictor of situational interest than content knowledge. The current analysis has several implications. First, the use of quests appears to aid student learning and problem solving by decomposing problems into smaller, more manageable units. Secondly, quest completions enhance students' situational interest as completing more quests is highly predictive of situational interest and further evidenced by responses to open-ended reflections from the students following their interaction with the environment. Consequently, the data from this study supports this hypothesis, and suggests quests could be a beneficial design tool for scaffolding problem solving.

The limitations of the study should be noted. Most importantly, re-conducting the current analysis with a control condition is imperative for confirming our current findings and implications. Until this study occurs, we cannot make valid claims about the benefits of integrating quest-like activities in similar environments. In addition, it will be important to more closely analyze each quest and subsequently revise each in order to realize the quest's greatest potential as a learning device for the particular concept in on which it is focused. Nonetheless, the results suggest lines of future investigation. As quests in narrative-centered learning environments are focused on one particular aspect of the curriculum, they could potentially form the foundation for an adaptive system targeting learners at the individual level. Quests could be unlocked and presented to students in real-time as the system automatically senses a student's lack of understanding of a certain topic. Moreover, as more advanced students might not need scaffolding, quests could be used as a tool only for less accomplished problem solvers by promoting appropriate challenge at the individual level. Moreover, since students could be challenged to repeat quests as to beat their preceding score, investigating the role of quests for promoting mastery learning is another venue for future research.

Acknowledgments. The authors wish to thank members of the IntelliMedia Group for their assistance. This research was supported by the National Science Foundation under Grant DRL-0822200 and the Graduate Research Fellowship Program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Kim, M.C., Hannafin, J.H.: Scaffolding problem solving in technology-enhanced learning environments (TELEs): Bridging research and theory with practice. *Computers & Edu.* 56, 403–417 (2011)
2. Anderson, L., Krathwohl, D.: *A Taxonomy For Learning, Teaching and Assessing*. Longman, New York (2001)
3. Alfieri, L., Brooks, P., Aldrich, N., Tenenbaum, H.: Does Discovery-Based Instruction Enhance Learning? *J. Edu. Psych.* 103(1), 1–18 (2011)
4. Kalyuga, S., Renkl, A., Paas, F.: Facilitating Flexible Problem Solving: A Cognitive Load Perspective. *Edu. Psych. Review* 22, 175–186 (2010)
5. Kirschner, P., Sweller, J., Clark, R.: Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. *Edu. Psychologist* 41(2), 75–86 (2006)
6. Mayer, R.: Should There Be a Three-Strikes Rule Against Pure Discovery Learning? The Case for Guided Methods of Instruction. *American Psychologist* 59(1), 14–19 (2004)
7. Rowe, J.P., Shores, L.R., Mott, B.W., Lester, J.C.: A Framework for Narrative Adaptation in Interactive Story-Based Learning Environments. In: *Working Notes of the Intelligent Narrative Technologies III Workshop*, Monterey, CA (2010)
8. Jonassen, D.H.: *Learning to solve complex, scientific problems*. Lawrence Erlbaum Associates, Mahwah (2007)
9. Liu, M., Toprac, P., Yuen, T.: What factors make a multimedia learning environment engaging: A case study. In: Zheng, R. (ed.) *Cognitive Effects of Multimedia Learning*, pp. 173–192. Idea, Hershey (2009)
10. Ketelhut, D.J.: The impact of student self-efficacy on scientific inquiry skills: An exploratory investigation in River City, a multi-user virtual environment. *J. Sci. Edu. & Tech.* 16, 99–111 (2007)
11. Hmelo-Silver, C.E.: Problem-Based Learning: What and How do Students Learn. *Edu. Psych. Review* 16(3), 235–266 (2004)
12. Parker, L., Lepper, M.: Effects of Fantasy Contexts on Children’s Learning and Motivation: Making Learning More Fun. *J. Personality & Social Psych.* 62(4), 625–633 (1992)
13. O’Brien, H., Toms, E.: What is user engagement? A conceptual framework for defining user engagement with technology. *J. American Society for Information Sci. & Tech.* 59(6), 938–955 (2008)
14. Schiefele, U.: Topic interest, text representation, and quality of experience. *Contemp. Edu. Psych.* 21(1), 3–18 (1996)
15. Wade, S., Buxton, W., Kelly, M.: Using think-alouds to examine reader-text interest. *Reading Research Quarterly* 34(2), 194–216 (1999)
16. Schraw, G.: Situational interest in literary text. *Contemp. Edu. Psych.* 22, 436–456 (1997)

Exploring Inquiry-Based Problem-Solving Strategies in Game-Based Learning Environments

Jennifer Sabourin, Jonathan Rowe, Bradford W. Mott, and James C. Lester

North Carolina State University, Raleigh, North Carolina, USA
{jlrobiso, jprowe, bwmott, lester}@ncsu.edu

Abstract. Guided inquiry-based learning has been proposed as a promising approach to science education. Students are encouraged to gather information, use this information to iteratively formulate and test hypotheses, draw conclusions, and report their findings. However, students may not automatically follow this prescribed sequence of steps in open-ended learning environments. This paper examines the role of inquiry behaviors in an open-ended, game-based learning environment for middle grade microbiology. Results indicate that students' quantity of information-gathering behaviors has a greater impact on content learning gains than adherence to a particular sequence of problem-solving steps. We also observe that information gathering prior to hypothesis generation is correlated with improved initial hypotheses and problem-solving efficiency.

Keywords: Problem solving, Inquiry-based learning, Game-based learning.

1 Introduction

Inquiry-based learning has been a focus of recent attention in both traditional classrooms [1,2] and intelligent tutoring systems [3,4,5], particularly in science education. There is evidence that inquiry-based learning may only be effective under particular conditions. Students typically need to have some background knowledge in order to learn new material in an inquiry-based setting [1, 2], and they may also require explicit guidance during inquiry-based learning in order to avoid floundering [1,2,5]. There is further evidence that providing guidance about appropriate inquiry behaviors can improve students' future inquiry skills [5].

A variety of approaches to inquiry-based learning have been explored in the intelligent tutoring systems community. For example, Woolf *et al.* have developed the inquiry environment *Rashi*, which supports inquiry skills in a variety of different domains including biology and geology [4]. Students use an inquiry notebook and hypothesis editor to record their observations, reason about findings and support or reject hypotheses. In the *Invention Lab*, students are encouraged to “invent” equations that explain the relationships between variables [3]. *River City* and *Crystal Island* both embed inquiry-based learning within interactive science mysteries in which students are encouraged to gather information about patient symptoms and diagnose a spreading disease in open-ended virtual environments [5,7].

A promising platform for promoting inquiry-based learning is digital game environments. Game-based learning environments have been used for a range of domains, including negotiation skills [8], foreign languages [9], and policy argumentation [6]. Devising effective methods for guiding inquiry-based learning in game environments requires an understanding of students' inquiry strategies in digital games. This paper examines students' inquiry behaviors within a game-based learning environment, as well as inquiry behaviors' relationships with problem solving and learning.

2 CRYSTAL ISLAND Learning Environment

Our work on problem-solving behaviors is situated in CRYSTAL ISLAND, a game-based learning environment for middle grade microbiology [7]. The premise of CRYSTAL ISLAND is that a mysterious illness is afflicting a research team stationed on a remote island. The student plays the role of a visitor who is drawn into a mission to save the research team from the outbreak. The student explores the research camp from a first-person viewpoint and manipulates virtual objects, converses with characters, and uses lab equipment and other resources to solve the mystery. The student is expected to gather information regarding patient symptoms and relevant diseases, form hypotheses based on her findings, use virtual lab equipment and a diagnosis worksheet to record their findings, and share her conclusion with the camp's nurse.

A range of in-game information gathering behaviors are available to students: they can converse with virtual characters about microbiology concepts; they can discuss symptoms and possible transmission sources with sick patients; and they can read virtual posters and books to narrow down which illnesses match the patients' symptoms. As students work towards solving the mystery, they have two primary mechanisms to specify and test their hypotheses. The first mechanism is a virtual laboratory instrument that enables students to test food objects to determine if they are contaminated with pathogens, mutagens or carcinogens. The second method is a diagnosis worksheet that serves as a graphic organizer for recording findings and hypothesized diagnoses. A camp nurse will review the diagnosis worksheet to determine its correctness and provide feedback. This paper examines two primary problem-solving tasks that are critical for solving the mystery: achieving a positive test with the laboratory instrument, and submitting a correct diagnosis worksheet to the camp nurse. In particular, this work investigates how different problem-solving strategies for these tasks relate to content learning gains and in-game problem solving performance.

3 Procedure

A study was conducted with 450 eighth grade students from two North Carolina middle schools. All of the students interacted with the CRYSTAL ISLAND environment. After removing instances of incomplete data, the final corpus included data from 400 students. Of these, there were 194 male and 206 female participants. The average age of the students was 13.5 years ($SD = 0.62$). At the time of the study, the students had not yet completed the microbiology curriculum in their classes.

Participants interacted with CRYSTAL ISLAND in their school classroom, although the study was not part of their regular classroom activities. During the week prior to using CRYSTAL ISLAND, students completed several personality questionnaires and a researcher-generated curriculum test consisting of 19 questions created by an interdisciplinary team of researchers assessing microbiology concepts covered in CRYSTAL ISLAND. During the study, participants were given approximately 55 minutes to work on solving the mystery. Immediately after solving the mystery, or after 55 minutes of interaction, students moved to a different room in order to complete several post-study questionnaires including the curriculum post-test.

In order to understand how students approach problem solving in the game, we consider four key milestones in CRYSTAL ISLAND's problem-solving process: *first laboratory test*, *positive laboratory test*, *first diagnosis worksheet check*, and *correct diagnosis worksheet check*. We are interested in identifying what in-game behaviors typically precede problem-solving milestone achieved by students, and what behaviors occur after completing milestones. Two hypotheses guide this investigation. It is hypothesized that students who spend more time gathering data and reviewing resources prior to their first laboratory test or diagnosis worksheet check will be more effective at solving the mystery (Hyp. 1) and have higher learning gains (Hyp. 2) than students who attempt the problem solving milestones without having gathered much background information. Data gathering behaviors in the context of CRYSTAL ISLAND include talking with characters, viewing posters, reading books, and taking notes.

4 Results

Of the 400 students in the corpus, 320 students were able to perform a positive lab test and 124 students were able to arrive at a correct diagnosis. In our investigation of laboratory test milestones and diagnosis worksheet milestones, we limit our analyses to these respective subsets of students.

4.1 Hypothesis 1 – More Effective Problem Solving

Pearson correlations were calculated to investigate the relationships between different information gathering behaviors and initial problem solving milestones. Metrics of effective problem solving include total number of attempts (i.e., tests conducted with the laboratory instrument or submissions of the diagnosis worksheet) and total time to achieve a successful result (i.e., time taken to perform a lab test that results positive or submit a complete and correct diagnosis worksheet).

Laboratory Tests. Prior to their first laboratory test, students read an average of 1.5 books in the game, looked at 3.9 posters, took 2.7 notes and talked to 3.7 unique virtual characters. On average, 7.3 minutes elapsed between students conducting their first lab test and conducting a positive test. During this time they ran an average of 5.2 total tests. A series of Pearson correlations revealed that students who talked to more unique characters took less time to achieve a successful test, $r(318) = -0.27$, $p < .001$ and ran fewer total tests, $r(318) = -0.14$, $p = .012$. Similarly, students who viewed more posters took less time to achieve a successful test, $r(318) = -0.18$, $p = 0.002$ and ran fewer tests, $r(318) = -0.11$, $p = 0.05$. The number of books read and notes taken were not observed to be significantly correlated with the problem solving metrics.

Table 1. Correlations of data-collection behaviors prior to first diagnosis check. * and ** indicate statistical significance at $p < .05$ and $.01$, respectively.

	<i>Total Problem-Solving Time</i>	<i>Total Number of Attempts</i>	<i>Correctness of First Submission</i>
Books	-0.35**	-0.29**	0.44**
Posters	-0.41**	-0.43**	0.47**
Notes	-0.22*	-0.19*	0.25*
Characters	-0.36**	-0.12	0.28*

Diagnosis Worksheet. Prior to their first diagnosis worksheet check, students read an average of 3.2 books, looked at 7.3 posters, took 3.1 notes and talked to 5.2 unique characters. Students took an average of 10.4 minutes to submit a correct diagnosis after their first attempt, and made an average of 3.5 attempts. Correlations revealed that prior information gathering was associated with more effective problem solving behaviors. Table 1 shows medium-strong correlations between many of the information gathering behaviors, problem-solving time, and number of worksheet checks.

Overall, Hypothesis 1 was supported. Increased data-collection behavior prior to problem-solving attempts was correlated with more effective problem solving. Students spent less time and made fewer total attempts than those who did not engage in information gathering behaviors prior to problem solving.

4.2 Hypothesis 2 – Better Learning Gains

Correlations were calculated between students' information gathering behaviors prior to their first laboratory test and first diagnosis check and normalized learning gains. However, there was no correlation between any of these metrics. The absence of an observed relationship prompted further investigation. When examining the **relationships** between student learning gains and total information gathering behaviors over the entire session, several significant correlations were observed. Conversations with characters, $r(398) = .26$, $p < .001$, looking at posters, $r(398) = .18$, $p < 0.001$, and reading books $r(398) = .18$, $p < .001$ were all positively correlated with normalized learning gains. This suggested that the total number of investigative actions was more associated with students' learning outcomes than when the behaviors were performed.

In order to further investigate this trend, we grouped students into early and late investigators based on the proportion of their information gathering behaviors that occurred prior to their first test or diagnosis check. T-tests between these groups yielded interesting findings. First, it appears that while early investigators are completing more information-gathering prior to problem solving, they are not completing more information gathering across the interaction (Figure 1). Specifically, prior to the first test or first diagnosis check, early investigators have completed significantly ($p < 0.001$) more information gathering behaviors than late investigators. However, at the time of a successful test late investigators have actually completed significantly more information gathering behaviors than their peers, $t(318) = 3.23$, $p = 0.001$. Alternatively, there is no difference in total investigative behaviors between early and late investigators at the time of a successful diagnosis check.

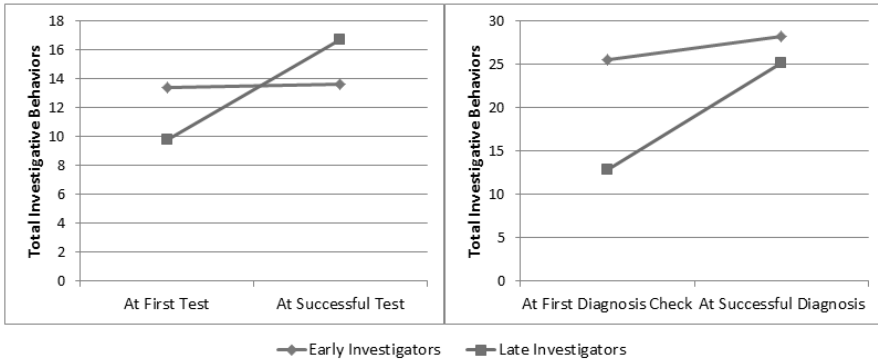


Fig. 1. Change in total investigative behaviors for early and late investigators

Together these findings suggest that no support was observed for Hypothesis 2. Increased information gathering behaviors prior to problem solving does not lead to better learning gains. Instead, total investigative behaviors, and not their timing, is what is important for microbiology content learning. There is evidence that early and late investigators still engage in the same amount of total data-collection behaviors, which accounts for the lack of difference in learning gains between these two groups.

5 Discussion

These findings suggest that students who do not automatically employ effective problem-solving strategies in open-ended game-based learning environments, and problem-solving strategy-use can experience distinct impacts on in-game problem solving and content learning gains. A possible explanation for this study's findings is as follows: the curriculum test primarily assessed microbiology concepts, as opposed to science problem-solving strategies. Students who gathered background information throughout the session benefitted from increased exposure to microbiology content, and these benefits were revealed by the curriculum test. However, gathering information prior to formulating and testing hypotheses was evidence of problem-solving skill. This strategic knowledge was primarily assessed by in-game performance, and not the curriculum test. This explains why effective problem-solving strategy use did not necessarily yield improved performance on a curriculum post-test, but it was associated with improved in-game problem solving outcomes.

The results point toward several promising directions for future work. First, the observation that both early and late investigators perform a comparable number of total information-gathering behaviors raises questions about whether the late investigators learned how to improve their inquiry skills. In fact, there is evidence from other learning systems that repeated exposure to game-based inquiry environments may improve students' inquiry skills [5, 8]. Another important area for future work will be closely examining those students who were unable to complete CRYSTAL ISLAND's problem-solving milestones, and identifying which features separate them from students who

were more successful. It will particularly important to determine what patterns of inquiry behaviors these students exhibit in order to devise intelligent scaffolding techniques to guide their problem solving.

Acknowledgments. The authors wish to thank members of the IntelliMedia Group for their assistance, Omer Sturlovich and Pavel Turzo for use of their 3D model libraries, and Valve Software for access to the Source™ engine and SDK. This research was supported by the National Science Foundation under Grants REC-0632450, DRL-0822200, and IIS-0812291. This material is based upon work supported under a National Science Foundation Graduate Research Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Additional support was provided by the Bill and Melinda Gates Foundation, the William and Flora Hewlett Foundation, and EDUCAUSE.

References

1. Kirschner, P.A., Sweller, J., Clark, R.E.: Why Minimal Guidance during instruction does not work: An analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. *Ed. Psychologist* 41, 75–86 (2006)
2. Alfieri, L., Brooks, P., Aldrich, N., Tenenbaum, H.: Does Discovery-Based Instruction Enhance Learning. *Journal of Education Psychology* 103(1), 1–18 (2011)
3. Roll, I., Aleven, V., Koedinger, K.R.: The Invention Lab: Using a Hybrid of Model Tracing and Constraint-Based Modeling to Offer Intelligent Support in Inquiry Environments. In: Aleven, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6094, pp. 115–124. Springer, Heidelberg (2010)
4. Woolf, B.P., et al.: Critical Thinking Environments for Science Education. In: Proceedings of the 12th Intl. Conference on Artificial Intelligence in Education, pp. 515–522 (2005)
5. Ketelhut, D.J.: The impact of student self-efficacy on scientific inquiry skills: An exploratory investigation in ‘River City’, a multi-user virtual environment. *Journal of Science Education and Technology* 16(1), 99–111 (2007)
6. Easterday, M.W., Aleven, V., Scheines, R., Carver, S.M.: Using Tutors to Improve Educational Games. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS, vol. 6738, pp. 63–71. Springer, Heidelberg (2011)
7. Rowe, J.P., Shores, L.R., Mott, B.W., Lester, J.C.: Integrating Learning, Problem Solving, and Engagement in Narrative-Centered Learning Environments. *International Journal of Artificial Intelligence in Education* (2011)
8. Kim, J., Hill, R.W., Durlach, P., Lane, C., Forbell, E., Core, M., Marsella, S., Pyanadath, D., Hart, J.: BiLAT: A game-based environment for practicing negotiation in a cultural context. *Intl. Journal of Artificial Intelligence in Education* 19(3), 289–308 (2009)
9. Johnson, W.L.: Serious Use of a Serious Game for Language Learning. *International Journal of Artificial Intelligence in Education* 20(2), 175–195 (2010)

Real-Time Narrative-Centered Tutorial Planning for Story-Based Learning

Seung Y. Lee, Bradford W. Mott, and James C. Lester

Department of Computer Science, North Carolina State University
Raleigh, NC 27695, USA
{sylee, bwmott, lester}@ncsu.edu

Abstract. Interactive story-based learning environments offer significant potential for crafting narrative tutorial guidance to create pedagogically effective learning experiences that are tailored to individual students. This paper reports on an empirical evaluation of machine-learned models of narrative-centered tutorial planning for story-based learning environments. We investigate differences in learning gains and in-game performance during student interactions in a rich virtual storyworld. One hundred and eighty-three middle school students participated in the study, which had three conditions: *Minimal Guidance*, *Intermediate Guidance*, and *Full Guidance*. Results reveal statistically significant differences in learning and in-game problem-solving effectiveness between students who received minimal guidance and students who received full guidance. Students in the full guidance condition tended to demonstrate higher learning outcomes and problem-solving efficiency. The findings suggest that machine-learned models of narrative-centered tutorial planning can improve learning outcomes and in-game efficiency.

Keywords: Narrative-centered learning environments, Game-based learning environments, Dynamic Bayesian Networks.

1 Introduction

Recent years have witnessed significant growth in research on interactive story-based learning environments that create engaging and pedagogically effective learning experiences [1,2]. These environments promote students' active participation in engaging story-based problem-solving activities. A number of researchers have explored story-based learning environments for education and training. For example, story-based learning environments can support science education [3], social behavior education [4], and training [5].

Story-based learning environments actively observe students interacting within the storyworld to determine the most appropriate time to intervene with the next tutorial action to perform in service of guiding students' learning experiences. Through this process, story-based learning environments create effective narrative-centered tutorial planning by managing the story structure and scaffolding student interaction.

Given the potential that story-based learning environments have shown, we have developed two empirically driven models of tutorial planning: *tutorial intervention*

planning [6] and *tutorial action planning* [7]. The tutorial intervention model determines when the next tutorial action should occur. The tutorial action model determines which narrative-centered tutorial action to perform. Both models were developed using empirically driven methods. By utilizing a corpus of human interactions within a story-based learning environment, dynamic Bayesian networks (DBN) were learned to model the two types of narrative-centered tutorial planning.

This paper reports on an empirical evaluation of the machine-learned models of narrative-centered tutorial planning for real-time interaction with a story-based learning environment. We investigate differences in learning gains and in-game performance during student interactions. Analyses reveal that the proposed approach offer significant potential for creating efficient learning processes and effective learning outcomes.

2 CRYSTAL ISLAND Story-Based Learning Environment

CRYSTAL ISLAND is a virtual learning environment developed for the domain of microbiology for eighth grade science education featuring a science mystery [3]. To devise accurate computational models of narrative-centered tutorial planning, a Wizard-of-Oz (WOZ) data collection was conducted with a customized version of CRYSTAL ISLAND. Wizards provide the tutorial and narrative planning functionalities while interacting with students in the environment. Throughout the corpus collection, detailed trace data was collected for all wizard decision-making and all navigation and manipulation activities within the virtual environment. The resulting corpus of trace data was utilized to learn the narrative-centered tutorial planning models.

2.1 Integrated Real-Time Model

To explore the real-time effectiveness of the machine-learned models of narrative-centered tutorial planning, the intervention model and the action model were integrated into the CRYSTAL ISLAND story-based learning environment (Fig 1). The environment is identical to the WOZ-enabled CRYSTAL ISLAND except non-player character interactions are driven by the tutorial planning models. Students interact with non-player characters to receive environmental information (e.g., How do you operate the testing equipment? or Where is the library?), and microbiology concepts (e.g., What is a waterborne disease?) using multimodal dialogue. Students select their questions using a dialogue menu and characters respond with spoken language.

Actions generated by the narrative-centered tutorial planning models are primarily initiated via the camp nurse. For example, when the model determines that it is an appropriate time to intervene to help the student examine patient symptoms to solve the mystery, the model directs the camp nurse to walk to the student. The camp nurse, using spoken language, informs the student that she should examine patients to determine their current symptoms. The camp nurse then guides the student to the infirmary to examine the patients.

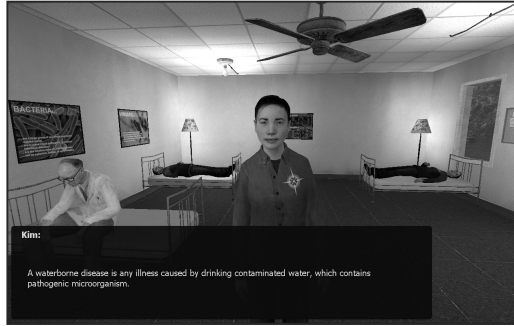


Fig. 1. CRYSTAL ISLAND story-based learning environment with integrated models

To solve the mystery, students complete a *diagnosis worksheet* to organize their hypotheses and record findings about patient symptoms and testing results. Once students have completed their diagnosis worksheets with the source and cause of the illness, they can submit their solutions to the camp nurse for review.

3 Empirical Study

Three experimental conditions were crafted to evaluate the effectiveness of the real-time narrative-centered tutorial planning: *Minimal Guidance*, *Intermediate Guidance*, and *Full Guidance*. Outcomes of the conditions were compared to determine the effectiveness of utilizing our machine-learned models.

Minimal Guidance. Students experience the storyworld controlled by a minimal narrative-centered tutorial planning model. This is a base model that includes the actions that must be achieved by the system (i.e., the user cannot achieve them without the system taking action). The minimal guidance model in this condition is not machine-learned; rather, it simply makes decisions once all pre-conditions are met for an action to be taken.

Intermediate Guidance. Students experience the storyworld controlled by an intermediate narrative-centered tutorial planning model. This is an ablated model inspired by the notions of *islands* [8]. Islands are intermediate plan steps through which all valid solution paths must pass. They have preconditions describing the intermediate world state, and if the plan does not satisfy each island's preconditions, the plan will never achieve its goal. Islands must occur at some intermediate time for achieving the overall goals. In our version of CRYSTAL ISLAND, the transitions between narrative arc phases represent "islands" in our narrative. Each arc phase consists of a number of potential tutorial action decisions; however, the phases are bounded by specific tutorial action decisions that define when each phase starts and ends. We employ these specific tutorial action decisions as our islands. The intermediate guidance tutorial planning employs only eight tutorial action decisions.

Full Guidance. Students experience the storyworld controlled by the full narrative-centered tutorial planning model. The model actively monitors students interacting

within the storyworld to determine when it is appropriate to intervene with the next tutorial decisions to guide students. The model has full control of the tutorial intervention decisions (i.e., determining when to intervene) and tutorial action decisions (i.e., determining what the intervention should be). The full guidance tutorial planning model employs all 15 of the tutorial action decisions described in previous work [7].

3.1 Study Method

A total of 183 students interacted with CRYSTAL ISLAND. Participants were all eighth-grade students from a North Carolina public school ranging in age from 12 to 15 ($M = 13.40$, $SD = 0.53$). Twelve of the participants were eliminated due to hardware and software issues. Another twenty-one participants were eliminated due to incomplete data on either their pre-test or post-test. Among the remaining students, 68 were male and 82 were female.

Students were given 45 minutes to solve CRYSTAL ISLAND's science mystery. Immediately after solving the mystery, or 45 minutes of interaction, whichever came first, students exited the CRYSTAL ISLAND learning environment and completed the post-test. The post-test consisted of the same items as the pre-test, which was completed several days prior to the intervention. The post-test was completed by the students within 30 minutes. In total, the students' sessions lasted no more than 90 minutes.

4 Results

An investigation of overall learning found that students' CRYSTAL ISLAND interactions yielded positive learning outcomes. A matched pairs t -test between post-test and pre-test scores indicates that the learning gains were significant, $t(149) = 2.03$, $p < .05$. Examining the learning outcome for each condition it was found that students' CRYSTAL ISLAND interactions in the *Full Guidance* condition yielded significant learning gains, as measured by the difference of post-test and pre-test scores. A matched pairs t -test revealed that students in the *Full Guidance* condition showed statistically significant learning gains. Students in the *Intermediate* and *Minimal Guidance* conditions did not show significant learning gains (Table 1).

Table 1. Learning gains and t -test statistics

Conditions	Gain Avg.	SD	t	p
<i>Full</i>	1.28	2.66	2.03	< 0.05
<i>Intermediate</i>	0.13	2.69	0.19	0.84
<i>Minimal</i>	0.89	3.12	1.23	0.22

In addition, there was a significant difference between the conditions in terms of learning gains. Controlling for pre-test scores using ANCOVA, the learning gains for the *Full* and *Minimal Guidance* conditions were significantly different, $F(2, 99) = 38.64$, $p < .001$ and the *Full* and *Intermediate Guidance* conditions were also significantly different, $F(2, 100) = 40.22$, $p < .001$. Thus, students in the *Full*

Guidance condition achieved significantly higher learning gains than the students in the other two conditions.

We also conducted in-game problem-solving performance analyses to more closely investigate the effectiveness of the narrative-centered tutorial planning model. In order to compare the behavior of students problem-solving performances among the conditions, we investigated the students' gameplay efficiency by analyzing whether they solved CRYSTAL ISLAND's science mystery and their game completion time. Table 2 reports the game play performance for each condition.

Table 2. In-game problem-solving performances

Conditions	Solved Mystery	Completion Time (s)	
		Mean	SD
<i>Full</i>	92.73 %	1724	417.01
<i>Intermediate</i>	85.42 %	1761	445.66
<i>Minimal</i>	70.21 %	2229	461.55

To analyze the difference in the number of students who solved the mystery among the conditions, a chi-square test was performed. The results showed that the correlation is significant, (likelihood ratio, $\chi^2 = 9.37$, Pearson, $\chi^2 = 9.47$, $p < .01$), indicating that the number of students who solved the mystery varied significantly among the conditions. We also examined the differences in time it took students to solve the mystery. An ANOVA test was performed to investigate the differences among the conditions. The test revealed that differences were significant, $F(2, 122) = 15.13$, $p < .001$, which implied that the total time it took to solve the mystery varied significantly among the different conditions. Tukey's pairwise comparison tests further indicated that the *Full* and *Minimal Guidance* conditions are significantly different ($p < .001$), as well as the *Intermediate* and *Minimal Guidance* conditions ($p < .001$). However, Tukey's test did not reveal any significant differences between the *Full* and *Intermediate Guidance* conditions.

5 Conclusion

Creating narrative-centered tutorial planning is critically important for achieving pedagogically effective story-based learning experiences. We have presented an empirical evaluation of machine-learned models of narrative-centered tutorial planning and investigated differences in learning gains and in-game performance during student interactions with a story-based learning environment. It was found that students in a full guidance condition exhibited significant learning gains and problem-solving performances. Furthermore, a detailed analysis of the differences in learning and in-game problem-solving performance among the conditions showed that there were statistically significant differences between students who received full guidance and students who received intermediate or minimal guidance. The findings suggest that integrated machine-learned models of narrative-centered tutorial planning for real-time interaction can improve learning outcomes and in-game efficiency.

Acknowledgments. The authors wish to thank members of the IntelliMedia Group for their assistance and Valve Corporation for access to the Source™ SDK. Special thanks to Joe Grafsgaard, Kate Lester, and Megan Mott for assisting with the study. This research was supported by the National Science Foundation under Grant DRL-0822200. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Johnson, W.L., Wu, S.: Assessing Aptitude for Learning with a Serious Game for Foreign Language and Culture. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 520–529. Springer, Heidelberg (2008)
2. Rowe, J.P., Shores, L.R., Mott, B.W., Lester, J.C.: Integrating Learning and Engagement in Narrative-Centered Learning Environments. In: Alevén, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6095, pp. 166–177. Springer, Heidelberg (2010)
3. Rowe, J., Mott, B., McQuiggan, S., Robinson, J., Lee, S., Lester, J.: Crystal Island: A Narrative-Centered Learning Environment for Eighth Grade Microbiology. In: Workshop on Intelligent Educational Games at the 14th International Conference on Artificial Intelligence in Education, Brighton, U.K., pp. 11–20 (2009)
4. Aylett, R.S., Louchart, S., Dias, J., Paiva, A., Vala, M.: FearNot! - An Experiment in Emergent Narrative. In: Panayiotopoulos, T., Gratch, J., Aylett, R.S., Ballin, D., Olivier, P., Rist, T. (eds.) IVA 2005. LNCS (LNAI), vol. 3661, pp. 305–316. Springer, Heidelberg (2005)
5. McAlinden, R., Gordon, A., Lane, C., Pynadath, D.: UrbanSim: A Game-Based Simulation for Counterinsurgency and Stability-Focused Operations. In: Workshop on Intelligent Educational Games at the 14th International Conference on Artificial Intelligence in Education, Brighton, U.K., pp. 41–50 (2009)
6. Lee, S.Y., Mott, B.W., Lester, J.C.: Director agent intervention strategies for interactive narrative environments. In: St, M., Thue, D., André, E., Lester, J.C., Tanenbaum, T., Zammito, V. (eds.) ICIDS 2011. LNCS, vol. 7069, pp. 140–151. Springer, Heidelberg (2011)
7. Lee, S.Y., Mott, B.W., Lester, J.C.: Modeling Narrative-Centered Tutorial Decision Making in Guided Discovery Learning. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS, vol. 6738, pp. 163–170. Springer, Heidelberg (2011)
8. Riedl, M., Stern, A., Dini, D., Alderman, M.: Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training. *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning* 3(1) (2008)

An Interactive Teacher's Dashboard for Monitoring Groups in a Multi-tabletop Learning Environment

Roberto Martinez Maldonado¹, Judy Kay¹, Kalina Yacef¹, and Beat Schwendimann²

¹ School of Information Technologies, ² Faculty of Education and Social Work
University of Sydney, Sydney, NSW 2006, Australia
{roberto, judy, kalina}@it.usyd.edu.au,
beat.schwendimann@sydney.edu.au

Abstract. One of the main challenges for teachers in facilitating and orchestrating collaborative activities within multiple groups is that they cannot see information in real time and typically see only the final product of the groups' activity. This is a problem as it means that teachers may find it hard to be aware of the learners' collaborative processes, partial solutions and the contribution of each student. Emerging shared devices have the potential to provide new forms of support for face-to-face collaboration and also open new opportunities for capturing and analysing the collaborative process. This can enable teachers to monitor students' learning more effectively. This paper presents an interactive dashboard that summarises student data captured from a multi-tabletop learning environment and allows teachers to drill down to more specific information when required. It consists of a set of visual real-time indicators of the groups' activity and collaboration. This study evaluates how teachers used the dashboard determine when to intervene in a group. The key contributions of the paper are the implementation and evaluation of the dashboard, which shows a form of learner model from a concept mapping tabletop application designed to both support collaborative learning and capture traces of activity.

Keywords: interactive tabletop, ubiquitous learning environment, collaborative learning, group modelling, data mining, teacher's dashboard, concept mapping.

1 Introduction and Related Work

Working effectively in collaborative settings is increasingly important both for education and work [3]. Given the importance of these skills, teachers ought to encourage enhanced performance by providing effective feedback and implementing strategies to help students to be more aware about their collaborative interactions. One of the main challenges for teachers in *orchestrating* multiple groups working face-to-face is that they need to determine the right moment to intervene and divide their time effectively among the groups[4]. Often teachers only see the final product that does not reveal the processes students followed [15]. This means teachers cannot act effectively as facilitators for the learning of group skills. This is a problem because teachers may find hard to evaluate the collaborative processes, such as the symmetry of participation [3], high quality partial solutions or students' individual contributions.



Fig. 1. Left: Class view of the teacher's dashboard displayed in a handheld device while a group of students build a concept map. Right: The multi-tabletop learning scenario.

Emerging pervasive shared devices, such as interactive multi-touch tabletops, have the potential to support face-to-face collaboration by providing a shared space through which students can have access to digital content while they build a joint solution. Tabletops also open new opportunities for capturing learners' digital footprints offering teachers and researchers the possibility to inspect the collaborative process and recognise patterns of behaviour. However, teachers often do not use quantitative information about student performance to change their strategies, suggesting that teachers need real time information carefully selected and effectively presented [16].

This paper presents a teacher-driven design, implementation, and evaluation of a dashboard for guiding teachers' attention by showing summaries of real time data captured from a tabletop environment (Figure 1). Stephen Few [7] defines dashboards as "*a visual display of the most important information needed to achieve one or more objectives; consolidated on a single screen so the information can be monitored at a glance*". Our dashboard shows a set of visual indicators of collaborative activity generated by means of group models and a data mining technique exploiting tabletop data including: amount and symmetry of learners' physical and verbal activity, the progress of the group towards the goal, the interactions among learners, and domain specific indicators. The main goal is to help teachers gain awareness by visualising selected information that would otherwise remain invisible so they can determine which groups need their attention right away and whether or not to intervene.

There has been significant research exploring data captured from educational table-tops. Fleck *et al.* [8] analysed the conversations that occur among learners working at interactive table-tops and highlighted that both verbal interactions and physical touches ought to be considered to study collaboration. Martinez *et al.* [13] showed how touch data captured from these devices make it possible to analyse collaborative learning, by, for example, mining sequential patterns of interaction that are followed by high achieving groups. VisTaco [17] is a tool that visualises the low-level logged touches of users using distributed table-tops to help researchers to study group dynamics. Verbal participation around non-interactive table-tops has been modelled to create visualisations of patterns of conversation in group decision making [2]. There is also

significant research on designing visual models that reveal associations between observable patterns and quality of group work. Erickson *et al.* [6] showed the benefits of visually representing the chat conversation of a group for self-regulation. Donath [5] displayed participation in the visualisation of online group activities using a Loom visualisation. Kay *et al.* [9] created a set of visualisations to identify anomalies in online team work by mirroring aspects such as participation, interaction and leadership. The most similar research to ours was conducted by AlAgha *et al.* [1] who built a tool through which teachers can interact with groups and monitor multi-tabletop classrooms. Our work goes beyond previous work by introducing a novel approach to model and visualise aspects of collaboration unobtrusively captured from an interactive tabletop environment to support teacher guidance.

2 The Tabletop-Based Learning Environment: Concept Mapping

This study used an updated version of a collaborative concept mapping tabletop application [11] (Figure 2). *Concept mapping* [14] is a technique through which learners can represent their understanding about a topic in a graphical manner. A concept map includes short words that represent objects, processes or ideas (called *concepts*, e.g. protein, milk). Two concepts can be linked to create a statement (called *proposition* e.g. milk *contains* protein).

Fifteen university students participated in the case study. They were assigned to groups of three and knew each other. First, learners were asked to read the same text about the learning domain (healthy nutrition) and build their individual concept maps in private using a desktop tool (CMapTools [14]). Then, learners came to the tabletop to integrate their perspectives into a collaborative concept map (see Figure 2, right). The activity was semi-structured in four stages: i) individual concept mapping (external to the tabletop); ii) collaborative brainstorming of the concepts for the joint map; iii) adding propositions that learners had in common, and iv) the discussion phase, where learners create the rest of the propositions, by negotiating different views. They had 30 minutes for building individual maps and up to 30 minutes for the collaborative stage at the tabletop. All sessions were video recorded. At the tabletop, learners could add concepts from individual lists of concepts from the individual stage; create new links and concepts, edit propositions and have access to their individual maps.

The tabletop hardware itself cannot distinguish between users. An overhead depth sensor (www.xbox.com/kinect) was used to track the position of each user and automatically identify who did each touch. Frequency of individual verbal contributions was recorded through a microphone array (www.dev-audio.com) located above the tabletop and which distinguishes who is speaking [10].



Fig. 2. The collaborative concept mapping tabletop application. Left: Two propositions. Center: Three learners working together. Right: Integrating propositions from the individual map.

3 The Interactive Teacher's Dashboard

It is challenging to define ways to present the information about group collaboration in a manner that is readily understood and useful for educators. For this reason we decided to include teachers experienced in classroom collaboration in early stages of the dashboard design. Features that classroom experts believed should be in a truly effective educational awareness tool included features for: identifying learners who are not contributing to the group who are dominating and controlling the activity; groups that work independently; or that do not understand the task. The dashboard was designed to enable teachers to determine whether groups or individual learners need attention, by showing the symmetry of activity, degree of interaction with others' contributions and overall progress of the task. Four teachers were involved in the *teacher-driven* design process that consisted of an iterative series of interviews, prototypes and empirical evaluations of both the visualisations and the structure of the dashboard. The final result was a dashboard with 2 levels of detail: 1) the *class level*, shows very summarised information about each of the groups so teachers can use it in real time to see several groups at once during a classroom session (Figure 1, right), and ii) the *detailed group level*, that permits in depth exploration of a specific group's activity.

3.1 The Class Level: Accumulated Summaries of Each Group Activity

The *class level* of the dashboard aims to give minimal information needed for a teacher to gain an overview of the overall activity of each group. This layer displays sets of three visualisations per group. We now explain the design of each of these.

Mixed radar of participation. Groups in which learners participate asymmetrically are often associated to cases of free-riding or disengagement while collaborative groups tend to allow the contribution of all members [3]. This radar models the cumulated amount and symmetry of physical and verbal participation (Figure 3 - 1). The triangles (red and blue) depict the number of touches and amount of speech by each learner. Each coloured circle represents a student. The closer the corner of the triangle is to the circle, the more that student was participating. If the triangle is equilateral it means that learners participated equally.

Graph of interaction with others' objects. Studies with students working at tabletops have confirmed that interacting with what others' have done may trigger further

discussion that is beneficial for collaboration [8]. This graph models the cumulated number of interactions by each learner with other students’ objects at the tabletop (Figure 3-2). The size of the circles indicates the amount of physical activity (touches) by each learner. The width of the lines that link these circles represents the number of actions that the learners performed on the concepts or links created by other learners.

Indicator of detected collaboration. This visualisation shows the “level of collaboration” detected by the system as a summary of group health. It is based on a mathematical model developed by Martinez *et al.* [12] using the data mining prediction Best-First tree algorithm. It classifies each block of half a minute of activity according to a number of features that can be captured from collocated settings. They are: number of active participants in verbal discussions, amount of speech, number of touches and symmetry of activity measured with an indicator of dispersion (Gini coefficient). The system labels each 30 second episode as one of three possible values: Collaborative, Non-collaborative, or Average. The visualisation shows the accumulation of these labelled episodes. The arrow bends to the right if there are more “collaborative” episodes or to the left if there are more “non collaborative” episodes (see Figure 3-3).

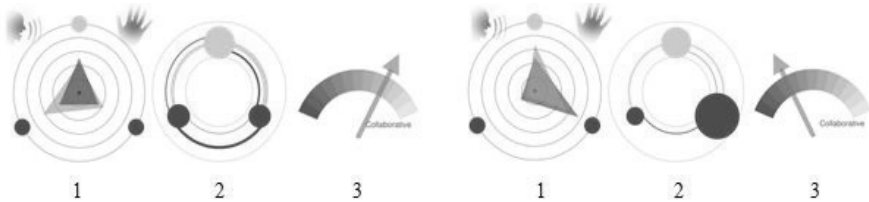


Fig. 3. Overview visualisations. Left: a balanced group (Group A). Right: a group in which one member (red circles) was completely disengaged from the activity (Group D).

3.2 The Detailed Group Level: Detailed Timeline Summaries for a Specific Group

The group level visually depicts information over time for *post-mortem* analysis. This level is accessed by touching the set of visualisations of a specific group in the *class level*. It includes the next five visualisation types.

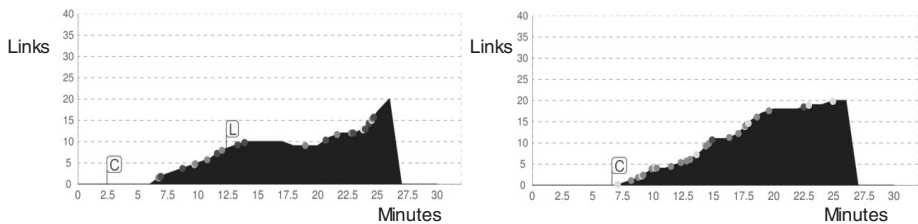


Fig. 4. Evolution of the group map. Left: A group with a dominant student (red) and a low participant student (yellow) (Group C). Right: A group with a low participant (red) (Group D).

Evolution of the group map. This visualisation shows the contributions of group members towards the group map, by displaying the number of propositions (links) created and their authors, along the time line (Figure 4). The small coloured circles indicate a “create link” event generated by the learner identified by that colour. In this way a teacher can become aware of dominant participants, see patterns of alternating contributions or whether all members contribute to the concept map evenly. The red flags (C, L) indicate the stages that students explicitly started: The first stage is brainstorming starting from minute 0 (not flagged). C= adding propositions learners have in *Common*, L= *Main Linking* phase. This is the only visualisation of the dashboard that is coupled with the concept mapping task.

Timeline of interaction with other learners' objects. This visualisation depicts the amount of interaction by each learner with others' objects. Each coloured horizontal line represents a learners' timeline. Each vertical line represents an interaction of that learner with other learners' objects. Figure 5 (left) shows the interactions of a group in which one learner (Alice, red coloured) dominated the physical interactions with her peers (Bob and Carl, green and yellow). Figure 5 (right) shows a group where learners hardly built upon other's ideas, as there are very few interactions.

Radars of verbal and physical participation in the timeline. These visualisations model the amount and symmetry of verbal (Row 1, Figure 6) and physical participation (Row 2, Figure 6) of each group member. Similarly to the cumulative radars described in the previous section, if the corner of the triangle is closer to the centre (black dot), that means the corresponding learner's activity was low.

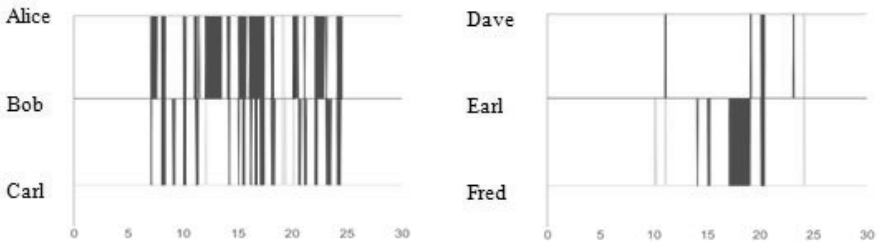


Fig. 5. Timeline of interaction with other learners' objects. Left: A group with a dominant learner (Group C). Right: group members that worked independently (Group B).

Contribution charts. These visualisations model the dimension of the concept map in the tabletop in terms of propositions. They also show the distribution of the individual contribution to the group concept map. The size of the charts indicates the number of links in the concept map. In the dashboard, these visualisations cover 4-5 minutes of activity. Therefore multiple visualisations are shown in the timeline (Row 3, Figure 6).

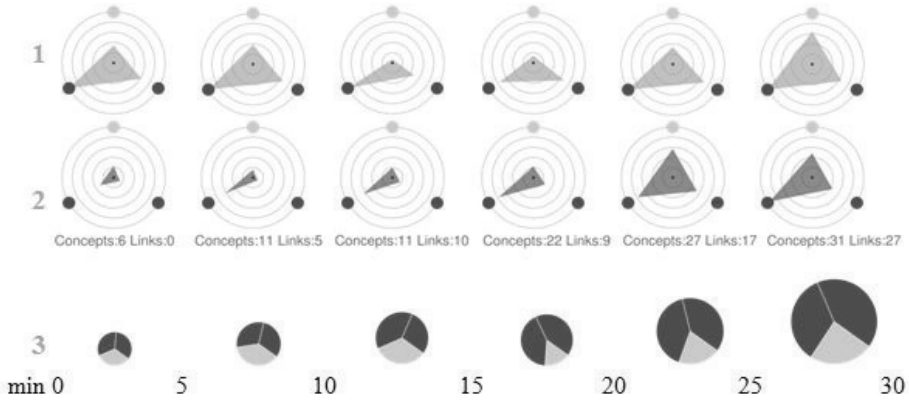


Fig. 6. Radars of verbal participation (Row 1), radars of physical participation (Row 2) and Contribution charts (Row 3) of a group with a dominant student-red coloured (Group C)

4 Evaluation

We aimed to evaluate two research questions: (Q1) *Is the class level of the dashboard useful for teachers to decide when to intervene or which groups need their attention?* (Q2) *Which visualisations (in both levels) do teachers use to decide whether groups need attention?*

Eight teachers experienced in small-group classroom collaboration participated in the evaluation sessions. None had been involved in the design of the dashboard. The data recorded from four groups, each with three students, was used. Groups were cross-distributed among teachers so that each teacher monitored three groups at the same time and each group was monitored by six teachers. The system simulated the real time generation of data for the teacher, as if he or she was monitoring three groups during 30 minutes. This version of the dashboard presents up to three groups at the same time. In parallel, each group video was manually analysed by an external person to diagnose groups' collaboration and have a baseline reference of group performance. Based on these observations, groups can be described as follows: Group A performed best in terms of collaboration. Students discussed their ideas, worked together to build the group concept map. They completed the task sooner than the other groups and their final solution was simpler. By contrast, members of Group B worked independently most of the time, building three different concept maps rather than combining perspectives into a shared map. Group C was distinguished by the dominance of a single student, who led the discussion, took most of the decisions and ended up building most of the group map without considering others' perspectives. In Group D, only two learners collaborated to merge their ideas. The third learner did not contribute to the group effort and had lower levels of participation – *free-riding*.

The evaluation recreated the classroom orchestration loop documented by Dillenbourg *et. al.* [4]: teachers monitor the classroom, compare it to some desirable state, and intervene to mentor students. This was adopted as follows: (1) First, teach-

ers were asked to think aloud as they were looking at the *class level* of the dashboard, verbalising their perception of each visualisation. (2) Then, they were asked to state whether each group was collaborating. (3) As appropriate, they would select the visualisations that indicated that a group might have *anomalies* in terms of collaboration. (4) As appropriate, they would choose one group (or none) that they would attend to, indicating which visualisations helped them to take such decision. (5) As a response, the system drills down from the *class level* to the selected *detailed group level* of the dashboard. (6) Then, teachers were requested to think aloud, stating the visualisations that helped them to confirm possible *anomalies* and whether they would talk with the group members or provide corrective feedback. If the teacher decided to intervene they had to wait at least 2 minutes in this layer without viewing other groups (simulating the time taken to talk with the group). Teachers followed this loop throughout the 30 minutes duration of the trials. Finally, they were asked to answer a short questionnaire to validate that they understood the visualisations. Data captured from the teacher dashboard usage sessions were recorded and analysed.

5 Results and Discussion

(Q1) *Is the class level of the dashboard useful for teachers to decide when to intervene or which groups need their attention?* This research question drove the study. Our objective is to help teachers recognise potential issues within the groups so they can be more aware about which group needs attention. Table 1 shows the two main evaluation aspects: which group teachers would visit next and why (*attention*), and if they would either intervene or let the group continue working (*intervention*). During the experiment *attention* was indicated when teachers navigated from the *class level* to the *detailed group level* of the dashboard. *Interventions* were indicated when, after analysing the group level of the dashboard, teachers felt that the group still needed to take corrective actions to improve collaboration.

Results indicated that teachers would focus most of their attention on groups B and D (investing 44% and 40% of their time on average in them). They correctly identified independent work and the presence of a free-rider as their major issues. They indicated interventions would have served to encourage students to work more collaboratively and share their ideas with others (on average 4 interventions out of 7 moments of attention and 3 interventions out of 6 moments of attention respectively per teacher). Group B claimed a similar degree of attention (13% of intervention out of the 31% of attention per tutor). In fact, the difference in the *attention* across these three groups was not significant ($p > 0.05$). However, for all of the tutors, Group A was clearly performing well and teachers would not have intervened (average of 2 visits and 0.7 interventions per tutor). The attention provided to other groups compared with Group A was statistically significant ($p < .00027$, two-tailed). Inter-tutor agreement was calculated to examine how different the observations. Table 1, Column k (Cohen's kappa) shows that the 6 tutors who monitored each group agreed on *which* group needed intervention and *when* they needed it either at the beginning, in the middle or by the end of the task- $k > 0.4$.

Table 1. Teachers attention and interventions per group. Att= Average number of times each tutor decided to monitor that group. Att%=Average proportion of moments dedicated to that group. Int=Average number of interventions. Int%=Average proportion of interventions. k= Inter tutor agreement (Cohen's kappa).

Group	Attention		Interventions		K	Observations based on the videos
	Att	Att%	Int	Int%		
A	2 (s=1)	15% (s=7)	1 (s=0.5)	4% (s=3.4)	0.7	Even group
B	7 (s=2)	44% (s=7)	4 (s=1.4)	21% (s=6)	0.4	Independent work
C	5 (s=1)	31% (s=6)	2 (s=0.6)	13% (s=3)	0.5	Dominant student
D	6 (s=3)	40% (s=13)	3 (s=1.7)	19% (s=8)	0.5	Free-rider

(Q2) *Which visualisations (at both levels) do teachers use to decide whether groups need attention?* Based on the think aloud analysis of the *class level* visualisations, we found that teachers agreed on the usefulness of the *mixed radar of participation* and the *chart of interactions with others' objects* graphs. These provided them with enough information to identify possible problems within certain groups. Some tutors indicated that the third graph, *indicator of detected collaboration*, was useful only to *confirm* their observations using the first two charts. Table 2 shows that teachers obtained more information from the two first visualisations (85 and 65 detected issues) and started to use them from the beginning of the activity. They identified the main anomalies of groups B, C and D describing the main problems with the groups: independent work and a low participant for Group B, a dominant student in Group C and a free-rider in Group D. They were not concerned about Group A (Table 1, 15% for *Attention*). Four out of 6 tutors indicated that Group A progressed quickly and finished the activity quickly, so in a real scenario they would have encouraged them to explore more ideas to complete their work. Teachers indicated that the detailed timeline level of the dashboard provided information about the *progress* of each group. All agreed that this level would become an important tool for after-class analysis but the *class level* of the dashboard provides enough information to identify possible anomalies during a classroom session. Table 2 shows that tutors tended to use all the *timeline* visualisations in combination to detect *issues* (usage between 22 and 36). However, it does not provide useful information during the first 10 minutes of the activity while the *class level* provides rich information from beginning to end of the activity (Table 2, column Min 10). Our analysis indicates that teachers could identify the major groups anomalies based on the *class level* and confirm them after looking at the *detailed group level*. Visualisations were understood by teachers (96% of correct answers in post-study questionnaires) and helped them divide their attention effectively according to groups' needs. Quantitative data does not provide details of group's collaboration but it provided information for teachers to infer whether groups were potentially engaged in non-collaborative activity.

Table 2. Potential group anomalies identified by teachers using each visualisation

Visualisation	Total	Min 10	Min 20	Min 30
Level 1 – Class				
Mixed radar of participation (audio and touches)	85	36	23	26
Chart of interactions with others' objects	65	18	29	18
Indicator of detected collaboration	26	8	6	12
Level 2 – Detailed group				
Evolution of the group map	22	1	8	13
Timeline of interactions with other's objects	35	3	18	14
Radars of verbal participation in the timeline	31	8	13	10
Radars of physical participation in the timeline	36	7	15	14
Contribution charts	26	7	7	12

6 Conclusions and Further Work

The goal of this research is to present real time data from interactive tabletops, combined with data mining results, in an interactive dashboard that helps teachers monitor group activities at a multi-tabletop learning environment. We present the design and evaluation of the teacher dashboard that shows information at two levels: a class summary and a detailed group timeline. Evaluation results indicate that the dashboard allowed teachers to effectively detect which groups encountered problems in terms of collaboration. The *class level* of the dashboard provided information from the beginning of the activity and was used as a decision making tool to help teachers manage their attention and interventions. The *detailed group level* shows chronological information that was considered effective for assessing task progress after class. Our evaluation is limited to pre-recorded data for the purpose of repeatability. Most of the visualisations contained in the dashboard can be generalised to other domains. A follow-up study can include a real study that analyses reactions from students to teacher's interventions. Future research will evaluate this tool in a real classroom and explore ways to integrate the dashboard into teachers' strategies and experience.

References

1. Alagha, I., Hatch, A., Ma, L., Burd, L.: Towards a teacher-centric approach for multi-touch surfaces in classrooms. In: Interactive Tabletops and Surfaces, ITS 2010, pp. 187–196 (2010)
2. Bachour, K., Kaplan, F., Dillenbourg, P.: An Interactive Table for Supporting Participation Balance in Face-to-Face Collaborative Learning. *IEEE Transactions on Learning Technologies* 3, 203–213 (2010)
3. Dillenbourg, P.: What do you mean by 'collaborative learning'? In: Collaborative Learning: Cognitive and Computational Approaches. *Advances in Learning and Instruction*, pp. 1–19. Elsevier Science (1998)

4. Dillenbourg, P., Zufferey, G., Alavi, H., Jermann, P., Do-Lenh, S., Bonnard, Q.: Classroom orchestration: The third circle of usability. In: CSCL 2011, pp. 510–517 (2011)
5. Donath, J.: A semantic approach to visualizing online conversations. *Communications ACM* 45, 45–49 (2002)
6. Erickson, T., Smith, D., Kellogg, W., Laff, M., Richards, J., Bradner, E.: Socially translucent systems: social proxies, persistent conversation, and the design of “babble”. In: SIGCHI 1999, pp. 72–79. ACM (1999)
7. Few, S.: *Information dashboard design: The effective visual communication of data*. O’Reilly Media, Inc. (2006)
8. Fleck, R., Rogers, Y., Yuill, N., Marshall, P., Carr, A., Rick, J., Bonnett, V.: Actions Speak Loudly with Words: Unpacking Collaboration Around the Table. In: *Interactive Tabletops and Surfaces, ITS 2009*, pp. 189–196 (2009)
9. Kay, J., Maisonneuve, N., Yacef, K., Reimann, P.: The Big Five and Visualisations of Team Work Activity. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006. LNCS*, vol. 4053, pp. 197–206. Springer, Heidelberg (2006)
10. Martinez, R., Collins, A., Kay, J., Yacef, K.: Who did what? who said that? Collaid: an environment for capturing traces of collaborative learning at the tabletop. In: *Interactive Tabletops and Surfaces, ITS 2011* (2011)
11. Martinez, R., Kay, J., Yacef, K.: Collaborative concept mapping at the tabletop. In: *Interactive Tabletops and Surfaces, ITS 2010*, pp. 207–210 (2010)
12. Martinez, R., Wallace, J.R., Kay, J., Yacef, K.: Modelling and Identifying Collaborative Situations in a Collocated Multi-display Groupware Setting. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011. LNCS*, vol. 6738, pp. 196–204. Springer, Heidelberg (2011)
13. Martinez, R., Yacef, K., Kay, J., Kharrufa, A., Al-Qaraghuli, A.: Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. In: *EDM 2011*, pp. 111–120 (2011)
14. Novak, J., Cañas, A.: *The Theory Underlying Concept Maps and How to Construct and Use Them*. In: 2006-01 TRIC (ed) Florida Institute for Human and Machine Cognition (2008)
15. Race, P.: *A briefing on self, peer & group assessment*. Learning and Teaching Support Network, York (2001)
16. Segedy, J., Sulcer, B., Biswas, G.: Are ILEs Ready for the Classroom? Bringing Teachers into the Feedback Loop. In: Aleven, V., Kay, J., Mostow, J. (eds.) *ITS 2010. LNCS*, vol. 6095, pp. 405–407. Springer, Heidelberg (2010)
17. Tang, A., Pahud, M., Carpendale, S., Buxton, B.: VisTACO: visualizing tabletop collaboration. In: *Interactive Tabletops and Surfaces, ITS 2010*, pp. 29–38 (2010)

Efficient Cross-Domain Learning of Complex Skills

Nan Li, William W. Cohen, and Kenneth R. Koedinger

School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh PA 15213 USA
{nli1,wcohen,koedinger}@cs.cmu.edu

Abstract. Building an intelligent agent that simulates human learning of math and science could potentially benefit both education, by contributing to the understanding of human learning, and artificial intelligence, by advancing the goal of creating human-level intelligence. However, constructing such a learning agent currently requires significant manual encoding of prior domain knowledge; in addition to being a poor model of human acquisition of prior knowledge, manual knowledge-encoding is both time-consuming and error-prone. Recently, we proposed an efficient algorithm that automatically acquires domain-specific prior knowledge in the form of deep features. We integrate this deep feature learner into a machine-learning agent, SimStudent. To evaluate the generality of the proposed approach and the effect of integration on prior knowledge, we carried out a controlled simulation study in three domains, fraction addition, equation solving, and stoichiometry, using problems solved by human students. The results show that the integration reduces SimStudent's dependence over domain-specific prior knowledge, while maintains SimStudent's performance.

Keywords: deep feature learning, learner modeling, transfer learning.

1 Introduction

Education in the 21st century will be increasingly about helping students not just to learn content but also to become better learners. In order to achieve this goal, we need to better understand the process of human knowledge acquisition and how students are different in their abilities to learn. Hence, a considerable amount of research (e.g., [6,1,5,7]) has been carried out in building intelligent agents that model human learning of math and science. Although such agents are able to produce intelligent behavior requiring less knowledge engineering than before, agent developers still need to encode a nontrivial amount of domain-specific prior knowledge. Such manual encoding of prior knowledge can be time-consuming and error-prone. Moreover, providing domain-specific prior knowledge to the intelligent agents is less cognitively plausible, as students do not necessarily know such prior knowledge before class. An intelligent system that models automatic

knowledge acquisition without domain-specific prior knowledge could be helpful both in reducing the effort in knowledge engineering intelligent systems and in advancing the cognitive science of human learning.

Previous work in cognitive science [2] showed that one of the key factors that differentiates experts and novices in a field is that experts view the world in terms of deep functional features (e.g., coefficient and constant in algebra, molecular ratio in stoichiometry), while novices only view it in terms of shallow perceptual features (e.g., integer in an expression). We [3] have recently developed a learning algorithm that acquires deep features automatically with only domain-independent knowledge (e.g., what is an integer) as input. We integrate this deep feature learning algorithm into a machine-learning agent, *SimStudent* [5], to let it have this major component of human expertise acquisition. To evaluate how deep feature learner affects learning performance as well as prior knowledge requirement, we carried out a controlled simulation study in three math and science domains: fraction addition, equation solving, and stoichiometry.

2 A Brief Review of SimStudent

SimStudent is a machine-learning agent that inductively learns skills to solve problems from demonstrated solutions and from problem solving experience. In the rest of this section, we will briefly review SimStudent. For full details, please refer to [4]. In this paper, we will use stoichiometry as an illustrative example. Stoichiometry is a branch of chemistry that deals with the relative quantities of reactants and products in chemical reactions. In the stoichiometry domain, SimStudent is asked to solve problems such as “How many moles of atomic oxygen (O) are in 250 grams of P_4O_{10} ? (Hint: the molecular weight of P_4O_{10} is $283.88\text{ g }P_4O_{10} / \text{mol }P_4O_{10}$).”

During the learning process, given the current state of the problem (e.g., *1 mol COH₄ has ? mol H*), SimStudent first tries to propose a plan for the next step (e.g., (*bind ?element (get-substance “? mol H”)*) (*bind ?output (molecular-ratio “1 mol COH₄” ?element)*)) based on the skill knowledge it has acquired. If it finds a plan and receives positive feedback, it continues to the next step. If the proposed next step is incorrect, the tutor sends negative feedback to SimStudent and demonstrates a correct next step. Then, SimStudent attempts to learn or modify its skill knowledge accordingly. If it has not learned enough skill knowledge and fails to find a plan, a correct next step is directly demonstrated to SimStudent.

Based on the demonstration, SimStudent learns a set of production rules as its skill knowledge. The left side of Figure 1 shows an example of a production rule learned by SimStudent in a readable format¹. A production rule indicates “where” to look for information in the interface, “how” to change the problem state, and “when” to apply a rule. For example, the rule to “calculate how many moles of H are in 1 mole of COH_4 ” shown at the left side of Figure 1 would be read as “given the current value (*1 mol COH₄*) and the question (*? mol H*),

¹ The actual production rule uses a LISP format.

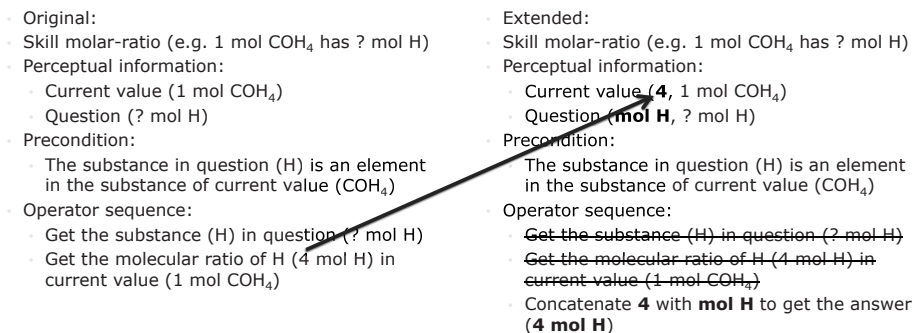


Fig. 1. Original and extended production rules for divide in a readable format

when the substance in question (H) is an element in the substance (COH_4), then get the substance in question (H), and compute the molecular ratio of H (4 mol H) in COH_4 .

3 A Brief Description of Integrating Deep Feature Learning into SimStudent

To learn the “how” part in the production rules, SimStudent requires a set of operator functions given as prior knowledge. For instance, (*molecular-ratio ?val1 ?val2*) is an operator function. It generates the number of moles of an individual substance that each mole of input substance has, based on molecular ratio of input substance. There are two groups of operator functions: domain-specific operator functions (e.g., (*molecular-ratio ?val1 ?val2*)) and domain-general operator functions (e.g., (*copy-string ?val*)). Domain-specific operator functions are more complicated skills, which human students may not know in advance.

Many of the domain-specific operator functions are extraction operators that extract deep features from the input. In order to reduce SimStudent’s dependence on such domain-specific operator functions, we use a deep feature learner [3] to acquire the deep features automatically, and then extend the “where” (perceptual information) part to include these deep features as needed. As presented at the right side of Figure 1, in addition to the original current value *1 mol COH₄* and the question *? mol H*, SimStudent automatically adds the molecular ratio of H (4) into the perceptual information part. Then, the “how” (operator sequence) part does not need the three domain-specific operators any more. Instead, SimStudent can directly concatenate the molecular ratio (4) with the rest part in question (*mol H*).

Here are a few more examples to demonstrate how the extended “where” part enables the removal of domain-specific operator functions, while maintaining efficient skill knowledge acquisition. Figure 2 shows the parse trees of example input strings acquired by the deep feature learner. The deep features are associated with nonterminal symbols in the parse trees.

In fraction addition, one of the important operator functions in this domain is getting the denominator of the addend (i.e., (*get-denominator ?val*)).

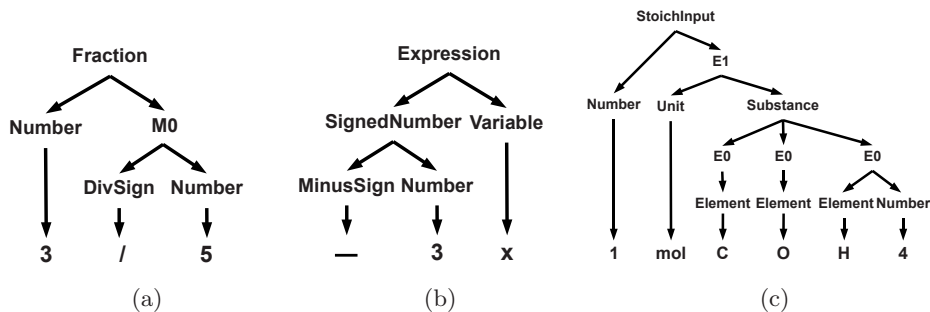


Fig. 2. Example parse trees learned by the deep feature learner in three domains, a) fraction addition, b) equation solving, c) stoichiometry

Figure 2(a) shows an example parse tree for $3/5$. The extended SimStudent can directly get the denominator 5 from the non-terminal symbol *Number* in rule $M0 \rightarrow 1.0, DivSign, Number$. Then, the operator function (*get-denominator ?val*) is replaced by a more general operator function (*copy-string ?val*). Another important domain-specific operator function in equation solving is getting the coefficient of some expression (i.e., (*get-coefficient ?val*)). With the deep feature learner, the coefficient of an expression can be extracted by directly taking the signed number (i.e., *SignedNumber*) in rule $Expression \rightarrow 1.0, SignedNumber, Variable$. Again, the domain-specific operator function (*get-coefficient ?val*) is replaced by the domain-general operator function (*copy-string ?val*). As mentioned before, (*molecular-ratio ?val0 ?val1*) is a domain-specific operator function used in stoichiometry. Instead of programming this operator function, after integrated with deep feature learning, the output can now be generated by taking the “*Number*” in grammar rule $E0 \rightarrow 0.5 Element, Number$, and then concatenating with the unit *mol* and the individual substance “*Element*”. Thus, the original operator function (*molecular-ratio ?val0 ?val1*) is replaced by the domain-general operator function concatenation (i.e., (*concat ?val2 ?val3*)).

4 Experimental Study

To further quantitatively evaluate the amount of required prior knowledge encoding and the learning effectiveness of SimStudent, we carried out a controlled simulation study in the above three domains: fraction addition, equation solving, and stoichiometry.

Methods: We compare three versions of SimStudent: two original SimStudents without deep feature learning, and one extended SimStudent with deep feature learning. One of the original SimStudents is given both domain-general and domain-specific operator functions (*O+Strong Ops*). The other is given only domain-general operator functions (*O+Weak Ops*). The extended SimStudent is also only given domain-general operator functions (*E+Weak Ops*).

Table 1. Number of training problems and testing problems presented to SimStudent

Domain Name	# of Training Problems	# of Testing Problems
Fraction Addition	40	6
Equation Solving	24	11
Stoichiometry	16	3

In each domain, the three SimStudents are trained on 12 problem sequences over the same set of problems in different orders. Both training and testing problems are gathered from classroom studies on human students. SimStudent is tutored by automatic tutors that are similar to those used by human students. The number of training and testing problems is listed in Table 1.

We evaluate the performance of SimStudent with two measurements. We use the number of domain-specific and domain-general operator functions used in three domains to measure the amount of prior knowledge engineering needed. In addition, we count the number of lines of Java code developed for each operator functions, and use this as a secondary measurement to assess the amount of knowledge engineering. To assess learning effectiveness, we define a *step score* for each step in the testing problem. Among all next steps proposed by SimStudent, we count the number of next steps that are correct, and compute the step score as the number of correct next steps proposed divided by the total number of correct steps plus the number of incorrect next steps proposed. This measurement evaluates the quality of production rules in terms of both precision and recall.

Experimental Results: Not surprisingly, only the original SimStudent given the strong set of operator functions (*O+Strong Ops*) uses domain-specific operator functions. Across three domains, it requires at least as many operator functions as the extended SimStudent without domain-specific operator functions (*E+Weak Ops*). Moreover, since domain-specific operator functions are not reusable across domains, the original SimStudent with domain-specific operator functions (*O+Strong Ops*) requires nearly twice as many operator functions (31 vs. 17) as that of the extended SimStudent (*E+Weak Ops*) needed. The total number of lines of code required for the operator functions used by the extended SimStudent (*E+Weak Ops*) is 645, whereas the total number of lines of code programmed for the original SimStudent (*O+Strong Ops*) is 6789, which is more than ten times the size of the code needed by the extended SimStudent.

Learning curves of the three SimStudents are presented in Figure 3. Across three domains, without domain-specific prior knowledge, the original SimStudent (*O+Weak Ops*) is not able to achieve a step score more than 0.3. Given domain-specific operator functions, the original SimStudent (*O+Strong Ops*) is able to perform reasonably well. It obtains a step score around 0.85 in equation solving. However, its performance is still not as good as the extended SimStudent. Given all training problems, the extended SimStudent (*E+Weak Ops*) performs slightly better than the original SimStudent with domain-specific prior knowledge in equation solving. It (*E+Weak Ops*) achieves significantly ($p < 0.0001$) better step scores than the original SimStudent given domain-specific operator functions

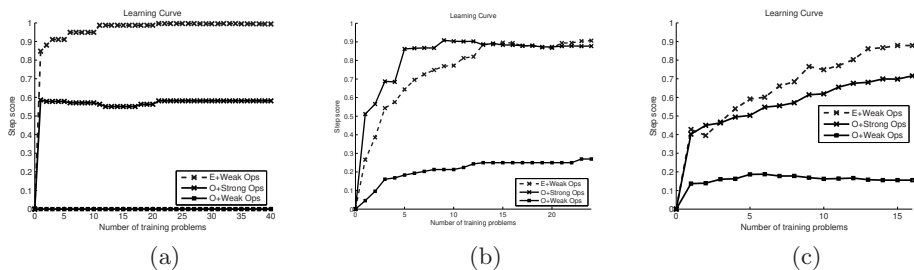


Fig. 3. Learning curves of three SimStudents in three domains, a) fraction addition, b) equation solving, c) stoichiometry

(*O+Strong Ops*) in two other domains. Hence, we conclude that the extended SimStudent acquires skill knowledge, which is as or more effective than the original SimStudent, while requiring less prior knowledge engineering.

5 Concluding Remarks

To summarize, we presented a novel approach that integrates a deep feature learner into a simulated student, SimStudent, and demonstrated with examples how the integrated deep feature learner reduces prior knowledge engineering effort across three domains. We then carried out a controlled simulation study to quantitatively measure the amount of prior knowledge engineering and the learning efficiency, and showed that the extended SimStudent achieved better or comparable performance than the original SimStudent, without requiring encoding of domain-specific prior knowledge.

References

1. Anzai, Y., Simon, H.A.: The theory of learning by doing. *Psychological Review* 86(2), 124–140 (1979)
2. Chi, M.T.H., Feltovich, P.J., Glaser, R.: Categorization and representation of physics problems by experts and novices. *Cognitive Science* 5(2), 121–152 (1981)
3. Li, N., Cohen, W.W., Koedinger, K.R.: A Computational Model of Accelerated Future Learning through Feature Recognition. In: Alevan, V., Kay, J., Mostow, J. (eds.) *ITS 2010*. LNCS, vol. 6095, pp. 368–370. Springer, Heidelberg (2010)
4. Li, N., Cohen, W.W., Koedinger, K.R.: Integrating representation learning and skill learning in a human-like intelligent agent. Tech. Rep. CMU-MLD-12-1001, Carnegie Mellon University (January 2012)
5. Matsuda, N., Lee, A., Cohen, W.W., Koedinger, K.R.: A computational model of how learner errors arise from weak prior knowledge. In: *Proceedings of Conference of the Cognitive Science Society* (2009)
6. Neves, C.M., Anderson, J.R.: Knowledge compilation: Mechanisms for the automatization of cognitive skills, pp. 57–84 (1981)
7. VanLehn, K.: *Mind Bugs: The Origins of Procedural Misconceptions*. MIT Press, Cambridge (1990)

Exploring Two Strategies for Teaching Procedures

Antonija Mitrovic, Moffat Mathews, and Jay Holland

Intelligent Computer Tutoring Group, University of Canterbury, Christchurch, New Zealand
{tanja.mitrovic,moffat.mathews,jay.holland}@canterbury.ac.nz

Abstract. Due to high cost and complexity of Intelligent Tutoring Systems (ITS), current systems typically implement a single teaching strategy, and comparative evaluations of alternative strategies are rare. We explore two competing strategies for teaching database normalization. Each data normalization problem consists of a number of tasks, some of which are optional. The first strategy enforces the procedural nature of the data normalization by providing an interface that requires the student to complete the current task (i.e. a part of the problem) before attempting the next one. The alternative strategy provides more freedom to the student, allowing him/her to select the task to work on. We performed an evaluation study which showed that the former, more restrictive strategy results in better problem-solving skills.

Keywords: teaching strategies, procedural tasks, evaluation.

1 Introduction

Ideally, ITSs should support multiple teaching strategies and adapt them for each student. Current ITSs typically implement a single teaching strategy, due to high development costs. Different teaching strategies might require a lot of development work; for example, the system's interface might need to be changed to support a different style of interaction. There are also difficulties in the evaluation of ITSs. For those reasons, evaluating competing teaching strategies for the same domain is rare.

Many factors influence ITS design, such as the limited capacity of working memory [1], the cognitive load [2] and the nature of the task. Instructional tasks can be arranged on a spectrum from strictly procedural (sequential), in which the student needs to learn a well-defined algorithm, to non-procedural, in which students are free to start from any part of the problem or apply actions in any order [3]. The solution search space for sequential tasks is much smaller than that of non-sequential tasks [4], as the student only has to concentrate on the solution space for a part of the task rather than the whole task. An example non-procedural task is software design: the student does not have to start at a particular point and there is no sequence they must follow; the solution search space is much higher as they keep track of what they have done, the consequences of what they have done, and what is left to do.

So, how should one teach procedures to novices? In this paper, we explore whether there is a difference between forcing students to adhere to the sequence of actions or leaving them to answer problems steps in any order they wish. Our hypothesis is that

students taught via the non-sequential method would be less efficient and solve fewer tasks, while students in the sequential method group would tackle more problems, and also more complex ones, have a higher rate of success, and be more efficient.

We discuss data normalization in the following section. Section 3 then presents two versions of NORMIT, implementing the sequential and a less-restrictive strategy. We present the study and its results in Section 4, and end with conclusions.

2 Data Normalization

Data normalization is the technique of refining an existing relational database schema in order to ensure that all relations are of high quality [5]. Normalization is usually taught via a series of lectures that introduce the relevant concepts followed by paper-based exercises. Students find data normalization very difficult [6, 7], as it is very theoretical and requires a good understanding of the relational data model, various types of keys (primary, candidate, foreign keys and superkeys), Functional Dependencies (FD), normal forms and normalization algorithms.

Data normalization is a procedural technique: the student goes through a number of tasks to analyze the quality of a database. Each problem consists of a relation schema and a set of FDs (which does not have to be complete). For example, the student might be given a relation $R(A, B, C, D, E)$ (typically the semantics of the attributes id not given) and the set of FDs: $\{A \rightarrow B, AB \rightarrow C, D \rightarrow AC, D \rightarrow E\}$.

The normalization procedure as implemented in NORMIT consists of eleven tasks described below. Please note that we refer to elements of the procedure as *tasks* rather than *steps*, as each of them contains a number of actions the student has to perform, including in some cases relatively complex algorithms. Therefore we refer to them as tasks to make it clear that the tasks are relatively complex compared to what is generally assumed by a step in the ITS research. The first eight tasks are necessary to determine the highest normal form the relation is in. If the relation is not in Boyce-Codd Normal Form (BCNF), the student needs to apply the relational synthesis algorithm to derive an improved database schema via tasks 9-11.

1. Identify the candidate keys for the given table. There may be one or more keys in a table; e.g. the only key in the above problem is D.
2. Find the closure of a given set of attributes. In the above example, to make sure that D is the key of relation R, we could determine that its closure consists of all attributes of relation R.
3. Identify prime attributes. Prime attributes are those attributes that belong to any candidate keys. In the above problem, D is the only prime attribute.
4. Simplify FDs by applying the decomposition rule, if necessary. In this task, a FD with more than one attribute on the right-hand side (RHS) is replaced with as many FDs as there are attributes on RHS. In the above problem, $D \rightarrow AC$ would be replaced with two FDs: $D \rightarrow A$ and $D \rightarrow C$.
5. Determine the normal forms for the given relation.
6. If the student specified that the relation is not in 2NF, he/she needs to identify FDs that violate that form (i.e. partial FDs).

7. If the student specified that the relation is not in 3NF, he/she needs to identify FDs that violate that form (i.e. transitive FDs).
8. If the student specified that the relation is not in BCNF, he/she will be asked to identify FDs that violate that form.
9. For relations that are not in BCNF, reduce LHS of FDs. This task checks whether some of the attributes on the LHS can be dropped while still having a valid FD.
10. Find minimal cover (i.e. the minimal set of FDs).
11. Decompose the table by using the minimal cover.

3 Two Versions of NORMIT

NORMIT [8, 9] teaches data normalization in a task-by-task manner, showing only one task at a time which the student needs to complete before moving on to the next task. The student can submit a solution at any time, which the system then analyses and presents feedback. At any point during the session, the student may change the problem, review the history of the session, examine the student model or ask for help on the current task. The system currently contains 50 problems and new problems can be added easily. NORMIT is a constraint-based tutor, and its knowledge base is represented as a set of 82 (problem-independent) constraints. Each constraint is relevant for a particular task of the procedure. Some constraints are purely syntactic, while others compare the student's solution to the ideal solution (generated by the problem solver). The short-term student model consists of a list of violated/satisfied constraints for the current attempt, while the long term model records the history of usage for each constraint. Please see [8] for information about NORMIT.

The original version of NORMIT enforces the procedural nature of the data normalization by forcing the student to complete the current task before being able to move on to the next task. An alternative strategy would allow the student to work on any task of the procedure in any order. To implement that strategy, we developed a less restrictive interface which shows all the tasks on a single page, thus allowing the student to approach the problem in different ways. In order to work on a particular task, the student clicks the *Edit* button which expands the page by adding specific elements for that task. The functionality provided by the modified interface is essentially the same as in the original tutor, but the interaction is slightly different. We also had to modify the system's knowledge base to support this new style of interaction. In the original version of NORMIT, constraints are task-specific: the very first test in each constraint specifies the task the constraint is relevant for. In the new version, the student is free to select the task, and therefore the constraints cannot be restricted to specific task. There are 75 constraints in the non-procedural version of NORMIT.

4 Evaluation Study

We performed an evaluation study with the students enrolled in an introductory database course at the University of Canterbury. Our hypothesis was that procedural version of the tutor would result in higher learning in terms of problem-solving skills and conceptual knowledge. Prior to the experiment, the students had four lectures on

normalization. The study was conducted at the scheduled lab times on October 5th or 6th, 2011 (the students were divided into two streams). The session length was 100 minutes. The students in the control group used the original, procedural version of the system, while the experimental group used the new, non-procedural version. The participation was voluntary, and 33 students participated in the study. All students enrolled in the course were free to use the system after the study if they so wished.

The students were randomly assigned to one of the two conditions, and were given an online pre-test, with four multi-choice questions. The initial two questions required students to identify the correct primary key and the highest normal form for a given table. For the remaining two questions students needed to identify the correct definition of a given concept. A similar test was used as the post-test at the end of the sessions. Both tests were short on purpose as the session was of limited length. The consequence of short pre/post tests, however, is the limited coverage of the domain.

Table 1. Statistics from the study (standard deviations given in parentheses)

Group	Experimental (14)	Control (14)	Significant?
Pre-test mean (SD)	1.9 (1.3)	2.3 (1.3)	no
Post-test mean (SD)	3.3 (1)	3.5 (0.8)	no
Gain	1.5 (1)	1.2 (1.5)	no
Normalized gain	0.7 (0.4)	0.6 (0.5)	no
Improvement pre-to-post	t=5, t<0.01	t=2.9, p<0.01	
Time	68 (27)	74 (15)	no
Attempted problems	4.7 (2.2)	8.3 (3.3)	p<0.05
Solved problems	3.7 (2.4)	6.6 (3.7)	p<0.05
Attempts	33 (24)	101 (56)	p<0.01
Known at start	35 (15)	37 (17)	no
Learnt constraints	6.9 (5.9)	5.4 (3.7)	no
Used constraints	53 (18)	63 (20)	p=0.09
Problem complexity	2.3 (1.3)	4.4 (2.5)	p<0.01

We excluded data about students who interacted with the system for less than 10 minutes and/or have made no attempts at problems, which resulted in 14 students in each group (Table 1). There was no difference between the two groups on the pre-test and post-test performance, as well as on the gains, normalized gains and interaction time. Both groups improved significantly during the session (determined by comparing their pre/post test results by a matched t-test).

We then analyzed the learning behavior by examining the student logs. The control group students attempted and solved significantly more problems and made significantly more attempts than their peers. The latter is easy to explain: the control condition had to go through each task in order to solve problems, while the experimental condition participants could only work on a subset of tasks.

Another measure of learning is the number of constraints that were learnt during the session. To see whether a constraint is known at the start, we require that the student has applied it correctly on at least 4 out of 5 initial attempts at that constraint. As reported in Table 1, there was neither significant difference on the number of constraints known at start, nor on the number of learnt constraints.

The control group participants attempted and solved approximately twice as many problems as their peers. At the first look, it seems contradictory that they acquired the same number of constraints and achieved similar results at the post-test as the experimental group. We therefore looked deeper into the logs. We identified all constraints relevant for attempts and report them in the *Used constraints* row of Table 1. There was a marginal difference in favour of the control group. Therefore, the control group students used more constraints to solve problems than the experimental group. A deeper look at the problems solved provides another interesting observation. The average complexity of problems solved by the experimental group is just over 2, while the control group solved problems of significantly higher complexities (the last row of Table 1). Given that there was no difference in background knowledge of the two groups, we can conclude that the significant difference in the problem-solving accomplishments comes from the difference in the interfaces. The procedural version of the tutor provided more guidance which in turn enabled the students to solve more problems, and also more complex problems, in the same amount of time.

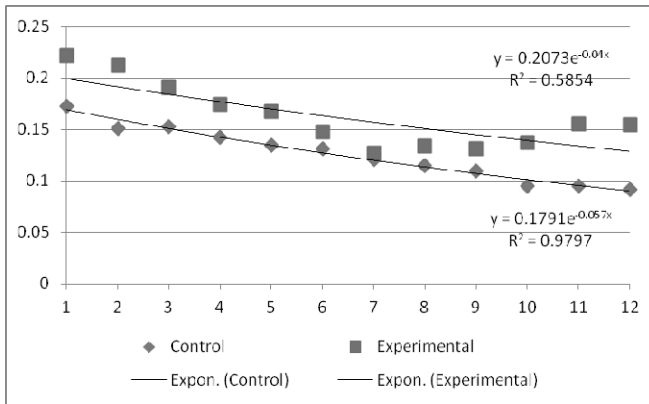


Fig. 1. Learning curves for the two groups

Figure 1 shows the learning curves for the two conditions (i.e. the proportion of violated constraints following the n^{th} occasion when a constraint was relevant, averaged across all students and all constraints). The R^2 fit to the exponential curve is good for the control, but is quite poor for the experimental group. The learning rate of the control group is also slightly higher. A closer inspection of the constraints learnt shows that the control group learnt more complex constraints, which is the consequence of higher average complexity of problems they attempted and solved.

5 Conclusions

There are many possible approaches to teach the same instructional domain. Due to high complexity of ITSs and high development costs, ITS developers usually implement only one teaching strategy. In this paper, we present two teaching strategies for data normalization, which differ in the amount of control students have in selecting

which part of the problem to work on. The first strategy requires the student to follow the procedure closely, working on one task at a time and completing it before attempting subsequent tasks, while the other gives full control to the student. Our hypothesis was that the former strategy would result in better learning.

Our study shows that both strategies resulted in significant improvement from pre- to post-test. There was no significant difference between the two groups on the post-test; however, the post-test was short and its questions are of different nature compared to the problems in the ITS. We also looked at how many new knowledge elements (i.e. constraints) students learnt during the study. Although there was no significant difference in the amount of newly acquired knowledge, there was difference in the kinds of constraints learnt. The procedural version resulted in significantly higher number of problems attempted and solved in comparison to the non-procedural strategy. The average complexity of problems solved is also significantly higher in the case of procedural strategy. Therefore, closer adherence to the procedural nature of data normalization did result in higher problem-solving success.

Our study was of short duration and small in terms of the participants. We plan to perform a bigger study in 2012 with NORMIT and also to conduct similar studies in other instructional domains.

References

1. Miller, G.A.: The Magical Number Seven, Plus or Minus Two. *The Psychological Review* 63, 81–97 (1956)
2. Sweller, J.: Cognitive Load Theory, Learning Difficulty, and Instructional Design. *Learning and Instruction* 4, 295–312 (1994)
3. Mitrovic, A., Weerasinghe, A.: Revisiting the Ill-Definedness and Consequences for ITSs. In: Dimitrova, V., Mizoguchi, R., du Boulay, B., Graesser, A. (eds.) *Proc. 14th Int. Conf. AIED*, pp. 375–382 (2009)
4. McCallum, A.K.: Learning to Use Selective Attention and Short-Term Memory in Sequential Tasks. In: *Proc. 4th Int. Conf. Simulation of Adaptive Behavior*, pp. 315–324 (1996)
5. Elmasri, R., Navathe, S.B.: *Fundamentals of database systems*. Addison-Wesley (2010)
6. Kung, H.-J., Tung, H.-L.: An alternative approach to teaching database normalization: A simple algorithm and an interactive e-Learning tool. *Journal of Information Systems Education* 17(30), 315–325 (2006)
7. Phillip, G.C.: Teaching database modeling and design: areas of confusion and helpful hints. *Journal of Information Technology Education* 6, 481–497 (2007)
8. Mitrovic, A.: The Effect of Explaining on Learning: a Case Study with a Data Normalization Tutor. In: Looi, C.-K., McCalla, G., Bredeweg, B., Breuker, J. (eds.) *Proc. Artificial Intelligence in Education*, pp. 499–506. IOS Press (2005)
9. Mitrovic, A.: Fifteen years of Constraint-Based Tutors: What we have achieved and where we are going. *User Modeling and User-Adapted Interaction* 22 (in print), <http://dx.doi.org/10.1007/s11257-011-9105-9>

Relating Student Performance to Action Outcomes and Context in a Choice-Rich Learning Environment

James R. Segedy, John S. Kinnebrew, and Gautam Biswas

Vanderbilt University, Nashville TN 37235, USA

{james.r.segedy, john.s.kinnebrew, gautam.biswas}@vanderbilt.edu

Abstract. This paper presents results from a recent classroom study using Betty's Brain, a choice-rich learning environment in which students learn about a scientific domain (*e.g.*, mammal thermoregulation) as they teach a virtual agent named Betty. The learning and teaching task combines reading and understanding a set of hypertext resources with constructing a causal map that accurately models the science phenomena. The open-ended nature of this task requires students to combine planning, targeted reading, teaching, monitoring their teaching, and making revisions, which presents significant challenges for middle school students. This paper examines students' learning activity traces and compares learning behaviors of students who achieved success with those who struggled to complete their causal maps. This analysis focuses on students' actions leading to changes in their causal maps. We specifically examine which actions led students to make correct versus incorrect changes to their causal map. The results of this analysis suggest future directions in the design and timing of feedback and support for similarly complex, choice-rich learning tasks.

Keywords: Metacognition, Monitoring, Learning Activity Traces, Sequence Analysis, Learning Environment.

1 Introduction

Betty's Brain is a learning-by-teaching environment where students teach a virtual agent, named Betty, about science topics by reading a set of hypertext resources and constructing a causal map (Figure 1) to model the relevant scientific phenomena [1]. Once taught, Betty (the Teachable Agent) can use her map to answer causal questions (*e.g.*, if cold temperatures increase, what happens to an animal's blood vessel constriction?) and explain those answers by reasoning through chains of links [1]. The student's goal is to teach Betty a causal map that matches a hidden, expert model of the domain using information from the resources. To gauge their progress towards this goal, students can make Betty take quizzes, which are sets of questions created and graded by a virtual mentor agent named Mr. Davis, who compares Betty's answers with those generated by the expert model. Thus, when Betty is unable to answer quiz questions correctly, the (human) students can use that information to discover Betty's (and their own) misunderstandings and correct them.

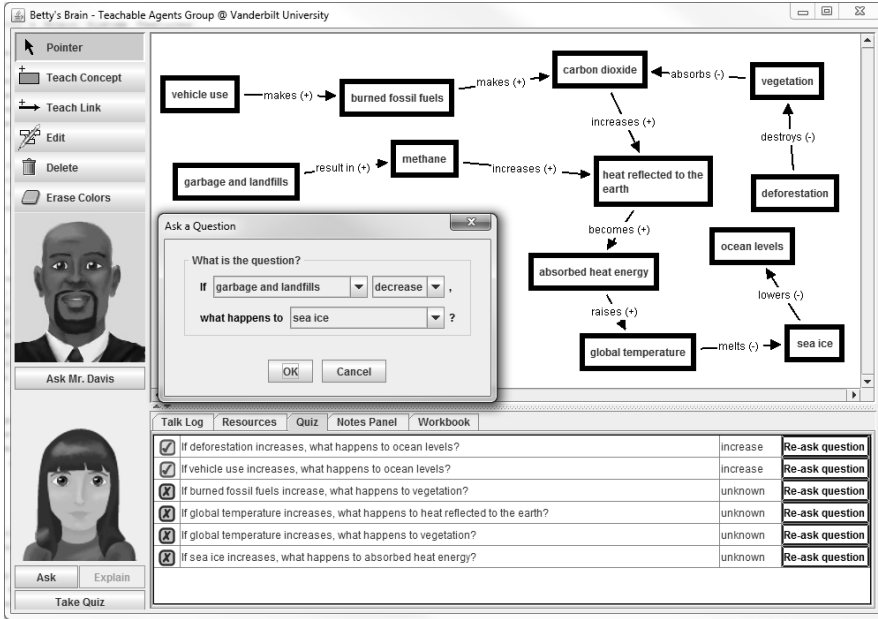


Fig. 1. Betty's Brain system with query window and quiz results

Overall, this is a complex, open-ended, and choice-rich learning task for middle school students. In the system, learners must choose how to effectively perform their teaching task: they must decide when and how to search for new information (reading the resources), how to teach this information to Betty (building the causal map), and when to monitor Betty's understanding (using quizzes, questions, and explanations). Achieving success at this task requires utilizing and coordinating a complex set of cognitive and metacognitive skills, some of which the student may not yet know.

Research has shown that middle school students do not have well-developed independent learning strategies; and as novices they often adopt suboptimal trial-and-error methods when they encounter difficult exploratory learning tasks [2, 3]. Research with Betty's Brain corroborates this finding, as many students struggle to teach Betty the correct causal map. Recent analyses show that these students (1) fail to systematically plan their teaching interactions, (2) struggle to discover and understand causal relations in the resources, and (3) misunderstand or misinterpret Betty's quiz results [4]. For example, a previous analysis illustrated that three-fifths of successful students' causal link edits (i.e., adding, deleting, or changing a link) improved their causal map scores¹, while less successful students' link edits increased their map scores only two-fifths of the time [5]. These results, in conjunction with other analyses of student behaviors, led us to hypothesize that these students experienced difficulties because they could not easily identify the causal relations contained in the resources.

¹ Causal map scores were calculated by subtracting the number of incorrect map links (i.e. those that were not present in the expert model) from the number of correct map links.

Our primary goal in this paper is to further explore our hypothesis through analysis of the outcome of causal map editing actions (i.e., whether or not the edit action led to a map that more closely matched the expert model) and the context of these actions (in terms of their relevance to recent preceding actions). We report results of this analysis on student learning activity traces from an experimental study recently conducted in an eighth-grade science classroom. These results suggest future directions in the design of feedback and support for similarly complex, choice-rich learning tasks.

2 Method

The current study was conducted in three eighth-grade middle Tennessee science classrooms taught by the same teacher. Due to attrition (caused by unsigned permission slips and student absences), we had complete data for 40 students at the end of the study. The study proceeded as follows: on day 1, the classroom teacher introduced students to thermoregulation in mammals. On day 2, they completed a pre-test. On days 3 and 4, the researchers instructed the students on causal reasoning and the Betty's Brain system. Students then spent five class periods independently using Betty's Brain with minimal intervention from the teachers and the researchers. Finally, all students took a post-test that was identical to the pre-test.

In this paper, we analyze students' learning activity traces collected from the system to investigate the edits they made to their causal map (specifically, adding, removing, or changing causal links). We categorize each edit along three dimensions determined by metrics for the consequences of the edit and its context (in terms of preceding actions). The first metric is the type of action performed directly before the edit (the prior action), which can be: (1) editing another link (L-EDIT), (2) editing (i.e., adding or removing) a concept (C-EDIT), (3) reading the resources (READ), (4) asking Betty to answer a causal question (QUER), (5) asking Betty to explain her answer to a question (EXPL), or (6) having Betty take a quiz (QUIZ). The second metric is whether the edit was relevant to any of the three actions directly preceding it. Actions in Betty's Brain are considered relevant to each other if they reference or operate on one of the same causal map links [5]. For example, two L-EDIT actions are relevant to each other only if they operate on the same link, and a QUER action is relevant to an L-EDIT action if the link edited was used in Betty's answer to the query. The third metric is the edit's correctness, where correct indicates that the link edit resulted in a map that more closely resembled the expert model. The proportion of a student's edits that fell into each category (the possible combinations of values across the three metrics) provided a score for the corresponding category. For example, each student was given a score equal to the percentage of their edits that were relevant, correct edits performed after a read (represented as READ → EDIT-relevant-correct).

3 Results

We begin by analyzing students' map scores, which measure the quality of the causal maps that they taught Betty. Similar to other recent studies with Betty's Brain, the students in this experiment achieved an average map score of 8.10 out of 15, with a standard deviation of 5.64. While some students were very successful at the task, others struggled to teach Betty the target material.

Building on previous analyses that used this data to illustrate marked differences between students with high and low map scores [4], we divide the students into three groups based on their map scores. Students in the low group taught Betty a map that achieved a score of 5 or below (out of a maximum score of 15). Students in the medium group taught Betty a map with a score of 6 to 10, and students in the high group taught Betty a map with a score of 11 to 15. The resulting low, medium, and high groups had 18, 6, and 16 students, respectively. The remaining analyses in this section compare the high group and the low group in order to highlight important differences in how they edited their causal maps.

Table 1. Editing behaviors exhibited proportionally more by the high group

	Edit Behavior	High Group Mean	Low Group Mean
1.	L-EDIT → L-EDIT-irrelevant-correct	11.6%	5.4%
2.	QUIZ → L-EDIT-relevant-correct	5.9%	1.9%
3.	READ → L-EDIT-irrelevant-correct	11.3%	7.6%

To assess the differences in students' editing behaviors, we compared the high and low groups' scores for each category of edit identified in Section 2. Table 1 shows the three edit behaviors with the largest difference in score between groups, where the behaviors were exhibited proportionally more often by the high group. Similarly, Table 2 shows the three edit behaviors with the largest difference that were exhibited proportionally more often by the low group. In Table 1, all three behaviors included correct edits, and in Table 2, all three behaviors included incorrect edits. This matches previous analyses indicating that successful students generally had a higher percentage of correct edits than students in the low group, and vice versa.

Table 2. Editing behaviors exhibited significantly more times by the low group

	Edit Behavior	High Group Mean	Low Group Mean
4.	READ → L-EDIT-irrelevant-incorrect	9.2%	15.7%
5.	C-EDIT → L-EDIT-relevant-incorrect	3.8%	7.6%
6.	QUER → L-EDIT-irrelevant-incorrect	2.6%	5.3%

One interesting result indicated by edit behavior 2 is that even though the high group students made mistakes when editing their maps, they were able to use the results of Betty's quizzes to correct those mistakes. In contrast, edit behavior 6 shows that students in the low group made irrelevant, incorrect link edits after queries. This suggests that these students had trouble using queries to explore and correct Betty's knowledge, and they may have had a misconception regarding how to use queries. To fully understand Betty's answers, students need to listen to her explanations, which provide insight into the causal links she used to answer the question. However, students in the low group often moved directly from querying Betty to editing the map.

Thus, they missed important opportunities to engage the science material at a deeper level and may have made an incorrect edit as a result.

A surprising result indicated by edit behaviors 1 and 3 is that many of the high group's correct link edits were not relevant to recent actions. We expect that students derive correct edits from their previous activities, such as reading or viewing quiz results. However, these patterns tend to contradict this intuition, at least with the definition of relevance used in this analysis. One possible explanation for the irrelevance of the link edit in behavior 1 may be an artifact of the Betty's Brain user interface: Students are able to simultaneously view the resources and edit the causal map, so an initial read action (i.e., accessing a page in the resources) may not fall within the relevance window for a subsequent edit, even though the page was still visible for the student to reference. Therefore, edit behavior 1 could indicate that the high group employed a strategy of opening a page in the resources and incrementally adding links, while continuing to reference the visible resource page. Alternatively, students could have used prior knowledge in combination with information from the resources to make irrelevant, but correct, edits. This could also explain the irrelevance of the edit in behavior 3 following a read action. The correct edit could have been informed by previous reads or prior knowledge (e.g., when the current page prompted the recall of that information).

Edit behavior 4 is analogous to edit behavior 3, except that the low group students were more likely to make an incorrect edit when it was not related to a preceding read. These students may have had difficulty in identifying causal links in the reading materials. Edit behavior 5 is difficult to interpret, as it only indicates that students in the low group more often made incorrect, but relevant, link edits directly after a concept edit. This could be a difference in the sequencing of concept and link edits between the two groups.

4 Discussion and Conclusion

In this paper, we have presented results from a recent classroom study that highlight how successful and unsuccessful students differ in their teaching behaviors in Betty's Brain. The results suggest that students who struggle with the teaching task often have difficulty in both finding causal relations as they read the resources provided in the system (edit behaviors 3 and 4) and monitoring Betty's understanding of the subject matter via quizzes and queries (edit behaviors 2 and 6). Further, these results illustrate the importance of considering both action outcomes (e.g., effect of an edit on the map score) and action context (e.g., preceding actions and relevance of the edit action) when analyzing learning activity sequences. The analysis presented here provides possible explanations for why students in the low group struggled to succeed.

One strong possibility is that many students in the low group approached the Betty's Brain learning task without a firm grasp of some of the cognitive skills necessary to achieve success in the system, such as identifying causal links in reading materials. These students may also have failed to fully understand how causal maps work during classroom instruction. Causal reasoning and careful reading are both difficult, complex skills that underlie the Betty's Brain learning task. However, the current feedback and scaffolding in Betty's Brain focus on metacognitive strategies for plan-

ning teaching interactions and monitoring the causal map's correctness (via queries and quizzes). These scaffolds may not be sufficient for helping students as they are struggling with reading and causal reasoning. The present analysis emphasizes this point with respect to students' reading abilities: students who were not able to teach Betty the correct map were more likely to incorrectly edit their map, even directly after reading the resources. This suggests that we need to augment the current Betty's Brain with direct support of causal reasoning and identifying causal links in reading materials.

An important limitation of this analysis is that it focused primarily on causal link edits and the actions directly preceding them. It is reasonable to assume that many of these edits were the results of learning that required multiple, coordinated actions. Further, some link edits marked as irrelevant may have been influenced by (relevant) actions that took place outside of the three-action window used to calculate relevance. In future analyses, we intend to investigate longer sequences of actions before edits.

As we move forward with this work, we will develop and incorporate tutorials into the system that support students in causal reasoning and reading skills. We will also introduce the notion of explicit pedagogical goals in the agents' reasoning mechanisms. For example, the mentor agent's goal will be a function of the student's state of knowledge, exhibited learning behaviors, and previous feedback delivered to the student. This will focus students on a skill until it produces a successful result, because all feedback delivered during that time period will focus on the same pedagogical goal. We believe that these refinements will provide important benefits to struggling students by helping them gain the necessary cognitive skills and then coordinate those skills in effective metacognitive strategies.

References

1. Leelawong, K., Biswas, G.: Designing learning by teaching agents: The Betty's Brain system. *International Journal of Artificial Intelligence in Education* 18(3), 181–208 (2008)
2. Azevedo, R.: Using hypermedia as a metacognitive tool for enhancing student learning? The role of self-regulated learning. *Educational Psychologist* 40(4), 199–209 (2005)
3. Schunk, D.H., Zimmerman, B.J.: Social origins of self-regulatory competence. *Educational Psychologist* 32(4), 195–208 (1997)
4. Segedy, J.R., Kinnebrew, J.S., Biswas, G.: Modeling learner's cognitive and metacognitive strategies in an open-ended learning environment. In: *AAAI Fall Symposium on Advances in Cognitive Systems*, Arlington, VA (2011)
5. Segedy, J.R., Kinnebrew, J.S., Biswas, G.: Supporting cognitive and metacognitive skills in complex, open-ended learning environments. *Journal of Educational Psychology* (in review)

Using the MetaHistoReasoning Tool Training Module to Facilitate the Acquisition of Domain-Specific Metacognitive Strategies

Eric Poitras, Susanne Lajoie, and Yuan-Jin Hong

Department of Educational and Counselling Psychology,
McGill University, Montreal, Quebec, H3A 1Y2
eric.poitras@mail.mcgill.ca, susanne.lajoie@mcgill.ca,
yuan-jin.hong@mail.mcgill.ca

Abstract. Learning through historical inquiry requires that students engage in domain-specific metacognitive strategies. For example, students need to be aware that causes of historical events are often unknown or uncertain and they need strategies for resolving such ambiguity. In this paper, we provide an overview of the theoretical, instructional, and empirical foundations of the MetaHistoReasoning Tool Training Module. This computer-based learning environment is designed to facilitate the acquisition of metacognitive strategies that are critical in learning through historical inquiry. We review findings pertaining to (1) the classes of self-explanations generated and (2) the accuracy of categorizations made by students. We discuss these findings in terms of developing an artificial pedagogical agent capable of appropriately delivering instructional explanations and effectively prompting self-explanations.

Keywords: metacognition, metacognitive tool, pedagogical agent, historical inquiry.

1 Introduction

Learning about complex historical events requires that students engage in metacognitive processes that are specific to that discipline. However, students often engage in dysregulated learning [1], since they fail to monitor and strategically control cognitive processes that are important to learning [2-3]. Specifically, students often fail to notice that the causes of historical events are unknown, uncertain, or unreported. They also fail to formulate explanations to gain better understanding [3].

As such, we designed the MetaHistoReasoning tool [4], a computer-based learning environment that serves as a metacognitive tool. The design of the environment is guided by a theory that accounts for domain-specific metacognitive strategies in learning through historical inquiry. The environment includes a training and inquiry module. We used example-based skill acquisition as an instructional model for the training module. The inquiry module is driven by inquiry-based learning principles. For the purposes of this study we only examine how students learn with the training module.

The MetaHistoReasoning tool training module supports students in terms of acquiring domain-specific metacognitive strategies in learning through historical inquiry [5]. These strategies include both metacognitive monitoring and control activities. In doing so, the training module provides students with examples of each type of metacognitive strategy. An artificial pedagogical agent prompts students to categorize these examples and provides corrective feedback. The agent also prompts students to explain the rationale and purpose of each strategy [6-7].

This study examines the effectiveness of the training model in facilitating skill acquisition. More specifically, we test a model of students' self-explanation activities in terms of predicting accuracy in categorizing examples of metacognitive strategies. In doing so, we address the following question: Does generating self-explanations in relation to the rationale and purpose of each domain-specific metacognitive strategy result in categorizing examples more accurately? Using categorization accuracy as an indicator of skill acquisition, we hypothesize that generating this specific type of self-explanation, which targets the learning-domain of the examples, is predictive of categorization accuracy. Based on our findings, we make recommendations for the modification of the training module with the goal of enhancing skill acquisition.

2 Methods

2.1 Participants

Eight undergraduate students (2 men and 6 women; Mean Age = 22; Mean GPA = 3.09) were recruited through a classified ad posted on the university website. Students were Anglophone non-history majors who had completed at least 2 years of their degree. A questionnaire was administered to assess their familiarity with historical inquiry and the historical topic [8]. The results confirmed that students were unfamiliar with both of these subjects, which is the population that is targeted by the MetaHistoReasoning tool. Students received a compensation of 40\$.

2.2 Computer-Based Learning Environment

The MetaHistoReasoning tool training module is a rule-based system wherein students progress through each phase as they become more proficient in categorizing examples. The exemplifying-domain of the examples refers to the Acadian Deportation. Students study these examples while going through three phases: the pre-training, training, and acquisition phase.

In the pre-training phase, students view an instructional video that introduces the topic of the examples. The video provides a brief overview of the strategies that are exemplified. The video also explains how to use the tools of the training module.

In the training phase, students study a categorized example of each strategy. In doing so, an artificial pedagogical agent defines and explains each strategy (i.e., "This example shows an historian asking a question. In doing so, the historian begins to search for the most important cause of the Acadian Deportation."). Alternatively, a brief description and example of each strategy is also available as a tool-tip that appears when the mouse cursor is positioned over an option from the list.

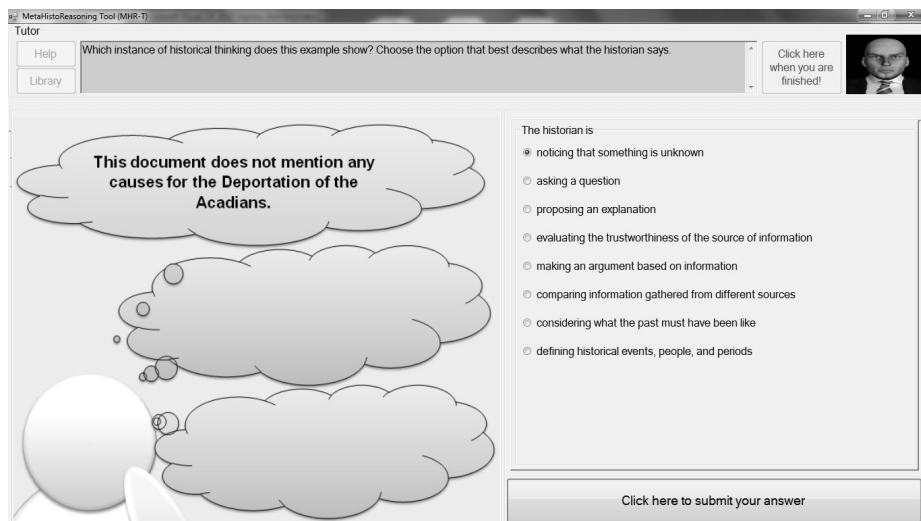


Fig. 1. The design of the MetaHistoReasoning tool training module

In the acquisition phase, students learn domain-specific metacognitive strategies by categorizing examples, receiving corrective feedback, and generating self-explanations (see Fig. 1). These strategies include (1) noticing unexplained events (e.g., the document does not mention any causes for the Deportation of the Acadians); (2) asking appropriate questions (e.g., What is the most important cause for the Acadian Deportation?); (3) formulating explanations (e.g., the Acadian Deportation was caused by British Governor Charles Lawrence’s discontent towards the Acadians); (4) evaluating the trustworthiness of sources (e.g., since anybody can edit or revise the website, it is considered less trustworthy); (5) gathering evidence (e.g., Governor Charles Lawrence’s declaration states that all citizens must bear arms in case of war); (6) corroborating evidence (e.g., the fact is mentioned in both a letter and the transcription of a council meeting); (7) contextualizing evidence (e.g., The Acadians must have felt apprehensive about the war); and (8) using substantive concepts (e.g., the Seven Years’ War is the first global conflict which involved both French and British).

The pedagogical agent supports students by providing categorization prompts (i.e., “Which instance of historical thinking does this example show? Choose the option that best describes what the historian says.”). Students make as many attempts as necessary to categorize an example by choosing correctly from amongst the strategies exemplified on an eight-option list. Students are then provided with corrective feedback (i.e., “Your answer is correct.”; “Your answer is incorrect, try again.”). The agent also supports students to explain the rationale of each strategy by providing a self-explanation prompt (i.e., “Explain how each instance of historical thinking relates to the historian’s goal, which is to explain why the Acadian Deportation occurred.”).

2.3 Measures

We collected data through on-line unobtrusive cognitive methodologies (i.e., log-file trace data and time-stamped video screen capture data; see [9]). The log-file records

events at a scale of milliseconds (10^{-4} seconds). Events recorded in the log-file trace data include the accuracy scores, self-explanations, time taken to categorize an example, number of attempts taken to categorize an example accurately, and the number of previous exposures to a similar type of example. The time-stamped video screen captures were used to corroborate log-file trace data by recording the sequence of entries.

2.4 Procedure

Students first completed a consent form, a demographic questionnaire, and a questionnaire that assessed familiarity with historical inquiry and the historical topic [8]. Students received instructions in learning with the training module through a video that automatically appeared on the computer screen. The video described the historical context surrounding the Acadian Deportation. The students were then shown how to use the training module. First, the students were taught to categorize examples by choosing from the multiple-choice options. Second, the students were shown how to write self-explanations. After learning with the benefit of the MetaHistoReasoning tool, students were debriefed and compensated for their participation.

2.5 Coding and Scoring

The unit of our analysis was the accuracy of students' categorizations (0 = accurate, 1 = inaccurate) – whether students identified the correct type of metacognitive strategy that was exemplified when choosing from a multiple-choice list of eight options. The predictor variables were the following: (1) the time spent categorizing an example; (2) the example category; (3) the fading threshold (i.e., baseline or auxiliary example); (4) the example difficulty (i.e., simple or complex example); (5) the amount of prior exposure to similar examples; (6) the count of categorization attempts; (7) the count of exemplifying-domain self-explanations; (8) the count of learning-domain simple self-explanations; and (9) the count of learning-domain elaborate self-explanations.

We adapted a coding scheme of self-explanation activities used in previous research for the purposes of this study [10]. *Exemplifying-domain self-explanations* paraphrased the contents of the examples. *Elaborate learning-domain self-explanations* involved both (1) relating skills with each other and (2) explaining each skill's contribution towards achieving the goal of explaining the Acadian Deportation. In contrast, *Simple learning-domain self-explanations* referred to only one of the two aspects mentioned for an elaborate explanation. Explanations that did not fit any of these categories were classified as *other*. The types of self-explanations were coded by two raters for the entire transcript the interrater agreement was substantial (i.e., interrater agreement of 89%). Disagreements were resolved through discussion.

3 Results

We compared a six- and nine-predictor logistic model (i.e., with and without self-explanations) in terms of their fit to the accuracy scores ($N = 517$ categorizations). The data were filtered for outliers (i.e., 4 cases were discarded) and the assumptions in relation to the binomial distribution and minimum observation to predictor ratio

were met. A test of the model with nine predictors against a constant-only model was statistically significant, $\chi^2(15, N = 513) = 155.199, p < .05$. This suggests that the nine-predictor model is effective in terms of classifying accurate and inaccurate example categorizations. Moreover, the Hosmer-Lemeshow (H-L) goodness-of-fit test was insignificant, $\chi^2(8, N = 513) = 5.551, p > .05$, which suggests that the second model was fit to the data well. A test of the nine-predictor model compared against the six-predictor model was statistically significant, $\chi^2(3, N = 513) = 29.900, p < .05$. This finding shows that adding the type of self-explanation generated by students to the predictors makes a significant contribution to distinguishing between accurate and inaccurate categorizations. The model correctly predicted 93.2% of accurate categorizations and 47.3% of inaccurate categorizations, for an overall rate of 81.7%.

Table 1 shows the parameters of the nine predictors in the second model that significantly contributed to the prediction of accuracy scores. Based on the model, the greater the amount of prior exposure to the example category, the more likely it is that a categorization is accurate. However, the greater the amount of time spent, attempts taken, and exemplifying-domain self-explanations, the more likely it is that a categorization is inaccurate. All other scores being equal, categorizing examples pertaining to contextualizing evidence, gathering evidence, and corroborating evidence were more likely to be inaccurate. The odds of inaccurately categorizing an example of contextualizing evidence was 25.074 times greater ($= e^{3.222}$), gathering evidence was 16.675 times greater ($= e^{2.814}$), and corroborating evidence was 5.869 times greater ($= e^{1.770}$). Moreover, complex examples were more likely to be categorized inaccurately, with an odds of .446 times greater ($= e^{-0.808}$).

Table 1. Nine-predictor logistic model (non-significant parameters excluded)

Predictor	β	$SE \beta$	Wald's χ^2	df	p	e^β (odd ratio)
Constant	-2.766	.758	13.327	1	.000*	N/A
Time to categorize	.043	.011	15.496	1	.000*	1.043
Attempt number	.312	.084	13.822	1	.000*	1.366
Amount of exposure	-.140	.045	9.734	1	.002*	.869
Example category			58.204	7	.000*	
Gathering evidence	2.814	.642	19.241	1	.000*	16.675
Contextualizing evidence	3.222	.633	25.922	1	.000*	25.074
Corroborating evidence	1.770	.641	7.623	1	.006*	5.869
Simple examples	-.808	.360	5.048	1	.025*	.446
Exemplifying-domain self-explanation count	.449	.118	14.486	1	.000*	1.567

4 Discussion

The aim of this study was to predict students' accuracy in categorizing examples of domain-specific metacognitive strategies based on students' self-explanation activities. Students were accurate in their categorizations on 75% of occasions. Students

were more accurate in categorizing examples when they had previous exposure to similar types of examples, which suggests that practice is critical in facilitating skill acquisition. Errors were more likely to occur when students categorized examples pertaining to gathering, corroborating, and contextualizing evidence. Errors were also more likely when students generated exemplifying-domain self-explanations. Learning-domain self-explanations had no impact on categorization accuracy.

These findings suggest that students require additional training in generating appropriate self-explanations. Furthermore, the model can be embedded in the environment to guide the delivery of instructional explanations designed according to empirically-based guidelines [6] and tailored to assist students in categorizing challenging examples. In making these design modifications, the training module stands to foster metacognitive activities that are critical in learning through historical inquiry.

References

1. Azevedo, R., Feyzi-Behnagh, R.: Dysregulated Learning with Advanced Learning Technologies. *Journal of e-Learning and Knowledge Society* 7(2), 9–18 (2011)
2. Greene, J.A., Bolick, C.M., Robertson, J.: Fostering Historical Knowledge and Thinking Skills Using Hypermedia Learning Environments: The Role of Self-Regulated Learning. *Computers & Education* 54, 230–243 (2010)
3. Poitras, E., Lajoie, S., Hong, Y.-J.: The Design of Technology-Rich Learning Environments as Metacognitive Tools in History Education. *Instructional Science* (2011)
4. Poitras, E., Lajoie, S., Nokes, J., Hong, Y.-J.: The MetaHistoReasoning Tool: Fostering Domain-Specific Metacognitive Processes While Learning through Historical Inquiry. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011*. LNCS, vol. 6738, pp. 609–611. Springer, Heidelberg (2011)
5. Poitras, E., Lajoie, S.: A Metacognitive Framework to Improve Learning through Historical Inquiry. *Contemporary Educational Psychology* (in preparation)
6. Renkl, A., Hilbert, T., Schworm, S.: Example-Based Learning in Heuristic Domains: A Cognitive Load Theory Account. *Educational Psychology Review* 21(1), 67–78 (2009)
7. Van Gog, T., Rummel, N.: Example-Based Learning: Integrating Cognitive and Social-Cognitive Research Perspectives. *Educational Psychology Review* 22, 155–174 (2010)
8. Hicks, D., Doolittle, P.E.: Multimedia-Based Historical Inquiry Strategy Instruction. Do Size and Form Really Matter? In: *Research on Technology in Social Studies Education*, pp. 127–154. Information Age Publishing, Greenwich (2009)
9. Azevedo, R., Moos, D.C., Johnson, A.M., Chauncey, A.D.: Measuring Cognitive and Metacognitive Regulatory Processes During Hypermedia Learning: Issues and Challenges. *Educational Psychologist* 45(4), 210–223 (2010)
10. Schworm, S., Renkl, A.: Computer-Supported Example-Based Learning: When Instructional Explanations Reduce Self-Explanations. *Computers & Education* 46(4), 426–445 (2006)

An Indicator-Based Approach to Promote the Effectiveness of Teachers' Interventions

Aina Lekira, Christophe Després, Pierre Jacoboni, and Dominique Py

LIUM, Université du Maine, Avenue Laënnec,
72085 Le Mans Cedex 9, France
{aina.lekira, christophe.despres,
pierre.jacoboni, dominique.py}@lium.univ-lemans.fr

Abstract. This paper deals with the feedback given to teachers in order they better manage their activities. We support teachers' activities, especially their interventions effectiveness, by giving them feedback about the effects of these interventions through an indicator-based approach. To investigate the benefits of the introduction of this aid, we conducted experimentations in the field of object-oriented programming. Experimental outcomes show that giving teachers information about the effects of their interventions increases their effectiveness qualitatively and quantitatively; it also has a positive impact on learners' ability to solve their problems.

Keywords: Indicators, Meta-indicators, Tutoring, Interventions, Synchronous monitoring.

1 Introduction

We deal with teachers' activities instrumentation in the context of mediated and synchronous tutoring within the framework of the support of teachers' interventions. Some attempts for that purpose have been made [1][2]. Research outcomes most often lead to models and tools design in order to enable teachers to monitor, supervise or evaluate learners' activities through indicators [3][4]. An indicator is a "*variable that describes 'something' related to the mode, the process or the 'quality' of the considered 'cognitive system' activity; the features or the quality of the interaction product; the mode or the quality of the collaboration [...]*" [5].

Our objective is to support teachers in these instrumented situations by giving them feedback about their work, especially their interventions. To reach this goal, we rely on an indicator-based approach that supplies teachers with the effects of their interventions; this supply is made through the study of the evolution of the indicator values which allowed to detect the critical situation at the root of the intervention.

The indicator-based approach has been implemented in the TEL system HOP3X [6]. We conducted two experimentations by using HOP3X, in the field of object-oriented programming. We compared a situation in which teachers have information about the effects of their interventions to another one in which this information is not supplied

and we observed that meta-indicators had a positive impact on teachers’ and learners’ activities.

The indicator-based approach is detailed in section 2. Hop3x, the TEL system used during the experimentations, is presented in section 3. Section 4 describes the experimentations. Their results are analyzed and discussed in section 5.

2 An Indicator-Based Approach

During the regulation of learners’ activities, teachers intervene when a learner is faced with a situation considered pedagogically interesting. To detect these situations, we rely on the information from indicators calculation. Assuming that teachers intervene because of indicator values identified as critical, we propose to give teachers feedback about the evolution of these indicator values through meta-indicators calculation. A meta-indicator is an indicator which gives information about the evolution of other indicators.

2.1 Categories of Indicators

Indicators about learners’ activities can give teachers information about learners’ progress, trails or productions. They reflect the gap between what learners have done and what teachers expect. We integrate this latter within the definition of an indicator through an acceptability domain of its value, which can be a value, a threshold, an interval or a set. Examples of indicators are presented in Tab.1.

Table 1. Examples of observation needs and corresponding indicators with the type and value of their reference

Observation need	Indicator	Reference type	Reference value
Check if the students master the concept of object sender/receiver	The number of parameters of the <i>equals</i> method	Value	1
Check if the students write a structured and commented program	Percentage per class of methods commented by <i>JavaDoc</i> comments	Threshold	>70
Verify/Ensure that a learner is not too late nor too early compared to the rest of the group	Relative progress of a learner compared to the average progress of the group	Interval	[-2, 2]
Verify if students master the concept of encapsulation in a given class X.	The visibility of the instance variables in class X.	Set	{private, protected}

2.2 Meta-indicators and Interventions

Indicators provide teachers with information about learners’ activities. When an indicator value doesn’t belong to its acceptability domain, the situation that it reflects is said “critical”. In this case, teachers may intervene. Thus, an intervention is linked to this/these indicator(s). To observe the effectiveness of teachers’ interventions, it is

necessary to follow the evolution of indicators values and verify if they change positively. To do that, meta indicators reflect the positive, negative or zero evolution of indicators values.

We also associate to an intervention the meta-indicators which follow the evolution of indicators at the root of the intervention. Thus, an intervention is successful if all the meta-indicators associated with it have evolved positively (“success”). Conversely, an intervention fails if all the meta-indicators associated with it have evolved unfavourably (deterioration or no change of the indicator value). Otherwise, we consider that an intervention has some effectiveness measured by the percentage of meta-indicators which reflect respectively positive evolution, negative evolution or no evolution of the indicator value (“success”, “no effect”, “improvement”, “deterioration”).

3 Hop3x: An Implementation of the Indicator-Based Approach

HOP3X is a track-based TEL system which aims at supporting learning programming [6]. We use it in object-oriented programming. HOP3X is composed of three applications:

- HOP3X-STUDENT allows learners to edit, compile and run codes and programs. It also allows them to call teachers for help if needed.
- HOP3X-TEACHER provides teachers with a real-time supervision of learners' activities. It allows teachers to visualize indicators and meta-indicators through a monitoring interface that uses a color code: red for the interventions that have failed, green for those which passed, orange green for those in which a majority of meta-indicators have evolved positively and orange red for the others.
- HOP3X-SERVER collects interaction tracks of the learning session participants and saves them as Hop3x events (e.g. a creation/suppression of a project or a file, a text insertion/deletion, a compilation, a run, an annotation, an audio intervention, etc.). These tracks allow indicators and meta-indicators calculation by TOOLUTL which uses the UTL meta-language and the DCL4UTL [7] associated language.

4 Experimentations Description

In order to measure the benefits of giving teachers feedback about their interventions, in the context of synchronous tutoring, we conducted two experimentations: one with available meta-indicators for teachers and another one without meta-indicators. These experimentations were carried out through two college years and dealt with practical work activities which are part of a course entitled “Object-oriented Programming and Java”. This course is provided to third-year undergraduate students of the Maine University (France), who are neophytes in Java programming. Before each learning session with HOP3X in which students practiced Java programming, students attended lectures and tutorials about the notions and concepts they would implement during practical work. The two experimentations had the same context: the same two teachers participated in the two experimentations, the same pedagogical scenario was used and the students involved had the same background and had taken the same

courses. In collaboration with the teaching team, we identified and defined 62 indicators that have been modeled with UTL. These indicators were available to teachers during the two experimentations.

The first experimentation (**experimentation 1**) was carried out from January to February 2010. It involved thirty-six students (**group 1**). The second experimentation (**experimentation 2**) was carried out from January to February 2011 and took place with forty-five students (**group 2**).

Regarding the amount of students' productions and teachers' interventions, there was no major difference between the two experimentations. On average, for a three-hour practical work, per student, there were 3995 events for group 1 and 4391 events for group 2. Concerning teachers' interventions, there were 84 interventions for group 1 (2.33 interventions per student) and 96 interventions for group 2 (2.13 interventions per student).

5 Experimental Outcomes and Discussion

Our analysis is twofold. First, we want to see if the meta-indicators improve teachers' performance and the effectiveness of their interventions. Second, we want to observe if this improvement of teachers' performance enhances learners' outcomes.

5.1 Contribution of Meta-indicators to Teachers' Activities

In this section, we study the benefits of giving teachers information about the effects of their interventions through the comparison of the results of experimentation 2 — in which meta-indicators were provided — and experimentation 1 — in which teachers had no available meta-indicators.

In this analysis, we deal with interventions which can be either unique i.e. only composed of the original intervention (we name it single intervention), either a sequence of interventions i.e. a series of interventions on the same subject. A sequence of interventions is successful if the last intervention is successful and the student eventually corrects the problems about which the teacher has intervened. Otherwise, it is a failure.

Meta-indicators provide teachers with information about the result (success or failure) of their interventions. In case of failure, they remind the teacher that the student has not yet solved the problem. Therefore, the teacher is encouraged to intervene again. Thus, we can assume that the supply of meta-indicators will increase the global rate of successful interventions (H1) and, in particular, it will increase the number of successful interventions because of re-interventions (H2).

As shown in Fig.1, 75.99% of all interventions (i.e. both single interventions and sequences of interventions) are successful for group 1. For group 2, this rate is 91.42%. These results show an improvement of the global rate of successful interventions that increases by 15.43 points. This result validates our first hypothesis H1: on the whole, teachers with available meta-indicators were better able to make their interventions efficient than those without available meta-indicators.

The explanation of this increase of the global rate of interventions that have succeeded is twofold. First, it is due to the improvement of teachers' re-interventions. Indeed, when the original interventions failed, there were 40.36% of cases in which teachers re-intervened in group 1. This rate is 75.02% in group 2. Meta-indicators are a permanent reminder of a possible failure of teachers' interventions. This permanent reminder fosters teachers to re-intervene: the rate of re-interventions in group 2 is 34.66 points higher compared to the group 1. As a consequence, the rate of successful sequences of intervention rises by 4.95 points — from 9.33% for group 1 to 14.28% for group 2 — and the rate of failed unique interventions decreases by 14.29 points — from 20.00% for group 1 to 5.71% for group 2. We explain this difference because of meta-indicators: sequence of interventions contains re-interventions which are triggered by meta-indicators. These results tend to validate our second hypothesis H2: there was an increase of the number of re-interventions and there was also a rise in the number of effective interventions through re-interventions.

Second, as shown in Fig.1, the rate of successful unique interventions has increased between group 1 and group 2. It cannot be due to meta-indicators because there was no re-intervention here. We explain this difference by the fact that the same teachers were involved in the two experimentations. After experimentation 1, teachers gained some expertise by remembering some effective interventions, so they could better target their choice of remediation strategies.

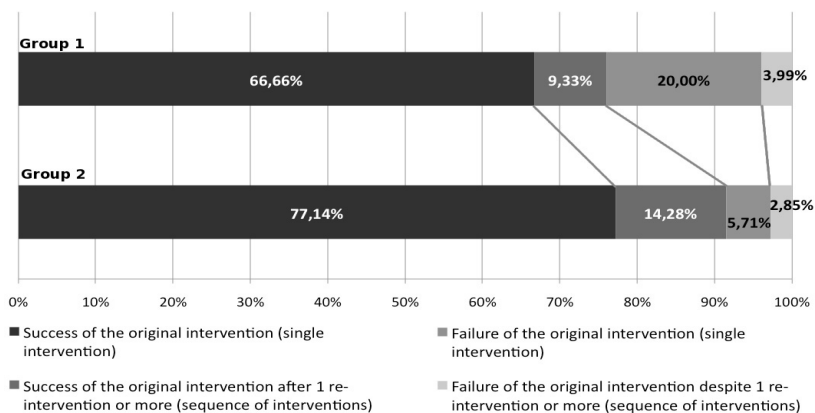


Fig. 1. Distribution of teachers' interventions for group 1 and group 2

5.2 Contribution of Meta-indicators to Learners' Activities

We want to see if the improvement of teachers' performance and the increase of their re-interventions has a positive impact in learners' activities. To reach this goal, we brought out critical situations (CS) i.e. situations in which indicators – about learners' activities – values were not acceptable. Among these situations, some have evolved positively (indicators values returned to normal at the end of the session) and others have not. In addition, among the CS, some have been resolved by students'

self-correction, others have not been the subject of an intervention since teachers had chosen not to treat them because they had more serious CS to deal with. Here, we are interested in the CS that have been the subject of an intervention.

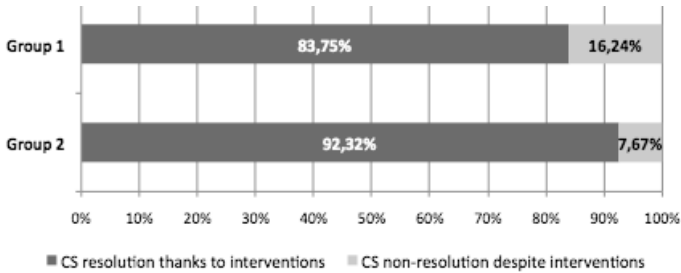


Fig. 2. Distribution of CS for group 1 and group 2

Fig.2 shows the distribution of CS which have been treated in interventions depending on their resolution at the end of the session. It shows that the rate of solved CS thanks to interventions increases from 83.74% to 92.32% between group 1 and group 2. This increase is probably due to the successful interventions, which increase with the introduction of meta-indicators (see section 6.1). However, for group 2, there was 7.67% of non-solved CS despite interventions. This category of non-solved CS despite interventions corresponds to interventions on learners who had great difficulty in programming.

This result tends to prove that the introduction of meta-indicators — which induces better interventions quantitatively and qualitatively (increase of the rate of re interventions and increase of the number of successful interventions) — has a positive impact on learners' performance because they were better able to solve their CS when teachers had feedback about their interventions.

6 Conclusion and Outlook

In this paper, we investigated the benefits of the aid provided to teachers when they manage their activities in real time, and in particular their interventions. Through a generic indicator-based approach, we provide teachers with information about learners' activities in using indicators calculated from learners' tracks. Relying on these indicators, teachers can intervene about critical situations and can have information about their intervention through meta-indicators. The indicator-based approach is reusable. It can be deployed in any track-based TEL system. It consists, on the one hand, in defining indicators in a given learning area and in categorizing them according to the fact that they have or not an acceptability domain, and on the other hand, in integrating the meta-indicators previously described.

Experimental results show that supplying teachers with information about their interventions improves the effectiveness of these interventions in a quantitative and qualitative ways. Moreover, these results also highlight that the improvement of teachers' performance has a positive influence on learners' performance.

Our mid-term objective consists in providing teachers with information which enables them to use their know-how acquired from one learner on others. To do that, we want to capitalize teachers' interventions depending on the measure of their effectiveness and suggest them effective interventions when they are in a similar context.

References

1. Pearce-Lazard, D., Poulouvassilis, A., Geraniou, E.: The Design of Teacher Assistance Tools in an Exploratory Learning Environment for Mathematics Generalisation. In: Wolpers, M., Kirschner, P.A., Scheffel, M., Lindstaedt, S., Dimitrova, V. (eds.) EC-TEL 2010. LNCS, vol. 6383, pp. 260–275. Springer, Heidelberg (2010)
2. Chen, W.: Supporting Teachers' Intervention in Collaborative Knowledge Building. *Journal of Network and Computer Applications* 29, 200–215 (2005)
3. Mazza, R., Dimitrova, V.: CourseVis: A Graphical Student Monitoring Tool for Facilitating Instructors in Web-Based Distance Courses. *International Journal in Human-Computer Studies* 65(2), 125–139 (2007)
4. Després, C.: Synchronous Tutoring in Distance Learning. In: *International Conference on Artificial Intelligence, Australia*, pp. 271–278 (2003)
5. ICALTS JEIRP: Interaction and Collaboration Analysis' Supporting Teachers and Students' self-Regulation, Deliverables 1, 2 and 3 (2004), <http://www.rhodes.aegean.gr/ltee/kaleidoscope-icalts>
6. HOP3X to learn Java Programming, <http://eiah.univ-lemans.fr/HOP3X/HOP3X.xml>
7. Pham Thi Ngoc, D., Iksal, S., Choquet, C.: Re-engineering of Pedagogical Scenarios Using the Data Combination Language and Usage Tracking Language. In: *10th IEEE International Conference on Advanced Learning Technologies, Tunisia*, pp. 681–685 (2010)

Limiting the Number of Revisions while Providing Error-Flagging Support during Tests

Amruth N. Kumar

Ramapo College of New Jersey,
Mahwah, NJ 07430, USA
amruth@ramapo.edu

Abstract. Error-flagging support provided during tests leads to higher scores, as reported in literature. Although many beneficial factors contribute to higher scores, one undesirable contributing factor is that students abuse error-flagging feedback to find the correct answer through trial and error even when the test is not multiple-choice in nature. A limit can be placed on the number of revisions allowed per problem to foil the trial and error approach. A follow-up study was conducted to examine whether limiting the number of revisions allowed per problem yielded the benefits of error-flagging feedback while alleviating its shortcomings. The study also considered the effects of error-flagging feedback on partial scores. The findings are: even with a limit placed on the number of revisions per problem, students revised more often and scored higher with rather than without error-flagging. When students solved problems incorrectly without revisions, their solution qualified for more partial credit when error-flagging support was provided. When a limit was placed on the number of revisions and students solved problems correctly with revisions, they did so with fewer revisions when error-flagging feedback was provided than when it was not. When students solved problems incorrectly with revisions, even with a limit placed on the number of revisions, they revised more often with error-flagging than without, scored more partial credit, but did not take more time than when error-flagging was not provided. A limit on the number of revisions may discourage students from using error-flagging feedback as a substitute for their own judgment. Overall, students solved problems faster with error-flagging feedback, even though revisions prompted by such feedback can cost time.

Keywords: Error-flagging, Testing, Adaptation, Evaluation.

1 Introduction and Experiment

In a recent study of online tests that do not involve multiple-choice questions [1], students scored better on tests with rather than without error-flagging support. A follow-up study [2] found that when error-flag feedback is provided, students save time on the problems that they already know how to solve, and spend additional time on the problems for which they do not readily know the correct solution. It also found that students may abuse error-flagging support to find the correct solution by trial and

error. The work reported herein was conducted as a follow-up to study 1) whether limiting the number of revisions allowed per problem would yield the benefits of error-flagging feedback while foiling abuse; 2) the effect of error-flagging feedback on partially correct solutions.

This work is of relevance to the tutoring systems community in that adaptive tutors often use an online pretest to prime the student model. Since error-flagging feedback helps students avoid inadvertent mistakes, tutors that provide error-flagging feedback during their pretest can build a more accurate student model that facilitates better adaptation of tutoring content.

For the current study, two problem-solving software tutors were used in fall 2011. The tutors were on predicting the behavior of `while` and `for` loops in introductory computer programming. The `while` loop tutor targeted 9 concepts; `for` loop tutor targeted 10 concepts. The tutors presented problems on these concepts, each problem containing a program whose output had to be identified by the student. Each software tutor went through pretest-practice-post-test protocol in 30 minutes. *Since this is a study of the effect of error-flagging feedback during testing, data from only the pretest portion of the tutor was considered for analysis.*

The evaluations were conducted online and *in-vivo*. The tutors were used in introductory programming courses at 11 institutions which were randomly assigned to one of two groups: A or B. A partial cross-over design was used: students in group A served as test subjects on `while` loop tutor and control subjects on `for` loop tutor, while students in group B served as control subjects on `while` loop tutor and test subjects on `for` loop tutor. All else being equal, error-flagging feedback was provided during pretest to students in the test group, but not the control group. Error-flagging feedback was provided before the student submitted the answer.

When solving a problem, students identified the output of a program, one at a time. Identifying each output consisted of entering the output string free-hand, and selecting from a drop-down menu, line number of the code that generated the output. Students could go back and delete a previously entered output by clicking on the “Delete” button paired with it.

When error-flagging feedback was provided, if an answer was incorrect, it was displayed on red background if incorrect, and green background if correct. When error-flagging support was not provided, the answer was always displayed on white background. When error-flagging support was provided, no facility was provided for the student to find out why the output was incorrect, or how it could be corrected. The online instructions presented to the students before using each tutor explained the significance of the background colors.

Whether or not the tutor provided error-flagging feedback, students had the option to revise their answer (e.g., “Delete” button described earlier) before submitting it. The interface always displayed the number of available revisions (maximum 3). If the student used up all available revisions, thereafter, the student could add additional outputs, but could no longer delete any previously identified outputs. These features were described in the instructions presented to the students at the beginning of each tutor.

2 Results

For analysis, only those students were considered who had used both `while` and `for` loop tutors. Only those students were considered who attempted most of the pretest problems: at least 6 of the 9 problems on `while` loop tutor and 6 of the 10 problems on `for` loop tutor. Students who scored 0 or 100% on either pretest were excluded. This left a total of 155 students - 126 students in group A and 29 students in group B. In order to factor out the effect of the difference in the number of problems solved by the students, the average score per pretest problem (range 0 \rightarrow 1.0) was considered for analysis rather than the total score.

Score Per Problem: A 2 X 2 mixed-factor ANOVA analysis of the score per pretest problem was conducted with the treatment (without versus with error-flagging support) as the repeated measure and the group (group A with error-flagging on `while` loop versus group B with error-flagging on `for` loop pretest) as the between subjects factor.

A significant main effect was found for error-flagging [$F(1,153) = 77.662, p < 0.001$]: students scored 0.541 ± 0.040 without error-flagging and 0.820 ± 0.024 with error-flagging (at 95% confidence level). The difference was statistically significant [$t(154) = -14.289, p < 0.001$]. The effect size (Cohen's d) is 1.323, indicating a large effect – test group mean is at 90th percentile of the control group. So, even with a limit placed on the number of revisions per problem, *students scored more with error-flagging support during tests than without.*

A large significant interaction was found between treatment and group [$F(1,153) = 26.441, p < 0.001$]. As shown in Table 1, the group with error-flagging scored statistically significantly more than the group without error-flagging on both `while` loop pretest [$t(153) = 3.414, p = 0.001$] and `for` loop pretest [$t(153) = -6.050, p < 0.001$]. Similarly, each group scored more with error-flagging than without [$t(125) = -16.378, p < .001$] for group A and [$t(28) = -1.912, p = .066$] for group B.

Table 1. Average pretest score with and without error-flagging

	<code>while</code> loop pretest	<code>for</code> loop pretest
Without error-flagging	0.704 ± 0.087	0.503 ± 0.043
With error-flagging	0.827 ± 0.027	0.789 ± 0.051

Time Per Problem: A 2 X 2 mixed-factor ANOVA analysis of the time per pretest problem was conducted with the treatment as the repeated measure and the group as the between subjects factor. A significant main effect was found for error-flagging [$F(1,153) = 6.581, p = 0.011$]: students spent 122.412 ± 7.455 seconds without error-flagging and 95.609 ± 6.150 seconds with error-flagging support. The difference was statistically significant [$t(154) = 6.582, p < 0.001$]. The effect size (Cohen's d) is 0.617, indicating a large effect – test group mean is at 73rd percentile of the control group. So, overall, *students solved problems faster with error-flagging feedback*, even though revisions prompted by such feedback can cost time.

A large significant interaction was observed between treatment and group [$F(1,153) = 21.456, p < 0.001$]. As shown in Table 2, the group with error-flagging solved problems faster than the group without error-flagging, but the difference was not statistically significant on either pretest. The difference with versus without error-flagging was significant for group A [$t(125) = 8.826, p < .001$], but not for group B.

Table 2. Average pretest time per problem with and without error-flagging

	while loop pretest	for loop pretest
Without error-flagging	102.913 ± 13.525	126.900 ± 8.455
With error-flagging	91.594 ± 5.960	113.051 ± 19.269

Number of Revisions: A 2 X 2 mixed-factor ANOVA analysis of the number of revisions was conducted with the treatment as the repeated measure and the group as the between subjects factor. A significant main effect was found for error-flagging [$F(1,153) = 50.711, p < 0.001$]: students revised an average of 1.26 ± 0.232 times without error-flagging and 3.90 ± 0.623 times with error-flagging support. The difference was statistically significant [$t(154) = -7.988, p < 0.001$]. The effect size (Cohen's d) is -0.885 , indicating a large effect – test group mean is at 82nd percentile of the control group. So, even with a limit placed on the number of revisions per problem, *students revised their answers more often with error-flagging support than without*. Both the groups revised more often with error-flagging than without, as shown in Table 3. The difference with versus without error-flagging was significant for group A [$t(125) = -6.354, p < .001$] as well as group B [$t(28) = -6.011, p < .001$].

Table 3. Number of revisions with and without error-flagging

	while loop pretest	for loop pretest
Without error-flagging	1.17 ± .584	1.29 ± .253
With error-flagging	3.70 ± .721	4.76 ± 1.086

As in the previous study [2], we considered **four cases** for comparing students with and without error-flagging support:

1. Students solved a problem correctly without any revisions – we compared the time students took to solve each problem.
2. Students solved a problem incorrectly without any revisions – we compared the partial score and time spent per problem.
3. Students solved a problem correctly with revisions – we compared the number of revisions and time spent per problem.
4. Students solved a problem incorrectly with revisions – we compared the partial score, time spent per problem and number of revisions.

The limit placed on the number of revisions per problem is expected to affect cases 3 and 4 only.

Case 1 – Problem solved correctly without any revisions: Univariate analysis of variance of the time spent per problem yielded a significant main effect for treatment [$F(1,1135) = 33.462, p < .001$]: students spent 91.56 ± 6.99 seconds per problem without and 67.9 ± 4.57 seconds with error-flagging support. This confirms the earlier result - *when error-flagging support is provided, students save the time they would have spent re-checking their solution.*

Case 2 – Problem solved partially or incorrectly without any revisions: ANOVA analysis of the time spent per problem yielded significant main effect for treatment [$F(1,1146) = 7.178, p = .007$]: students solved the problems in 136.48 ± 7.108 seconds per problem without and 117.42 ± 12.726 seconds with error-flagging support. ANOVA analysis of the partial score yielded a significant main effect for treatment [$F(1,1146) = 183.288, p < .001$]: students scored $0.209 \pm .021$ points per problem without error-flagging, and $0.495 \pm .037$ points per problem with error-flagging support. *So, even when students solved problems incorrectly without revisions, their solution qualified for more partial credit when error-flagging support was provided. In this study, they also solved the problems faster than when error-flagging was not provided.*

Case 3 – Problem solved correctly, with revisions: ANOVA analysis of the time spent per problem yielded no significant main effect for treatment: [$F(1,290) = 0.166, p = 0.684$]: whereas students solved problems correctly in an average of 92.91 ± 13.01 seconds without error-flagging and 97.74 ± 11.00 seconds with error-flagging, the difference was not statistically significant.

Analysis of the number of revisions yielded a significant main effect for treatment: [$F(1,290) = 20.44, p < .001$]: students revised their answers $1.49 \pm .178$ times without error-flagging, and $1.16 \pm .056$ times with error-flagging. *So, when a limit was placed on the number of revisions, students solved problems correctly with fewer revisions when error-flagging support was provided than when it was not.* We speculate that when students are made aware of the limit placed on the number of revisions allowed, they deliberate more before revising and therefore, need fewer revisions. Fewer revisions may also explain why students spent less time with rather than without error-flagging feedback.

Revisions still carry a time penalty – among the problems students with error-flagging support solve correctly, the problems solved without revisions take significantly less time (67.9 ± 4.575 seconds) than the problems solved with revisions (97.74 ± 11.0 seconds) [$t(891) = 5.794, p < .001$].

Case 4 – Problem solved partially or incorrectly with revisions: ANOVA analysis of the time spent per problem yielded no significant main effect for treatment [$F(1,265) = .024, p = 0.876$]: students spent about the same amount of time without (145.79 ± 19.62 seconds) as with error-flagging support (147.7 ± 13.07 seconds). ANOVA analysis of the number of revisions yielded significant main effect for treatment [$F(1,265) = 8.411, p = 0.004$]: students revised $1.42 \pm .155$ times without error-flagging and $1.73 \pm .12$ times with error-flagging. ANOVA analysis of the partial credit earned by students yielded a significant main effect for treatment

[$F(1,265) = 27.82, p < .001$]: students scored $.221 \pm .067$ points without error-flagging and $.435 \pm .043$ with error-flagging. *So, even when a limit is placed on the number of revisions, students revise more often with error-flagging than without, score more partial credit, but do not take more time than when error-flagging is not provided.*

Table 4 lists the percentage of problems that were solved correctly/incorrectly, with/without revisions in the two treatments. Prior study had reported that students with error-flagging feedback solved a third fewer problems correctly without revisions than with revisions, presumably because students were using error-flagging feedback as a substitute for their own judgment. With the introduction of a limit on the number of allowed revisions, students with error-flagging feedback solved nearly three times as many problems correctly without revisions than with revisions. This reversal suggests that *a limit on the number of revisions may discourage students from using error-flagging feedback as a substitute for their own judgment.* As in the prior study, we note that the percentage of students who solved problems incorrectly without any revisions is far smaller with than without error-flagging. In other words, students take advantage of error-flagging feedback to fix an incorrect answer. It is clear that students with error-flagging support revise their solution far more than those without error-flagging support, whether or not the solution eventually turns out to be correct. The objective of limiting the number of revisions allowed per problem is to minimize the amount of time students spend revising solutions that eventually turn out to be incorrect, and/or increase the partial credit students score in such cases. Case 4 above bears out that this objective was met.

Table 4. Percentage of problems solved correctly/incorrectly, with and without revision

	Solution never revised		Solution revised	
	Correct	Partial/Incorrect	Correct	Partial/Incorrect
Without Error-Flagging	33.08	57.63	3.95	5.34
With Error-Flagging	47.14	22.60	16.74	13.52

In conclusion, placing a limit on the number of revisions per problem did yield the benefits of error-flagging feedback while foiling abuse. Even with the limit, students revised more often and scored higher with rather than without error-flagging. When students solved problems incorrectly without revisions, their solution qualified for more partial credit when error-flagging support was provided. With the limit in place, when students solved problems correctly with revisions, they did so with fewer revisions when error-flagging feedback was provided than when it was not. When students solved problems incorrectly with revisions, even with the limit in place, they revised more often with error-flagging than without, scored more partial credit, but did not take more time than when error-flagging was not provided. A limit on the number of revisions discourages students from relying on error-flagging uncritically. Overall, students solved problems faster with error-flagging feedback, even though revisions prompted by such feedback can cost time. This makes the process of using a pretest prime the student model in an adaptive tutor more efficient.

Acknowledgments. Partial support for this work was provided by the National Science Foundation under grant DUE-0817187.

References

1. Kumar, A.N.: Error-Flagging Support for Testing and Its Effect on Adaptation. In: Alevan, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6094, pp. 359–368. Springer, Heidelberg (2010)
2. Kumar, A.N.: Error-Flagging Support and Higher Test Scores. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS(LNAI), vol. 6738, pp. 147–154. Springer, Heidelberg (2011)

Towards Academically Productive Talk Supported by Conversational Agents

Gregory Dyke, David Adamson, Iris Howley, and Carolyn Penstein Rosé

Carnegie Mellon University, Pittsburgh, PA, USA
{gdyke, dadamson, ihowley, cprose}@cs.cmu.edu

Abstract. In this paper, we investigate the use of conversational agents to scaffold on-line collaborative learning discussions through an approach called *academically productive talk*. In contrast to past work, which has involved using agents to elevate the conceptual depth of collaborative discussion by leading students in groups through directed lines of reasoning, this approach lets students follow their own lines of reasoning and promotes productive practices such as explaining, stating agreement and disagreement, and reading and revoicing the statements of other students. We contrast two types of academically productive talk support for a discussion about 9th grade biology and show that one type in particular has a positive effect on the overall conversation, while the other is worse than no support. This positive effect carries over onto participation in a full-class discussion the following day. We use a sociolinguistic style analysis to investigate how the two types of support influence the discussion and draw conclusions for redesign. In particular, our findings have implications for how dynamic micro-scripting agents such as those scaffolding academically productive talk can be used in consort with more static macro- and micro-scripting.

Keywords: conversational agents, discussion scaffolding, collaboration scripting.

1 Introduction

In recent years there has been a series of successful results in the area of conversational agents to support learning in chat environments [2][4][6-11]. Such agents have provided social support, affording the agents a more credible social standing in the group and helping to diffuse tension and create a productive learning environment. Furthermore, they have provided conceptual support, designed to elicit more depth by leading students through directed lines of reasoning, referred to as *knowledge construction dialogues* (KCDs).

While KCDs have been shown to lead to increased learning gains, particularly in situations where the conversational agents also provide social support [8], the necessity of designing them statically, with a pre-defined line of reasoning in mind both makes them hard to adapt to new subject material and does not fully exploit the benefits of collaborative learners following their own spontaneous lines of reasoning.

We have therefore drawn on extensive work related to support of classroom discourse [12-14] and collaborative learning [3,15] to investigate the use by conversational agents of facilitation moves that promote *academically productive talk* (APT). The aim of APT facilitation moves is to increase the amount of *transactivity* [3], by dynamically reacting to student discussions, encouraging them to build on each other's reasoning. Furthermore, as APT refers both to learners social positioning to each other and their conceptual positioning to knowledge, this provides us with a theoretical framework to better integrate the social and conceptual support aspects of conversational agents.

In this paper, we analyse our first study involving an agent performing APT moves in the context of a 9th grade biology classroom. We contrast two forms of support (one in which the agent performs the facilitation and a second in which the agent prompts another student to perform these moves) and a null condition with no support. We show that the presence of APT moves is correlated with improved student reasoning but also discover that while the first form of APT support shows promise, the second produces much less reasoning than would be expected. In order to better understand how the agents shape the conversation, both productively and unproductively, we employ a linguistic style process analysis to inform the next iteration of development of academically productive talk agents.

2 Academically Productive Talk

The notion of Academically Productive Talk stems from frameworks that emphasize the importance of social interaction in the development of mental processes, and has developed in parallel to similar ideas from the computer-supported collaborative learning community. Michaels, O'Connor and Resnick [12] describe some of the core dialogic practices of Accountable Talk along three broad dimensions:

- Students should be accountable to the learning community, listening to the contributions of others and building on them to form their own.
- Students should be accountable to accepted standards of reasoning, emphasizing logical connections and drawing reasonable conclusions
- Students should be accountable to knowledge, making arguments which are based explicitly on facts, written texts or other public information.

In order to introduce such practices in the classroom where they do not exist, it is necessary both to introduce students to unfamiliar dialogic interaction forms and to provide teachers with the means to scaffold these interaction forms. Drawing on over 15 years of observation and study, Michaels, O'Connor and Resnick [12] propose a number of core "moves" that teachers can draw upon in order to encourage the development of academically productive classroom discussion, among which are:

1. Revoicing: "So let me see if I've got your thinking right. You're saying XXX?" (with time for students to accept or reject the teacher's formulation);
2. Asking students to restate someone else's reasoning: "Can you repeat what he just said in your own words?";

3. Asking students to apply their own reasoning to someone else's reasoning: "Do you agree or disagree and why?";
4. Prompting students for participation: "Would someone like to add on?";
5. Asking students to explicate their reasoning: "Why do you think that?" or "How did you arrive at that answer?" or "Say more about that".

These moves have in common that they encourage reasoning statements (where the reasoning is made explicit) and they encourage transactivity [3], in which a reasoning *operates on* previous reasoning statement.

3 An Agent to Facilitate Academically Productive Talk

In this study, 50 students in four 9th grade biology periods were involved in an activity about diffusion and osmosis over two 42-minute periods on consecutive days. On the first day, they went through a 20 minute discussion in groups of three, in which a conversational agent presented them with three similar experimental setups, asking them to make predictions, watch a video, record their observations and provide explanations. This agent also provided APT scaffolding according the condition to which the groups were assigned. Furthermore the students were assigned roles related to APT scaffolding, with each student being responsible for performing one type of scaffold when appropriate. On the second day, the students participated in full class discussions, led by their teacher, at the end of which they took a post-test. Our research goal was to evaluate two forms of APT support. Our educational goal was to prepare the students as well as possible for the second day's discussion so that they might each benefit from it as much as possible.

3.1 Agent Support for Academically Productive Talk

The APT conversational agent was setup to accomplish two roles, neither of which provided any conceptual support. The first was to guide and instruct students through each phase of the activity. The second was to provide various levels of scaffolding using three of the "moves" proposed for the scaffolding of APT: prompting students to restate each other's reasoning, asking students whether they are in agreement with each other or not, and asking students to further explicate their reasoning.

The levels of support formed the three experimental conditions of our study:

- **Unsupported:** provide no APT support (only guiding through phases of activity)
- **Direct:** directly prompt students using APT moves ("John, could you say what Ann said in your own words")
- **Indirect:** prompt students to fulfill their assigned role ("Susan, could you ask John to say in his own words what Ann said").

In a pilot study using human "wizards of Oz" to provide APT support, students reacted unfavorably to the tutors – we hypothesized that in such a social situation a computer agent might not have the authority and credibility to make APT move requests of the human participants. The Indirect condition was designed to mitigate this situation by prompting learners to fulfill a role which had already been assigned to them *in lieu* of the agent.

Student1	I think it's going to get heavier.
Tutor	Student2, do you agree with what Student1 just said?
Student2	Wait I'm confused, please explain this again.
Student1	The egg will get bigger... heavier
Tutor	Student3, do you agree with what Student1 just said?
Student3	I can't understand.
Student3	oh, ok, I get it.

In the example above, when the agent detects that a student has made a prediction, it tries to get the other students to challenge the prediction. In this case, the response is that both of the other students admit that they are confused. This is actually a productive response since voicing confusion can be a precursor to a useful clarification dialogue. If students don't voice their confusion, they are less likely to achieve clarity within the conversation. In the Indirect condition, the Tutor would have said: *Student3, check with Student2 if they agree with Student1.*

4 Analysis

In our analysis, presented below, we initially examine the students' conversations and the effect of the ATP support conditions, by coding utterances for accountable talk moves, reasoning, and transactivity. Reasoning moves We then examine the effect on participation in the following day's full class discussion and the learning outcome subsequent to that discussion. This shows that the Direct condition outperforms the None and the Indirect.

We then perform a more detailed process analysis of linguistic style, to investigate why the Indirect condition performs so poorly. We investigate specific areas in the conversations where Indirect seems different from the other two conditions and isolate some of the issues which will be a focal point for APT agent redesign.

4.1 Reasoning in Conversations

We first coded for APT moves (which follow a set template), reasoning (0.72κ inter-rater reliability), and transactivity (0.70κ).

Table 1. APT Moves, Reasoning, Transactivity per student, across all conditions

Condition	Student APT Moves	APT Moves (including tutor)	Reasoning	Transactivity
Unsupported	.56 (2.7%)	1.6 (1.8%)	1.6 (11%)	.55 (2.7%)
Indirect	1.2 (4.9%)	3.8 (3.6%)	.53 (3.8%)	.13 (1.1%)
Direct	.67 (6.4%)	4.25 (7%)	2 (17%)	.92 (5.1%)

It should first be admitted that, overall, these results are lower than we had expected, with little reasoning and transactivity, mainly because of the difficulty the students had in carrying out the activity. The biggest difference between conditions shows up in terms of explicit displays of reasoning. Here there is a marginal effect on total number of reasoning moves per session $F(2,42) = 2.46$, $p < .1$, whereby students in the Direct condition produce a significantly greater number of reasoning moves than students in the Indirect condition, with the Unsupported condition not being significantly different from either (this same effect is significant when considering reasoning moves as a percentage $F(2,42) = 4.47$, $p < .05$). We did not see any statistical relationship between the number or percentage of Academically Productive Talk moves from the tutor and either student reasoning displays or transactive moves, however, we did see a significant but weak correlation between total percentage of Academically Productive Talk moves in a chat transcript from any source and the percentage of student contributions that were explicit displays of reasoning $R^2 = .11$, $p < .05$. Given this result, and the non-significant trend of the Indirect condition having more APT moves (both from the students and from all participants), it is surprising that the Direct condition outperformed the Indirect condition in producing reasoning.

4.2 Effect on Full-Class Discussion Participation

We examined the effect on class participation by counting contributions to the teacher-led discussion. Because the data were far from normally distributed, we first did a log transformation on the counts of contributions. We then performed an ANOVA analysis to determine whether there was a significant effect of condition. Since there was also a big difference in participation (and ability) across class periods, we retained class session as an additional factor in the ANOVA analysis. Both class session ($F(3,21) = 7.0$, $p < .005$) and condition ($F(2,26) = 4.2$, $p < .05$) were statistically significant¹. A post-hoc analysis using t-tests demonstrated that students in both the Direct and Indirect conditions contributed to the whole group discussion significantly more frequently than students who had been in the Unsupported condition. In both cases the effect size was about .75 standard deviations.

Table 2. Classroom discussion participation by Period and Condition

	Unsupported	Indirect	Direct
Period 1	4.2 (3.7)	8.0 (5.9)	3.7 (2.1)
Period 3	N/A	19 (8.5)	60 (49.5)
Period 6	1 (0)	3.2 (2.1)	5.8 (5.3)
Period 9	1 (0)	20 (0)	7 (0)

¹ Because of the difficulty in indentifying participating students in our audio recordings of the class discussion, this data is incomplete and the analysis may not accurately reflect the effect of participation on discussion. On the other hand, there is no reason to assume that our ability to identify students was biased by condition.

4.3 Learning Gains

The major factor influencing post-test results was the class period. The performance of all but the first period was so poor that no results of any significance were observable. To increase statistical power, we examined the effect of condition only on the first period (grouping Direct and Indirect conditions into the Supported condition) and only on questions related to providing generic explanations (as opposed to fact recall and observation understanding). Students in the Supported conditions scored significantly higher than those in the Unsupported $F(1,46) = 4.3$, $p < .05$, with an effect size of 1.1sd.

Table 3. Post-test score on Explain for Period 1, by condition (mark is out of 4 points)

	Supported	Unsupported
Explain	2 (.7)	1.1 (.9)

4.4 Process Analysis of Linguistic Style

From the above analyses it is surprising that the Indirect condition produced such poor reasoning compared to the Direct. We therefore examined the conversations in greater detail. In addition to Transactivity, which shows how students reason and operate on each others' reasoning, we coded the discussions for Heteroglossia (0.77κ inter-rater reliability), which shows how participants frame their assertions. The Heteroglossia framework is operationalized from Martin and White's theory of engagement (Martin & White, 2005), and here we describe it as identifying word choice that allows or restricts other possibilities and opinions. This creates a rather simple divide in possible coding terms for contributions (among statements that are ontask assertions):

- Heteroglossic-Expand (HE) phrases tend to make allowances for alternative views and opinions (such as "She claimed that glucose will move through the semi-permeable membrane.")
- Heteroglossic-Contract (HC) phrases attempt to thwart other positions (such as "The experiment demonstrated that glucose will move through the semi-permeable membrane.")
- Monoglossic (M) phrases make no mention of other views and viewpoints (such as "Glucose will move through the semi-permeable membrane.")

Overall, we find a positive and strong correlation between the average percentage of HE contributions in a discussion and the percentage of a student's contributions that are explicit reasoning displays, $R^2 = .5$, $p < .0001$. We also see a significantly smaller percentage of student contributions that are Heteroglossic Expand $F(2,41) = 6.79$, $p < .005$ in the Indirect condition.

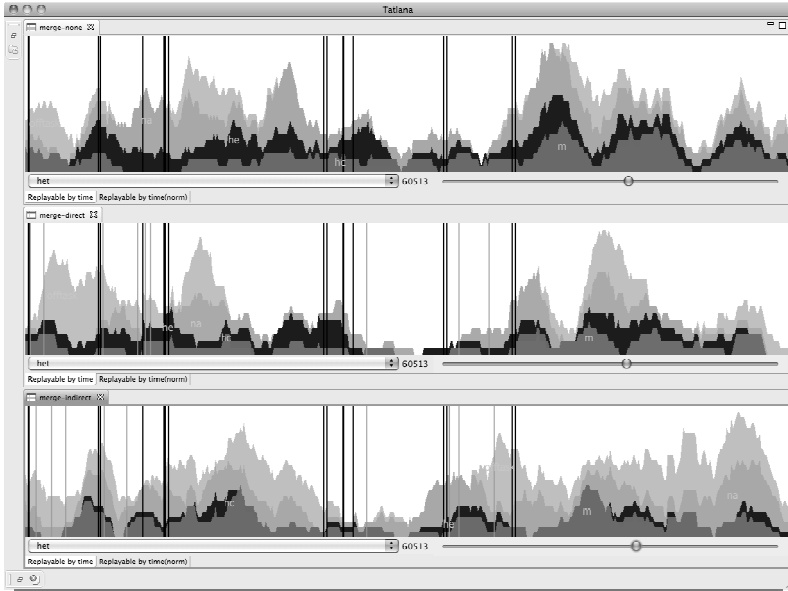


Fig. 1. Heteroglossia (M=red, HE=blue, HC=light blue, Non Assertion=orange, Offtask=cyan) distributed over time (horizontal axis) in Unsupported (top), Direct (center) and Indirect (Bottom). Dark bars indicate tutor turns present in all conditions and groups, green bars are the APT moves specific to individual conditions and groups. The dip in the middle is where the students watch the video.

To better understand what was happening in the indirect condition, we used Tatiana [5] to construct a visualization showing the running average distribution of heteroglossia codes over time within each of the three conditions (cumulating across the groups of each condition, cf. Fig. 1). We can see that during the prediction phase, before going to see the video but after several ATP moves by the agent, there is a marked lack of HE turns and a marked presence of M turns in the Indirect condition compared to the two others. The HE turns remain low throughout. By investigating these phases of the conversation more closely, we saw that HE statements tended to be predictions and explanations, whereas the M statements tended to be statements of incomprehension.

Closer examination revealed that this was often triggered by the agent's macro-scripting of the activity (instructions of what to do) interfering with its micro-scripting of the APT (e.g. Fig. 2). Furthermore, the agent frequently lost credibility as performing moves in inappropriate places was doubly harmful when the students followed its instructions (e.g. asking for a revoice when there was nothing useful to revoice).

Tutor	You should now move on to discussing Condition C
S041	The world is going to end in 2012
Tutor	S027, now would be a good time to ask S034 to build on what S041 is saying.
S034	I'm so confused!
S027	034, would you like to build onto what 041 is saying? And me too!
Tutor	When you agree, write down your predictions on your worksheet.

Fig. 2. Inappropriate timing of request for revoicing in the Indirect condition

5 Discussion and Implications for Redesign

The analysis of heteroglossia helped us identify the major way in which the Indirect discussions were different from those in the other two conditions. It also showed that lack of heteroglossic turns and presence of monoglossic turns could be an indicator of areas where the tutor was negatively impacting the discussion. Aside from the interference between static macro- and dynamic micro- scripting, we also saw that students were often frustrated when the agent was unable to provide them with any conceptual insight.

In redesign for re-conducting a similar study this year, the agent scripting issues were solved as discussed in [1] by implementing a better coordination algorithm so as to avoid collisions from different agent types. The activity was simplified so as to be better tailored to the students abilities. Last, and most importantly, the revoicing and addon APT moves were adopted because of their ability to provide simple conceptual nudges. The revoicing agent compares student utterances to a set of sentences that it knows to be true about the domain. When it finds a sentence which is semantically similar, it asks the students whether it is an appropriate revoice (e.g. Fig. 3). The addon agent knows about a list of concepts which ought to be discussed and will prompt struggling groups to discuss these concepts. Both these agents allow for rapid extension to new domains without the need to develop elaborate lines of reasoning.

This year's study has just been conducted and preliminary analysis shows that the amount of reasoning is greater by at least an order of magnitude than the amount shown in this condition. The agents also were considered more credible and useful interlocutors than in the study presented in this paper.

S25	it mixed with the water and it turned green because of how much glucose is in the water
Tutor	Would that be the same as saying "a glucose test strip changes from yellow to green when glucose is there?"
S25	Yes

Fig. 3. In this excerpt from the new revoicing agent, the tutor's utterance both reframes the student utterance more precisely, and leaves the responsibility for producing and evaluating reasoning with the student

6 Conclusion

In this paper, we presented a new kind of conversational agent for learning, based on the theoretical framework of Academically Productive Talk. Such agents are designed to behave as beneficial generic participants in collaborative learning discussion situations. We described a study in which two different implementations of such APT support are compared and contrasted with an Unsupported condition. While our activity proved to be slightly too difficult, the Supported conditions are shown to provide better learning outcomes and increased participation in subsequent classroom discussion. The Direct condition is shown to outperform the Indirect condition in increasing the amount of student reasoning. A process analysis of linguistic style is used to investigate this difference more closely, revealing several issues with the agents as implemented. In a promising redesign, we implemented new kinds of APT moves such as revoicing and adding on and a better coordination mechanism for loosely coupled agents. We believe APT agents open the doors to creating agents which can be reused in a variety of contexts with minimal adaptation effort. Furthermore, they provide new opportunities for controlled research into the effects and pertinence in context of various APT and other discussion scaffolding moves.

Acknowledgments. This work was funded by NSF SBE-0836012.

References

1. Adamson, D., Rosé, C.P.: Coordinating Multi-dimensional Support in Collaborative Conversational Agents. In: Cerri, S.A., Clancey, B. (eds.) ITS 2012. LNCS, vol. 7315, pp. 347–352. Springer, Heidelberg (2012)
2. Ai, H., Kumar, R., Nguyen, D., Nagasunder, A., Rosé, C.P.: Exploring the Effectiveness of Social Capabilities and Goal Alignment in Computer Supported Collaborative Learning. In: Alevén, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6095, pp. 134–143. Springer, Heidelberg (2010)
3. Berkowitz, M., Gibbs, J.: Measuring the developmental features of moral discussion. *Merrill-Palmer Quarterly* 29, 399–410 (1983)
4. Chaudhuri, S., Kumar, R., Howley, I., Rosé, C.P.: Engaging Collaborative Learners with Helping Agents. Submitted to *Artificial Intelligence in Education* (2009)
5. Dyke, G., Lund, K., Girardot, J.-J.: Tatiana: an environment to support the CSCL analysis process. In: *CSCL 2009*, Rhodes, Greece, pp. 58–67 (2009)
6. Howley, I., Chaudhuri, S., Kumar, R., Rosé, C.P.: Motivation and Collaboration On-Line. Submitted to *Artificial Intelligence in Education* (2009)
7. Howley, I., Mayfield, E., Rosé, C.P.: Missing Something? Authority in Collaborative Learning. In: *Proceedings of Computer Supported Collaborative Learning* (2011)
8. Kumar, R., Rosé, C.P., Wang, Y.C., Joshi, M., Robinson, A.: Tutorial Dialogue as Adaptive Collaborative Learning Support. In: *Proceedings of Artificial Intelligence in Education* (2007)
9. Kumar, R., Ai, H., Beuth, J.L., Rosé, C.P.: Socially Capable Conversational Tutors Can Be Effective in Collaborative Learning Situations. In: Alevén, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6094, pp. 156–164. Springer, Heidelberg (2010)

10. Kumar, R., Rosé, C.P.: Architecture for building Conversational Agents that support Collaborative Learning. *IEEE Transactions on Learning Technologies Special Issue on Intelligent and Innovative Support Systems for Computer Supported Collaborative Learning* (in press)
11. Martin, J., White, P.: *The Language of Evaluation: Appraisal in English*. Palgrave (2005)
12. Michaels, S., O'Connor, C., Resnick, L.B.: Deliberative discourse idealized and realized: Accountable talk in the classroom and in civic life. *Studies in Philosophy and Education* (2007)
13. Resnick, L.B., Bill, V., Lesgold, S.: Developing thinking abilities in arithmetic class. In: Demetriou, A., Shayer, M., Efklides, A. (eds.) *Neo-Piagetian Theories of Cognitive Development: Implications and Applications for Education*, pp. 210–230. Routledge, London (1992)
14. Resnick, L., O'Connor, C., Michaels, S.: *Classroom Discourse, Mathematical Rigor, and Student Reasoning: An Accountable Talk Literature Review* (2007)
15. Weinberger, A., Fischer, F.: A framework to analyze argumentative knowledge construction in computer supported collaborative learning. *Computers & Education* 46, 71–95 (2006)

Automatic Evaluation of Learner Self-Explanations and Erroneous Responses for Dialogue-Based ITSs

Blair Lehman¹, Caitlin Mills², Sidney D'Mello², and Arthur Graesser¹

¹Institute for Intelligent Systems, University of Memphis, Memphis, TN 38152
{balehman, a-graesser}@memphis.edu

²Department of Psychology, University of Notre Dame, South Bend, IN 46556
{cmills4, sdmello}@nd.edu

Abstract. Self-explanations (SE) are an effective method to promote learning because they can help students identify gaps and inconsistencies in their knowledge and revise their faulty mental models. Given this potential, it is beneficial for intelligent tutoring systems (ITS) to promote SEs and adaptively respond based on SE quality. We developed and evaluated classification models using combinations of SE content (e.g., inverse weighted word-overlap) and contextual cues (e.g., SE response time, topic being discussed). SEs were coded based on correctness and presence of different types of errors. We achieved some success at classifying SE quality using SE content and context. For correct vs. incorrect discrimination, context-based features were more effective, whereas content-based features were more effective when classifying different types of errors. Implications for automatic assessment of learner SEs by ITSs are discussed.

Keywords: self-explanations, automatic scoring, adaptive responses, ITSs, natural language understanding.

1 Introduction

Learning is a complex process that involves both the acquisition of new knowledge and integration of new content with existing knowledge. This task can be especially difficult when learners' mental models are rife with gaps, inconsistencies, and misconceptions. One method to facilitate the learning process is to have instructors provide explanations and guidance. Another method is to allow learners to construct and refine their own mental models. The latter method represents a more active form of knowledge construction. This type of active knowledge construction, in which learners are encouraged to engage in a form of self-instruction [1], can be contrasted with more shallow learning that involves the mere accumulation of facts [2-3].

Self-instruction can be completed through a number of learning activities; one such activity is self-explanation. Self-explanations (SE) are a representation of the learner's current knowledge about a concept and involve making inferences as well as integrating new information into existing knowledge structures [4]. SEs can also facilitate learning by causing learners to realize where gaps or inconsistencies exist in

their knowledge [5-6]. The impact of SEs on learning can be especially strong when learners are required to apply skills to new situations [5, 7].

The value of SEs as a means to diagnose learner knowledge and facilitate learning has been acknowledged for some time. Many studies have taken advantage of the SE effect (e.g., [5, 8, 9]). For example, Chi et al. [5] had learners study example problems on Newtonian physics and engage in a talk-aloud while studying. They found that higher achieving learners generated SEs at each step of the example problem while working to create a more refined understanding of the concept. Less successful learners, on the other hand, did not generate their own SEs while learning.

The benefits of SEs have also been studied in the context of intelligent tutoring systems (ITS). Many ITSs incorporate SEs as part of the learning process and some even train learners to become more adept self-explainers [6-7, 10-12]. iSTART, for example, is an ITS that provides learners with SE and reading strategy training [12]. By providing learners with examples of high quality SEs, practice generating SEs, and additional reading strategies, iSTART is able to increase learners' reading comprehension skills [13].

In addition to promoting SE use and training learners to generate higher quality SEs, ITSs must also be capable of evaluating the quality of learner-generated SEs. If an ITS can provide learners with opportunities to self-explain and automatically assesses the quality of their SEs, the ITS can adaptively respond to any gaps in the learner's knowledge and begin to correct problematic misconceptions.

The process of understanding natural language contributions from learners, however, is not a trivial task because the responses are often short, conversational, fragmented, and syntactically incorrect. In one study, Williams and D'Mello [14] used linguistic properties to assess the quality of learner responses during expert human tutoring sessions. The Linguistic Inquiry Word Count [15] was used to classify answers as correct, partially-correct, vague, or error-ridden. Although this approach did not use any content-dependent words, they were able to correctly classify 45.2% of learner responses.

Other studies have used a more content-dependent approach for assessing learner contributions. Litman, Moore, Dzikovska, and Farrow [16] used content word matching to analyze corpora from tutoring sessions with an ITS and human tutors. Use of a domain-specific glossary yielded some success; however, approximately half of the content words in learner responses were misclassified. In a series of studies, Graesser, Penumatsa, Ventura, Cai, and Hu [17] made use of Latent Semantic Analysis (LSA) [18] to model learner knowledge during interactions with an ITS. LSA is a method to semantically compare two texts using a bag of words approach and dimensionality reduction techniques. By comparing learner responses to expectations (ideal responses) and common misconceptions, they were able to model learner knowledge at a level that was comparable to unskilled human tutors.

Research on natural language understanding (NLU) techniques to assess learner responses has also revealed that a combination of algorithms may be an effective method for diagnosing learner knowledge. Alevan, Popescu, and Koedinger [19] used the combination of a geometry knowledge base (e.g., keywords, ideal responses) and a statistical text classifier (NaïveBayes). The knowledge base incorporated hierarchical

ordering for comparisons of learner responses to correct or partially-correct example responses. When only the knowledge base was used to discriminate between correct and incorrect learner responses, 59.5% of responses were correctly classified [20]. However, when the classification model included both the knowledge base and statistical classifier, classification improved to 61% [21]. The negligible increase, when the statistical classifier was included (59.5% vs. 61%), was attributed to the large number of potential classifications for each SE (167 labels). When semantic similarity between labels, or types of error-ridden answers, was taken into account and reduced the number of potential labels, accuracy greatly increased to 81%.

Rus, McCarthy, Lintean, Graesser, and McNamara [22] examined seven algorithms to assess the quality of learner SEs from iSTART interactions. iSTART presents learners with a text and then asks them to explain the text in their own words. The algorithms were either word-based, syntactic, or a combination of word and syntactic information. Word-based algorithms assessed word-overlap between learner SEs and the original text. Seventy-four percent of paraphrase SEs were correctly classified via a combination of the entailment index [23], synonymy index, word-overlap, and LSA (see [22] for details).

Past research on automatic classification of learner contributions has focused on the response content (i.e., the words in the response), while context from the learning session has largely been ignored. In the present paper we attempt to expand upon these past results by augmenting a semantic analysis of the response content with information about the context surrounding the response. Similar to past research, we test a model that uses a weighted word-overlap algorithm as the predictive feature (*SE Content* model). We build on past research by testing a *Context* model that incorporates features of the response characteristics (e.g., SE response time) and larger learning context (e.g., order of topic presentation, prior performance in the learning session). We compare the individual models to a *Combined* model (Content + Context). Finally, taking a somewhat different approach, we tested a *Word-Based* model that exclusively relies on the words in SEs as predictive features without a knowledge-based model of correct and incorrect answers. The models were tested on a corpus of learner SEs collected from a previous study involving tutorial sessions on scientific reasoning topics.

2 Method

2.1 Participants

Participants were 76 undergraduate students from a mid-south university in the US who participated for course credit. Participants completed four learning sessions, one on each of the scientific reasoning topics (experimenter bias, control group, random assignment, replication). Order of topics was counterbalanced across participants.

2.2 Learning Sessions

The learning sessions consisted of the human learners engaging in a dialogue with two animated pedagogical agents (tutor agent and peer student agent) while learning about scientific reasoning topics [24]. The dialogues consisted of learners discussing

the scientific merits of sample research studies with the tutor and peer agents. All studies had one subtle flaw and the learner had to identify this flaw by reflecting and deliberating on the quality of the methodologies of the studies.

The excerpt in Table 1 is an example trialogue between the two agents and the human learner. In this excerpt, the tutor agent (Dr. Williams) and the peer agent (Chris) are discussing a flawed study with the human learner (Bob). The study is flawed because the replication study used the same population as the original study.

Table 1. Excerpt of trialogue

Turn	Speaker	Dialogue Move
<p>A study showed that people who drank non-alcoholic beer can feel drunk. A professor ran this study with his class. He randomly assigned 50 students to drink the real beer and 50 the non-alcoholic beer. The professor was skeptical, so he ran a replication study two weeks later with the same method and same class. He found the same results again. So he concluded that people can feel just as drunk from non-alcoholic beer. <Research Study></p>		
1	Dr. Williams	I think we all need to go over the experiment some more. <Advance dialogue>
2	Dr. Williams	Alright, so I think that the real problem here is with the second experiment. <Assert information>
3	Chris	I disagree with you, Dr. Williams. I think the replication was done right. <Assert information>
4	Dr. Williams	Bob, was this a good or bad replication? Please type good or bad. <Forced-choice question>
5	Bob	bad <Response>
6	Dr. Williams	Tell me more about your thoughts on this Bob. <Pump for SE>
7	Bob	How can feel drunk drinking a non-alcoholic drink unless you had a alcoholic drink before taking the experiment. Its faulty and doesn't make sense. <Student SE>

Each learning session began with learners reading the research study and then beginning a trialogue with the agents. The discussion involved five trials. All trials consisted of the tutor (turn 2) and peer (turn 3) agents asserting their opinions, prompting the learner to intervene (turn 4), and obtaining the learner's response (turn 5). Learner responses were either correct or incorrect. Trials 3 and 4 also included pumps (turn 6) that required learner SEs (turn 7). Learners did not receive feedback on SE quality; the agents simply acknowledged learners' contributions (e.g., "Alright," "Okay"). This cycle was repeated in each trial, with each trial becoming more specific about the scientific merits of the study. The present paper will focus on Trials 3 and 4 because learners were asked to self-explain during these trials.

2.3 Procedure

Learners were tested individually over a two-hour session. First, learners signed an informed consent and completed the pretest. Next, learners read a short introduction on research methods. Learners then completed four learning sessions, one on each

scientific reasoning topic. Finally, learners completed the posttest and were fully debriefed. Pretest and posttest data is not relevant to the present analyses and will not be discussed any further.

2.2 Self-Explanation Coding

A total of 608 learner SEs were obtained from the learning sessions. Two human-raters coded the SEs as correct, partially-correct, or incorrect. A subset of the corpus was first coded to compute reliability ($kappa = .842$). The corpus was then divided evenly between the raters for coding. For the current analyses, partially-correct and incorrect SEs were collapsed into one category (incorrect) because there were very few instances of partially-correct SEs (8.72%). This yielded 36% correct responses and 64% incorrect responses.

Incorrect SEs were further coded for types of error-ridden reasoning. Learner SEs could be rated as Correct, Error Type 1, Error Type 2, Error Type 3, Unclassified, or Frozen Expression. Incorrect learner SEs that did not fit into one of the error type categories were grouped as *Unclassified*. Frozen expressions, SEs unrelated to the topic, were not included in the current analyses because a speech act classifier that can accurately identify these utterances has already been developed [25].

Table 2 shows an example of a correct response, different error types, and a frozen expression. Error types were unique to each scientific reasoning topic and trial. Errors could vary from focusing on superficial features of the study rather than methodological issues (see Error Type 2) to complete misunderstandings of the concept being discussed (see Error Type 1).

Table 2. Examples of SE response types for Trial 3 of the replication topic

Response Type	Example
Correct Answer	It was bad since the study used the same people to replicate the study. Different people should have been used so the accuracy of the data could have been confirmed more firmly.
Error Type 1	I think that it was a good replication of the first study; however, I do not think that the first study was executed properly.
Error Type 2	How can feel drunk drinking a non-alcoholic drink unless you had a alcoholic drink before. It doesn't make sense.
Error Type 3	The professor was careful to conduct random assignment. That helps to make it a good replication. And he used the same people.
Unclassified	It was conducted well but the longevity of the study could not make it very accurate.
Frozen Expression	I don't know.

2.3 Semantic Matching

In order to evaluate the semantic quality of learner SEs, we first needed to create expected responses and expected errors. Prototypical correct responses and prototypical erroneous responses (for each error type) were created by a content

expert (see Table 2 for an example). Prototypical correct and erroneous responses were unique to each of the eight individual questions (4 topics x 2 trials).

Learner SEs were compared to prototypical correct and erroneous responses using an inverse word frequency weighted overlap (IWFOW) algorithm. The IWFOW algorithm is a word-matching algorithm in which each overlapped word is weighted on a scale from 0 to 1, relative to its inverse frequency in the English language using the CELEX corpus [26]. The inverse frequency allows for higher weighting of lower frequency, more contextually relevant words (e.g., replication, bias), while higher frequency words (e.g., and, but) are given a lower weighting. Comparisons resulted in a match score between 0 and 1 (1 = perfect similarity).

3 Results and Discussion

3.1 Content, Context, and Combined Models

We tested three models to determine which SE features were most diagnostic of SE quality. The *Content Model* included the IWFOW match score (either to the prototypical correct or error type SE based on the classification task) and the number of words in the SE. The *Context Model* included SE response time, performance (correct or incorrect) and response time on the forced-choice question prior to the SE (see turn 4 in Table 1), and the order of topic presentation (e.g., first, second). These contextual features were selected because they are already logged by the learning environment and would not require additional processing for future SE classification. Finally, there was also a *Combined* model, which combined features from the two individual models.

Four classification algorithms from WEKA [27] were used to build and evaluate the models: NaïveBayes, IBk (nearest neighbor with $k = 10$), j48, and LogitBoost. The majority class algorithm (ZeroR) that classifies all SEs to the most prevalent group was used as the baseline comparison. Each algorithm was evaluated using 10-fold cross-validation. Two separate classification tasks were performed. The first task consisted of making a simple correct vs. incorrect discrimination, while the second task performed a fine-grained discrimination in terms of specific error types.

SEs were separated into eight groups based on scientific reasoning topic and trial. After removing frozen expression responses, there was an average of 71.9 responses per group ($SD = 2.42$; *Range* 69 to 75). The algorithms were evaluated on each SE group for both classification tasks. For each SE group the best algorithm (i.e., one out of the four algorithms that yielded the best performance) was selected. The best classification results were averaged across SE groups and constituted the Content, Context, and Combined models. Table 3 shows the results obtained for each classification task averaged across the eight groups.

We note that the Context model (74.0%) was the most successful for segregating correct from incorrect responses. Both the Content, $t(7) = 2.40$, $p < .05$, and Context models, $t(7) = 4.29$, $p < .01$, performed significantly better than the Baseline model. The Context model also significantly outperformed the Content model for correct-incorrect discriminations, $t(7) = 2.39$, $p < .05$. Both individual models outperformed the Baseline model for error type discriminations (Content: $t(7) = 8.02$, $p < .01$; Context: $t(7) = 2.69$, $p < .05$). However, it was the Content model that performed best

Table 3. Mean (SD) classification performance across groups

Model	Correct-Incorrect		Error Type	
	Accuracy (%)	Kappa	Accuracy (%)	Kappa
Baseline	64.6 (9.45)	.000 (.000)	43.3 (6.93)	.000 (.000)
Content	69.5 (6.74)	.248 (.080)	67.6 (4.44)	.501 (.108)
Context	74.0 (4.08)	.335 (.160)	50.3 (8.44)	.231 (.095)
Combined	74.3 (3.92)	.347 (.160)	67.4 (6.54)	.510 (.103)

for error discrimination (67.6%). Interestingly, the Content model was twice as more effective for error type classifications than the Context model, $t(7) = 4.70$, $p < .01$. Indeed, these models were differentially effective for different classification tasks.

When comparing correct and incorrect SEs, we found that learners with correct SEs took longer to self-explain, $t(14) = 3.14$, $p = .01$, and responded more accurately to the forced-choice question prior to self-explaining, $t(14) = 2.30$, $p < .05$. This suggests that learners who responded correctly took more time to thoughtfully construct a response. For erroneous SEs, error types only differed on match to the prototypical erroneous responses, $F(3) = 20.2$, $p < .01$, which is what could be expected. Furthermore, SEs that were grouped as *unclassified* had lower match scores to the prototypical erroneous responses.

Comparisons of the Combined model to the individual models were also quite informative. Combined models for both discrimination tasks outperformed the Baseline models (correct-incorrect: $t(7) = 2.86$, $p < .05$; error type: $t(7) = 8.26$, $p < .01$). However, the Combined model did not yield any noticeable improvements over the best performing individual model for either the correct vs. incorrect or error discrimination task (p 's $> .05$). The negligible improvement by the Combined models suggests that it may be beneficial for systems to not conduct a full classification model initially, but rather allot these resources only when needed. For example, if an SE is classified as correct, it is not necessary to conduct a full classification model and analyze the actual content of the SE.

3.2 Word-Based Models

We also attempted to classify SEs with only the words in responses as features. This was accomplished using the StringToWordVector package in WEKA to transform text strings (words) into numerical input using *tf-idf* (term frequency-inverse document frequency) weighting. The *tf-idf* weighting allows less frequent, more content-rich words to have higher weightings.

The same four classifiers were used to train the models and they were tested with ten-fold cross-validation. As in the previous analyses, SEs were separated by scientific reasoning topic and trial for classification. The best classifier for each individual SE group was then selected. The average classification accuracy (across the eight groups) for the correct vs. incorrect was 71.1% ($SD = 8.45$) with a kappa of .282 ($SD = .178$). For error discrimination, the average accuracy was 58.1%

($SD = 9.30$) with a kappa of .352 ($SD = .119$). The word-based models performed significantly better than the Baseline model for both discrimination tasks (correct vs. incorrect: $t(7) = 2.10, p < .1$; error type: $t(7) = 5.43, p < .01$).

These results suggest that while it is possible to classify SEs on the basis of words alone, the resultant models were less effective than the Content model (67.6% accuracy) for error classification. However, the word-based models were approximately equivalent to the Context model (74% accuracy) for correct vs. incorrect discrimination. This suggests that for fine-grained detection of learner errors, a knowledge-based approach of SE content is more appropriate [19-21].

4 Conclusion

Several ITSs have incorporated the assessment of learner natural language responses using NLU techniques such as LSA, word-overlap, and other linguistic features. We tested which response features (content, context, combination) were most effective at accurately assessing SE quality, both in terms of correct vs. incorrect discriminations and classifying different error types. We were able to achieve moderate success at SE classification with models that included either the response content or response context, but there were no improvements when the models were combined.

Previous work on the classification of learner contributions has focused on response content [16-17, 22]. We expanded these previous efforts by also incorporating features of the context. We found that the effectiveness of content- and context-based features differed depending on the discrimination task. More specifically, the context-based model was sufficient to make correct vs. incorrect discriminations but the content-based model was needed for more specific error type classification. An effective approach for classification systems, then, would be to initially use context-based features to determine whether an SE is correct or incorrect. If the SE is classified as incorrect, the content features can then be used to make a finer-grain distinction between types of erroneous responses.

One interesting and informative finding was that we were relatively successful at making a general correct vs. incorrect SE classification *without even considering* the actual SE response. The success of this context model, which incorporated the learner's prior performance and other informative parameters, suggests that it can be used to make predictive assessments of SE quality. This information can be used to decide the optimal time to ask learners to provide an SE. However, this conclusion should be taken with a modicum of caution because further empirical testing of this classification scheme will be necessary to determine how frequently SEs are misclassified and the impact this misclassification has on learning.

Automatic classification of SE quality and error-ridden reasoning has important implications for building adaptive and effective ITSs. Through the use of readily available context features as well as word-overlap comparisons, ITSs can use SEs to create a more accurate model of learner knowledge. ITSs can then use this information to provide individually tailored scaffolding based on errors identified in learner-generated explanations. This type of adaptive scaffolding will allow ITSs to more efficiently and effectively help learners to reach deeper levels of understanding.

Acknowledgement. The research was supported by the National Science Foundation (REC 0106965, ITR 0325428, HCC 0834847, DRL 1108845) and the Institute of Education Sciences (R305A080594). The opinions expressed are those of the authors and do not represent views of the NSF and IES.

References

1. Simon, H.: Problem solving and education. In: Tuma, D., Reif, F. (eds.) *Problem Solving and Education: Issues in Teaching and Research*. Erlbaum, Hillsdale (1979)
2. Graesser, A., Jeon, M., Dufty, D.: Agent technologies designed to facilitate interactive knowledge construction. *Discourse Processes* 45(4), 298–322 (2008)
3. Prosser, M., Trigwell, K.: *Understanding learning and teaching*. The Society for Research into Higher Education and Open University Press, Buckingham (1999)
4. Chi, M.: Self-explaining expository texts: The dual process of generating inferences and repairing mental models. In: Glaser, R. (ed.) *Advances in Instructional Psychology: Educational Design and Cognitive Science*, pp. 161–238. Erlbaum, Mahwah (2000)
5. Chi, M., Bassok, M., Lewis, M., Reimann, P., Glaser, R.: Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science* 13, 145–182 (1989)
6. Chi, M., de Leeuw, N., Chiu, M., LaVancher, C.: Eliciting self-explanations improves understanding. *Cognitive Science* 18, 439–477 (1994)
7. Renkl, A., Stark, R., Gruber, H., Mandl, H.: Learning from worked-out examples: The effects of example variability and elicited self-explanations. *Contemporary Educational Psychology* 23, 90–108 (1998)
8. McNamara, D.: SERT: Self-explanation reading training. *Discourse Processes* 38, 1–30 (2004)
9. Renkl, A.: Learning from worked-out examples: A study on individual differences. *Cognitive Science* 21, 1–29 (1997)
10. Alevin, V., Koedinger, K.: An effective meta-cognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor. *Cognitive Science* 26, 147–179 (2002)
11. Conati, C., VanLehn, K.: Toward computer-based support of meta-cognitive skills: A computational framework to coach self-explanation. *International Journal of Artificial Intelligence in Education* 11, 398–415 (2000)
12. McNamara, D., Levinstein, I., Boonthum, C.: iSTART: Interactive strategy trainer for active reading and thinking. *Behavioral Research Methods, Instruments, and Computers* 36, 222–233 (2004)
13. O'Reilly, T., Best, R., McNamara, D.: Self-explanation reading training: Effects for low-knowledge readers. In: Forbus, K., Gentner, D., Regier, T. (eds.) *Proceedings of the 26th Annual Meeting of the Cognitive Science Society*, pp. 1053–1058. Erlbaum, Mahwah (2004)
14. Williams, C., D'Mello, S.: Predicting Student Knowledge Level from Domain-Independent Function and Content Words. In: Alevin, V., Kay, J., Mostow, J. (eds.) *ITS 2010, Part II. LNCS*, vol. 6095, pp. 62–71. Springer, Heidelberg (2010)
15. Pennebaker, J., Francis, M., Booth, R.: *Linguistic Inquiry and Word Count (LIWC)*. Erlbaum, Mahwah (2001)
16. Litman, D., Moore, J., Dzikovska, M., Farrow, E.: Using natural language processing to analyze tutorial dialogue corpora across domains and modalities. In: Dimitrova, V., Mizoguchi, R., DuBoulay, B., Graesser, A. (eds.) *Proceedings of 14th International Conference on Artificial Intelligence in Education*, pp. 149–156. IOS Press, Amsterdam (2009)

17. Graesser, A., Penumatsa, P., Ventura, M., Cai, Z., Hu, X.: Using LSA in AutoTutor: Learning through mixed-initiative dialogue in natural language. In: Landauer, T., McNamara, D., Dennis, S., Kintsch, W. (eds.) *Handbook of Latent Semantic Analysis*, pp. 243–262. Lawrence Erlbaum, Mahwah (2007)
18. Landauer, T., McNamara, D., Dennis, S., Kintsch, W. (eds.): *The handbook of latent semantic analysis*. Erlbaum, Mahwah (2007)
19. Alevin, V., Popescu, O., Koedinger, K.: Towards tutorial dialog to support self-explanation: Adding natural language understanding to a cognitive tutor. In: Moore, J., Redfield, C., Johnson, W. (eds.) *Proceedings of the 10th International Conference on Artificial Intelligence in Education*, pp. 246–255. IOS Press, Amsterdam (2001)
20. Popescu, O., Koedinger, K.: Towards understanding geometry explanations. In: Rose, C., Freedman, R. (eds.) *Building Dialogue Systems for Tutorial Applications, Papers of the 2000 AAAI Fall Symposium*, pp. 80–86. AAAI Press, Menlo Park (2000)
21. Alevin, V., Popescu, O., Koedinger, K.: Pilot-Testing a Tutorial Dialogue System That Supports Self-Explanation. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) *ITS 2002. LNCS*, vol. 2363, pp. 344–354. Springer, Heidelberg (2002)
22. Rus, V., McCarthy, P., Lintean, M., Graesser, A., McNamara, D.: Assessing student self-explanations in an intelligent tutoring system. In: McNamara, D., Trafton, J. (eds.) *Proceedings of the 29th Annual Conference of the Cognitive Science Society*, pp. 623–628. Erlbaum, Mahwah (2007)
23. Rus, V., McCarthy, P.M., Graesser, A.C.: Analysis of a Textual Entailer. In: Gelbukh, A. (ed.) *CICLing 2006. LNCS*, vol. 3878, pp. 287–298. Springer, Heidelberg (2006)
24. Lehman, B., D’Mello, S.K., Strain, A.C., Gross, M., Dobbins, A., Wallace, P., Millis, K., Graesser, A.C.: Inducing and Tracking Confusion with Contradictions during Critical Thinking and Scientific Reasoning. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011. LNCS*, vol. 6738, pp. 171–178. Springer, Heidelberg (2011)
25. Olney, A., Louwerse, M., Mathews, E., Marineau, J., Hite-Mitchell, H., Graesser, A.: Utterance classification in AutoTutor. In: Burstein, J., Leacock, C. (eds.) *Building Educational Applications using Natural Language Processing: Proceedings of the HLT - NAACL Conference 2003 Workshop*, pp. 1–8. Association for Computational Linguistics, Philadelphia (2003)
26. Baayen, R., Piepenbrock, R., Gulikers, L.: *The CELEX lexical database (Release 2) [CD-ROM]*. University of Pennsylvania, Linguistic Data Consortium, Philadelphia (1995)
27. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: *The WEKA data mining software: An update*. *ACM SIGKDD Explorations Newsletter* 11(1), 10 (2009)

Group Composition and Intelligent Dialogue Tutors for Impacting Students' Academic Self-efficacy

Iris Howley, David Adamson, Gregory Dyke, Elijah Mayfield, Jack Beuth,
and Carolyn Penstein Rosé

Carnegie Mellon University, Pittsburgh, PA

{ihowley, dadamson, gdyke, emayfiel, beuth, cprose}
@andrew.cmu.edu

Abstract. In this paper, we explore using an intelligent dialogue tutor to influence student academic self-efficacy, as well as its interaction with group self-efficacy composition in a dyadic learning environment. We find providing additional tutor prompts encouraging students to participate in discussion may have unexpected negative effects on self-efficacy, especially on students with low self-efficacy scores who have partners with low self-efficacy scores.

Keywords: Intelligent Dialogue Tutors, Collaborative Learning, Self-efficacy, Group Composition, and Discourse During Learning Interactions.

1 Introduction

We know from past research that academic self-efficacy, which is a student's perception of his academic capabilities, is beneficial in individual learning contexts (Zimmerman, 1999) as well as in collaborative learning contexts in which higher group-level self-efficacies are associated with behaviors that support learning (Howley et al, 2011). If the connection is causal, and if we can improve a student's self-efficacy, then the student may reap the associated increased learning and persistence benefits.

In this paper we test this causal connection. Specifically, we leverage conversational agents that have been used successfully as dynamic support for collaborative learning in earlier work (Kumar et al, 2007) as well as theories of discussion moves hypothesized to increase student perception of competence (Michaels, O'Connor, & Resnick, 2008) to provide opportunities for students to take a more authoritative role in a conversation in order to test the effect of that manipulation on self-efficacy. While it is possible to give a student the opportunity to participate more authoritatively, they may choose not to take it or may find themselves unable to take it. The effect of these choices in the face of these uncertainties is an open question. Furthermore, the reactions the student receives from his teammates may have also have an effect on a student's willingness to pursue a discussion opportunity, so it is also necessary to control the group's self-efficacy composition to investigate this issue systematically.

2 Prior Work

The work in this paper revolves around theories from linguistics and social psychology, most notably self-efficacy and a behavioral construct known as authoritativeness.

We focus on Bandura's (1977) theory of self-efficacy and define academic self-efficacy as a student's perceptions of her academic capabilities, interpreted from previous mastery experience, vicarious experience, verbal and social persuasions, and emotional and physiological states. Bandura (1997) also introduces collective efficacy as several individuals' combined perception of the group's capabilities to perform given tasks. Wang & Lin (2007) further investigate this group disposition in collaboration, where they report that individual student self-efficacy predicts the group's collective efficacy, and collective efficacy predicts use of high-level cognitive skills in discussion, as well as group performance.

Along with self-report measures of self-efficacy, we examine behavioral data using a framework for looking at authoritativeness of knowledge presentation. Authoritativeness provides researchers with a lens for examining students' ownership over knowledge through their behavior, rather than through self-report. For our purposes, an "authoritative" statement is a presentation of knowledge without seeking external validation for the knowledge. The Authoritativeness Framework we introduce in this paper is rooted in Martin's Negotiation Framework (Martin, 1992), from the systemic functional linguistics community. A more thorough discussion of our efforts in making this framework replicable is described in Mayfield & Rosé (2011), while our approach for automatically coding chat transcripts with the framework is further explained in Howley et al. (2011).

Our formulation of the Authoritativeness framework is comprised of two dimensions with six and three codes respectively, and is based on principles from the Negotiation framework. For this paper, we will focus on three moves in particular:

- K1, or 'primary knower'. A 'primary knower' move includes a statement of fact, an opinion, or an answer to a factual question, such as 'yes' or 'no'. It only counts as 'primary knower' if it is not presented in such a way as to elicit an evaluation from another participant in the discussion. An example: "This is the end."
- K2, or 'secondary knower'. A 'secondary knower' move includes statements where the speaker is not positioned as authoritative on the current topic, such as asking a question eliciting information, or presenting information in a context where evaluation is the expected response or formulated in such a way as to elicit feedback. An example: "Is this the end?"
- 'o' or 'other' encompassing conversational moves that do not fit within the bounds of the prior two codes described. An example: "So..."

3 Method

The data for this experiment was gathered in order to examine how student academic self-efficacy, learning, and behavior might be affected by targeted prompts from an intelligent dialogue tutor, while also manipulating the partner's self-efficacy in order to look closer at the influence of a peer's self-efficacy. 104 undergraduate students from a thermodynamics class at a private American university participated in the study by attending one of six computer lab sessions, in which time was strictly controlled. Students were given a pre-questionnaire, software training and practice (60

minutes), pretest (10 minutes), the experimental manipulation (40 minutes), and then the posttest and post-questionnaire (15 minutes).

Students were semi-randomly assigned to pairs according to a median split on their course self-efficacy scale in order to achieve homogeneous high self-efficacy pairs, homogenous low self-efficacy pairs, and heterogeneous pairs. After being assigned to pairs, each partner was randomly assigned a goal to design either an eco-friendly power plant or a power- proficient power plant. In all conditions, a tutor agent participated with the students in the chat in order to offer support. The lab session took place in a single computer lab, in which each student had her own computer and partners did not sit next to each other. The experimental manipulation took place during an online collaborative design discussion and consisted of modifying tutor behaviors only. In all other respects, the student experience in all conditions was the same.

Students used Cyclepad (Forbus et al, 1999), a computer software simulator that students use to design simulated power plant designs through a graphical interface. Specifically, students must consider trade-offs between power output and environmental friendliness in designing a Rankine cycle, which is a type of heat engine. The intelligent dialogue tutor was implemented through the Bazaar agent authoring framework (Adamson & Rosé, 2012), allowing the software agent to guide and time discussions, with additional social behaviors. Student dyads collaborated through the ConcertChat software (Stahl, 2006) which enables communication through a chat window and a whiteboard for sharing graphical information.

The experimental manipulation was a 3X3 between-subjects design. Each student pair was randomly assigned to one of nine conditions. The first independent variable manipulated tutor behavior toward high and low self-efficacy students within each pair. The three variations of the tutor behavior were: “target high” (targeting the high self-efficacy student with additional prompts for explanation), “target low” (targeting the low self-efficacy student), and “neutral” (no additional targeted prompts). An example of the tutor’s targeting behavior is “student08, I don't get it - why can't t-max be any higher?” Targeted students received two such prompts, while untargeted students received one, and students in the neutral condition received none. Task-related information such as conceptual hints and timing reminders were kept constant across all three tutoring behaviors leaving the only manipulation to be this targeting behavior. Additionally, all conditions also included context-less mini acknowledgements or encouragements such as “What do you think, student14?” In homogeneous self-efficacy teams, the student with the higher (or lower) self-efficacy score received targeted context questions as shown in Table 1. That is, in a homogenous low self-efficacy pair that was assigned to “target low” the student with the lower self-efficacy score would receive these additional contextual prompts for participation. In the case that both students had identical self-efficacy scale scores, the student target would be selected at random.

The second independent variable contrasted the three team composition types (homogeneous high self-efficacy, homogeneous low self-efficacy, and heterogeneous self-efficacy), where the median split for the original assignment of “high” and “low” was determined from other similar studies, although for analysis we later reassigned the median split value to be that of this study’s cohort. As outcome measures, we

examined academic self-efficacy both before and after the experimental activity. The pre- and post-questionnaires consisted of scales for measuring collective efficacy, mastery-related beliefs (said to predict self-efficacy), and self-efficacy, constructed via the guidance in Bandura (2006). 35 isomorphic multiple choice and short answer questions were used to test analytical and conceptual knowledge on both the pre- and post-tests. And finally, a process analysis examining changes in chat behavior over time was performed.

4 Results

Data was analyzed with respect to factors including: gender, self-efficacy, dialogue tutor targeting, and self-efficacy team composition. Upon reassigning the self-efficacy median split value to match that of this study's cohort median split (i.e., after the completion of the study), our sample consisted of: 14 pairs of homogeneous high self-efficacy, 15 pairs of homogeneous low self-efficacy, and 23 pairs of heterogeneous academic self-efficacy. When analyzing this data with respect to the targeting condition, we look at individual students as "targeted", "untargeted" (if the student's partner received targeted prompts from the intelligent tutor), or "neutral" (if neither partner received targeted behaviors from the dialogue tutor).

When looking at the tutor's effect on post- academic self-efficacy, we found a significant effect of team composition type, $F(2, 97) = 4.91$, $p = 0.0093$. Specifically, students who were in homogeneous low self-efficacy groups ended with a self-efficacy score significantly lower than the homogenous high self-efficacy groups and heterogeneous groups, even with controlling for initial self-efficacy. The interaction term between team composition type and whether the student was targeted, untargeted, or in a neutral condition was not significant, although we did find that targeted students in homogeneous low self-efficacy pairs did significantly worse than all other combinations of tutor-conditions and team composition types (except for students in neutral-tutor homogeneous low condition who were indistinguishable from either group). This result is the opposite of what we expected. These results suggest that prompting low self-efficacy students for further participation may not be the ideal method for improving activity self-efficacy in situations where both partners have low self-efficacies compared to the rest of their classmates. More investigation is necessary to make stronger claims, but future designs should take this into account.

Investigating the relationship between students' collective efficacy and self-efficacies showed that collective efficacy is significantly positively correlated with both students' pre- ($r(104) = 0.54$, $p < 0.0001$) and post- self-efficacy ($r(104) = 0.59$, $p < 0.0001$).

In order to look at what effect the intelligent dialogue tutor has on conversational behavior, we examine authoritativeness that has been automatically coded through the process described in Mayfield & Rosé (2011). As a validation of the automatic coding, we tested our agreement and found an inter-rater reliability with the automated coding scheme of 0.65, which is close to a robust confidence in non-random agreement.

When looking at overall counts of authoritative codes, we find a significant main effect of team composition and targeting condition on K2 moves that is superseded by the interaction term between team composition and the targeting condition on K2 moves, $F(2, 98) = 6.01$, $p = 0.0034$. A post hoc analysis reveals that targeted students in homogeneous low self-efficacy groups had significantly more K2 moves than every other group. Additionally, team composition type is a significant predictor of “other” moves, $F(2,98) = 3.71$, $p = 0.028$ as well as authoritative (the number of primary knower moves over the total number of knowledge authority moves), $F(2, 98) = 3.52$, $p = 0.033$. Students in homogeneous low self-efficacy groups were significantly less authoritative than students in heterogeneous groups, with homogeneous high groups being indistinguishable from either. Students in homogeneous low self-efficacy groups also had significantly more “other” moves than students in homogeneous high groups.

While it may be expected that homogeneous low self-efficacy dyads would perform worse than their heterogeneous counterparts, it is interesting that the manipulation appears to have an impact on student behavior within these low self-efficacy pairs. When we look at K2 moves, we find that it is negatively correlated with pre- to post- self-efficacy residuals $r(104) = -0.255$, $p = 0.009$. This is consistent with our previous results from the self-efficacy analysis.

The tutor targeting conditions, team composition, authoritative, and pre- and post- individual self-efficacies did not have a significant effect on learning, but collective self-efficacy is marginally positively correlated with learning, $F(1, 101) = 3.11$, $p = 0.081$. One might think that perhaps students with lower collective efficacies were at a disadvantage because their group has access to less knowledge; however, there was no significant correlation between pretest scores and collective efficacy, nor between pretest scores and self-efficacy.

5 Conclusions

The majority of our results involved students in homogeneous low self-efficacy dyads. Targeted students in homogeneous low groups ended with self-efficacies lower than predicted compared to all other groups. Students in the low self-efficacy groups had more secondary knower authoritative moves and lower authoritative scores.

These results point to some important caveats for future work in this area. Dyad self-efficacy composition must be taken into consideration, especially since much of our results concern students in homogeneous low self-efficacy pairs. Simply providing opportunities for students to participate more in the discussion may not harm the untargeted students in the pair, but it does not seem to have the desired effect for the targeted student. And so, future work should control for self-efficacy team composition, as well as consider the dynamics within homogeneous low self-efficacy groups.

With regards to authoritative, we found that targeted students in homogeneous low self-efficacy groups had significantly more K2 moves, but we do not yet know if secondary knower moves are beneficial or desirable. Future work should look more specifically at how students propose knowledge for evaluation, and if there is a beneficial or harmful side effect to doing so.

Acknowledgment. This work was supported by the Pittsburgh Science of Learning Center (#SBE 0836012) and a Graduate Training Grant from the Department of Education (#R305B040063).

References

1. Adamson, D., Rosé, C.P.: Coordinating Multi-dimensional Support in Collaborative Conversational Agents. In: Cerri, S.A., Clancey, B. (eds.) ITS 2012. LNCS, vol. 7315, pp. 347–352. Springer, Heidelberg (2012)
2. Bandura, A.: Self-Efficacy: Toward a unifying theory of behavioral change. *Psychological Review* 84, 191–215 (1977)
3. Bandura, A.: *Self-efficacy: The exercise of control*. Freeman, New York (1997)
4. Bandura, A.: Guide for constructing self-efficacy scales. *Self-efficacy Beliefs of Adolescents*, 307–334 (2006)
5. Forbus, K.D., Whalley, P.B., Evrett, J.O., Ureel, L., Brokowski, M., Baher, J., Kuehne, S.E.: CyclePad: An articulate virtual laboratory for engineering thermodynamics. *Artificial Intelligence* 114(1-2), 297–347 (1999)
6. Howley, I., Mayfield, E., Rosé, C.P.: Missing something? Authority in collaborative learning. In: *Proceedings of Computer-Supported Collaborative Learning 2010* (2011)
7. Kumar, R., Rosé, C.P., Wang, Y.C., Joshi, M., Robinson, A.: Tutorial dialogue as adaptive collaborative learning support. In: *Proceedings of Artificial Intelligence in Education* (2007)
8. Martin, J.: *English Text: System and structure*. Benjamins, Amsterdam (1992)
9. Mayfield, E., Rosé, C.P.: Recognizing authority in dialogue with an integer linear programming constrained model. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1018–1026 (2011)
10. Michaels, S., O'Connor, C., Resnick, L.: Deliberative discourse idealized and realized: Accountable Talk in the classroom and in civic life. *Studies in Philosophy and Education* 27(4), 283–297 (2008)
11. Stahl, G.: Analyzing and designing the group cognitive experience. *International Journal of Cooperative Information Systems, IJCIS* (2006)
12. Veel, R.: Language, knowledge, and authority in school mathematics. In: Christie, F. (ed.) *Pedagogy and the Shaping of Consciousness: Linguistics and Social Processes*, Continuum (1999)
13. Wang, S., Lin, S.: The effects of group composition of self-efficacy and collective efficacy on computer-supported collaborative learning. *Computers in Human Behavior* 23(5), 2256–2268 (2007)
14. Zimmerman, B.J.: Self-efficacy: An essential motive to learn. *Contemporary Educational Psychology* 25, 82–91 (1999)

How Do They Do It? Investigating Dialogue Moves within Dialogue Modes in Expert Human Tutoring

Blair Lehman¹, Sidney D'Mello², Whitney Cade¹, and Natalie Person³

¹ Institute for Intelligent Systems, University of Memphis, Memphis, TN 38152
{balehman, wlcade}@memphis.edu

² Departments of Psychology and Computer Science, University of Notre Dame
Notre Dame, IN 46556
sdmello@nd.edu

³ Department of Psychology, Rhodes College, Memphis, TN 38112
person@rhodes.edu

Abstract. Expert human tutors are widely considered to be the gold standard for increasing student learning. While not every student has access to an expert tutor, it is possible to model intelligent tutoring systems after expert tutors. In an effort to achieve this goal, we have analyzed a corpus of 50 hours of one-to-one expert human tutoring sessions. This corpus was coded for speech acts (dialogue moves) and larger pedagogical strategies (dialogue modes). Using mixed-effects modeling, we found that expert tutors differentially used dialogue moves depending on the dialogue mode. Specifically, tutor posed questions, explanations, and motivational statements were predictive of different dialogue modes (e.g., Lecture, Scaffolding).

Keywords: expert tutoring, speech acts, dialogue, ITSs, pedagogical strategies.

1 Introduction

Expert human tutors have widely been considered the gold standard for learning, with Bloom [1] reporting a 2 sigma (or approximately 2 letter grade) learning gain over traditional classroom instruction. Novice human tutors typically only achieve a gain of 0.4 sigma [2], while intelligent tutoring systems (ITSs) produced a 1 sigma learning gain over traditional classrooms [3]. A recent meta-analysis by VanLehn [4], however, reported a more modest effect for expert tutors ($d = .79$). Interestingly, ITSs had a comparable impact on learning ($d = .76$). Despite this more modest learning effect, the pedagogical practices of expert tutors are still effective and there might be advantages associated with building ITSs that model the strategies of expert tutors.

So what exactly are these strategies that make expert tutors so effective? Unfortunately, many of the studies have relied on a small sample ($N = 2$) and the definition of expert status has varied widely (e.g., [5-6]). For example, college professors [5] and graduate students have been used as expert tutors [6]. These two groups may be considered experts, but there is a lack of consensus on what constitutes expertise.

Despite these concerns, past research has been able to identify some of the strategies that make expert tutors effective. Lepper and Woolverton [7] have suggested that there are three key elements to this effectiveness: individualization, immediacy, and interactivity. Through modeling and monitoring student knowledge, tutors have the ability to dynamically adapt to the needs of individual students [8]. Over the course of a tutoring session, tutors can target the specific knowledge deficits and misconceptions, and construct just-in-time interventions for each student.

Although these broad strategies hint at why expert tutors are effective, a more detailed analysis of tutoring strategies is needed. There is, however, a question about the level of analysis. Past studies have analyzed tutoring sessions at the speech act level [9-10], problem-solving episode [11], and the larger pedagogical context [9, 12]. If the goal is to develop an ITS based on the strategies of expert tutors, it is necessary to understand these strategies at both a fine grained level and a more global level.

There have been few studies that investigated the interplay between different levels of tutorial dialogue [9, 13]. One study attempted to extract the larger pedagogical context from the speech acts of students and novice tutors using Hidden Markov Models [9]. In another study that took a more theory-driven approach, Cade et al. [12] created a dialogue mode coding scheme based on both learning theory (e.g., Modeling-Scaffolding-Fading paradigm, [14]) and observations from a corpus of expert tutoring sessions. Although this study yielded important insights into the strategies of expert tutors, it only considered tutorial dialogue at the mode level.

The present study addresses this issue via a multi-level analysis of tutorial dialogue in a corpus of 50 hours of one-to-one expert tutoring sessions. Previously, this corpus was coded at the dialogue move [10] and dialogue mode levels [12]. In the present paper we investigated the distribution of moves within each mode. Specifically, we sought to answer the question: How are dialogue modes manifested in dialogue moves? We will also test whether it is possible to discriminate between dialogue modes using dialogue moves as features.

2 Expert Tutoring Corpus

The corpus consisted of 50 tutoring sessions between ten expert tutors and 40 students [10, 12]. Expert status was defined as licensed to teach at the secondary level, five or more years of tutoring experience, employed by a professional tutoring agency, and recommended by local school personnel. The students were in middle or high school and having difficulty in a science or math course. All tutor-student pairs were working together prior to the study. Each session lasted approximately one hour.

Tutor-student dialogue was coded at two levels: dialogue moves [10] and dialogue modes [12]. Dialogue moves varied in length from one-word acknowledgements to lengthy explanations. Dialogue modes were longer, pedagogically distinct phases that consisted of both tutor and student contributions over multiple dialogue turns. The dialogue move ($kappa = .88$) and dialogue mode coding schemes ($kappa = .87$) were developed and coded independent of each other.

Tutor Dialogue Move Coding Scheme. The 26-item tutor dialogue move coding scheme [10] was divided into groups based on similar functions within the tutoring session: *direct instruction* (example, counterexample, preview, summary, provide correct answer, direct instruction), *question* (new problem, simplified problem, prompt, pump, hint, forced-choice), *feedback* (positive, neutral, negative), *motivational statement* (humor, attribution, general motivation, solidarity), *conversational “Okay”*, and *off-topic*.

Student Dialogue Move Coding Scheme. The 16-item student dialogue move coding scheme was divided into eight groups based on the function of each move: *answer* (correct, partially-correct, vague, error-ridden, none), *question* (common ground, knowledge deficit), *misconception*, *metacomment*, *work-related action* (think aloud, read aloud, work silently), *socially motivated action* (social coordination, acknowledge), *gripe*, and *off-topic*.

Dialogue Mode Coding Scheme. An 8-category coding scheme was used to code dialogue modes [12]. The coding scheme for dialogue modes consisted of *Introduction*, *Lecture*, *Clarification*, *Modeling*, *Scaffolding*, *Fading*, *Off-topic*, and *Conclusion*. *Lecture*, for example, involved the tutor explicitly delivering information to the student with fewer student responses, while *Scaffolding* involved collaborative problem solving between the tutor and student.

The present paper focused on *Lecture*, *Clarification*, *Modeling*, *Scaffolding*, and *Fading* because these modes have predominantly pedagogical functions, whereas the remaining dialogue modes (*Introduction*, *Conclusion*, *Off-topic*) involved social and rapport building dialogue [15].

3 Results and Discussion

3.1 Dialogue Moves Predicting Dialogue Modes

Mixed-effects logistic regressions [16] were used to investigate whether dialogue move groups (e.g., feedback) and individual dialogue moves (e.g., positive feedback) could predict the presence (1) or absence (0) of each dialogue mode. Mixed-effects modeling is the recommended analysis for the present data set because of the repeated and nested structures in the data (e.g., moves embedded within modes). There were a total of 47,318 observations (dialogue moves) in the corpus.

In each model, the random effects were the tutor, student, domain (math or science), and order of the dialogue move within the tutoring session. The fixed effects were either move groups or individual moves. Separate models were constructed for tutor and student moves to isolate their independent contributions. For each mode five models were tested: random effects only, move groups (tutor or student), and individual moves (tutor or student). The lme4 package in R [17] was used to perform the requisite computation.

For all modes, models with fixed effects fit the data significantly better than the random effects only models ($p < .001$). Table 1 shows the pattern of significant ($p < .05$) predictors, using move groups as fixed effects. However, instances in which individual moves differed from move groups are discussed below.

Table 1. Dialogue move group patterns for dialogue modes

	Lecture	Clarify	Model	Scaffold	Fade
Tutor Dialogue Move Groups					
Direct Instruction	+	+	+	-*	-
Question	-			+	-
Feedback	-		-	+	+
Motivational Statement			+	-	
Comprehension Gauging Question	+	+	+	-	-
Conversational OK	+		-*	-	
Off-Topic	-	-	-	-	
Student Dialogue Move Groups					
Answer Quality	-	-	-	+	+
Misconception				+	
Metacomment	+			-	
Question	-*			+	
Work-Related Action	-		-	+	+
Socially Motivated Action	+	+	+	-	-
Gripe	-	+	-		
Off-Topic	-	-	-	-	

+ = positive predictor; - = negative predictor; blank = non-significant predictor; * = $p < 0.1$

Overall, a contrast between a transmission model of learning [18] and a more collaborative interaction was revealed. Specifically, in *Lecture*, *Clarification*, and *Modeling* the tutor provided the majority of information and requested little information from the student. A different pattern emerged for *Scaffolding* and *Fading*. Tutors supplied less information and instead asked questions and provided feedback. Similarly, students asked and answered questions during *Scaffolding*. This profile of *Scaffolding* suggests that students were engaged in problem solving with the guidance of the tutor.

During *Fading*, tutor transmission of information became almost non-existent. Although tutor questions were a negative predictor of *Fading*, posing new problems was a significant positive predictor. For student moves, *Fading* was predicted by answers and work related actions. This suggests that during *Fading*, tutors took on a passive role and allowed students to apply their knowledge. Overall, these findings suggest that there is a connection between these two levels of tutorial dialogue.

3.2 Discriminating Between Dialogue Modes

Next, we attempted to discriminate between *Lecture*, *Clarification*, *Modeling*, and *Scaffolding* with dialogue move groups. *Clarification* and *Modeling* were collapsed into one category due to similar pedagogical functions (referred to as *Modeling*). To account for unequal distributions, we downsampled to create more equal mode distributions (*Lecture* = .352; *Modeling* = .322; *Scaffolding* = .325).

Twelve models were tested using dialogue move groups and tutoring session context to discriminate between dialogue modes. Each model was trained and evaluated using discriminant function analyses. Classification accuracy (correct) and kappa scores (see Table 2) were computed using leave-one-out cross validation.

Table 2. Classification results

Context	Dialogue Move Groups					
	Tutor		Student		Tutor + Student	
	Correct	Kappa	Correct	Kappa	Correct	Kappa
None	39.1%	.090	39.5%	.072	43.6%	.147
Move Order	44.5%	.171	43.9%	.154	46.7%	.201
Domain	63.6%	.450	63.2%	.444	63.6%	.451
Domain + Move Order	66.7%	.497	66.7%	.497	67.0%	.503

The results indicated that models combining tutor and student move groups ($kappa = .147$) were the most effective at classifying modes (Tutor + Student). When the context of the tutoring session was added, classification accuracy improved ($kappa = .503$). In particular, inclusion of the tutoring session domain improved performance the most. These findings suggest that the tutoring session context, particularly the domain, is an important element to consider when generating tutorial dialogue.

4 Conclusion

There have been a number of studies investigating the strategies of expert tutors [5-8], but tutorial dialogue has rarely been analyzed at different levels within a single study. In the present paper we examined tutorial dialogue at two levels. While the patterns found were expected based on theories of learning and pedagogy (e.g., [7, 14]), it is important to find evidence that expert tutors actually use these practices. This paper confirmed that some of these ‘ideal tutorial strategies’ (e.g., Modeling-Scaffolding-Fading) are indeed implemented by more accomplished human tutors.

It is important to briefly consider the implications of our findings for ITSs. ITSs already manage tutorial dialogue at both a local and global level [19] and are effective in achieving learning gains at rates comparable to human tutors [4]. However, the dialogue of most ITSs is informed by learning theories or the practices of novice human tutors, not *expert* human tutors. The present findings can inform ITS dialogues in several important ways. First, expert tutors seem to use a balance of information transmission and collaborative problem solving. Second, the patterns of moves can be used to detect when transitions between modes should occur. Although the present analyses do not address transitions between modes, this has been previously analyzed [12]. Finally, the content of the tutoring session (i.e., domain) seems to have an impact on tutorial dialogue. Future research will need to further examine how strategies differ and under what circumstances different strategies should be deployed to further improve learning.

Acknowledgement. The research reported here was supported by the Institute of Education Sciences (R305A080594) and the U. S. Office of Naval Research (N00014-05-1-0241). The opinions expressed are those of the authors and do not represent views of the IES, the U.S. DoE, the ONR, or DoD.

References

1. Bloom, B.: The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher* 13(6), 4–16 (1984)
2. Cohen, P., Kulik, J., Kulik, C.: Educational outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal* 19, 237–248 (1982)
3. Corbett, A., Anderson, J., Graesser, A., Koedinger, K., VanLehn, K.: Third generation computer tutors: Learn from or ignore human tutors? In: *Proceedings of the 1999 Conference of Computer-Human Interaction*, pp. 85–86. ACM Press, New York (1999)
4. VanLehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist* 46(4), 197–221 (2011)
5. Evens, M., Spitkovsky, J., Boyle, P., Michael, J., Rovick, A.: Synthesizing tutorial dialogues. In: *Proceedings of the 15th Annual Conference of the Cognitive Science Society*, pp. 137–142. Lawrence Erlbaum Associates, Hillsdale (1993)
6. Fox, B.: Cognitive and interactional aspects of correction in tutoring. In: Goodyear, P. (ed.) *Teaching Knowledge and Intelligent Tutoring*, pp. 149–172. Ablex, Norwood (1991)
7. Lepper, M., Woolverton, M.: The wisdom of practice: Lessons learned from the study of highly effective tutors. In: Aronson, J. (ed.) *Improving Academic Achievement: Impact of Psychological Factors on Education*, pp. 135–158. Academic Press, Orlando (2002)
8. Chi, M., Siler, S., Jeong, H.: Can tutors monitor students' understanding accurately? *Cognition and Instruction* 22(3), 363–387 (2004)
9. Boyer, K., Phillips, R., Ingram, A., Ha, E., Wallis, M., Vouk, M., et al.: Investigating the relationship between dialogue structure and tutoring effectiveness: A Hidden Markov Modeling approach. *International Journal of Artificial Intelligence in Education* (in press)
10. Person, N., Lehman, B., Ozgun, R.: Pedagogical and motivational dialogue moves used by expert tutors. Paper Presented at the 17th Annual Meeting of the Society for Text and Discourse, Glasgow, Scotland (2007)
11. VanLehn, K., Siler, S., Murray, C., Yamauchi, T., Baggett, W.: Why do only some events cause learning during human tutoring? *Cognition and Instruction* 21(3), 209–249 (2003)
12. Cade, W.L., Copeland, J.L., Person, N.K., D'Mello, S.K.: Dialogue Modes in Expert Tutoring. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) *ITS 2008. LNCS*, vol. 5091, pp. 470–479. Springer, Heidelberg (2008)
13. D'Mello, S., Olney, A., Person, N.: Mining collaborative patterns in tutorial dialogues. *Journal of Educational Data Mining* 2(1), 1–37 (2010)
14. Rogoff, B., Gardner, W.: Adult guidance of cognitive development. In: Rogoff, B., Lave, J. (eds.) *Everyday Cognition: Its Development in Social Context*, pp. 96–116. Harvard University Press, Cambridge (1984)
15. Cade, W., Lehman, B., Olney, A.: An exploration of off topic conversation. In: Burstein, J., Harper, M., Penn, G. (eds.) *NAACL-HLT 2010 Conference*, pp. 669–672. Association for Computational Linguistics (2010)
16. Pinheiro, J., Bates, D.: *Mixed-effects models in S and S-PLUS*. Springer, New York (2000)
17. Bates, D., Maechler, M.: *lme4: Linear mixed-effects models using Eigen and Eigenpack* [computer software] (2010), <http://cran.r-project.org/>
18. Dillon, T.: *Questioning and teaching: A manual of practice*. Teachers College Press, New York (1988)
19. VanLehn, K.: The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education* 16(3), 227–265 (2006)

Building a Conversational SimStudent

Ryan Carlson, Victoria Keiser, Noboru Matsuda, Kenneth R. Koedinger,
and Carolyn Penstein Rosé

School of Computer Science
Carnegie Mellon University

{rcarlson,keiser,noboru.matsuda,cprose}@cs.cmu.edu, koedinger@cmu.edu

Abstract. SimStudent, an intelligent-agent architecture that generates a cognitive model from worked-out examples, currently interacts with human subjects only in a limited capacity. In our application, SimStudent attempts to solve algebra equations, querying the user about the correctness of each step as it solves, and the user explains the step in natural language. Based on that input, SimStudent can choose to ask further questions that prompt the user to think harder about the problem in an attempt to elicit deeper responses. We show how text classification techniques can be used to train models that can distinguish between different categories of student feedback to SimStudent, and how this enables interaction with SimStudent in a pilot study.

1 Introduction

Teachable agents take advantage of the learning-by-teaching paradigm, allowing the user to take on the role of tutor while the agent plays the tutee role. In such setups, the tutor can gain experience listening and responding to displayed thought processes from the tutee. Additionally, the tutor-in-training is not in danger of harming the tutee's learning. These advantages make teachable agents useful tools which can be integrated into learning environments ranging from video games to homework help sessions [1].

Incorporating natural language input in a chat environment with intelligent tutors has had success in the past. Conversational agents have been used in qualitative physics tutoring [7], and tutorial dialogue agents have reported some success in replicating knowledge construction dialogues found in human-human tutoring interactions [4]. The empty box awaiting input forces students to explicitly express their ideas and to identify areas with which they have difficulty. Moreover, it greatly increases how expressive students can be in their dialogues with the agent. This open-ended setup has also been leveraged to reinforce student reflection during thermodynamics tutoring [8]. As we will see, while natural language input allows for thoughtful, complete responses to questions posed by the teachable agent, it also opens the door for off-task behavior.

Once we allow natural language interaction between user and agent, we immediately find a need for a mechanism to process and interpret the user's input. In order to respond reliably and realistically to user input, one technique is to classify the input into one of several discrete categories and react accordingly.

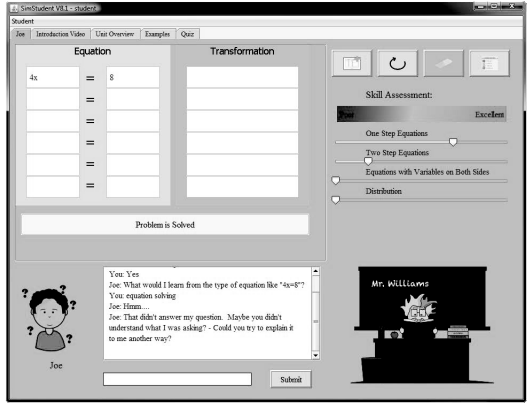


Fig. 1. SimStudent interface and sample input from a problem solving session

Machine learning can be a powerful tool to generate a model that automatically classifies the data based on a coding scheme. While the results can be difficult to interpret, machine learning offers the ability to sift through massive amounts of data and produces a classifier with minimal human intervention. Without a machine learning step to automatically model the discussion space, an agent is often limited to a much more static conversation structure.

2 System Building

The application of this research revolves around an architecture called SimStudent, which supports a computational model of human learning [5]. TuTalk [2], a dialogue-building system, was integrated into the SimStudent architecture to provide richer interaction with the student. SimStudent can ask users questions, allowing them to show off how well they understand the material, and can follow up based on the user responses. For example, a response like “because you need to combine like terms” may be sufficient, while “because I said so” will probably warrant further examination and prompting. A small set of codes have been developed to categorize the user responses and determine SimStudent’s next statement or question. Our task is to build a model to predict which of these codes a student’s statement falls into, and thus determine the appropriate next move for SimStudent. Note that we refer to *SimStudent* and *agent* interchangeably; similarly with *student* and *user*.

2.1 Interaction with SimStudent

As discussed, SimStudent is an algebra-solving teachable agent, pictured in Figure 1. The agent is featured as an avatar in the lower left corner and starts off with some basic knowledge about how to solve one-step equations. Students

1. *[Student instructs SimStudent to solve the problem $x + 3 = 2x$ by writing the left and right hand sides of the equation in the equation box.]*
2. **SimStudent (SS):** What will doing the problem $x + 3 = 2x$ help me learn?
3. **Student (S):** Your gonna need to combine like terms in this problem
4. **SS:** Okay!
5. *[SimStudent puts “subtract 3” in the top cell of the Transformation column.]*
6. **SS:** I entered “subtract 3” for the transformation. Do you think that would be a good move?
7. *[Student clicks the No button (which appears in the interface when appropriate).]*
8. **SS:** But I put “subtract 9” for $y + 9 = 6$. Why doesn’t subtract 3 work now?
9. **S:** Youre subtracting the wrong thing!!!!
10. **SS:** Hmm, I still don’t understand – could you explain this to me in more detail?
11. **S:** subtract the variable x to get the variable on one side

Fig. 2. A sample conversation between a student and SimStudent. Italicized lines indicate non-language actions. Creative spelling is typical of the data we are processing.

choose problems for it to attempt and then help the agent through the process. SimStudent solves the equation step-by-step, asking the student to verify each step’s correctness. SimStudent will sometimes follow up with a question about the current problem, at which point the student responds in natural language. Our model evaluates this input and classifies it into one of a small set of codes. Based on that evaluation, SimStudent responds in order to foster a stronger dialogue and force the student to think about the relevant concepts for the problem at hand. Thus, the set of codes we have chosen are in a one-to-one correspondence with the next SimStudent action. This differs from previous work in that we are essentially classifying SimStudent’s responses based on the student input, rather than forming a representation of what the student has said.

In order to better illustrate the type of conversation the student and agent are engaging in, we can look at a sample discussion, shown in Figure 2. The interaction begins with the student setting up an equation for the agent to solve. SimStudent asks a question about the importance of the student’s choice, and she responds with a complete response, so SimStudent moves on. The agent makes a mistake at step 6 and when it comes time for the student to explain the rationale (step 9) she indicates that SimStudent is subtracting the wrong term. But since the goal is to prompt the student towards fuller, more complete responses, SimStudent asks the student to better explain herself (step 10) and she does in the following step.

2.2 Coding Scheme

Our coding scheme uses seven different codes, broken up into three broader categories: ON TARGET & HELPFUL, RESPONSIBILITY ORIENTED, and PUNT. The data were coded by two independent coders (cohen’s kappa = 0.75). The first category deals with student responses that are relevant to the problem at hand. They must reference facts about the equation. If the student uses concepts in her explanation (e.g. “because you should have combined like terms”), then the

response is coded as *deep*. These responses are the only type which requires no follow-up from SimStudent. If the response is not concept-oriented but is procedural (e.g. “add 6”) then the appropriate code is *action*. While a hypothetical human tutee might know what to do after this type of hint, she would not know why, and thus SimStudent should prompt for a more complete response. If the statement is relevant to the problem but is either vague or is missing some crucial information (e.g. “add”, “pay attention to the negative”), it should be assigned the *insufficient* code.

The second category of responses is concerned with assigning responsibility to either the student or the teachable agent. If the student accepts responsibility for an error that was made, it is labeled as a *mistake*. Additionally, the same code is used if the student aligns herself with SimStudent (e.g. “my bad”, “we made a mistake”). If the student deflects responsibility or does not align herself with the agent (e.g. “you’re wrong”, “it was wrong”), the code should be *blame*.

The final category consists of responses unrelated to the content of the questions asked by SimStudent. If the student admits that she simply doesn’t know the answer we use the code *doesn’t know* (e.g. “idk”, “i have no idea get me out of here!”). Finally, the catchall code is *unhelpful* which consists of a large, heterogeneous set of responses that range from the bogus to the almost relevant (e.g. “shut up and go away”, “look at the equation”). They are met with SimStudent asking the user to explain themselves in another way.

3 Exploring the Data

Our training set came from a study of 141 students interacting with SimStudent from December 2-6, 2010. The original, raw data can be found through DataShop [3]. The data was collected from 7-10 grade-level students using SimStudent at school. The school volunteered its students and the students were not compensated. Contrast this with a pilot study run in December 2011 that we use as our test set. Nine students, grades 6-11, were recruited using local fliers. This was thus an opt-in program and the students were compensated for their time.

Before training a model, some initial preprocessing steps were necessary. This involved basic spell-checking that ensured instances of the same word (e.g. “because”, “becuase”, “beecause”) were treated as the same. Additionally, all equation-specific details were removed from the user input and were replaced with consistent tags. Specifically, every equation was replaced with EQN, and so on for expressions, numbers, and variables. This allowed for features to encode patterns involving, say, a number, without relying on that number’s value.

To train our model we used SIDE [6], a text mining tool kit. For each transaction we extracted unigrams (i.e., individual words), bigrams (i.e., pairs of words that occur contiguously in the text), and punctuation from the user input. Furthermore, all the words are stemmed to remove suffixes that indicate tense, number, etc. Once all of the features were extracted, we trained our model using the LibLinear package of Weka [9] with L2-regularized logistic regression, which aids in avoiding overfitting during training.

Two challenges are introduced in the training data. First, since half of the entries are *unhelpful*, we expect the model to bias towards that class value when strong evidence does not favor a different class. Additionally, the *unhelpful* class is poorly defined, making it difficult to find strong predictive features. Thus, the model will generally classify instances as *unhelpful* unless they squarely match up with other classes. If the *unhelpful* class is no longer a large majority class in the test set as is the case with the second study, we expect to run into problems.

The pattern of results turns out to be exactly as we expect. Using 10-fold cross-validation, which means averaging over 10 iterations of training on 90 percent of the data and testing on the other 10 percent, the model achieves an error rate of 17.9% (0.73 kappa). However, when we train on all of that data and test on the test set from the smaller study, the error rate increases to 50.7% (0.34 kappa).

Examining the features which most prominently contribute to error in cross-validation can help to make sense of these models and explain their performance. The single largest contributor to error results from incorrectly distinguishing between *unhelpful* and *insufficient* responses (5.5 percent of the error). As an example, the unigram *problem* is often used in generic and irrelevant answers like “you got a problem like it wrong before.” The feature is in 17 percent of the instances where the model correctly classifies the instance as *unhelpful*. While this feature does have some predictive power, it can also be deceptive as it is present in 22 percent of the transactions that confuse *unhelpful* and *insufficient*.

We present a methodology below that is simplistic, but which dramatically improves performance in practice. As we have noted, the *unhelpful* class is ill-defined and occurs often in the training set but only rarely in the test set. Our technique is to eliminate this class from the training set and train a new model, evaluating it on the test set. Training a classifier on the other six codes and then checking its performance on the test set guarantees we classify every unhelpful transaction incorrectly. Since the test set contains considerably fewer transactions coded as *unhelpful*, failing at these codes does not make a large impact. Eliminating this crucial code from the training set increases the overall ability of the model to accurately classify input in the test set. Cross-validation on this altered training set produces an error rate of 21.5 (0.71 kappa). While these results are slightly worse when comparing cross-validation across the two models, we are most interested in the performance on the test set. And here our technique proves valuable. The model’s error rate drops on the test set from 50.7% all the way to 30.3% (0.52 kappa), which is much better than the original model’s results.

4 Discussion

The work we have presented here does not benefit from a mere quirk of this data. In any intelligent tutoring environment where the user is interacting with an agent in an open-ended fashion (be it in text, speech, or another modality), there is going to be the possibility of off-task or otherwise unhelpful input. Moreover,

if these agents are being deployed into the classroom, educators will often know how motivated, engaged, or otherwise likely to provide on-topic responses the students are. Under these circumstances, our work can provide a simple first step to enhance the interactive experience of the tutoring session.

We have shown that a model generated using data where approximately half of the responses were deemed unhelpful has an error rate of under 20 percent so long as the data is similarly distributed. We drew this conclusion from using cross-validation with very little feature engineering. In a practical application, this model would be used with the lower achievers or the less motivated. SimStudent will often need to ask the user to rephrase or might direct her to the unit overview. On the other hand, we can say that more motivated, higher achievers are much less likely to move off task and will pattern much closer to the volunteers in the pilot study. Given this information, we can raise classification accuracy on such a set of new data dramatically. In our tests, we saw the error rate drop significantly with a simultaneous rise in kappa statistic. These results show promise that a very simple and computationally inexpensive methodology can greatly improve the interaction experience.

Acknowledgements. This research was supported by National Science Foundation Award DRL-0910176 and the Institute of Education Sciences, U.S. Department of Education (Grant R305A090519 to Carnegie Mellon University). The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education. This work is also supported in part by the Pittsburgh Science of Learning Center, which is funded by the National Science Foundation Award No. SBE-0836012. Additionally, this work is supported by Graduate Training Grant awarded to Carnegie Mellon University by the Department of Education (R305B090023).

References

1. Blair, K., Schwartz, D., Biswas, G., Leelawong, K.: Pedagogical agents for learning by teaching: Teachable Agents. *Educational Technology* 47(1), 56 (2007)
2. Jordan, P., Ringenber, M., Hall, B.: Rapidly Developing Dialogue Systems that Support Learning Studies. In: *Workshop on Teaching with Robots, Agents, and NLP* (2006)
3. Koedinger, K.R., Baker, R.S.J., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J.: A data repository for the EDM community: The PSLC DataShop. In: *Handbook of Educational Data Mining*, pp. 1–21. CRC Press, Boca Raton (2010)
4. Kumar, R., Rosé, C.P., Aleven, V., Iglesias, A., Robinson, A.: Evaluating the Effectiveness of Tutorial Dialogue Instruction in an Exploratory Learning Context. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006*. LNCS, vol. 4053, pp. 666–674. Springer, Heidelberg (2006)
5. Matsuda, N., Cohen, W.W., Koedinger, K.R., Keiser, V., Raizada, R., Yarzebinski, E., Watson, S.P., Stylianides, G.: Studying the Effect of Tutor Learning using a Teachable Agent that asks the Student Tutor for Explanations. In: *Proceedings of the International Conference on Digital Game and Intelligent Toy Enhanced Learning*, Digitel (2012)

6. Mayfield, E., Rosé, C.P.: An interactive tool for supporting error analysis for text mining. In: Proceedings of the NAACL HLT 2010 Demonstration Session, pp. 25–28. Association for Computational Linguistics (2010)
7. Rosé, C.P., Jordan, P., Ringenberg, M., Siler, S., VanLehn, K., Weinstein, A.: Interactive conceptual tutoring in Atlas-Andes. In: Proceedings of AI in Education 2001 Conference, pp. 151–153 (2001)
8. Rosé, C.P., Torrey, C., Alevan, V., Wu, A., Forbus, K.: CycleTalk: Toward a Dialogue Agent That Guides Design with an Articulate Simulator. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) ITS 2004. LNCS, vol. 3220, pp. 401–411. Springer, Heidelberg (2004)
9. Witten, I.H., Frank, E., Hall, M.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Elsevier (2011)

Predicting Learner's Project Performance with Dialogue Features in Online Q&A Discussions

Jaebong Yoo and Jihie Kim

Information Sciences Institute, University of Southern California
{jaebong, jihie}@isi.edu

Abstract. Although many college courses adopt online tools such as Q&A online discussions, there is no easy way to evaluate their impact on learning. In this paper, we investigate a predictive relation between characteristics of discussion contributions and student performance. For the modeling dynamics of conversational dialogue, speech acts (Q&A dialog roles that participants play) and emotional features covered by LIWC (Linguistic Inquiry and Word Count) were used. These dialogue information is used for correlation and regression analyses for predicting the performance of learners (173 student groups). Our current results indicate that the number of answers provided to others, the degree of positive emotion expressions, and how early students exchange information before the deadline correlate with project grades. This finding confirms the argument that in assessing student online activities, we need to capture how they interact, not just what they produce.

Keywords: Online discussions, group projects, speech act classifiers.

1 Introduction

Online asynchronous discussions (OADs) have become increasingly popular tools for university-level engineering courses in supporting students' communication and collaboration. As recent studies have pointed to OADs as a promising strategy for collaboration and higher-order thinking, researchers have also sought to understand the predictive relationship between discussion participation and learning [1].

There exists a large body of research on assessing OADs using quantitative or qualitative methods. Quantitative approaches use some statistical information such as message frequencies (the number of initials and replies, the number of messages read, thread lengths, and response time from the previous messages) and correlate them with course grades [2]. It is widely acknowledged that this approach provides at best a rough analysis of online activities on a surface level. Qualitative approaches such as content analysis have gained considerable attention in the past decade [3]. Such approaches reveal latent semantic information in the transcript from the discussion boards for knowledge building or critical thinking. However, such results have not been fully used for explaining or predicting student performance.

In this study, we extend the scope of existing qualitative methods by employing a relatively large corpus of student discussion contributions and relating dialogue features that capture the dynamics of Q&A conversation to student performance. For

modeling dynamics of conversational dialogue, Speech Acts (SAs) [4] and Linguistic Inquiry Word Count (LIWC) [5] were used. The SAs define roles that individual messages play within the discussion, such as *Sink* (information seeking act) and *Source* (information providing act), and can provide hints on how the student is contributing to the class. The LIWC has been used in capturing emotional and psychological features, and predicting student knowledge [6]. For effective data processing, we apply machine learned classifier (Speech Acts) and automatic text processing (LIWC).

2 Methods

2.1 Participants

Our work takes place in the context of an undergraduate Operating Systems course discussion board in the Computer Sciences department at the University of Southern California. The course is held every semester and taught by the same instructor for the past 15 semesters. We studied recent eight semesters from the same course. Among the 240 groups enrolled, 173 groups (370 students) were active (posted more than 3 messages). In Table 1, ‘group’ participation means participation by at least one member of the group. Our analysis focuses on the active groups and treats each group as a unit. All the group members receive the same grade that takes 40% of the final grade.

Table 1. Forum Participation of Individual and Group by Semester ($N = 240$)

	Year	2006	2006	2007	2007	2008	2009	2010	2010
	Semester	Spring	Fall	Spring	Fall	Fall	Spring	Spring	Fall
PR	Individual	0.62 36/58	0.55 46/83	0.40 21/53	0.65 77/119	0.50 58/115	0.48 26/54	0.49 49/99	0.41 57/140
	Group	0.83 24/29	0.78 32/41	0.67 14/21	0.78 39/50	0.73 37/51	0.80 20/25	0.73 32/44	0.55 42/77

* PR (Participation Ratio) = # of students participated / # of students enrolled in the discussion forum

2.2 Procedures

Data Collection. The class used phpBB (2006 ~ 2009) and then Moodle (2010 ~ current). The 8 semesters’ discussion data have been collected from the discussion boards.

Data Preprocessing. In order to generate meaningful features, appropriate cleaning and normalization have to be performed. For SA classifiers, our data preprocessing step fixes common typos and abbreviation, converts contracted forms to their full forms, and transforms informal words to formal words. For example, “we’re” was modified to “we are” in M1 and “dont” should be converted to “do not” in M2 as shown in Figure 1. As another example, “ya”, “yea”, and “yup” are all substituted by “yes.” For the LIWC measures that rely on the number of words, we remove quotes (repetition of previous message content inside the current post) using a text comparison tool called “google-diff-match-patch.” Also, a large block of code that appears inside the text cannot be recognized by LIWC, which expects normal English text as input. For detecting programming content, we developed a set of regular expressions

to identify variable assignments, function definitions, function calls, comments and etc. The programming content in M1 is replaced with a code tag as shown in Figure 1.

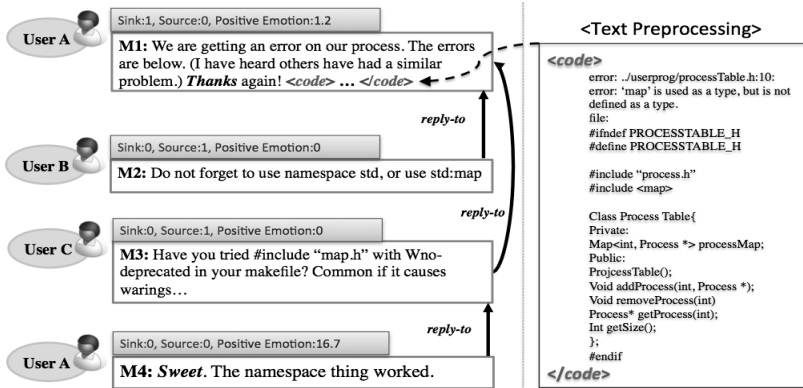


Fig. 1. An Example discussion thread:

Variable Generation. From the above data pre-processing steps, the input data for SA classifiers and LIWC are produced. The results from SA classifiers and LIWC are as follows: Sink/Source classification and 80 LIWC metrics. Among 80 metrics, we selected 20 metrics that are relevant to student project forums. The variables used for our analysis are summarized below.

Predicting Learner’s Performance. After extracting data for selected variables, correlation and regression analyses were performed against project grades.

2.3 Discussion Analysis Variables

The categories of variables below were used for the study are:

- Traditional quantitative metrics (3 variables): the number of total/initial/replies
- Qualitative and quantitative metrics in discussion dialogue (4 variables):
 - Sink: Sink as a message which requests information from others
 - Source: Source as a message which provides information to others
 - APTTPD (Average duration between the Posting Time and Project Deadline)
 - Degree of technical term use: % of operating systems technical terms in text
- Qualitative variables from LIWC (20 variables): word count, words/sentences, words>6 letters, tense (past, present, future), negations, swear words, positive/negative emotions, insight, causation, discrepancy, certainty, tentative, inhibition, see, time, achievement, assent. The further details of these variables, see [5].

3 Speech Act Classifier

The discussion threads can be viewed as a special case of human conversation, and we adopted the theory of speech acts (SAs) [7] to classify patterns of student’s interaction. As SAs are important variables for characterizing student discourse, in order

to improve the classification accuracy, we investigated several ways of optimizing features and eliminating irrelevant or redundant information [8]. The SA classifiers were built through three steps: feature generation, feature selection, and classification.

- Feature generation: we used standard n-grams features as well as metadata of messages such as their absolute/relative positions, author change information, and their previous metadata information. There are 30,044 features in the training corpus.
- Feature selection: we applied four filter-based feature selection algorithms (i.e. Chi-square, InfoGain, GainRatio, and ReliefF) to reduce the dimensionality of the input data because filters can handle a large number of features efficiently.
- Feature classification: Finally, the SA classifier is built with the selected optimal features. The 2006 spring and 2007 fall semester discussion data (898 messages) were randomly divided into two datasets: training dataset (628 messages) and test dataset (270 messages). All the threads were annotated by hand beforehand: the Kappa scores of Sink and Source were 92.92% and 95.95% respectively.

We optimized the features as we crease the number of selected features from 100 to 4000 with an increment of 100. Note that SVM is less sensitive to the change of the number of selected features so we chose it for developing the final SA classifiers. The accuracies of the SA classifier for Sink and Source have improved from 87.1% to 93.2% and from 86.1% to 90.1% respectively by optimizing the features.

4 Results

4.1 Correlation Analysis

Table 2 shows that 5 out of 27 independent variables (described in Section 2.2) are significantly related to the project grade. The correlation analysis revealed that Sink did not have a significant coefficient in comparison to Source. We predicted that low performers ask more questions due to confusion or misunderstanding. However, students who tend to answer others' questions may have understood the topic better, and achieve better grades. Surprisingly, the simple statistical information such as Total and Reply was related to the learner's performance. Among LIWC variables, only Positive Emotion was positively correlated with the project grade.

4.2 Multiple Stepwise Regression Analysis

In order to identify which variables explain the variance of our model, multiple stepwise regression analysis was conducted with normalized project grade as the dependent variable. An analysis of variance test suggests that the regression model is significant, $F(3, 169) = 17.08, p < 0.001$, with 32% variance in student's performance being explained by three predictors. The result of the multiple stepwise regression is summarized in Table 3. One pair of Total and Reply was automatically dropped off from the analysis because Source includes them conceptually. Source has the largest regression coefficient, $B = .47 (p < .001)$. It implies the more information students provide to other students, the better grade they achieve. APTTPD has the second largest regression coefficient, $B = .20 (p < .001)$. This is consistent with recent findings from other researchers [9] that high procrastinators tend to get lower grades. Among

the LIWC variables, the only significant variable was Positive Emotion. This suggests that LIWC that has been used mainly for behavioral or social psychology research may be not the best tool for analyzing our technical Q&A discussion data.

Table 2. Correlation between Variables and Grade among Learners

Grade	Category	Variables	Correlation	
	Traditional Model	Quantitative metric	Total	.17*
		Quantitative metric	Initial	.13
		Quantitative metric	Reply	.17*
	Our Model	Speech act	Sink	.03
		Speech act	Source	.22**
		Procrastination	APTTD	.21**
		Technical terms	Technical terms	.08
	LIWC	Linguistic	Word count	-.08
			Words/sentence	-.09
			Words>6 letters	-.10
			Past tense	-.09
			Present tense	.10
			Future tense	.08
			Negations	-.11
			Swear words	-.09
		Psychological	Positive Emotion	.16**
			Negative Emotion	.10
			Insight	-.01
			Causation	.05
			Discrepancy	.16
			Tentative	.10
Certainty			.05	
Inhibition			-.02	
See			.06	
Time			.01	
Personal concerns	Achievement	-.02		
Spoken category	Assent	.02		

N = 173; **p* < .05; ***p* < .01

Table 3. Summary of Multiple Stepwise Regression Analysis

Variables in the Equation			
Variable	B	Std. Error	Beta
Source	.47	.06	.48***
APTTD	.20	.07	.20**
Positive Emotion	.02	.01	.13*
Note: R ² =.32 <i>N</i> = 173; * <i>p</i> < .05; ** <i>p</i> < .01; *** <i>p</i> < .001			

5 Summary and Future Work

We have investigated how quantitative and qualitative features of student online Q&A discussions are related to student project performance. For modeling dynamics of

conversational dialogue, speech acts and emotional and psychological features were used. In order to generate meaningful features using machine classifiers and automatic text processing tools, the raw discussion data was processed with various noise reduction and normalization steps. As SAs are an important variable for characterizing student discourse, to improve SA classifiers, we identified an optimal feature set for the classification. The final Sink/Source classifier accuracies reached 93.2% and 90.1% respectively.

The current results indicate that qualitative dialogue features such as the degree of information provided to others and how early students discuss their problems before the deadline are important factors in explaining the project grade. Other quantitative characteristics or local textual variables do not seem to contribute much. We plan to perform more comprehensive analysis with diverse conversational or collaborative discussion features including number of conversation partners, degree of interactions with teachers, etc. as well as additional textual features.

Acknowledgement. This work is supported by the National Science Foundation, REESE program (award #1008747).

References

1. Koschmann, T.E.: CSCL: Theory and practice of an emerging paradigm 1996. Lawrence Erlbaum Associates, Inc. (1996)
2. Palmer, S., Holt, D., Bray, S.: Does the discussion help? The impact of a formally assessed online discussion on final student results. *British Journal of Educational Technology* 39(5), 837–858 (2008)
3. De Wever, B., Schellens, T., Valcke, M., Van Keer, H.: Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. *Computers & Education* 46(1), 6–28 (2006)
4. Ravi, S., Kim, J.: Profiling student interactions in threaded discussions with speech act classifiers. IOS Press (2007)
5. Pennebaker, J.W., Francis, M.E., Booth, R.J.: Linguistic inquiry and word count: LIWC 2001. Lawrence Erlbaum Associates, Mahway (2001)
6. Williams, C., D'Mello, S.: Predicting Student Knowledge Level from Domain-Independent Function and Content Words. In: Alevan, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6095, pp. 62–71. Springer, Heidelberg (2010)
7. Searle, J.R., Bierwisch, M.: Speech act theory and pragmatics, vol. 10. Springer (1980)
8. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. Morgan Kaufmann Publishers, Inc. (1997)
9. Michinov, N., et al.: Procrastination, participation, and performance in online learning environments. *Computers & Education* 56(1), 243–252 (2011)

Interventions to Regulate Confusion during Learning

Blair Lehman¹, Sidney D’Mello², and Arthur Graesser¹

¹ Institute for Intelligent Systems, University of Memphis, Memphis, TN 38152
{balehman, a-graesser}@memphis.edu

² Departments of Psychology and Computer Science, University of Notre Dame
Notre Dame, IN 46556
sdmello@nd.edu

Abstract. Experiences of confusion have been found to correlate with learning, particularly for learning at deeper levels of comprehension. Previously, we have induced confusion within learning environments that teach critical scientific reasoning. Confusion was successfully induced with the presentation of contradictory information and false feedback. Next, we would like to regulate experiences of confusion to increase learning. In the current paper, we propose a series of experiments that investigate potential interventions to help regulate confusion during learning. Specifically, these experiments will address the impact of feedback specificity and emotional support.

Keywords: confusion, contradiction, false feedback, affect, tutoring, intelligent tutoring systems, scaffolding, learning.

1 Introduction

Learning is an emotional experience and confusion is one emotion that plays a particularly important role in learning [1]. Learners experience confusion when they are confronted with an anomaly, contradiction, or system breakdown; and are uncertain about how to proceed. Although confusion has been correlationally linked to learning, it is unlikely that the mere experience of confusion promotes deep learning. Instead, confusion creates opportunities for learning because it causes students to stop, reflect, and begin active problem solving to resolve their confusion. These cognitive activities enable learners to work through confusion and acquire a deeper understanding of complex topics [2]. Hence, our working hypothesis is that learning can be increased if intelligent tutoring systems (ITSs) can capitalize on the benefits of confusion.

To take advantage of the benefits of confusion, an ITS must include events that *trigger* confusion in learners, *track* and monitor learner experiences of confusion, and provide support so that learners can *regulate* confusion. Previously we have conducted experiments to address the induction and tracking of confusion [3]. However, we have not yet addressed the third task: regulating confusion. We propose a series of three experiments that will test the effectiveness of interventions to regulate learner experiences of confusion in order to increase positive learning outcomes.

2 Previous Research

We have experimented with confusion induction techniques within learning environments that promoted the learning of scientific reasoning concepts (e.g., experimenter bias, replication). In these experiments, learners engaged in either a triologue with two pedagogical agents (tutor and student) (Experiments 1, 2, 3, and 5) or a dialogue with one agent (tutor) to diagnose flaws in hypothetical research studies (Experiment 4). Confusion was induced through the presentation of contradictory information by two agents in Experiment 1, 2, 3, and 5 [3]. Confusion was successfully induced when the two agents presented opposing opinions and asked the learner to pick one side. In Experiment 4, we induced confusion using false system feedback. After learners attempted to diagnose the flaw in a study, the tutor agent delivered either accurate or inaccurate feedback. We found that learners who responded correctly but received negative feedback (e.g., “That’s wrong.”) were more confused than learners that received accurate feedback.

In addition to confusion induction, we also investigated methods to track learner confusion. The accuracy of learner responses immediately following the manipulations was used to track confusion in the contradictory information experiments, such that incorrect responses were indicative of being in a state of confusion. In the false feedback experiment, learners were asked to self-report experiences of confusion after receiving feedback. Through response quality and strategically placed self-report probes, we have been able to track learner confusion with minimal interruption to the learning process.

There is evidence that partial or complete resolution of confusion can increase learning, particularly at deeper levels of understanding [4]. The two systems discussed above do not currently provide any support for the regulation or resolution of learner confusion. However, increased learning was still found in both experiments. We expect that interventions that help learners regulate and potentially resolve confusion will further increase learning.

3 Future Research Plans

Confusion regulation interventions will be investigated within an ITS that discusses scientific reasoning topics. Learners will engage in dialogues with two agents (tutor and student) while diagnosing the flaws in research studies. Confusion will be induced through the presentation of contradictory information by the two agents and tracked through a combination of response accuracy and strategically placed self-report probes. We will compare interventions based on *feedback specificity* and *emotional support* to help learners regulate their confusion.

In previous experiments learners have provided self-explanations (SEs) after they diagnosed the flaw in a research study, but were not given feedback about SE quality. We hypothesize that elaborated feedback on SE quality will facilitate confusion regulation. To test this hypothesis we will test the impact of feedback specificity on confusion regulation (Proposed Experiment 1). Learner-generated SEs will first be classified as correct or incorrect and then incorrect SEs will be further classified based

on the type of error present. We have already developed mechanisms to facilitate the classification of learner SEs [5]. The tutor agent will either provide *no feedback, non-elaborated feedback only* (e.g., “That’s correct”), or elaborated feedback. For elaborated feedback the tutor agent will first deliver feedback and then either provide the correct answer (*feedback + correct answer*) or correct the specific error that was present in the SE (*feedback + error correction*). Feedback that is tailored to specific errors is expected to facilitate confusion regulation and ultimately improve learning [6].

Proposed Experiment 2 will investigate emotionally supportive interventions in response to induced confusion. It is hypothesized that learners view confusion as indicative of failure and is a threat to their self-concept of intelligence [7]. We will test two types of emotional support to help learners change these negative attributions of confusion. In one condition the tutor agent will serve as an “encouraging and supportive mentor” for the learner by providing *general encouragement* (e.g., “I know you can figure this out!”). To specifically address learner misgivings about confusion, the tutor agent will explain the benefits of confusion (*confusion reappraisal*). We expect that directly targeting learner beliefs about confusion will be more effective for confusion resolution than general encouragement. Finally, Proposed Experiment 3 will compare the most effective interventions from Experiments 1 and 2.

Acknowledgement. The research was supported by the National Science Foundation (REC 0106965, ITR 0325428, HCC 0834847, DRL 1108845) and the Institute of Education Sciences (R305A080594). The opinions expressed are those of the authors and do not represent views of the NSF and IES.

References

1. Calvo, R., D’Mello, S. (eds.): *New Perspectives on affect and learning technologies*. Springer, New York (2011)
2. VanLehn, K., Siler, S., Murray, C., Yamauchi, T., Baggett, W.: Why do only some events cause learning during human tutoring? *Cognition and Instruction* 21(3), 209–249 (2003)
3. Lehman, B., D’Mello, S.K., Strain, A.C., Gross, M., Dobbins, A., Wallace, P., Millis, K., Graesser, A.C.: Inducing and Tracking Confusion with Contradictions during Critical Thinking and Scientific Reasoning. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011*. LNCS, vol. 6738, pp. 171–178. Springer, Heidelberg (2011)
4. D’Mello, S., Graesser, A.: Inducing and tracking confusion and cognitive disequilibrium with breakdown scenarios. *Memory and Cognition* (in review)
5. Lehman, B., Mills, C., D’Mello, S., Graesser, A.: Automatic Evaluation of Learner Self-Explanations and Types of Erroneous Responses for Dialogue-Based ITSs. In: Cerri, S.A., Clancey, B. (eds.) *ITS 2012*. LNCS, vol. 7315, pp. 544–553. Springer, Heidelberg (2012)
6. Fedor, D.: Recipient Responses to Performance Feedback: A Proposed Model and its Implications. *Research in Personnel and Human Resources Management* 9, 73–120
7. Dweck, C.: The development of ability conceptions. In: Wigfield, A., Eccles, J. (eds.) *Development of Achievement Motivation*, pp. 57–88. Academic Press, San Diego (2002)

Using Examples in Intelligent Tutoring Systems

Amir Shareghi Najar¹ and Antonija Mitrovic²

¹ University of Canterbury (Intelligent Computer Tutoring Group)

`amir.shareghinajar@pg.canterbury.ac.nz`

² University of Canterbury (Intelligent Computer Tutoring Group)

`tanja@cosc.canterbury.ac.nz`

Abstract. In the past five years, researchers studied the use of examples compared to tutored problems in different domains, yet the results are not conclusive to decide whether they could replace tutored problems or not. Due to different results in using examples for Intelligent Tutoring System (ITS), there is still a research potential to investigate examples' effects on learning compared to tutored problems. We plan to expand this area to Constraint Based Modeling (CBM) tutors using SQL-Tutor. The result of this study would allow us to develop more effective and efficient ITSs.

Keywords: Examples, Problem-Solving, SQL-Tutor.

1 Introduction

Previous studies found considerable benefit in applying examples in education. Most of the prior studies on supported or unsupported problem-solving indicate that the primary benefit of applying examples in learning is that novices who receive examples learn more efficiently than those who learn by solving problems. Another significant benefit is that using examples improves learning gain. This has been shown for novices when the examples were compared to untutored problems (e.g. [3, 8]). On the other hand, a few studies found no significant difference in learning achievement between examples and tutored problem-solving conditions. Nevertheless, Schwonke et al. [9] show higher learning gain in conceptual knowledge for the example condition. We would like to emphasize that the reviewed studies' are limited to a small number of domains under specific conditions; therefore, there is still a need for further research to gain a better understanding of the examples' application compared to tutored problems.

In our future work, we would like to expand this area to constraint-based modelling tutors. For this purpose, we chose SQL-Tutor developed by the Intelligent Computer Tutoring Group (ICTG) at the University of Canterbury [6]. SQL is a well-defined domain with ill-defined tasks [7]. This makes our study different from the prior studies as they were implemented on well-defined domains with well-defined tasks (e.g. Geometry and Algebra).

In addition, we also plan to investigate the effect of examples with self-explanation (SE). McLaren and Isotani [5] performed a study with three conditions: examples only, problems only and example/problem pairs. They show that

using examples alone is more efficient than the other two conditions in the Stoichiometry domain. However, they have not considered the effect of SE, which was only used after examples, but not after problems. We believe that SE is valuable for learning also after problems. Our hypothesis is that example/problem pairs lead to faster learning and a better learning gain than examples only or problems only conditions. Aleven and Koedinger [1] explain from two studies' results that using tutored problem-solving with self-explanation significantly improves the learning gain, because self-explanation reduces the shallowness of procedural knowledge and provides better integration between verbal and visual declarative knowledge. "Procedural knowledge is implicit knowledge that is now available to awareness whereas declarative knowledge is explicit knowledge that we are aware of in visual or verbal form" [1].

2 Future Work

We plan to provide SE after each problem, but the SE questions must be different from those that were provided in the example conditions. According to Schwonke et al. [9] the conceptual transfer is significantly associated with visual mapping activities (the process of translating from visual thinking onto paper or electronic paper), and procedural transfer is strongly related to principle-based self-explanations; moreover, in Schwonke's study, the students in the example condition had more conceptual transfer than in the problem condition. Hence, in the prior study by McLaren, they reinforced procedural transfer of examples by providing SE. This made the example-SE pair ideal, and it might be the reason that why they found the examples only condition was more efficient than the other conditions. In our study, we plan to use conceptual SE to reinforce learning from problem-solving and procedural transfer to support examples. Overall, we aim to have a better comparison between examples and problems when they both have been reinforced with suitable SE.

The study will have three conditions: Problem-Problem, Example-Problem and Example-Example pairs. Students in the control group will have to solve six isomorphic problem pairs (six question pairs is the average maximum number of SQL questions that a student can solve in 90 minutes). When they solve a tutored problem, then they have to answer a conceptual SE question; if they could not give the right answer, system will give them the answer. In the example-example condition they must follow the same routine as the students in control group, but they have to deal with procedural SE. Therefore, after reviewing each example students need to provide the right answer for the procedural SE question, and if they couldn't give the correct answer then they have to go back to the example in order to find the answer. The last group (example-problem) must first read the example and then they will be asked a procedural SE question. Then they need to solve a problem and answer to a conceptual SE question. In our study, students will answer self-explanation questions by selecting right answers from multiple choices.

According to Atkinson et al. [2], intra-example features describes a single worked-out example design. In this study, we will design examples grounded

on multi-media learning theory [4]. This will improve learning from examples compared to problem solving; however, it would not bias our result as our goal is to investigate the efficiency of example-problem pairs condition compared to problem-problem and example-example only conditions. We expect the example-problem condition to improve learning gain and time more than the example-example and problem-problem pairs.

To recapitulate, the aim of this research is not only to investigate the benefits of example-based strategy in SQL-Tutor, but also to find an ideal approach to present examples in Intelligent Tutoring Systems.

References

- [1] Aleven, V.A., Koedinger, K.R.: An effective metacognitive strategy: learning by doing and explaining with a computer-based cognitive tutor. *Cognitive Science* 26(2), 147–179 (2002)
- [2] Atkinson, R.K., Derry, S.J., Renkl, A., Wortham, D.: Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research* 70(2), 181–214 (2000)
- [3] van Gog, T., Kester, L., Paas, F.: Effects of worked examples, example-problem, and problem-example pairs on novices' learning. *Contemporary Educational Psychology* 36(3), 212–218 (2011)
- [4] Mayer, R.E.: *Multimedia Learning*, 2nd edn. Cambridge University Press, New York (2009)
- [5] McLaren, B., Isotani, S.: When Is It Best to Learn with All Worked Examples? In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011*. LNCS, vol. 6738, pp. 222–229. Springer, Heidelberg (2011)
- [6] Mitrovic, A.: An intelligent SQL Tutor on the web. *Int. J. Artif. Intell. Ed.* 13, 173–197 (2003)
- [7] Mitrovic, A., Weerasinghe, A.: Revisiting ill-definedness and the consequences for ITSs. In: *Proceeding of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pp. 375–382. IOS Press, Amsterdam (2009)
- [8] Rourke, A., Sweller, J.: The worked-example effect using ill-defined problems: Learning to recognise designers' styles. *Learning and Instruction* 19(2), 185–199 (2009)
- [9] Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Aleven, V., Salden, R.: The worked-example effect: Not an artefact of lousy control conditions. *Computers in Human Behavior* 25, 258–266 (2009)

Semi-Supervised Classification of Realtime Physiological Sensor Datastreams for Student Affect Assessment in Intelligent Tutoring

Keith W. Brawner¹, Robert Sottolare¹, and Avelino Gonzalez²

¹ United States Army Research Laboratory
Human Research and Engineering Directorate
Learning in Intelligent Tutoring Environments Laboratory
Simulation and Training Technology Center, Orlando, FL 32826
{Keith.W.Brawner, Robert.Sottolare}@us.army.mil

² University of Central Florida
Department of Electrical Engineering and Computer Science
Orlando, FL 32826
Gonzalez@ucf.edu

Abstract. Famously, individual expert tutoring holds the promise of two standard deviations of improvement over classroom-based instruction. Current content-scaling techniques have been able to prove one standard deviation of improvement. However, just as expert tutors take the motivation and emotional state of the student into account for instruction, so too must computer instructors. Differences between individuals and individual baselines make this difficult, but this information is known across one training session. The construction of assessing modules in realtime, from the available performance and sensor datastreams, skirts these problems, but is technically difficult. This research investigates automated student model construction in realtime from datastreams as a solution from which to base pedagogical strategy recommendations.

Keywords: Intelligent Tutoring, Affective Computing, Datastream Mining.

1 Background, Research, and Direction

Artificial Intelligence is a collection of methods that are used to solve problems. The most frequent problem solved is the automation of decision making, based upon the classification of inputs. The classification problem can be separated into two categories: unsupervised and supervised. Supervised classification problems have training data with provided 'answers', known as 'labels', and testing data. Unsupervised artificial intelligence problems attempt to classify data without knowing the true class of the observation.

Physiological data presents a unique problem to the realm of classification. One of the overwhelming trends in the field of psychology is that all people are different,

known as individual differences. As such, the observed behavior of individuals varies widely. This trend represents itself well among physiological sensors as well [1]. Psychology studies relating to physiological measurements frequently involve the ‘baseline’ of an individual in order to correct for this problem. This is, inherently, an unsupervised learning problem. For example, galvanic skin responses (GSR), which are specific to the individual, must be learned without explicit second-by-second updates on the person's emotions, due to impracticality.

While there have been many studies that use physiological data in order to establish meaning among individuals or groups [2], the problem of individual differences forces the researcher to evaluate each individual individually. While this approach is helpful to psychology researchers, a different approach must be taken for an intelligent tutoring system. If an engineered system was to respond to the needs of its user, this data would have to be parsed, interpreted, and recommended for action in real-time. Because of individual differences, day-to-day variations, inter-day variations, sensor placements, and a host of other issues, baseline measurements cannot be stored for the individual [3]. Establishment of the meaning of these sensors measurements must be made as close to instantaneously as possible. This presents its own problems, starting with the ideas that the data can be of potentially infinite length, and all points and trends on a new individual are unknown.

Intelligent Tutoring comes in many forms. It can be a virtual world where the student can play and practice skills, a computer-led classroom presentation, a computer-human mixed-discussion activity, or other teaching methods. The two fundamental inputs to the human tutor are the assessments of knowledge and the assessments of the affect of the student [4]. Expert human tutors achieve learning gains of two sigma, or roughly two letter grades [5]. Web-based computer tutors, which perform only one of these assessments, have been shown to produce one sigma of learning gain [6]. In order to increase the effectiveness of computer-based learning activities, the intelligent tutor should mirror the approach of human tutoring, and account for the affect of the person being trained [7].

All of the above describes the effort of the author to solve part of a problem which is not only important, but novel. Intelligent tutoring systems should respond to the needs of their students, by assessing their affect, from sensor data taken from the student in realtime, and classified along with self assessments and performance measures. This research addresses this issue through the comparison of supervised against unsupervised methods of machine learning on a dataset of wide-ranging sensors.

This research will develop realtime, unsupervised or semi-supervised methods of affect detection. These models will be directly compared against the supervised linear regression tree models built from validated benchmarks collected in another experiment using low-cost sensors as measurement and high-cost EEG as a moment-by-moment ground truth [8]. The three main thrusts of this research are:

- Group classification models of sensor data are impractical or nonexistent
 - Individual classification models must be built
 - Shown via literature
- Offline individual models of sensor-based affect are not reusable

- Models must be built in realtime
- Shown via literature
- Realtime-constructed models are comparable to their offline counterparts
 - Making them usable in Intelligent Tutoring Systems
 - Shown via experiments and artificial intelligence datastream development [9]

References.

1. Baker, R.S.J.d.: Mining Data for Student Models. In: Nkambou, R., Bourdeau, J., Mizoguchi, R. (eds.) *Advances in Intelligent Tutoring Systems*. SCI, vol. 308, pp. 323–337. Springer, Heidelberg (2010)
2. D’Mello, S.K., Graesser, A.C.: Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Model. User-Adapt. Interact.* 20(2), 147–187 (2010)
3. Bersak, D., McDarby, G., Augenblick, N., McDarby, P., McDonnell, D., McDonal, B., Karkun, R.: Biofeedback using an Immersive Competitive Environment. In: *Online Proceedings for the Designing Ubiquitous Computing Games Workshop, UbiComp 2001* (2001)
4. Scandura, J.: What TutorIT Can Do Better Than a Human and Why: Now and in the Future. In: *Tech., Inst., Cognition and Learning*, vol. 8, pp. 175–227. Old City Publishing (2011)
5. Bloom, B.S.: The 2 sigma problem. The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher* 13(6), 4–16 (1984)
6. Verdú, E., Regueras, L.M., Verdú, M.J., De Castro, J.P., Pérez, M.A.: Is Adaptive Learning Effective? A Review of the Research. In: Qing, L., Chen, S.Y., Xu, A., Li, M. (eds.) *Proceedings of the 7th WSEAS International Conference on Applied Computer & Applied Computational Science (ACACOS 2008)*, pp. 710–715. WSEAS Press, Stevens Point (2008)
7. Woolf, B., Burleson, W., Arroyo, I., Dragon, T., Cooper, D., Picard, R.: Affect-Aware Tutors: Recognizing and Responding to Student Affect. *International Journal of Learning Technology* 4, 129–164 (2009)
8. Carroll, M., Kokini, C., Champney, R., Sottolare, R., Goldberg, B.: Modeling Trainee Affective and Cognitive State Using Low Cost Sensors. In: *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (IITSEC)*, Orlando, FL (November 2011)
9. Brawner, K., Gonzalez, A.: Realtime Clustering of Unlabeled Sensory Data for User State Assessment. In: *Proceedings of International Defense & Homeland Security Simulation Workshop of the I3M Conference, Rome, Italy* (September 2011)

Detection of Cognitive Strategies in Reading Comprehension Tasks

Terry Peckham

Laboratory for Advanced Research in Intelligent Educational Systems
University of Saskatchewan
tep578@mail.usask.ca

Abstract. In this paper we discuss the detection of cognitive strategies used in a reading comprehension task. By mining the results of student interaction data we have been able to determine various cognitive strategies employed by the students that are positive for learning and others that are negative. Some of these strategies are associated with the Bloom level of the student's task. This could be useful to learning environments in directly supporting students' learning metacognitive skills.

Keywords: Bloom's Taxonomy, data mining, metacognition, reading comprehension.

1 Introduction

Reading comprehension is critical in life-long learning [4] and in the workplace, where self-regulated learning (SRL) is the key. Azevedo, et. al. [3] demonstrated that it is possible to identify student metacognitive strategies in a SRL environment by using a tracing methodology that analyzed the data collected by the environment rather than self-reported measures which are commonly used within educational environments. This recording of student interaction data with course content at a fine grained level of detail along with the pedagogical framework of the course allows for the automated classification of students as effective or ineffective in their use of cognitive skills. Some of the skills are associated with a particular level of Bloom's taxonomy¹ [2]. Bloom's taxonomy of the cognitive domain provides a pedagogical hierarchy of six levels of cognitive strategies ranging in difficulty from the simple recall of facts at the low end of the scale to analysis, synthesis, and evaluation at the high end of the scale. There are times when it is more expedient to use a different cognitive strategy than the one normally employed by a student to solve a given task. If we are able to capture cognitive strategies from student usage data, we can inform a student model and/or provide feedback to the student to help them use a better strategy, thereby scaffolding metacognition appropriate to the Bloom level of the student's task. This would extend current work on metacognition in intelligent tutoring systems (ITS).

¹ Bloom's Taxonomy Levels, in increasing order of difficulty, are comprised of Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation.

2 The Experiment

An experiment was designed to look for patterns of student behavior in a reading comprehension task. Student interaction with a learning environment was designed to emulate hypermedia courses offered in post-secondary institutions where content is presented along with questions about that content. The students could view the content and/or questions in any order or manner they chose with no constraints applied in their interaction with the system. All the interactions/events with the content and questions were recorded and time-stamped.

The students were tasked with reading content and then answering various questions at different levels of Bloom's Taxonomy. The first experimental condition had the students answering lower level Bloom questions on a single document. In the second condition students answered questions at higher Bloom levels with multiple documents. The participants were adult students enrolled in a grade 12 Adult Education English course. There were 17 participants for the first experimental condition and 11 for the second with an average age of twenty six.

3 Results

The timestamp data was processed so that reading, scanning and scrolling navigation times could be calculated for each of the interactions/events. The time cutoffs used to distinguish reading from scanning from scrolling fit with other document navigation research [1]. Multi-dimensional K-Means clustering was applied to the data collected with respect to the reading, scanning and scrolling times. The following clusters proved to be statistically interesting with respect to the Bloom level:

- Light Reading Cluster: 50% reading: 30% scanning: 20% scrolling (50:30:20)
- Light Medium Reading Cluster: (60:30:10)
- Heavy Medium Reading Cluster: (70:20:10)
- Heavy Reading Cluster: (80:10:10)

Two other clusters, Medium Scrolling (10:10:60) and Medium Scanning (20:60:20), did appear in the clustering searches but the number of data points that were placed into these categories was often quite small. Because of their small size they were omitted from the statistical analysis.

An ANOVA was performed on each of the clusters as it relates to each level of Bloom found within the experiment. All of the clusters were statistically significant with the exception of Bloom's level five (due to a small sample size). For example, those students who were classified as light readers for Bloom level 1 questions were significantly different in their reading strategies from those clustered as light medium readers for the same Bloom level. A Tukey-Kramer analysis was used to help control for unequal sample sizes in this type of analysis. When a test was performed to see if the clusters were significantly different without using Bloom in the analysis, no significant differences between the clusters were found. Interestingly, the granularity

of the Bloom level is still significant if we group the clusters into high and low level Bloom rather than at each individual level.

Next we analyzed how the clusters were related to the Bloom level. The Light Reading and Heavy Medium Reading clusters were not found above Bloom Level 3 and the Heavy Reading cluster was found at all Bloom levels. As the Bloom level increases the amount of Heavy Reading increases while the amount of Light Reading decreases. The Light Medium Reading cluster was only found at Bloom level 1 and 2. There were several students that used the same cognitive strategy despite the difficulty of the task while there were others who adopted different strategies for different difficulty levels. Those participants that chose a Heavy Reading style did not complete all of the questions as time became a factor. Similarly, those that used a light reading style completed on time or early but performed poorly on some of their tasks where higher level Bloom skills were needed. Those who varied their strategies depending on Bloom level tended to perform better.

4 Conclusions

This experiment demonstrates that not all students choose the correct cognitive strategy to solve various tasks. Since we have been able to detect these inconsistencies automatically we have the potential to update a student model, inform the student about their metacognitive strategies and/or suggest appropriate pedagogical tasks that could be useful for a student attempting to improve weak metacognitive skills. Future research will explore the patterns found in reading comprehension, further enquire about the relationship between the selection of reading content and questions that need to be answered, and look at how these patterns can be exploited by an ITS.

References

1. Alexander, J., Cockburn, A.: An Empirical Characterization of Electronic Document Navigation. In: Proceedings of Graphics Interface 2008, Windsor, On., Canada, pp. 123–130 (2008)
2. Anderson, L.W., Krathwohl, D.R., Bloom, B.S. (eds.): Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Longman, New York (2000)
3. Azevedo, R., Moos, D., Johnson, A., Chauncey, A.: Measuring Cognitive and Metacognitive Regulatory Processes During Hypermedia Learning: Issues and Challenges. *Educational Psychologist* 45(4), 210–223 (2010)
4. Sporer, N., Brunstein, J., Kieschke, U.: Improving student's reading comprehension skills: Effects of strategy instruction and reciprocal teaching. *Learning and Instruction* 19, 272–286 (2009)

The Effects of Adaptive Sequencing Algorithms on Player Engagement within an Online Game

Derek Lomas¹, John Stamper¹, Ryan Muller¹, Kishan Patel², and Kenneth R. Koedinger¹

¹ Carnegie Mellon University

{derekloomas, jstamper, rmuller, koedinger}@cs.cmu.edu

² Dhirubhai Ambani Institute of Information and Communication, Gujarat, India
kishan_patel@daiict.ac.in

Abstract. Using the online educational game *Battleship Numberline*, we have collected over 8 million number line estimates from hundreds of thousands of players. Using random assignment, we evaluate the effects of various adaptive sequencing algorithms on player engagement and learning.

Keywords: games, number sense, engagement, adaptive sequencing.

1 Introduction

Number line estimation accuracy is highly correlated with math achievement scores in grades K-8 (Siegler, Thompson, Schneider, 2011). To promote practice with number line estimation, we have developed *Battleship Numberline*, a game involving estimating the location of ships on a number line. Using this game, we have collected over 8 million number line estimates from several hundred thousand online players. The order of instructional items in the game is typically presented at random, but we hypothesize that an adaptive sequence will result in an improved learning experience. Adaptive instructional sequences are best known for increasing the efficiency of learning [2]. However, Pavlik et al. [3] reported that students tended to choose an adaptive sequence of foreign language instructional items over a random sequence of items. We further explore this phenomenon by investigating whether adaptive sequences can increase motivation to engage in a learning activity.

2 Adaptive Sequences

Conati et al. [1] describe using Bayesian Knowledge Tracing (BKT) to promote learning in an educational game. However, many games use far simpler algorithms to promote learning and player interest; for instance, they may require a player to perform flawlessly on a level before progressing to the next. Could simpler adaptive algorithms achieve comparable performance to Bayesian Knowledge Tracing? Specifically, could they produce comparable learning (pre-post test gain) and player engagement (duration of intrinsically-motivated play)?

In our implementation of BKT, we developed a knowledge component model with five knowledge components (KC). The parameters for the model were developed

based on data collected from a prior classroom study involving 150 students in 4th-6th grade. These parameters included the probability of existing knowledge (LO), learning rates (T), and the probability for slipping (S) and guessing (G). The sequencing algorithm worked by randomly choosing an item belonging to the KC with the highest probability of being known, so long as it was below the threshold of .9 probability of being known. When a KC exceeded .9, it was removed from the sequence. Once all KCs in the level exceeded .9, the level was over.

The Difficulty Ladder (dLadder) is an adaptive sequencing algorithm that requires mastery of easier items before allowing progress to more difficult items. Based on the same dataset from which the BKT parameters were derived, the items in the instructional sequence were divided into 5 bins of difficulty, each with 4 items. Players began in the easiest bin; if they were correct twice in a row, they advanced to the next more difficult bin. If they were incorrect twice in a row, they went back to the previous, less difficult bin. When the player completed the hardest bin, the level was over. A high performing player could complete the ladder in only 10 trials.

Naïve ITS is based on the idea that a successful response tends to generate more learning than an unsuccessful response. To promote success, if a player gets an item incorrect, they are given another opportunity to attempt the item after a delay of one other item. The delay of one trial facilitates working memory retrieval without making the task trivially easy (as it might be if there was no delay). Once the player gets every item correct at least once, the level is over.

The random sequence randomly presents (without replacement) one of 20 different fractions. Unlike the adaptive sequences, the random sequence is not affected by the player's prior performance.

3 Experiment 1: Structure, Participants and Metrics

The adaptive sequencing experiment involved randomly assigning 1087 players to one of sixteen different level sequences representing four different experimental conditions (BKT, Difficulty Ladder, Naïve ITS, & Random) with four different pre/posttest form combinations (A-B, B-C, C-D, D-A). Each level sequence consisted of a pretest level, a level with one of four sequencing algorithms, a post-test level, and then additional levels of the same sequencing algorithm (so that patterns of extended play could be compared over the different algorithms). The pre/post tests involve four fraction estimation problems, presented fully within the context of the game.

Our participants are anonymous online players who freely access our game through the educational portal Brainpop.com. Despite this anonymity, we can infer from the demographics of Brainpop.com that our users are likely to be third to eighth grade students, probably playing in a classroom setting. Brainpop.com offers a number of different educational games. We assume that students are free to stop playing *Battle-ship Numberline* at any time; indeed, over 50% of students play less than 10 trials.

In this study, we define engagement as the number of trials that a player chooses to play, as this is believed to reflect the players intrinsic motivation to participate in the gameplay sequence. We measure learning as the gain from pretest to posttest.

Table 1. Initial conditions of experiment

	Completed Pretest	Pretest Av.	Av. # of Trials	Median # of Trials	% playing > 40 Trials
BKT	265	23% (.42)	25(30)	14	20%
DLadder	267	23% (.42)	29(35)	16	25%
NaiveITS	279	26% (.44)	30(37)	16	24%
Random	276	23% (.42)	24(22)	15	21%

Table 2. Here, data is presented only for the players that completed the posttest. Gain is significant from pre to post test over all conditions ($p < .02$, $p < .01$, $p < .001$) using a paired t-test.

	Completed Posttest	Pretest Av.	Posttest Av.	Median # of Trials
BKT	0	n/a	n/a	n/a
DLadder	22	46% (.50)	65%(.48)	30.5
NaiveITS	55	31% (.46)	47%(.50)	49
Random	103	25% (.43)	37%(.48)	28

4 Discussion

The data presented here suggests a modest effect from the sequencing algorithms. Unfortunately, learning gains are impossible to compare directly, without statistically correcting for the substantial rates of attrition. Our BKT algorithm apparently set the bar too high—no players in this sample actually completed the level, despite some players completing more than 100 trials. Future work will involve tuning the parameters of the BKT algorithm, developing more comparable measures of learning, and validating our online engagement construct in a classroom setting.

References

1. Conati, C., Zhao, X.: Building and evaluating an intelligent pedagogical agent to improve the effectiveness of an educational game. In: Proceedings of the 9th International Conference on Intelligent User Interfaces, pp. 6–13. ACM (2004)
2. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modelling and User-Adapted Interaction* 4(4), 253–278 (1995)
3. Pavlik Jr., P., Bolster, T., Wu, S.-M., Koedinger, K., MacWhinney, B.: Using Optimally Selected Drill Practice to Train Basic Facts. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 593–602. Springer, Heidelberg (2008)
4. Siegler, R.S., Thompson, C.A., Schneider, M.: An integrated theory of whole number and fractions development. *Cognitive Psychology* 62(4), 273–296 (2011)

A Canonicalizing Model for Building Programming Tutors

Kelly Rivers and Kenneth R. Koedinger

Carnegie Mellon University

Abstract. It is difficult to build intelligent tutoring systems in the domain of programming due to the complexity and variety of possible answers. To simplify this process, we have constructed a language-independent canonicalized model for programming solutions. This model allows for much greater overlap across different students than a basic text model, which enables more self-sustaining hint generation methods in programming tutors.

Keywords: canonicalization, programming tutors, abstract syntax trees.

1 Introduction

Though interest has continually been shown in creating intelligent tutors for programming topics, few solutions have been found that have been applied to widespread classes [1]. This is partially due to constraints already existing in the classroom such as programming language, development environment, and curriculum choices. We aim to simplify the tutor-building process by creating a language-independent method for turning students' programs into canonicalized models which can be more easily examined and compared than text programs. We also discuss ideas for self-sustaining hint generators that would not require as much instructor input.

2 Model Creation

Our model is based on **abstract syntax trees** (ASTs). ASTs represent the underlying structure of a program by branching complex statements out into smaller sub-statements. They are commonly used in program transformations, which means that modules already exist for creating and modifying ASTs from text for many different programming languages; they're also constructed from basic programming concepts, so they can be made equivalent across languages.

Once a student's program has been converted into an AST, we can gather relevant information on what data structures and algorithms the student is using by examining the tree. This information can later be used to unearth basic problems. For example, a student uncomfortable with variables might try to write an entire program in one line rather than use any assignments, while another student might write code after a return statement without realizing that it isn't being run.

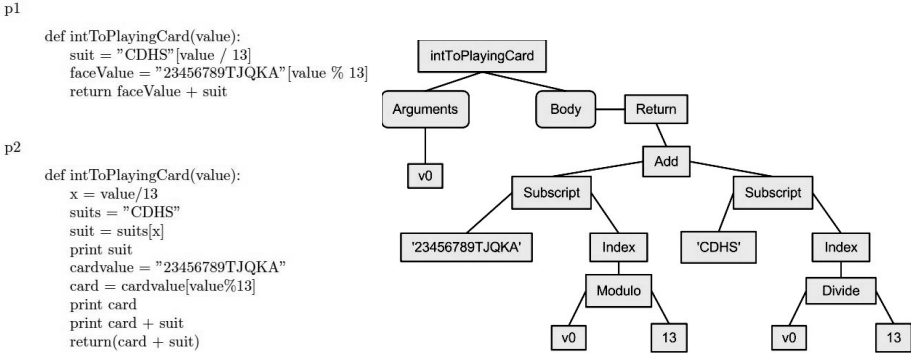


Fig. 1. Above, the two programs shown canonicalize to the same model

At that point, canonicalizing functions can be run over the AST to change it into a format more likely to match other students’ submissions. These functions are commonly used in compiler optimizations and result in trees which can be shown to be semantically equivalent [2], so they will not change the student’s output. The functions we use currently include:

- Collapsing constant operations
- Propagating expressions assigned to variables
- Using De Morgan’s laws to propagate the *not* op inside boolean statements
- Normalizing the direction of comparisons
- Ordering commutative operators with a strict comparison function
- Removing unreachable and unused code
- Inlining helper functions

We did preliminary testing of this model using solutions to basic programming problems taken from an introductory programming course composed of around five hundred students. A median of 70% of the students could be mapped to common solution groups (where groups were composed of 2 to 300 students). Fig. 1 demonstrates how this includes submissions that look completely different on a textual level. We are currently extending the model to work for more complicated problems, and results have been promising (a median of 25% of the students map to groups in multi-function problems using control structures).

Next, we plan to utilize machine learning algorithms to determine the best methodology for creating a clustering of canonicalized models, using unit test results, tree substructures, tokens, and any other information which proves useful to construct the clustering algorithm. We are also considering using text mining techniques on the tokens of the canonicalized abstract syntax trees. We plan to verify the correctness of the resulting algorithms by checking it against original grades and results from unit tests run on the original submissions.

3 Hint Generation and Future Work

The next steps involve experimenting with different ways to generate hints based off of canonicalized models. A few approaches which could be adapted include:

Model Driven: Basic hints could be created based entirely on the student's underlying model and the canonicalizing functions used to create it. This would look for the typical red flags of bad code- unreachable statements, infinite loops, etc.- to give suggestions for improvement. It could also be trained to look for typical novice mistakes, such as off-by-one errors and stylistic mistakes.

Data Driven: In this approach (inspired by work done on creating automatic hints in a logic tutor [3]), the clustering of models would be used in conjunction with compile-time data about how previous programs changed over time until they reached a solution. The solutions found by other students whose models were closest in the clustering would be used to determine the optimal next step for the student asking for a hint.

Crowd Driven: Instead of being programmatically based, this option uses crowd-sourcing amongst students to slowly build a database of hints. Students would type quick conceptual explanations of how they had fixed a problem after progressing past a state; these statements could then be re-used as hints for future students stuck at the same state. A simple voting system could bring the best hints to the top, and a filtering mechanism could keep the explanations from giving away exact solutions.

We plan to continue work on this concept using a corpus of final submissions from the introductory programming course at our university. If this method is successful, we hope to use it in a system for programming instructors requiring little input or upkeep, which would be ideal for the large-scale courses which have become popular recently; in such an environment, solutions would be submitted rapidly enough to provide tutoring for complex problems. We also hope to explore how canonicalization could be used as a method for grouping submissions (for purposes such as general grading and plagiarism detection) and how canonicalizing functions should best be classified for instructor use.

Acknowledgements. This work was supported in part by Graduate Training Grant awarded to Carnegie Mellon University by the Department of Education (# R305B090023).

References

1. Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M., Paterson, J.: A survey of literature on the teaching of introductory programming. *ACM SIGCSE Bulletin* 39(4), 204–223 (2007)
2. Xu, S., Chee, Y.S.: Transformation-Based Diagnosis of Student Programs for Programming Tutoring Systems. *IEEE Transactions on Software Engineering* 29(4), 360–384 (2003)
3. Barnes, T., Stamper, J.: Toward Automatic Hint Generation for Logic Proof Tutoring Using Historical Student Data. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) *ITS 2008*. LNCS, vol. 5091, pp. 373–382. Springer, Heidelberg (2008)

Developmentally Appropriate Intelligent Spatial Tutoring for Mobile Devices

Melissa A. Wiederrecht and Amy C. Ulinski

University of Wyoming, Laramie WY 82072, USA
{mwiederr,aulinski}@uwyo.edu

Abstract. Given the centrality of spatial reasoning to the STEM disciplines, it is commonly acknowledged that it is of great importance that researchers determine effective ways to train our next generation to be spatially literate. Early childhood has been shown to be a very important time in a child’s development of spatial skills and it is also known that certain types of interventions can help children develop higher levels of spatial ability. However, teaching young children comes with unique challenges, such as Developmentally Appropriate teaching and open-ended instruction and play. We propose that an Intelligent Tutoring System might be useful to address these challenges and present an initial research plan to design one to teach young children spatial skills in a Developmentally Appropriate, open-ended play-based manner.

Keywords: early childhood, developmentally appropriate, open-ended play, spatial reasoning, intelligent tutoring systems, mobile learning.

1 Motivation and Related Works

Spatial reasoning is well known to be of vital importance in the disciplines of science, technology, engineering, and mathematics (STEM) [16]. It is also known that spatial reasoning is a skill that can be learned with practice [12]. In particular, spatial reasoning has been shown to be malleable in very young children [12], and it is recognized to be very important that we find ways to encourage the development of spatial thinking in children in the preschool years [12].

Mobile devices have been shown to offer unique attributes that can help benefit education for both adults and children: among other benefits, they encourage “anywhere, anytime” learning, fit with learning environments, and enable a personalized learning experience [14]. However, to take full advantage of the opportunities that mobile devices offer to education, developers must be careful to design software that fulfills standards of good educational quality.

For young children (ages 3-8), the gold standard of educational quality is a concept called Developmentally Appropriate (DA) Practice. The National Association for the Education of Young Children (NAEYC) position statement on Technology and Young Children specified that DA software should be 1) age appropriate, 2) individually appropriate, and 3) culturally appropriate, and should

engage “children in creative play, mastery learning, problem solving, and conversation” [10]. Others also recommend that DA software should “match the child’s current level of understanding and skills, while growing with the child” [6], and should be open-ended to allow the children control over their environment [15].

More specifically for systems designed to teach spatial skills, it has been shown that children learn spatial skills better in the context of a story [1]. Further, several learning trajectories have been identified which specify the stages that young children go through in the development of certain spatial skills [13] that we propose could be adapted for use in a computer tutoring setting.

Given the effectiveness of Intelligent Tutoring Systems (ITSs) in many domains for providing individualized instruction for adults and children, we will investigate the feasibility and effectiveness of an ITS for providing a DA learning experience to enhance early childhood spatial education. Several ITSs have been developed to teach spatial skills in the past [3,5,9]. Other systems without intelligent capabilities have been developed to teach spatial skills to adults [8,11,7] and to children [2]. Work has also been done on systems designed for open-ended play for children [4]. However, it has yet to be investigated how the benefits of an ITS may be utilized to teach spatial skills in a DA manner to young children.

2 Initial Research Plan

We will investigate the feasibility and effectiveness of a mobile, story-based ITS to teach spatial skills to young children in a DA manner utilizing spatial reasoning learning trajectories. We will answer the following:

1. What measures can we use to evaluate children’s spatial abilities on mobile devices during open-ended play?
2. What tasks will provide data for our measures and allow for inquiry-based, open-ended-play-based developmentally appropriate interactions?
3. How can we develop useful child-centric student models using data from these tasks and measures?
4. Based on our student model, what methods of intervention most effectively guide the children to a higher level of understanding of spatial reasoning?
5. Finally, how well do children learn spatial reasoning from our system compared to more traditional approaches both on and off the computer?

We will select measures based on proven spatial ability tests and determine their effectiveness for our application. Then we will adapt proven spatial ability tasks to fit our child-centered requirements and to provide data for our chosen measures. This data will be organized into a student model which will be constructed based on existing tutoring systems strategies and adapted to fit the child-centered requirements. Next we will utilize knowledge from early childhood education and spatial cognition about how to best intervene to encourage learning of spatial concepts. Finally, we will conduct studies comparing the same types of instruction with our system, with non-intelligent computer-based systems, and with traditional non-computer instruction to determine the relative effectiveness of our system.

3 Conclusion

We hope that our work on determining the feasibility and effectiveness of an Intelligent Tutoring System designed to teach young children spatial reasoning skills in a Developmentally Appropriate, open-ended, play-based manner will provide a beneficial, accessible, and enjoyable way for children to learn these skills that will be vital to their future success in the STEM disciplines.

References

1. Casey, B., Erkut, S., Ceder, I., Young, J.M.: Use of a storytelling context to improve girls' and boys' geometry skills in kindergarten. *Journal of Applied Developmental Psychology* 29, 29–48 (2008)
2. Clements, D.H., Sarama, J.: Effects of a Preschool Mathematics Curriculum: Summative Research on the Building Blocks Project. *Journal for Research in Mathematics Education* 38, 136–163 (2007)
3. Connell, M.W., Stevens, D.A.: A computer-based tutoring system for visual-spatial skills: dynamically adapting to the user's developmental range. In: *ICDL 2002* (2002)
4. Creighton, E.: Jogo: an explorative design for free play. In: *IDC 2010*, Barcelona, Spain, June 9–12 (2010)
5. Fournier-Viger, P., Nkambou, R., Mayers, A.: Evaluating spatial representations and skills in a simulator-based tutoring system. *IEEE Transactions on Learning Technologies* 1 (2008)
6. Haugland, S.: Children's Home Computer Use: An Opportunity for Parent/Teacher Collaboration. *Early Childhood Education Journal* 25 (1997)
7. Martin-Dorta, N., et al.: A 3D Educational Mobile Game to Enhance Student's Spatial Skills. In: *2010 10th IEEE International Conference on Advanced Learning Technologies* (2010)
8. Martín-Gutiérrez, J., Contero, M., Alcañiz, M.: Evaluating the Usability of an Augmented Reality Based Educational Application. In: Alevén, V., Kay, J., Mostow, J. (eds.) *ITS 2010*, part I. LNCS, vol. 6094, pp. 296–306. Springer, Heidelberg (2010)
9. Mengshoel, O.J., Chaeuahan, S., Kim, Y.S.: Intelligent critiquing and tutoring of spatial reasoning skills. *Artificial Intelligent for Engineering Design, Analysis and Manufacturing* 10, 235–249 (1996)
10. National Association for the Education of Young Children (NAEYC). *NAEYC Position Statement: Technology and Young Children - Ages 3 through 8* (1996)
11. Nesbitt, K., Sutton, K., Wilson, J.: Improving Player Spatial Abilities for 3D Challenges. In: *IE 2009*, Sydney, Australia, December 17–19 (2009)
12. Newcombe, N.S., Frick, A.: Early education for spatial intelligence: why, what, and how. *Mind, Brain, and Education* 4, 102–111 (2010)
13. Sarama, J., Clements, D.H.: *Early childhood mathematics education research: learning trajectories for young children*. Routledge (2009)
14. Shuler, C.: *Pockets of potential: using mobile technologies to promote children's learning*. The Joan Ganz Cooney Center at Sesame Workshop, New York (2009)
15. Snider, S.L., Badget, T.L.: "I have this computer, what do I do now?" Using technology to enhance every child's learning. *Early Childhood Educ. J.* 23 (1995)
16. Wai, J., Lubinski, D., Benbow, P.: Spatial ability for STEM domains: aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of Educational Psychology* 101, 817–835 (2009)

Leveraging Game Design to Promote Effective User Behavior of Intelligent Tutoring Systems

Matthew W. Johnson, Tomoko Okimoto, and Tiffany Barnes

University of North Carolina at Charlotte, Charlotte NC 28223, USA

Abstract. We propose developing a mobile device application that will leverage game-play mechanics to incentivize optimal spacing for second language vocabulary acquisition. Through the collection and analysis of user log-data, we intend to investigate the effects of pervasive studying, studying vocabulary words for short intervals, many times throughout a day. This investigation will provide insight into new strategies of studying second language vocabulary, which may be more efficient.

Keywords: Game Design, Spacing, Second Language Acquisition, Pervasive Games.

1 Introduction

Intelligent tutoring systems and digital games have a number of aspects in common, for example, the feedback and update loops present. However, we can leverage certain aspects of digital games to improve intelligent tutoring systems. One important aspect of learning a second language is repetition and consistency, but repetitively reviewing vocabulary can reduce motivation and interest. We hypothesize that 1) we can use game design and mechanics to maintain consistent levels of motivation and interest, 2) that we can use game play mechanics to incentivize optimal spacing of language items for students, to make second language (L2) acquisition more efficient, and 3) that we will be able to investigate the effects of pervasive studying, studying for many short intervals in a day, by using a mobile device delivery method.

The Japanese language has three alphabet systems used in the written language, Kanji, Hiragana and Katakana. Hiragana and Katakana each have 46 characters and Kanji are the Chinese characters often used to represent words, of which there are 2,136 that make up the $J\bar{o}y\bar{o}$ or ‘regular-use Chinese characters’ [3]. Next, when writing the order of those 2,136 characters the pronunciation may change, and many characters have multiple pronunciations depending on where in the word it appears. These characteristics of the Japanese language make it difficult for L2 learners to study, learn and become literate. The Japanese language has a standardized test, known as the JLPT, Japanese Language Proficiency Test, which is comprised of four levels. The JLPT divides the $J\bar{o}y\bar{o}$ Kanji into the four levels of the JLPT, level four being the easiest and level one being the most difficult.

2 Method

We propose the development of an Intelligent Tutoring System and Digital game hybrid built for mobile devices, which will allow for players to login and practice vocabulary words in a pervasive fashion. The software will have two phases; the first phase is the review phase and is inspired from the works in the field of Intelligent Tutoring Systems. In phase one, players will study and review vocabulary words, during this phase they will earn points, used in phase two. The second phase will be the game phase where the players will spend points earned in the first phase to improve one's character or achieve other game-based goals.

2.1 Gamification

Players will choose their ability level, based on the JLPT, to define the set of words to study. Crothers and Suppes show that the number of words to study at one time is dependent on word difficulty, but for lists with limited difficult words, 100 words seem to be optimal[2], this will act as our Open-list. As players consistently answer Kanji items correctly, the game will move those questions from the Open-list to a review-list. Crothers and Suppes also discovered that people are capable of remembering 100 words within seven repetitions, and at 80% retention for lists with 216 words after six repetitions[2]. Other research has shown studying word-pairs is an effective method for vocabulary acquisition [4]. Techniques developed for ITSs, like student modeling, can be incorporated into the ITS phase to improve word selection.

We will apply two game mechanics to motivate consistent play, ongoing motivation, and optimal spacing. The first mechanic is to incorporate spacing, e.g. every 12 hours, players can perform a Kanji-quiz in the ITS phase in order to gain points. Researchers have studied spacing in many ways and with proper spacing people are less likely to forget items they have studied [1,5]. The ITS phase will have a review and quiz stage. The review stage will always be available and give players an unlimited time to review Kanji. During the quiz stage, students can gain points for correctly answering questions. After the interval has expired, the player can again perform the quiz in order to gain more points. We can calculate points based on many player attributes, like success rate on Kanji-items, the number of item attempts, and item difficulty. By limiting the availability of the point gaining opportunity, players will value that opportunity and are likely to maximize their point earning potential. The most efficient performance is scoring points as soon as they are available to the player, i.e. at every interval.

The game phase will use the points earned to reward players. Depending on the theme or 'skinning' of the game we could see varying degrees of interest from the audience, so the game should provide a setting, which our audience, college students studying the Japanese language, have an interest. Ideally, the game-portion would also provide an educational experience but that is not required. We can also offer points when players log in consecutively over days, and maintain

desired spacing. Lastly, we can incorporate an achievement method to motivate review, where players can gain point bonuses for overcoming thresholds, for example reviewing 1000 items or for consecutively answering questions correctly.

By making the system work on mobile devices like iOS or Android, we can allow the player the ability to quickly pick up and play for even short periods, making their study-time more accessible. To facilitate this, the shortest play-experience can be as quick as reviewing a single item and ending a play session.

2.2 Study Design

We propose the following study to investigate whether these game play mechanics have a positive effect on learning gains, consistency and motivation through the course of a player's study of the Japanese language. In collaboration with the Japanese department at my University, students will be provided access to either a game play version or strictly the ITS version of the software. Next we will provide logins to students so we can log their user-data for later analysis. Through our collaboration we can administer pre-tests and post-tests on Kanji to understand the types of affects the game had on student progress in learning Japanese Kanji.

3 Conclusion

We propose the development of a pervasive ITS-game for mobile devices so we can investigate the affects of game-play mechanics that incentivize optimal spacing. Furthermore we will collect log data to better understand player behavior and gain insight into learning gains of pervasive studying. Through data analysis we can gain a better understanding of what methods players use for learning a second language. A meaningful advantage this approach has over other approaches is we will be able to collect and analyze data from players and monitor their study habits over long periods, including entire semesters, and even over multiple semesters through the cooperation of the University's Japanese language department.

References

1. Atkinson, R.C.: Optimizing the learning of a second-language vocabulary. *Journal of Experimental Psychology* 96, 124–129 (1972)
2. Crothers, E.J., Suppes, P.: Experiments in second-language learning. Academic Press, New York (1967)
3. MEXT: Tech. rep., Ministry of Education, Culture, Sports, Science and Technology (2010), http://www.mext.go.jp/b_menu/hakusho/nc/k19811001001/k19811001001.html
4. Nation, I.S.P.: Beginning to learn foreign vocabulary: A review of the research. *RELJ Journal* 13(1), 14–36 (1982), <http://rel.sagepub.com/content/13/1/14.abstract>
5. Pavlik, P.I., Anderson, J.R.: Practice and forgetting effects on vocabulary memory: An activation based model of the spacing effect. *Cognitive Science* 29, 559–586 (2005)

Design of a Knowledge Base to Teach Programming

Dinesha Weragama and Jim Reye

School of Electrical Engineering & Computer Science
Queensland University of Technology, Brisbane, Australia
d.weragama@qut.edu.au, j.reye@qut.edu.au

Abstract. Programming is a subject that many beginning students find difficult. This paper describes a knowledge base designed for the purpose of analyzing programs written in the PHP web development language. The aim is to use this knowledge base in an Intelligent Tutoring System that will provide effective feedback to students. The main focus of this research is that a programming exercise can have many correct solutions. This paper presents an overview of how the proposed knowledge base can be utilized to accept different solutions to a given exercise.

Keywords: knowledge base design, Intelligent Tutoring System, program analysis, PHP.

1 Introduction

Programming is a very difficult subject for many beginning students. This paper describes a knowledge base designed to support an Intelligent Tutoring System (ITS) that will teach programming to novices. Since PHP is a popular web development language, it has been selected as the concrete language for this ITS.

A computer programming problem very rarely has a unique solution. This is illustrated by the three simple program examples in Table 1, each of which is superficially different, but each of which has the same overall effect of setting the variable y to 0 when x has an integer value that is greater than 10, and to 1 in all other instances.

Therefore, the proposed ITS should be capable of analyzing different student solutions to a given problem and providing constructive feedback. The strength of the proposed knowledge base is that it is capable of supporting many alternative solutions to a single programming exercise.

Table 1. Programs to illustrate different solutions to a given programming task

Program a	Program b	Program c
<pre>if (\$x>10) \$y=0; else \$y=1;</pre>	<pre>if (\$x>=11) \$y=0; else \$y=1;</pre>	<pre>if (\$x<=10) \$y=1; else \$y=0;</pre>

2 Knowledge Base

The knowledge base in this system has been designed as a set of predicates in first order logic, together with associated rules and actions. Currently, it is capable of handling key aspects of assignment statements, conditional statements, arrays, for loops, functions and HTML form processing.

Fig. 1 shows a Object Relational Modeling (ORM) [1] diagram that contains some key objects and predicates in the knowledge base. Once a student submits their answer to an exercise, the program code is first parsed into an abstract syntax tree (AST). The AST is then processed node by node, creating corresponding instances of predicates. The knowledge base also contains a large number of rules that are used to derive more predicates in order to analyze the student’s solution. Some programming statements, such as the assignment statement, are modeled as actions. When such a statement is encountered, the corresponding action is performed, resulting in the creation of more predicates.

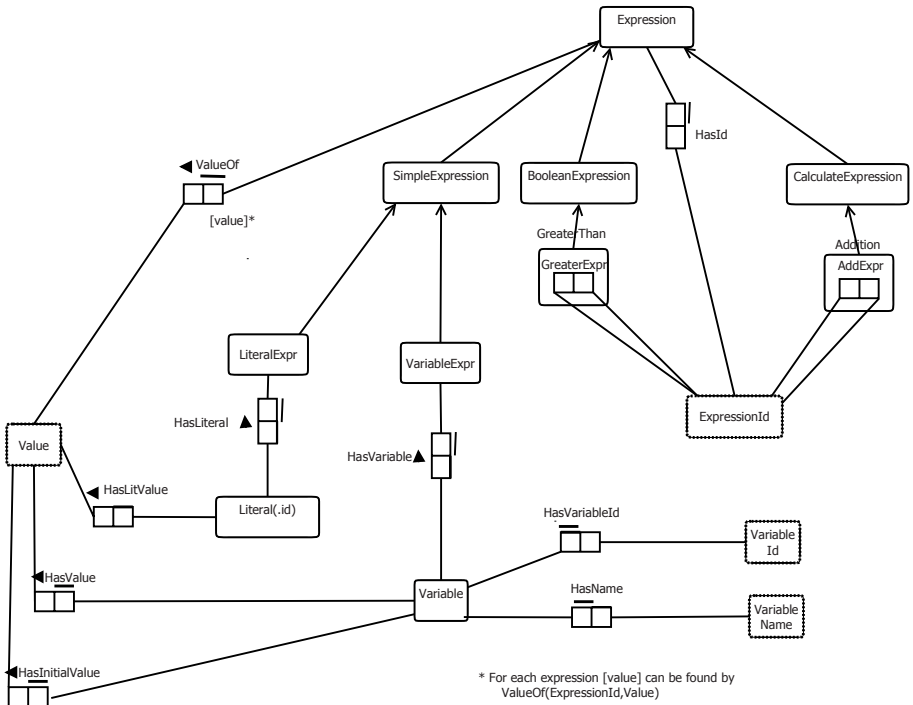


Fig. 1. ORM diagram of key predicates

The formal program specification for each exercise contains a goal which is a combination of the above predicates. Once the student’s program is converted to predicates using the above method, the resultant set of predicates is known as the final

state. This final state is then compared against the goal to identify any predicates that are missing or are unnecessary. This information can then be used to provide feedback that is appropriate to the missing predicates.

As mentioned earlier, a programming exercise does not have a unique solution. The knowledge base makes use of different rules to convert different types of programs into a standardized set of predicates that corresponds with the goal state.

If statements are handled in the knowledge base as implications. The condition checked in the if statement, such as `GreaterThan(x,y)` implies the predicates that result from the statements under the if part. Looping statements are modeled using sub-plans. The functionality of the desired loop is specified as a set of predicates representing the precondition and the post-condition of the loop. When the predicates derived from the program match these conditions, the loop is taken to be correct.

3 Discussion and Related Work

Programming is a difficult task for beginning students. Many ITSs have been built to teach programming to novices. The knowledge base of these ITSs have been developed using numerous methods. Some of the better known methods are model tracing [2], Constraint Based Modeling [3] and the PROUST system [4]. Although many other ITSs have been developed to teach programming, none of them are in widespread use. Therefore, it is obvious that more research needs to be carried out in this area.

This research aims to address some of these difficulties by creating a knowledge base that is capable of accepting alternative solutions to a given programming exercise. It attempts to manage the flexibility that is available to students when they write programs in traditional programming languages, by handling variations of order and logical equivalence. The proposed knowledge base is also capable of addressing different algorithms to a certain extent. For example, it can handle different types of loops although it cannot replace a loop with recursion. As the system is aimed at teaching introductory programming, it does not try to solve all the complexities of programming.

References

1. Halpin, T.A., Morgan, T.: Information modeling and relational databases. Elsevier/Morgan Kaufman Publishers, Burlington, MA (2008)
2. Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R.: Cognitive tutors: Lessons learned. *The Journal of Learning Sciences* 4, 167–207 (1995)
3. Ohlsson, S., Mitrovic, A.: Constraint-based knowledge representation for individualized instruction. *Computer Science and Information Systems* 3, 1–22 (2006)
4. Johnson, W.L., Soloway, E.: PROUST: Knowledge-based program understanding. *IEEE Transactions on Software Engineering* SE-11, 267–275 (1985)

Towards an ITS for Improving Social Problem Solving Skills of ADHD Children

Atefeh Ahmadi Olounabadi and Antonija Mitrovic

University of Canterbury (Intelligent Computer Tutoring Group)

Atefeh.Ahmadi@pg.canterbury.ac.nz,

Tanja.Mitrovic@canterbury.ac.nz

Abstract. The major problem of ADHD (Attention Deficit Hyperactivity Disorder) is the lack of social skill and personal relationships, which leads to peer rejection and society isolation. As the result, they often develop depression and other mental disorders. Effective educational software for ADHD children is of great societal importance as evidenced by the high proportion of this disability in the population (8% to 10% [5]). The main aim of this research is to develop an ITS for ADHD children to teach them social problem-solving skills. The proposed system will enable children to solve everyday problems, which leads to a better life in which there is no peer rejection as well as a strong foundation for their adulthood.

Keywords: ADHD Children, Social Skills, Problem-Solving Skills, Computer-Based Training, Intelligent Tutoring Systems.

1 Introduction

ADHD is a developmental disorder composed of different difficulties with unknown etiology [1]. People with ADHD simply cannot control their behavior. Inattention, hyperactivity and impulsivity are the three symptoms of ADHD [2-3]. ADHD people also have major problems in their relationships with other people around them which might be taken into their adulthood in lack of proper treatment [3]. This disorder has been diagnosed as the most common childhood behavior disorder affecting 8% to 10% of children [4-5]. Both assessment and therapy are needed for this disorder best before the age of seven, as untreated ADHD has significant impact on the child, their immediate family and the whole society [5]. Moreover the probability of performing risky actions like dangerous driving [5-6] or crime commitment [7] is high amongst ADHD adults. Untreated ADHD children have problems in higher education. Their problems in personal relationships, social skills, time management and self-organization lead to society isolation which may lead to depression or other mental problems [8]. So having a way of helping ADHD children to control their disorder, we equipped them with a well-organized foundation for their future.

2 Method

There are three main elements for social skills: social intake, internal processing and social output. Traditional problem-solving strategies do not work well for ADHD

children. The reason is they do not practice the learnt lessons in the real life, so they have short term effects. On the other hand, due to mental disorders, ADHD children learn very hard and forget about lesson learnt easily. Also, new approaches have to be tailored for them to be applicable. Centre of social success in Dallas introduced a method for problem solving called POPS [9]. It is an abbreviation for: Problem, Options, Pick, and Solve or Start again. Applying POPS, children are asked to define a problem. Then they are given some options. They are asked to pick an option and try it. If the chosen option is able to choose the problem, the process ends, otherwise they have to start again. In Social Autopsy, children are asked to give their solution options themselves even if they have an adult's support. ADHD children normally cannot give any justification for their actions, especially the ones who have hyperactivity or impulsivity symptoms. Giving their own solution options is a hard task for them especially when they have to be flexible enough to change it without any help. In my project, I am going to adopt an integration of POPS and Social Autopsy and develop a software system according to this new approach specific to ADHD children.

The first step in designing system is to find out what social skills 8 to 12 years old children should know. The social context is another important factor that has to be considered. After choosing the skill they like to practice, the child will be asked to define the problem context. The problem context is any different places where the child could be during the day and therefore is another important factor that has to be considered. The system will then select a problem with an animated scenario to help children to imagine themselves in the real situation. The child's progress will be tracked and recorded with each session to monitor improvements or difficulties with each task. It also helps in choosing the next appropriate problem for the child. Going through different phases of the system depends on successive scores of the previous phases. The learning process is multi-level and is divided to three phases with increasing level of difficulty in each phase.

Phase 1: System poses a problem to the child. When s/he becomes familiar enough with the question, system will give her/him a list of solution options. The child chooses one option. Then system will ask for a justification for her/his choice with a supporting list of justifications. The system provides feedback for each step in this phase. An example: Imagine the child has selected the "Requesting Help" skill in the context of school yard. A problem could be: "Your mom was supposed to come and collect you after school, but she is late and you are worried. Who is the best person to get help from?" This scenario would be an animated and colorful view and the child can see a figure as a symbol of him/her in that environment. The child has to click on the right object which in this case is the school's principal. If a wrong object was clicked, the system asks for a justification which in this phase is given as a pop down menu.

Phase 2: Once the child has got enough practice and success in stage one, they enter phase 2. In this phase again problems are given to the child, but instead of making options available, s/he has to come up with options themselves. They also have to give justification for each choice.

Phase 3: This phase is an advanced mode which will be open-ended, so that children have to enter not only the solution options, but also their own problem to the system

and go through the social problem solving skills independently like the real life. The system will not provide a lot of feedback in this phase.

Pre-test and post-test are being done by psychologists who measure certain factors using pre-designed standard tests. Additional related factors such as response time, interaction time or correctness rate will be logged so that children's behavior can be studied while they are working with the system. Furthermore, children will work with two versions of the system; a version without feedback, and an adaptive version with feedback. This is to evaluate effectiveness of the training in particular.

The software system should be attractive enough to absorb ADHD child's attention. The object of the displayed scenario will be moved to different places each time, so if the child have a better performance next time when s/he works with the system we can make sure s/he has not memorize the object's place. Therefore we evaluate the child's improvement with more confidence. The proposed system will be developed specifically for ADHD children. Using this system they can become good social problem solvers.

References

1. Parsons, T., Bowerly, T., Buckwalter, J., Rizzo, A.: A Controlled Clinical Comparison of Attention Performance in Children with ADHD in a Virtual Reality Classroom Compared to Standard Neuropsychological Methods. *Child Neuropsychology* 13, 363–381 (2007)
2. Excoffier, E.: What is Child Attention Deficit Hyperactivity Disorder? *Revue Du Praticien* 56(4), 371–378 (2006)
3. Cho, B., Ku, J., Jang, D., Kim, S., Lee, Y., Kim, I., Lee, J., Kim, S.: The Effect of Virtual Reality Cognitive Training for Attention Enhancement. *Cyber Psychology and Behaviour* 5(2) (2002)
4. Slate, S., Meyer, T., Burns, W., Montgomery, D.: Computerized Cognitive Training for Severely Emotionally Disturbed Children with ADHD. *Behavior Modification* 22(3), 415–437 (1998)
5. Anton, R., Opris, D., Dobrea, A., David, D., Rizzo, A.: Virtual Reality in the Rehabilitation of Attention Deficit/Hyperactivity Disorder. *Instrument Construction Principles. Journal of Cognitive and Behavioural Psychotherapies* 9(2), 235–246 (2009)
6. Thompson, A., Molina, B., Pelham, W., Gnagy, E.: Risky Driving in Adolescents and Young Adults with Childhood ADHD. *Journal of Paediatric Psychology* 32(7), 745–759 (2007)
7. Fletcher, J., Wolfe, B.: Long-Term Consequences of Childhood ADHD on Criminal Activities. *Journal of Mental Health Policy* 12(3), 119–138 (2009)
8. Harpin, A.: The Effect of ADHD on the life of an individual, their family and community from Preschool to Adult Life. *Archives of Disease in Childhood* 90(2), I2–I7 (2005)
9. Attention Deficit Disorder Association (ADDA), Empowering ADHD Children to Become Better Social Problem Solvers, <https://www.adda-sr.org/reading/Articles/Istreempowering.html>

A Scenario Based Analysis of E-Collaboration Environments

Raoudha Chebil¹, Wided Lejouad Chaari¹, and Stefano A. Cerri²

¹Laboratory of Optimization Strategies and Intelligent Computing (SOIE)
National School of Computer Studies (ENSI) – Manouba University
Campus de la Manouba, 2010 Manouba, Tunisia
{raoudha.chebil,wided.chaari}@ensi.rnu.tn

²The Montpellier Laboratory of Informatics, Robotics, and Microelectronics (LIRMM),
University of Montpellier 2 & the National Center for Scientific Research
161, Rue Ada - F-34095 Montpellier Cedex 5, France
stefano.cerri@lirmm.fr

Abstract. Collaboration is the basis for conceiving, coordinating and implementing the tasks associated to complex goals. Collaboration is pervasive: there is practically no human challenging domain that is not influenced by collaborative processes, in particular Education in formal and informal settings. For these reasons we have to consider collaboration at a distance as a “new” key phenomenon that deserves to be studied, thus modeled in order as much as possible to foresee its effects. In the global village, synthetically represented by “the Web”, many collaborative contexts exist; each with its properties.

In all these different contexts where the collaboration quality depends on some particular property like the collaboration goal, tasks, and constraints, classical performance evaluation methods are not adequate and cannot be applied directly. In this paper, we discuss a new e-collaboration evaluation approach based on the analysis of scenarios.

Keywords: e-collaboration, performance evaluation, scenario based analysis.

1 Problem Position

Frequently, two or many persons in different ends of the world have common interests and need to collaborate. Such a requirement imposes multiple constraints, but at the same time could produce very interesting results promoting a significant progress in the concerned domain. Thanks to the existing technologies, collaborative work can be encouraged with a minimum of constraints. Actually, a wide range of collaborative platforms is available offering services more and more sophisticated and adapted to all the needs. Despite all this technological wealth in perpetual growth, its exploitation is still limited and slowed by a weak reliability level. The improvement of this situation can't be ensured without the application of largely validated performance evaluation methods permitting to detect and eventually solve the existing problems.

In the literature, there are many works on e-collaboration performance evaluation developing different ideas generally without any validation [3][4]. This explains the

lack of standards for performance evaluation and the frequency of subjective statements on e-collaboration performance. In this paper, we present a scenario based analysis of e-collaboration environments as a first step of a new, hopefully well founded, performance evaluation approach that we are currently studying.

2 E-Collaboration Analysis

At the end of this analytical phase, we wish to obtain the abstractions representing the reality in a significant way. We considered that the best mean to ensure this purpose, is to start from observing real e-collaboration cases like e-learning sessions [1], vote scenarios [2] and virtual meetings of research teams; this way we choose an empirical approach to experimentation. In the following, we present the preliminary conclusions we have obtained.

2.1 Observing Results

The observation showed that despite their diversity, all e-collaboration scenarios are supported by a communication tool permitting to participants to work with each other. The collaborator’s interactions available in any scenario generate interesting knowledge and expertise exchanges responsible of the sub-goal satisfaction and so the accomplishment of the global goal. From this common description of e-collaboration scenarios shown in Figure 1, we can cite the following elements as their most important constituents: collaborators, e-collaboration tool, interactions, sub-goals and global goal. The aspects related to individual exchanges can be considered as the e-collaboration kernel and deserve a more elaborated description as suggested hereafter.

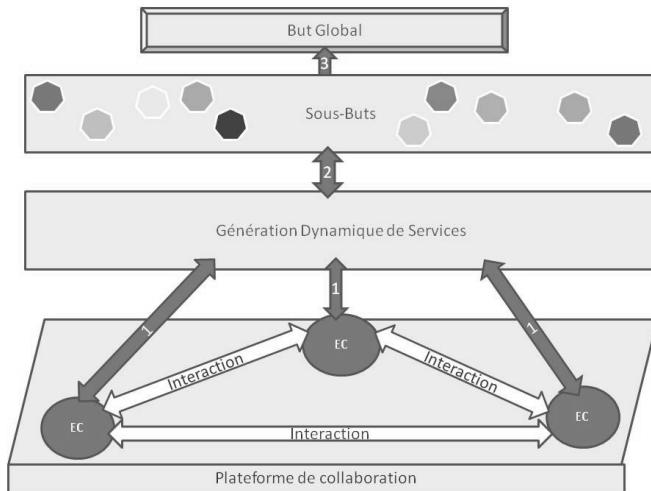


Fig. 1. Formalization of e-collaboration scenarios

In every e-collaboration scenario, exchanges are ensured by series of “communication moves” between the collaborators. In order to communicate with collaborator B, collaborator A needs to interact with his/her computer which needs to interact on its turn with the recipient’s computer. To access to the received information, collaborator B has also to communicate with his/her computer. From this description, three types of interactions [5] can be identified during an e-collaboration session: Computer to Computer Interaction; Collaborator to Computer Interaction; Collaborator to Collaborator Interaction.

The previously described work [5] was based on particular scenarios and provided a general formalization of e-collaboration sessions. This generalization is useful as it will be a starting point for the reusable evaluation method to propose. Once the most important e-collaboration constituents are available, the principal aspects having to be evaluated can be hopefully determined (measured): the platform’s performance, the accomplishment of the global goal in terms of sub goals and the quality of the exchanges.

3 Discussion

The three cited evaluating aspects haven’t the same importance. In fact, thanks to technological progress, e-collaboration platforms performances are continuously improved and nowadays are generally satisfactory. In addition, evaluating goal’s accomplishment is easy to carry out in any e-collaboration scenario. The last point concerning exchanges is estimated to be the most difficult to deal with; so our future analysis will be focused on it.

References

1. Eisenstadt, M., Komzak, J., Cerri, S.A.: Peer conversations for e-learning in the grid. In: 1st International ELeGI Conference on Advanced Technology for Enhanced Learning, Vico Equense, Naples (2005)
2. Business Process Model and Notation, V1.1 (January 2008), Standard document: <http://www.omg.org/spec/BPMN/1.1/PDF>
3. Steves, M., Scholtz, J.: A framework for evaluating collaborative systems in the real world. In: Proceedings of the 38th Annual Hawaii International Conference (2005)
4. Westphal, et al.: Measuring collaboration performance in virtual organizations. In: Establishing The Foundation of Collaborative Networks (2007)
5. Chebil, R., Lejouad Chaari, W., Cerri, S.A.: An E-Collaboration New Vision and Its Effects on Performance Evaluation. International Journal of Computer Information Systems and Industrial Management Application 3, 560–567 (2011)

Supporting Students in the Analysis of Case Studies for Ill-Defined Domains

Mayya Sharipova

University of Saskatchewan, Department of Computer Science

Abstract. Computer science students often must take a professional ethics course, but sometimes find the qualitative nature of such a course to be challenging. To this end, we have built a prototype system called Umka that helps such learners in analyzing the case studies commonly used in this kind of course by : (i) directly critiquing (with various kinds of feedback) the arguments of a learner about issues that arise in a case study; and (ii) supporting collaboration among multiple learners as they discuss these issues. The key technology underlying Umka is the use of latent semantic analysis (LSA) augmented with the structured interface for the "diagnosis" of students' arguments. Umka was tested in a proof-of-concept experiment, in which we assessed the accuracy of the LSA technique in diagnosing a learner's argument, and explored the pedagogical effectiveness of the support provided by Umka for various types of learners. Preliminary conclusions are drawn that are promising, and further experiments are planned in the future. It is the longer term goal of our research to develop techniques that can be used to create tools to support learners in a number of ill-defined educational domains.

Keywords: ethics education, ill-defined domain, latent semantic analysis, supporting learner argumentation and discussion.

Introduction. Rendering support to learners working on domain-specific tasks constitutes a main part of any ITS. An effective support system should be pedagogically effective, helping students to achieve learning goals in a personalized way. This makes the construction of an effective support system a fairly complicated task. For ill-defined domains [1] such as professional ethics, this task is further complicated by the absence of a uniform set of guidelines to solve problems, the non-existence of a single right solution and the reliance on the natural language interaction in solving the problems.

How can learners be effectively supported in ill-defined domains? What types of support are pedagogically effective and for what categories of learners? How can these support types be effectively realized? To begin to answer these questions, at least for the domain of professional ethics for computer science students, we have built a prototype system called Umka, and using the system conducted a small experiment with 23 students.

System's Description. The system's domain knowledge consists of case studies representing some ethical dilemma. For every case study there are possible

ways of resolving the dilemma, and predefined arguments for and against a particular resolution. An argument can be a “good” argument or a misconceived argument. For good arguments the system stores hints attached to them, and for misconceived arguments - challenging questions to correct the misconception.

The mechanism for matching student arguments against system arguments and other students’ arguments is based on Latent Semantic Analysis (LSA), helped by the structure of the interface. LSA has been used in other ITSs including Autotutor [2] for evaluating students answers against predefined system answers. LSA works as the “bag of words” model, and therefore is not very effective in distinguishing negative arguments from positive ones. The interface allows this distinction to be clearly identified by the system, by forcing the student to place his/her argument either into the arguments FOR or arguments AGAINST windows.

Umka provides different support types available on demand by the student. When the student works individually with the system, it gives various kinds of feedback to the arguments of the student: (1) feedback that the student argument is good, when the argument was closely matched with a good system argument; (2) feedback that the argument is original, if the system was not able to find any close match; (3) asking challenging questions for the argument, if the argument was closely matched with a misconceived system argument; (4) providing a counterargument to the student’s argument, if a similar argument was found in the system’s knowledge base but on the opposite side of analysis, for vs against; (5) giving hints on good system arguments that the student had not yet considered in his or her analysis; and (6) giving guidance on the steps of ethical analysis.

Umka also supports collaboration among multiple learners by (1) suggesting that the student consider similar, different, and counterarguments of other students, where “similar” means arguments closely matched by LSA, “different” means arguments of other students that the student had not considered in his analysis, found quite far from his arguments in the LSA semantic space, and “counterargument” means an argument similar to the student argument but on the opposite side of analysis, for vs against; and by (2) showing arguments of all students semantically grouped based on the LSA similarity measure.

Experiment. In the experiment students were given an Intellectual Property case study representing dilemma as to whether or not to make a copy of a copyrighted software for a friend. In the first part of the experiment students worked individually, identifying arguments for and against copying or not copying, and the system was giving feedback on these arguments. In the second part of the experiment students could see and comment on the arguments of other students, and the system was supporting this collaboration.

The goal of our experiment was to answer two questions: (i) how effective is the cross-interaction of LSA and the interface in finding good matches?; and (ii) what types of support are pedagogically effective, and which are preferred by different categories of students? Our findings are:

- The cross-interaction of LSA and the interface was proved to be fairly effective, with an average precision of LSA 0.5 when compared to human expert judgement, which was around 4 times as high as the precision of random matching, and higher than keyword search precision. Moreover, students found 62% of support messages relevant.
- Various support types seem to produce unequal pedagogical effects. Thus, students found it most helpful to see arguments of all students semantically clustered. The most frequently used support type was the system suggestions to consider similar, different or counterarguments of other students. The type of support that affected students most was hints on ideas students hadn't yet considered.
- We also discovered that different categories of students indeed preferred different support types. Thus, male students were more responsive to counterarguments than female students, while females used guidance about ethical analysis more than males. Students who find themselves well-versed in the ethical issues of computing appreciated feedback on a good idea and presentation of all semantically clustered ideas more than their counterparts. And finally, students who hadn't taken an ethics course were more affected by arguments of other students suggested by the system than students who had taken an ethics course.

Conclusion. The results from our first study look promising, and we would like to follow them up with subsequent studies moving to more complex case studies, considering other support types and different domains. Future steps for the research are enhancing LSA with other methods such as textual entailment, work on the automatic expansion of the case library by extracting novel ideas of students not present in the system, and personalization of the environment by adapting to specific features of learners, following up the different behaviour patterns seen in this study. It is the longer term goal of our research to develop techniques that can be used to create tools to support learners in almost any ill-defined educational domain, where argument, qualitative analysis, and interaction are key pedagogical practices.

Acknowledgements. The author wishes to thank the Natural Sciences and Engineering Research Council of Canada for their funding of this research project.

References

1. Lynch, C., Ashley, K., Pinkwart, N., Aleven, V.: Concepts, structures, and goals: Redefining ill-definedness. *International Journal of Artificial Intelligence in Education* 19(3), 253–266 (2009)
2. Graesser, A., Lu, S., Jackson, G., Mitchell, H., Ventura, M., Olney, A., Louwerse, M.: Autotutor: A tutor with dialogue in natural language. *Behavior Research Methods* 36(2), 180–192 (2004)

Using Individualized Feedback and Guided Instruction via a Virtual Human Agent in an Introductory Computer Programming Course

Lorrie Lehmann, Dale-Marie Wilson, and Tiffany Barnes

The University of North Carolina at Charlotte
9201 University City Blvd.
{ljllehman,dwilso1,tiffany.barnes}@uncc.edu

Abstract. Students taking introductory courses in higher learning often hold misconceptions of how well they understand the material they will be tested on. One common phrase from students is, “I know the material, but I just do poorly on the tests.” We propose an automated system to keep the students informed of their progress in how well they understand the knowledge components of a course in a timely manner along with providing customized help via a virtual human agent to increase their performance on tests.

Keywords: meta-cognition, adaptive help, virtual humans.

1 Introduction

In this paper we propose a unique approach to help students in an introductory programming class become more aware of which concepts they need more practice with and offer them a virtual human agent who will give them guided instruction on their specific weak points. It has been shown that meta-cognition skills are important for effective learning [1] and we seek to help students develop these skills while improving their performance in programming.

Over the course of team-teaching six semesters of our “Introduction to Computer Science Course, ITCS 1212”, we have used clicker quizzes to mark attendance in lectures. This semester we are using the results of these quizzes to provide individual feedback to each student through email and individualized practice sessions with a virtual human agent, Dr. Chestr. Our goal is to determine if providing the students with the specific topics they answered incorrectly and pointing them to online help with a virtual human who will guide them in these topics will increase performance on lecture tests.

2 Background

Dr. Chestr, a Computerized Host Encouraging Students to Review, is a virtual human with a game-show host personality that is designed to help students review C++ programming concepts. Dr. Chestr is implemented using Haptek’s People Putty, a

text-to-speech engine and is connected to a MySQL database holding over 300 questions all directly related to topics covered in lecture. Dr. Chestr can run over an Internet connection using various browsers. Dr. Chestr has a voice component and is able to read the questions and provide verbal feedback to student responses. He is programmed to have a playful yet intelligent personality. Student progress is recorded in the database. Dr. Chestr was used in a pilot program at The University of North Carolina, where participation was voluntary, to study students' reactions to the virtual game show host personality. The study reported a high degree of usability [2]. Since then we have added more questions to the database and are now incorporating student clicker results with the use of the agent. We have used clicker quizzes in the past semesters for attendance purposes. Students did receive feedback at the time of the quiz, but there was no follow up and it was evident that many students did not put much importance on the correctness of their responses.

3 Proposed Study

Each lecture session includes a clicker quiz comprised of four to six questions. The results of the quizzes are read into a script and email is generated weekly and sent to each student stating which topics they missed on the quiz and how they did relative to the other students in the class. One question on each quiz asks the students to rate their understanding of the concepts. Some quizzes cover previous lecture material and some cover the material covered in the current lecture. The students are directed to the Dr. Chestr link and the virtual human offers questions on the missed topics. In addition, the student may practice with any other course topics. Time spent using the agent and student scores are stored in the database.

We predict that the individual timely feedback will provide students with a more realistic appraisal of how much they are understanding and will prompt many of them to seek help using the virtual human at their convenience and to participate more in lecture as the semester progresses. We will be able to measure changes in lecture test results from the same semester last spring where we had the same distribution of students (majors vs. non-majors), approximately the same class size of 320 students and the same instructors to determine what effects the use of individualized feedback and the virtual human have on test performance. The virtual human will provide a non-threatening agent for practice that students can use anytime using a web browser. In a user study last year [2] students expressed a general like for the virtual character's personality. We will log how often each student uses the virtual tutor and which questions the student chooses for practice. We also hope to help students become more effective learners by becoming more aware of their misconceptions as the semester progresses, and to become more aware during lecture what concepts are confusing.

4 Future Work

Once this system is fully implemented we would like to use it to provide individual study guides to students to prepare for each test and the final exam. We currently provide general study sessions, which provide a review of all the major topics. The

individual study guides would be built using the results of clicker quizzes along with the student's responses on previous tests. The guides would be delivered via the virtual human when the student logs into the system.

References

1. Wagster, J., Tan, J., Biswas, G., Schwartz, D.: How Metacognitive Feedback Affects Behavior in Learning and Transfer. In: Workshop on Metacognition and Self-Regulated Learning, International Conference on Artificial Intelligence in Education (2007)
2. Wilson, D., Sakpal, R.: Dr. C.H.E.S.T.R.: Computerized Host Encouraging Students to Review. In: Computer Games, Multimedia and Allied Technology 2009 Conference, Singapore, Japan (2010)

Data-Driven Method for Assessing Skill-Opportunity Recognition in Open Procedural Problem Solving Environments

Michael John Eagle and Tiffany Barnes

The University of North Carolina at Charlotte
9201 University City Blvd. Charlotte, NC 28213
{mjeagle,tiffany.barnes}@uncc.edu

Abstract. Our research goal is to use data-driven methods to generate the basic functionalities of intelligent tutoring systems. In open procedural problem solving environments, the tutor gives users a goal with little to no restrictions on how to reach it. Knowledge components refer to not only skill application, but also applicable skill-opportunity recognition. Syntax and logic errors further confound the results with ambiguity in error detection. In this work, we present a domain independent method of assessing skill-opportunity recognition. The results of this method can be used to provide automatic feedback to users as well as to assess users problem solving abilities.

Keywords: Educational Data Mining, Interaction Network.

1 Introduction

To generate knowledge components in open procedural problems we must first address the assumptions of the Bayesian knowledge-tracing model[2]. First, we must be able to address each interaction as correct or incorrect. Second, we must be able to assign to each interaction a single knowledge component. For open procedural problems, both of these assumptions are challenging. As each interaction represents a step towards a goal, it is difficult to address the correctness of an individual step. While errors in the application of actions can be easily marked, errors in obtaining the correct solution require special attention.

The next challenge is the classification of each interaction to individual knowledge components. The open nature of the environment makes it possible for each interaction to provide opportunities to apply several skills. Furthermore, the skills needed for an interaction include action-application, action-opportunity recognition, and problem-solving skill. We can assess the action-application knowledge components by using legal/illegal action application attempts. Previous work in [1] generated automated feedback from student data; this work is extended here by the addition of automated generation of knowledge components, as well as the addition of other heuristics for suggesting next steps. We

introduce interaction networks, a data structure generated on previously collected tutor-data, as well as metrics and algorithms which and be performed on the structure to generate knowledge tracing and hint feedback.

2 Interaction Network

We model a solution attempt as a path graph of states (vertices) and actions (edges). We use *case* to refer to individual students, as well as student specific information. We create the interaction network for a problem by conjoining the set of all the path graphs. We use *state* to describe the state of the software environment, representing enough information so the program's state could be regenerated in the interface. We use *actions* to describe user interactions and their relevant parameters. We also store the set of all cases who visited any particular state-vertex or action-edge, allowing us to count frequencies and connect case specific information to the interaction network representation. This representation results in a connected, directed, labeled multi-graph with states as vertexes, directed action edges to connect the states, and cases that provide additional information about states and edges.

To build the interaction network for a problem we combine the interaction sequences, or solution attempts, from each case into one network. States are combined when they are considered equal. In different tutors and interfaces, two states could be considered equal as long as the screen looks the same, or all the same actions have been performed, regardless of order, but in other cases, states arrived at by taking the same actions in a different order could be considered distinct. Frequency information, as well as information about which cases have visited, is embedded into the edges and vertexes. This results in a network which represents the interactions of a large number of users in a relatively small space.

As actions are responsible for the state transitions, it is reasonable to use these labels to denote the skill needed for the interaction, as a starting point. For any state in the interaction network there are multiple out-edge actions, as well as multiple successor states. We use metrics generated from this graph in order to address action-opportunity recognition and problem-solving knowledge components; we can also use these as a way to assess correctness in interactions.

2.1 Methods

New interactions can be evaluated by using an interaction network built on previously collected data. On each interaction, we look up the state, action, and resulting-state information in the interaction network. We then update the student model. For each student interaction we update the model as follows:

```

if (action is legal) {
  actionKC is updated as correct
  for each(action otherAction in state actions)
    if(otherAction.value > currentAction) {
      otherAction is updated as incorrect    } }

```

Selecting an action shows evidence that the user can recognize that a action is applicable. However, selecting a non-optimal action is evidence that the user did not recognize that another action was applicable to the problem state. This poses the challenge of defining optimality for the actions; which is made more difficult by our goal of domain independence. We have identified several potential metrics for assessing the value of actions: Shortest Path; Fastest real-world time path; Retention path (did not dropout); and Error avoidance path.

2.2 Example Case: The Deep thought Tutor

In order to assess these metrics we used data from Deep Thought, a propositional logic tutor in which students are tasked with performing first-order logic proofs [3]. Students are given a set of premises and a desired conclusion; the student must then use basic logic axioms to prove the conclusion. As the student works through the proof, the tutor records each interaction. We model the application of axioms as the actions. We model the state of the logic tutor as the conjoined set of each premise and derived proposition.

For example a student starts at state $A \vee D, A \rightarrow (B \wedge C), \neg D \wedge E$, where each premise is separated by a comma. The student performs the interaction $SIMP(\neg D \wedge E)$, applying the simplification rule of logic to the premise $\neg D \wedge E$ and derives $\neg D$. This leads to the resulting-state of $A \vee D, A \rightarrow (B \wedge C), \neg D \wedge E, \neg D$. Errors are actions performed by students that are illegal operations of logic and the tutor this results in a loop. For example: The student is in state $A \vee D, A \rightarrow (B \wedge C), \neg D \wedge E, \neg D$. The student performs the interaction $SIMP(A \vee D)$ in an attempt to derive A . The resulting-state would remain $A \vee D, A \rightarrow (B \wedge C), \neg D \wedge E, \neg D$, the log-file would mark this edge as an error.

The results were promising, considering that the method requires no domain specific knowledge. When comparing next best steps with the preexisting MDP method of hint generation [1], the suggested next step overlapped around 85% of the time. While further study is needed to determine the differences between the suggested hints, this result provides some measure of convergent validity. Qualitative analysis of the knowledge components showed moderate success, with the knowledge component values raising as would be expected over steps. In future work we will expand this analysis quantitatively in order to evaluate the model's prediction of student errors.

References

1. Barnes, T., Stamper, J.: Toward Automatic Hint Generation for Logic Proof Tutoring Using Historical Student Data. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 373–382. Springer, Heidelberg (2008)
2. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction* 4, 253–278 (1994)
3. Croy, M.J.: Graphic interface design and deductive proof construction. *J. Comput. Math. Sci. Teach.* 18, 371–385 (1999)

How Do Learners Regulate Their Emotions?

Amber Chauncey Strain¹, Sidney D’Mello², and Melissa Gross¹

¹ Institute for Intelligent Systems, University of Memphis
365 Innovation Drive, Memphis, TN, 38152

² University of Notre Dame, 384 Fitzpatrick Hall, Notre Dame, IN, 46556
{dchuncey@memphis.edu, magross@memphis.edu, sdmello@nd.edu}

Abstract. In an online survey, one hundred and thirteen college students were asked to describe the emotion regulation strategies they frequently use during learning. We found that learners tend to report using certain strategies more frequently than others, and that generally the strategies that are used most often are considered by learners to be the most effective. We discuss the implications of these findings for the development of intelligent tutoring systems that train and scaffold effective strategies to help learners regulate their emotions.

Keywords: emotion regulation, intelligent tutoring systems.

1 Introduction

There is a complex interplay between emotion and cognition during learning and problem solving [1]. Researchers are now testing emotion regulation (ER) strategies to help learners regulate their emotions so they might pursue more positive trajectories of thought and feeling. The present study analysed the types of reappraisal strategies that are commonly used during learning with an eye for implementing a subset of these strategies in next generations ITSs.

2 Method and Results

One hundred and thirteen (N=113) participants from a large public U.S. university were recruited for this experiment. The key online material for this study was an open-ended ER strategy questionnaire. This questionnaire was a six-item measure that provided definitions and examples of emotion regulation strategies that are commonly used in the literature (situation selection/modification, attentional deployment, cognitive change, suppression) [2]. After the description of each strategy was presented, participants were asked to describe a time they used that particular strategy during learning. In particular, participants were prompted to describe the specific way in which they used the strategy, and whether they thought that strategy was effective.

We used a subset of participants’ responses on the open-ended emotion regulation questionnaire to develop a coding scheme to identify the types of reappraisal strategies learners use. The strategies we identified were: *quiet-seeking/stimulation seeking* (seeking out a quiet/stimulating place to study), *self-reward* (providing oneself with rewards for accomplishing goals), *prioritizing* (selecting the order in which to accom-

plish tasks in a way that will minimize negative emotions), *taking a break* (disengaging from the learning task and engaging in a non-academic task), *strategy use* (engaging in a learning strategy that might help minimize negative emotions), *positive/negative rumination* (choosing to attend to positive/negative feelings), *self-talk* (giving oneself a sense of reassurance by talking through the emotion), *value focus* (thinking about the personal value of the task), *role play* (imagining or acting out a particular role other than the role of a student or learner), and *making a game* (making a game of the learning task so that it has elements of fun or competition).

After the coding scheme was developed, two trained coders independently coded each response for the type of reappraisal strategy used, and obtained an inter-rater agreement of 97%. Results indicated that quiet seeking was the most frequently used ER strategy, along with taking a break, positive and negative rumination, and making a game. Interestingly, we also found that with the exception of negative rumination, learners reported that each of the most frequently used ER strategies were also the most effective, indicating that learners are perhaps metacognitively aware of which strategies are the most beneficial and tend to engage more frequently in those strategies.

3 Discussion

While more research in this area is certainly needed, our study serves as an initial point towards gaining knowledge about the types of reappraisal strategies that are used in real learning contexts. The next step is to implement a subset of these strategies in ITSs and other advanced learning technologies.

Acknowledgments. This research was supported by the NSF (ITR 0325428, HCC 0834847, DRL 1108845). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

References

1. Linninbrink, L.A.: The role of affect in student learning: A multi-dimensional approach to considering the interaction of affect, motivation, and engagement. In: Schutz, P.A., Pekrun, R. (eds.) *Emotion in Education*, Amsterdam, pp. 13–36 (2007)
2. Gross, J.: Emotion regulation. In: Lewis, M., Haviland-Jones, J., Barrett, L. (eds.) *Handbook of Emotions*, 3rd edn., pp. 497–512. Guilford, New York (2008)

A Model-Building Learning Environment with Explanatory Feedback to Erroneous Models

Tomoya Horiguchi¹, Tsukasa Hirashima², and Kenneth D. Forbus³

¹ Graduate School of Maritime Sciences, Kobe University, Japan
horiguti@maritime.kobe-u.ac.jp

² Department of Information Engineering, Hiroshima University, Japan
tsukasa@isl.hiroshima-u.ac.jp

³ Qualitative Reasoning Group, Department of Electrical Engineering and Computer Science,
Northwestern University, USA
forbus@northwestern.edu

Abstract. Many model-building learning environments (MBEs) have been developed to support students in acquiring the ability to build appropriate models of physical systems. However, they can't explain how the simulated behavior of an erroneous model is unnatural. Additionally, they can't create any feedback when the model is unsolvable. We introduce a MBE which overcomes these problems with two technical ideas: (1) *robust simulator* which analyzes the consistency of a model and relaxes some constraints if necessary, and (2) *semantics of constraints* which is a systematic description of physical meanings of constraints and provides heuristics for explaining the behavioral unnaturalness.

Keywords. model-building learning environment, qualitative reasoning, error-awareness/correction, robust simulator, semantics of constraints.

1 Introduction

Many model-building learning environments (MBEs) have been developed to support students in acquiring the ability to build the appropriate model of physical systems [1,2,4]. In MBEs, students build a model by combining (GUI-based) components. Then the behavior of their model is simulated to give feedback to the students. However, the feedback given by these systems is insufficient when students build an erroneous model because they can't explain how the simulated behavior is unnatural nor how to correct the error. Additionally, when a model includes inconsistent constraints, these systems can't create feedback themselves. Helping students identify and correct the errors in their models is necessary because it is a difficult task for them (and sometimes even for teachers).

2 The Method

In order to solve these problems, we use the framework of *Error-based Simulation* (EBS) [5,6]. In the EBS framework, if an erroneous model is unsolvable, simulation occurs by relaxing the constraint(s) responsible for the inconsistency. The

unnaturalness of the behavior is judged by identifying what correct constraint(s) it violates or relaxes (i.e., how the behavior differs from the correct one). Using this framework, we developed a MBE which can create appropriate feedback for students' erroneous models. It has two technical features. (1) The *robust simulator* (RSIM) can analyze the consistency of a model represented by qualitative differential equations and inequalities, and relaxes some constraints if necessary. It is implemented by using LTRE, a Logic-based Truth Maintenance System (LTMS) coupled to a forward-chaining Rule Engine [3]. (2) The *semantics of constraints* (SOC) is a systematic description of physical meanings of constraints that provides heuristics for explaining the behavioral unnaturalness. It is a hierarchy of *constraint classes* (CC), each of which stands for a role of constraints in modeling physical systems.

Figure 1a shows an example of a model built in our MBE prototype. It represents the qualitative relation between the amounts of water in two containers and the flow rate of water through a pipe connecting them at their bottom (figure 1b). Because it is over-constrained, the RSIM tries to relax some constraint(s). According to SOC, it finds that the constraint 'the total amount of water is conserved' is the most fundamental and produces the most unnatural behavior if relaxed. Therefore, it relaxes the constraint to produce the most 'motivating' simulation, and explains what the behavior means. Figure 1c is an image of the simulation showing the total amount of water unnaturally increases.

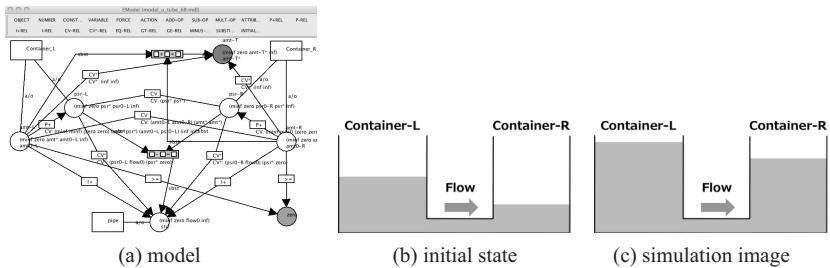


Fig. 1. Model of two containers (erroneous)

References

1. Biswas, G., Schwartz, D., Bransford, J.: Technology Support for Complex Problem Solving - From SAD Environments to AI. In: Forbus, K.D., Feltovich, P.J. (eds.) Smart Machines in Education, pp. 72–97. AAAI Press (2001)
2. Bredeweg, B., Linnebank, F., Bouwer, A., Liem, J.: Garp3 — Workbench for qualitative modelling and simulation. Ecological Informatics 4(5-6), 263–281 (2009)
3. Forbus, K.D., de Kleer, J.: Building Problem Solvers. MIT Press (1993)
4. Forbus, K.D., Carney, K., Sherin, B., Ureel, L.: Qualitative modeling for middle-school students. In: Proc. of QR 2004 (2004)
5. Hirashima, T., Horiguchi, T., Kashihara, A., Toyoda, J.: Error-Based Simulation for Error-Visualization and Its Management. Int. J. of Artificial Intelligence in Education 9(1-2), 17–31 (1998)
6. Hirashima, T., Imai, I., Horiguchi, T., Toumoto, T.: Error-Based Simulation to Promote Awareness of Errors in Elementary Mechanics and Its Evaluation. In: Proc. of AIED 2009, pp. 409-416 (2009)

An Automatic Comparison between Knowledge Diagnostic Techniques

Sébastien Lallé^{1,2}, Vanda Luengo¹, and Nathalie Guin²

¹Laboratoire informatique de Grenoble (LIG METAH), Université Joseph Fourier,
110 av. de la Chimie, BP 53, 38041 Grenoble cedex 9, France
{sebastien.lalle,vanda.luengo}@imag.fr

²Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205, F-69622, France
Nathalie.guin@liris.univ-lyon1.fr

Abstract. Previous works have pointed out the crucial need for comparison between knowledge diagnostic tools in the field of Intelligent Tutoring Systems (ITS). In this paper, we present an approach to compare knowledge diagnostics. We illustrate our proposition by applying three criteria of comparison for various diagnostic tools in geometry.

Keywords: knowledge diagnostic, student modeling, comparison.

In the field of Intelligent Tutoring Systems (ITS), knowledge diagnostic use data collected from an ITS during interactions with the learner in order to infer the skills mastered or not by the student. Student modeling is a complex task: skills refer to a particular domain (mathematics, medicine...) and so models are often designed for one ITS and some specific tasks. Thus a comparison between two knowledge diagnostic processes is very difficult.

Our work aims at assisting a user who wants to compare existing knowledge diagnostic tools in a particular domain, like benchmarks in computer sciences. Authors such as [1] have pointed out the crucial need for comparison as a way to improve knowledge diagnostic evaluation. We plan to give the same activity traces (i.e. record of all interactions of the student with an ITS in a particular domain) as input to various diagnostic tools, and to evaluate in an automatic way a set of criteria on their outputs, in order to assist the user in his comparison.

1 Approaches

As said previously, student modeling techniques are domain-dependent. Like Wenger [2], we identify two levels: Behavior level, where the student answers are parsed and evaluated by the ITS with respect to the domain, and Knowledge level, where the current state of the knowledge is diagnosed. The knowledge level seems less domain-dependent than the Behavior level, as theoretical and generic models (such as cognitive tutors) are more and more used. If skills are still specific to the domain,

their representation and the way they are diagnosed may be more generic. We then assume that the Behavior level has already been done by the ITS in order to produce enriched traces. We work on the Knowledge level diagnostic, which takes as input these enriched traces.

More precisely, we define a knowledge diagnostic technique as a couple composed of a diagnostic model and a computer tool that implements this model. The diagnostic model is a particular way to represent and infer students' knowledge state (like Knowledge Tracing, constraint-based, bug libraries). These models can be implemented using various computer tools that impact the diagnostic (like Bayesian network, logic, plan recognition). We can now define the comparison as a set of criteria that can be applied on the results of each diagnosis technique.

2 Application of Some Criteria

We realized a prototype comparing some diagnostic techniques applied to the domain of geometry of areas, using data stored in Datashop¹. In Datashop, the Behavior diagnostic is already provided (students' answers are evaluated as correct or incorrect). We implemented five diagnosis techniques; then we applied three examples of criteria using Datashop's data: a) the accuracy of the prediction at time t of the answer of the student at time $t+1$, b) the correlation with a reference (gold-standard) and c) the number of skills diagnosed as mastered/non-mastered for all the students. These criteria allow getting information about the quality of the diagnostics (a and b) and their confidence (c). This work show how an automatic comparison can be performed. Taking for instance three techniques based on Knowledge Tracing, using Hidden-Markov Model, AFM and Fuzzy Logic, the accuracy of the prediction is respectively of 58.8%, 57.8% and 30.3% (in cross-validation). Another technique using Dynamic Bayesian Network gives 35.9%, and a misconceptions-based diagnostic implemented with IF/THEN rules gives 49.2%, for the same traces.

To conclude, we have presented some notions that allow comparing knowledge diagnostic techniques, using criteria of comparison. These criteria can help an ITS designer to evaluate various techniques (which one is the more accurate? which skills are correctly diagnosed?). The criteria are generic and are applied in an automatic way, so that the ITS designer gets immediate results for his/her own ITS, domain and data. As shown above, we managed to apply some criteria on Datashop's traces and their results vary depending on the knowledge diagnostic technique, i.e. diagnostic models and the implementation tools.

References

1. Mitrovic, A., Koedinger, K., Martin, B.: A Comparative Analysis of Cognitive Tutoring and Constraint-Based Modeling. In: Brusilovsky, P., Corbett, A.T., de Rosis, F. (eds.) UM 2003. LNCS, vol. 2702, pp. 313–322. Springer, Heidelberg (2003)
2. Wenger, E.: Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge. Morgan Kaufman Publishers (1987)

¹ <https://pslcdatashop.web.cmu.edu/>

The Interaction Behavior of Agents' Emotional Support and Competency on Learner Outcomes and Perceptions

Heather K. Holden

United States Army Research Laboratory
Learning in Intelligent Tutoring Environments (LITE) Laboratory
Simulation Training and Technology Center, Orlando, FL 32826
heather.k.holden@us.army.mil

Abstract. Pedagogical agents, visual 'tutor' representations embedded within computer-based learning environments, exhibit lifelike appearance, persona, and social characteristics in an attempt to establish an ideal learner-agent relationship. This article reports on a study to assess the impact and interaction behavior of a pedagogical agent's emotional support and competency on learner's self-efficacy, performance, and agent perceptions (i.e., perceived intelligence and trust of the agent).

Keywords: Pedagogical Agents, Virtual Tutors, Intelligent Tutoring Systems.

1 Introduction and Methodology

The intention of an intelligent tutoring system (ITS) is to provide learners with customized, computer-based instruction through the utilization of artificial intelligence resources. Pedagogical agents are often added to the ITS interface to establish a personal relationship and emotional connection with the learner. Thus, the aim of the learner-agent relationship is to emulate the same benefits as the human relationship in one-to-one tutoring as found in Bloom's two-sigma problem [1]. A central component of human one-to-one tutoring as well as general teaching/learning is social interaction. Social interaction builds trust, influences learners' motivation to learn [2], and attributes to learners' cognitive and affective development [3].

A 2x2 mixed-design experiment was created to investigate the impact of the independent variables (i.e., emotional support and competency) on learners' Sudoku Self-Efficacy (SSE), perceptions of the agent's intelligence and trustworthiness, and performance/subjective knowledge acquisition. This study used an adult sample of convenience consisting of 35 volunteers (21 males / 14 females). For the experimental testbed, a learning environment was developed to teach participants how to play the game Sudoku with a pedagogical agent/virtual tutor, Audie, an animated Microsoft Agent that resembles a computer. Participants were randomly assigned to interact with one of four experimental versions of Audie [e.g., Emotionally-Supportive and Competent (ESC), Emotionally-Supportive Only (ESO), Competent Only (CO), and Neither Emotionally-Supportive or Competent (NESC)]. The two hypotheses that were found to be favorably supported are: (**H₁**) Learners who work with an ESO virtual tutor will have higher self-efficacy in a learned task than those

who work with a CO tutor and (**H₂**) Learners who work with an emotionally supportive (i.e., ESO or ESC) tutor will perceive the virtual tutor as more intelligent than it really is.

2 Results and Conclusions

One-way between-groups and repeated measures ANOVA found that there were no significant differences between the agent conditions in regard to learners' Sudoku Self-Efficacy (SSE). However, there was a significant relationship between learners' post-measures of SSE and their perceived trust (PT) in the agent/tutor ($r = .368$, $p = .029$). As expected, participants of the emotional supportive only (ESO) condition reported the highest post-experiment self-efficacy ratings for all conditions, thus supporting **H₁**. Furthermore, the ESO condition was the only group to collectively increase learners' SSE throughout the experiment.

In addition, the agent type had a very large effect on learners' perceived intelligence (PI) of the agent. Subjects who worked with the CO agent reported the highest PI ratings among the experimental groups. A comparison between the ESO and NESC groups were used to test and support **H₂**. Although not statistically significant, subjects of the ESO condition reported higher PI of the agent (approximately 4.0 points higher on average) than the subjects of the NESC condition group. However, the agents in both groups had the same level of intelligence. Interestingly, the ESC agent condition, which combined high emotional support and high competency, had a negative impact on the agent's PI. This is seen with reductions in perception scores throughout the progression of the experiment.

The results of this study provide insight on learner's responses to the interaction behavior between two essential agent characteristics. Ultimately, this study could lead to better methods of manipulating these independent variables for targeted learners and domains. Identifying the optimal degree of an agent's characteristics can (a) maximize learners' trust and acceptance of both the learning environment and pedagogical agent and (b) increase learners' readiness to learn, self-efficacy towards the domain, and the effectiveness of their learning experiences. Future work could utilize this study's findings to investigate how agent characteristics impact learners' trust/acceptance of the intelligent tutoring system (ITS) the agent is embedded within, thereby increasing our understanding of learners' ITS acceptance, expectations and future usage intentions. Future studies can also assess the impact of agent characteristics on learners' real-time and predictive cognitive and affective states.

References

1. Bloom, B.: The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher* 13(6), 4–16 (1984)
2. Baylor, A.: Beyond Butlers: Intelligent Agents as Mentors. *Journal of Educational Computing Research* 22(4), 373–382 (2000)
3. Kim, Y., Baylor, A.: A Social-Cognitive Framework for Pedagogical Agents as Learning Companions. *Educational Technology Research and Development* 54(6), 569–596 (2006)

Accuracy of Tracking Student's Natural Language in Operation ARIES!, A Serious Game for Scientific Methods

Zhiqiang Cai¹, Carol Forsyth¹, Mae-Lynn Germany¹, Arthur Graesser¹,
and Keith Millis²

¹Institute for Intelligent Systems, University of Memphis, Memphis, TN, USA
{zcai, cmfrsyth, mlgerman, graesser}@memphis.edu

²Department of Psychology, Northern Illinois University, Dekalb, IL, USA
kmillis@niu.edu

Abstract. *OperationARIES!* is an ITS that uses natural language conversations in order to teach research methodology to students in a serious game environment. Regular expressions and Latent Semantic Analysis (LSA) are used to evaluate the semantic matches between student contributions, expected good answers and misconceptions. Current implementation of these algorithms yields accuracy comparable to human ratings of student contributions. The performance of LSA can be further perfected by using a domain-specific rather than a generic corpus as a space for interpreting the meaning of the student generated contributions. ARIES can therefore accurately compute the quality of student answers during natural language tutorial conversations.

Keywords: Serious game, natural language processing, latent semantic analysis.

1 Introduction

Operation ARIES! (ARIES for short) is an Intelligent Tutoring System which uses natural language conversations between a human student and two pedagogical agents in order to teach students scientific methodology in a game-like atmosphere. Both Latent Semantic Analysis [LSA, 1] and Regular Expressions[2] are used to accurately compare the semantic overlap between the student's input and pre-defined ideal answers and misconceptions. Regular expressions focus more on key words or phrases whereas LSA attempts to uncover inferential aspects of the meaning of the human input by employing a statistical pattern matching algorithm that captures the meaning of words and world knowledge in a high dimensional semantic space. In order for the serious game to be successful, it is imperative that the system accurately categorizes student input which enables appropriate responses.

2 Analyses

The goal of the analyses conducted is to evaluate and improve the performance of the language processing implemented in ARIES. In a previous study [3], we discovered

that the language processing within ARIES is not only comparable ($r = .667$) but also not significantly different from expert human raters ($r = .686$). In viewing the unique contributions of each algorithm, the regular expressions contributed mostly to this success, thus motivating the researchers to improve the performance of LSA by selecting domain specific corpora based on the ARIES E-book, *The Big Book of Science*. The selection process used an algorithm which assigned a *keyness* value to each word in the E-book based on the relative frequency of occurrences of the word in the documents of the E-book and a reference corpus. In the present analyses, 892 contributions resulting from 21 student's interactions with ARIES were analyzed using an LSA space consisting of documents selected based on the keyness values. The match scores, a value from 0-1 representing the results from LSA, derived from this comparison were then compared to ratings made by two human experts.

The language processing within ARIES was found to optimally perform using a domain-specific corpus rather than a generic corpus. In the first analysis, the match score using corpora selected from Wikipedia selected with keyness values correlated at a significantly higher rate with expert human raters than TASA, ($r = .493$ for Wikipedia, $r = .425$ for TASA, Chi square = 3.369, $p=0.066$). The next goal was to determine whether the additional contribution was due to domain-specificity or Wikipedia itself. Therefore, a random space was generated by randomly selecting documents from Wikipedia. The result of the comparison between the two spaces and the expert ratings showed that the domain-specific space ($r = .493$) slightly outperformed the random space ($r = .483$), but not significantly. In order to probe the difference between the two types of spaces, varying numbers documents (ranging from 500 to 16,000) were extracted from Wikipedia, one sample using *keyness* and the other using random. This analysis led to the discovery that 1000 documents selected using *keyness* values derived from the domain-specific book produced optimal match scores as compared to expert human ratings. Regardless of the number of documents, domain-specific selections always out-performed the generic corpora. These findings have relevance not only to the developers of ARIES but also for other researchers using natural language processing in an ITS.

Acknowledgements. This research was supported by the Institute for Education Sciences, U.S. Department of Education, through Grant R305B070349. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

References

1. Landauer, T., McNamara, D., Dennis, S., Kintsch, W. (eds.): Handbook of Latent Semantic Analysis. Erlbaum, Mahwah (2007)
2. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice-Hall, Upper Saddle Creek (2008)
3. Cai, Z., Graesser, A.C., Forsyth, C., Burkett, C., Millis, K., Wallace, P., Halpern, D., Butler, H.: Trialog in ARIES: User Input Assessment in an Intelligent Tutoring System. In: Proceedings of the 3rd IEEE International Conference on Intelligent Computing and Intelligent Systems, pp. 429–433. IEEE Press, Guangzhou (2011)

Designing the Knowledge Base for a PHP Tutor

Dinesha Weragama and Jim Reye

School of Electrical Engineering & Computer Science
Queensland University of Technology, Brisbane, Australia
d.weragama@qut.edu.au, j.reye@qut.edu.au

Abstract. Programming is a subject that many beginning students find difficult. This paper describes a knowledge base designed for the purpose of analyzing programs written in the PHP web development language. The aim is to use this knowledge base in an Intelligent Tutoring System. The main emphasis is on accepting alternative solutions to a given problem.

Keywords: knowledge base design, Intelligent Tutoring System, program analysis, PHP.

1 Introduction

Programming is a very difficult subject for many beginning students. This paper describes a knowledge base designed to support an Intelligent Tutoring System (ITS) that will teach programming to novices using the PHP web development language.

A computer programming problem very rarely has a unique solution. Table 1 shows an example of two programs to find the maximum of a set of numbers stored in an array. In order to handle this type of situation, the proposed knowledge base should be capable of analyzing different student solutions to a given exercise.

Table 1. Programs illustrating different methods for finding the maximum in an array

Program a	Program b
<pre>\$max=\$marks[1]; for(\$i=2;\$i<=5;\$i++) if(\$marks[\$i]>\$max) \$max=\$marks[\$i];</pre>	<pre>\$maxpos=1; for(\$i=2;\$i<=5;\$i++) if(\$marks[\$i]>\$marks[\$maxpos]) \$maxpos=\$i; \$max=\$marks[\$maxpos];</pre>

2 Knowledge Base Design

The knowledge base in this system has been designed using the concepts of first order logic. Currently, it is capable of handling key aspects of assignment statements, conditional statements, arrays, for loops, functions and HTML form processing.

The formal program specification for each exercise contains a goal which is a combination of predicates. Such a goal specification for the program to find the maximum of an array is given in Fig 1. Once a student submits their answer to an exercise, the knowledge base converts this solution into a set of predicates using rules and actions. The final set of predicates is then compared against the goal to identify any predicates that are missing or are unnecessary. This information can then be used to provide appropriate feedback. The exact method of analysis is quite complex and the limited space in this paper does not allow for a detailed discussion.

Final Goal: $\text{HasValue}(\text{VID}_m, \text{VAL}_m) \wedge \forall j [(1 \leq j \leq 5) \rightarrow$
 $\exists \text{VID}_j, \text{VAL}_j, \text{KID}_j, \text{EID}_j$
 $\{ (\text{HasVariableId}(\text{HasElement}(\text{ARRID}_m, \text{KID}_j), \text{VID}_j)$
 $\wedge \text{HasKeyExpression}(\text{KID}_j, \text{EID}_j) \wedge \text{ValueOf}(\text{EID}_j, j) \wedge \text{HasValue}(\text{VID}_j, \text{VAL}_j)$
 $\wedge \text{LessThanOrEqual}(\text{VAL}_j, \text{VAL}_m) \wedge \text{VAL}_m \in \text{Array}(\text{ARRID}_m) \}]]$

Fig. 1. The goal for finding the maximum

3 Discussion and Related Work

Many ITSs have been built to teach programming. Some popular examples of methods used for modeling the knowledge base are model tracing [1], Constraint Based Modeling [2] and the PROUST system [3]. Although many other ITSs have been built to teach programming, none of them are in widespread use. Therefore, it is obvious that more research needs to be carried out in this area.

One of the main challenges faced is to be able to handle alternative solutions to a given problem. This research aims to create such a knowledge base. The current knowledge base is capable of handling many of the basic programming statements for a PHP program. However, since it is aimed at teaching introductory programming, it does not attempt to solve all the complexities of web programming.

References

1. Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R.: Cognitive tutors: Lessons learned. *The Journal of Learning Sciences* 4, 167–207 (1995)
2. Ohlsson, S., Mitrovic, A.: Constraint-based knowledge representation for individualized instruction. *Computer Science and Information Systems* 3, 1–22 (2006)
3. Johnson, W.L., Soloway, E.: PROUST: Knowledge-based program understanding. *IEEE Transactions on Software Engineering SE-11*, 267–275 (1985)

Domain Specific Knowledge Representation for an Intelligent Tutoring System to Teach Algebraic Reasoning

Miguel Arevalillo-Herráez¹, David Arnau², José Antonio González-Calero³,
and Aladdin Ayesh⁴

¹ Department of Computer Science, University of Valencia, Spain
Miguel.Arevalillo@uv.es

² Department of Didactics of Mathematics, University of Valencia, Spain
David.Arnau@uv.es

³ Department of Mathematics, University of Castilla la Mancha, Spain
Jose.GonzalezCalero@uclm.es

⁴ Faculty of Technology, De Montfort University, UK
aayesh@dmu.ac.uk

Abstract. Translation of word problems into symbolic notation is one of the most challenging steps in learning the algebraic method. This paper describes a domain-specific knowledge representation mechanism to support Intelligent Tutoring Systems (ITS) which focus on this stage of the problem solving process. The description language proposed is based on the concept of a hypergraph and makes it possible to simultaneously a) represent all potential algebraic solutions to a given word problem; b) keep track of the student's actions; c) provide automatic remediation; and d) unequivocally determine the current state of the resolution process. An experimental evaluation with students at a public school supports the use of the ITS in practice.

Keywords: ITS, algebra, knowledge representation, hypergraph.

In solving algebra-word based problems, the stage of translating the problem into algebraic notation is particularly difficult to teach [1-2]. The student's compulsion to calculate and the tendency to use non-algebraic solving paths have been identified as major factors that deflect students away from the algebraic method [1]. In this work, we have implemented an Intelligent Tutoring System (ITS) that focuses on the translation stage of the problem solving process. The ITS uses a domain-specific knowledge representation mechanism which makes it possible to represent all potential solutions to a word problem, without making any assumption on the resolution path that student may follow in the resolution process.

Algebraic knowledge on a word problem can easily be represented as a function of known quantities, unknown quantities and relations between them. In his work, Fridman [3] observes that the structure of the solution to a given word problem can be expressed as a set of interconnected ternary relations such that there is at least one unknown element in each; and relations are linked between them by at least one

unknown quantity. The knowledge representation mechanism used in the ITS presented in this paper uses trinomial graphs to represent the structure of the solution, and extends Fridman's notation by using directed edges to identify quantities at the left side of the relations. This representation is used to determine all valid student inputs at a given instant in time, and hence to judge on the correctness of any particular input. The reasoning engine allows the student to take any valid path that yields a correct solution, without imposing any restrictions on neither the number of symbols/equations used nor the order of the actions taken to translate the problem into symbolic notation. As in constraint based systems [4], no system intervention occurs unless a definite incorrect input is processed by the engine. When this happens, the student's incorrect input is stored for final reporting purposes. In addition, the system supports multiple readings for the same problem, by maintaining multiple concurrent instances of the knowledge base. The same reasoning engine has been used to build a problem solver. This module is able to automatically work out a solution to a word problem from the corresponding trinomial graph by using a deterministic and systematic approach.

The Graphical User Interface has been carefully designed to facilitate learning of the algebraic approach to problem solving, focusing on the translation of the problem statement into symbolic notation. Quantities are presented as elements which are required to define relations, and relations can only be defined by using elements which already exist. In this way, quantities need first be defined before they are used as part of a relation, partially forcing an algebraic approach. To implement this restriction, the student is not allowed to type the expressions directly. Instead, these are built by using a calculator-like graphical component that contains a button for each arithmetic operator and one more for each quantity which has already been defined.

Results of an evaluation performed at a public school in Spain using a control and an experimental group show positive effects on the group of student who used it; and support the argument that the use of more specific representations that exploit domain particularities may result in additional benefits to ITS in certain domains.

Acknowledgements. This work has been funded by the Spanish government through projects TIN2011-29221-C03-02 and EDU2009-10599 and the University of Valencia, through projects 79/FO11/31 and 107/DT11/34 from the Vicerektorat de Cultura, Igualtat i Planificació.

References

1. Stacey, K., MacGregor, M.: Learning the Algebraic Method of Solving Problems. *The Journal of Mathematical Behavior* 18(2), 149–167 (1999)
2. Croteau, E.A., Heffernan, N.T., Koedinger, K.R.: Why Are Algebra Word Problems Difficult? Using Tutorial Log Files and the Power Law of Learning to Select the Best Fitting Cognitive Model. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) *ITS 2004*. LNCS, vol. 3220, pp. 240–250. Springer, Heidelberg (2004)
3. Fridman, L.M.: Trinomial graphs (Russian). *Mat. Modeli Povedeniya* 3, 47–53 (1978)
4. Mitrovic, A., Martin, B., Suraweera, P.: Intelligent tutors for all: the constraint based approach. *IEEE Intelligent Systems* 22(4), 38–45 (2007)

Exploring the Potential of Tabletops for Collaborative Learning

Michael Schubert¹, Sébastien George², and Audrey Serna²

¹ Faculty of Psychologie, University of Tuebingen, Tuebingen, Germany
michael.schubert@uni-tuebingen.de

² Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France
{sebastien.george, audrey.serna}@insa-lyon.fr

Abstract. Digital tabletops, with their multi-touch surfaces, are innovation that could provide new opportunities for learning. They could support rich interactive collaborative activities while maintaining natural face-to-face communication. Nevertheless, we still know little about the potential learning outcomes and the situations they favor. We have explored how group activities on tabletops could encourage collaboration on the same tabletop but also between multiple tabletops. In particular, we focus on the potential of tabletops to favor learning during a brainstorming activity.

Keywords: Collaborative and Group Learning, Tabletop, Brainstorming, CSCL.

1 Introduction and Research Issues

Many different tabletops have been developed in the last few years to study either the user interface and applications [1] or different facets of the technological possibilities [2] bypassing the “one-user/one computer” paradigm. In our work, we focus on a collaborative situation involving several interconnected tabletops, while at the same time we keep the possibility of collaboration on each tabletop through multi-touch support. Subsequently, two levels of collaboration will be established: 1) collaboration through tabletops, and 2) collaboration at tabletops. One of the final issues of this work is to assess if this type of CSCL (Computer-Supported Collaborative Learning) environment reveal advantages in comparison to traditional CSCL environments.

We present a first pilot study to verify if using an adapted brainstorming tool in a multiple tabletops CSCL environment can outperform a traditional paper-pencil version in performing the same task: learning the procedure of a brainstorming, while performing a brainstorming. To achieve this goal, we suggest that the brainstorming tool have to be adapted not only to the task and to the phase in the learning session, but also to the specific learning environment (CSCL vs. traditional). We are not in favor of solely transposing a paper-version onto the tabletops, but we really want to improve and to take advantage of the inherent technological possibilities.

2 Application Design to Support a Brainstorming Activity

Just as during a paper-pencil brainstorming learning session, Post-its and a whiteboard were used. We tried to represent and transfer all functionalities of these tools onto our application on the tabletops. The goal was to keep the appearance and interactions as simple and intuitive as possible. The whiteboard was a large white screen, giving the possibility for the students to add their own Post-its onto the board. The application was developed to support the collaboration of two students on the same tabletop (*collaboration at tabletops*). The question of “territoriality” for respectively private and shared spaces has been carefully considered (as discussed in [3]). Hence, we propose an artificial space-switcher, so that each of the two users on a table could decide whether they prefer to work in their own private space or to work in a shared local space.

Another benefit lies in the global collaboration support (*collaboration through tabletops*). The whiteboard acted as a global shared space where every action on one whiteboard (one tabletop) was immediately reflected on all other whiteboards (other tabletops).

3 Evaluation and Conclusion

A comparative study showed that the collaborative learning activity might be increased using tabletops [4]. Even if we still have to prove the advantages of multiple tabletops to classic CSCL environments in general, it seems that tabletop applications could be a good means for the learners to reflect on their actions and thereby to favor the knowledge transfer. This interesting point should be tested in a broader context.

Acknowledgements. This research is undertaken within the framework of the SEGAREM (SERious GAMES and Mixed Reality) project. The authors wish to thank both the DGCIS for the fund and the partners of this project (from LIRIS lab, Symetrix and Total Immersion) for their collaboration.

References

1. Streitz, N.A., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., Steinmetz, R.: i-LAND: an interactive landscape for creativity and innovation. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI is the Limit, pp. 120–127. ACM, New York (1999)
2. Shen, C., Everitt, K., Ryall, K.: UbiTable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864, pp. 281–288. Springer, Heidelberg (2003)
3. Scott, S.D., Grant, K.D., Mandryk, R.L.: System guidelines for co-located, collaborative work on a tabletop display. In: Proceedings of the Eighth Conference on European Conference on Computer Supported Cooperative Work, pp. 159–178. Kluwer Academic Pub., Norwell (2003)
4. Schubert, M., Serna, A., George, S.: Using collaborative activities on tabletops to enhance learning and knowledge transfer. In: Proceedings of the 12th IEEE International Conference on Advanced Learning Technologies, July 4–6, Rome, Italy (to be published, 2012)

Modeling the Affective States of Students Using SQL-Tutor

Thea Faye G. Guia¹, Ma. Mercedes T. Rodrigo¹, Michelle Marie C. Dagami¹,
Jessica O. Sugay¹, Francis Jan P. Macam¹, and Antonija Mitrovic²

¹ Ateneo Laboratory for the Learning Sciences
Department of Information Systems and Computer Science, Ateneo de Manila University
Loyola Heights, Quezon City, Philippines

{theafayeguia, rhyzz_craig_08, f_macam}@yahoo.com,
{mrodrigo, jsugay}@ateneo.edu

² Department of Computer Science and Software Engineering, University of Canterbury
Private Bag 4800, Christchurch, New Zealand
tanja.mitrovic@canterbury.ac.nz

Abstract. We attempted to build models of affect of students using SQL-Tutor. Most exhibited states are engaged concentration, confusion and boredom. Though none correlated with achievement, boredom and frustration persisted. Using linear regression, we arrived at a parsimonious model of boredom.

Keywords: SQL-Tutor, observation, performance, models of affect, boredom.

Constraint-based tutors (CBT) are distinguished from other ITSs by knowledge representation. Others require detailed models while CBTs use constraints to limit this specificity [3]. A constraint identifies feature of correct solutions and specifying implicitly the solutions that violate it as incorrect. SQL-Tutor [2] is a CBT.

1 Methods

74 juniors in 3 sections from Ateneo de Manila University used SQL-Tutor for 60 minutes. Observations were carried out by a team of 4 observers who worked in pairs. One is an assistant instructor who was highly experienced in observations. Others are one undergraduate and two graduate students in training. Each pair observed 10 students per section. Every student was observed for twenty seconds. If two distinct states are seen, only the first was coded. Cohen's $\kappa=0.91$ which is considered to be a high level of agreement.

Learning science researches used features as indicators of learning. Learning indicators for SQL-Tutor that were based on these studies are: *SolvedProblems*, *AttemptedProblems*, *LearnedConstraints*, *ConstraintsUsed*, *SeenMessages*, *NumOfLogins*, *TotalTime*, *AvgTimeToSolve*, *TotalAttempts* and *AvgNumOfAttemptsPerSolvedProb*.

2 Results and Discussion

Engaged concentration (57.9%) was most common affect. Confusion (23.9%) and boredom (8.1%) followed. When correlated with achievement, none was significant. Using L [1], boredom persisted ($L=0.11$, $t(33)=2.3$, $p=0.03$). Frustration persisted marginally significant ($L=0.22$, $t(12)=2.18$, $p=0.05$). In linear regression models of two states, only boredom ($r=0.647$; $p<0.001$) was significant. It had -14.27 BiC' [4].

Table 1. Incidence of affective states and correlation with achievement

Affective state	Incidence	Correlation with achievement
Boredom	8.1%	-0.021
Confusion	23.9%	-0.006
Delight	4.1%	-0.320
Engaged concentration	57.9%	0.073
Frustration	2.1%	0.152
Neutral	3.9%	-0.262

Table 2. Model of boredom within SQL-Tutor

MODEL	r	p	BiC'
Boredom = $-0.002 * SeenMessages +$ $-0.002 * TotalTime +$ $0.031 * AvgTimeToSolve +$ $0.007 * TotalAttempts +$ -0.068	0.647	< 0.001	-14.27

3 Conclusion

We attempted to build models of affect of students using SQL-Tutor. Most exhibited states are engaged concentration, confusion and boredom. Though none correlated with achievement, boredom and frustration persisted. We built models of both states but only boredom was significant. Boredom can be predicted by amount of feedback received, total interaction time, average time per solved problem and total attempts.

References

1. D'Mello, S., Taylor, R.S., Graesser, A.: Monitoring affective trajectories during complex learning. In: Proc. 9th Annual Meeting of the Cognitive Science Study, pp. 203–208 (2007)
2. Mitrovic, A.: Learning SQL with a computerized tutor. In: Proc. 29th SIGCSE Technical Symposium on Computer Science Education, pp. 307–311 (1998)
3. Mitrovic, A., Ohlsson, S.: Evaluation of a constraint-based tutor for a database language. *Artificial Intelligence in Education* 10, 238–256 (1999)
4. Raftery, A.E.: Bayesian model selection in social research. *Sociological Methodology* 25, 111–163 (2003)

A Cross-Cultural Comparison of Effective Help-Seeking Behavior among Students Using an ITS for Math

Jose Carlo A. Soriano¹, Ma. Mercedes T. Rodrigo¹, Ryan S.J.D. Baker², Amy Ogan³,
Erin Walker⁴, Maynor Jimenez Castro⁵, Ryan Genato², Samantha Fontaine²,
and Ricardo Belmontez²

¹ Ateneo de Manila University, Loyola Heights, Quezon City, Philippines

² Worcester Polytechnic Institute, Worcester, MA

³ Carnegie Mellon University, Pittsburgh, PA

⁴ Arizona State University, Tempe, AZ

⁵ Universidad de Costa Rica, San Pedro, Costa Rica

josecarlosoriano@yahoo.com, mrodrigo@ateneo.edu,
rskbaker@wpi.edu, aeo@andrew.cmu.edu, erin.a.walker@asu.edu,
maynorj@gmail.com, rgenato@pi.edu, samanthajo@wpi.edu,
ricardobelmontez@wpi.edu

Abstract. We use educational data mining to arrive at models of help-seeking behaviors associated with learning from datasets from three countries: Costa Rica, the Philippines, and the USA. The models were then tested on each country's data to find out how effective help-seeking behavior varies across countries. This study found that models of effective help-seeking are not necessarily transferrable across specific pairs of cultures.

Keywords: Help-seeking, Cross-Cultural, Cognitive Tutors, Scatterplot.

Our objective was to find out whether effective help-seeking behavior is similar across cultures, as this would have implications on future efforts to develop meta-cognitive tutors, or tutors that try to incorporate tutoring effective help-seeking behavior. We do this by generating models of effective help-seeking for three countries and comparing them across cultures. This study made use of data collected from prior studies in Costa Rica [4], Philippines [5], and the USA. In these studies, data were extracted from logs produced by an ITS for generating and interpreting scatterplots [2]. From the scatterplot tutor logs, 17 help-seeking features were distilled. As in [1, 3], the features consisted of the frequency of semantic behaviors across all tutor use. We modeled effective help-seeking behavior by finding a set of related behaviors that led to the most learning for each country, and then created an additional 'universal' model from the combined data sets of the three countries. We quantified learning as student learning gains, as measured through a pre-test and post-test (e.g. post – pre). Our process for creating models of effective help-seeking for each culture involved several steps, very similar to that in [3]: feature engineering, feature selection, feature optimization, model creation, and model evaluation. In model evaluation, we tested each country's models to the data sets of the other countries, and got the correlation of the actual learning and the predicted learning.

The Philippine and USA models performed well on each other's data sets ($r=0.146$, 0.228 respectively). Interestingly, other automated detectors have been shown to generalize between students in the US and Philippines, for example a detector of carelessness in [5]. However, the Philippine, USA, and the universal model did not perform well for data from Costa Rica ($r=0.004$, -0.085 , -0.073 respectively). The collaborative behaviors seen in [4] for Costa Rican students may explain the difference in help-seeking behavior, as a more collaborative environment may make other students the main source of help while studying with ITSs, while only specific types of help that are not available from other students will be sought in the ITS.

In conclusion, we found that help-seeking behaviors do not necessarily transfer across specific pairs of countries. This exposes the possibility that the help-seeking model used by a meta-cognitive tutor may be effective in one culture but not in others. Hence, future work will be needed to determine how to develop models that can be used world-wide, perhaps involving data from a wide range of countries, or intelligent tutors adapting to help-seeking behaviors will need to have their models re-fit for the countries where they are used.

References

1. Alevan, V., McLaren, B.M., Roll, I., Koedinger, K.R.: Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence in Education* 16(2), 101–128 (2006)
2. Baker, R.S.J.d., Corbett, A.T., Koedinger, K.R., Evenson, S., Roll, I., Wagner, A.Z., Naim, M., Raspat, J., Baker, D.J., Beck, J.E.: Adapting to When Students Game an Intelligent Tutoring System. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006*. LNCS, vol. 4053, pp. 392–401. Springer, Heidelberg (2006)
3. Baker, R.S.J.d., Gowda, S.M., Corbett, A.T.: Automatically detecting a student's preparation for future learning: Help use is key. In: *Proceedings of the 4th International Conference on Educational Data Mining*, pp. 179–188 (2011)
4. Ogan, A., Walker, E., Baker, R., Rebolledo, G., Jimenez-Castro, M.: Collaboration in Cognitive Tutor Use in Latin America: Field Study and Design Recommendations. To appear in: *Proceedings ACM Computer-Human Interaction Conference* (2012)
5. San Pedro, M.O.C.Z., Baker, R.S.J.d., Rodrigo, M.M.T.: Detecting Carelessness through Contextual Estimation of Slip Probabilities among Students Using an Intelligent Tutor for Mathematics. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011*. LNCS, vol. 6738, pp. 304–311. Springer, Heidelberg (2011)

Emotions during Writing on Topics That Align or Misalign with Personal Beliefs

Caitlin Mills and Sidney D’Mello

University of Notre Dame, Notre Dame, Indiana, 46556
{cmills4, sdmello}@nd.edu

Abstract. We conducted a study where 42 participants wrote two essays on opposing stances about abortion (pro-choice and pro-life). Participants’ affective states were tracked at 15-second intervals via a retrospective affect judgment protocol. The results indicated participants experienced more boredom when writing essays that did not align with their positions on abortion, but were more engaged when there was alignment. Participants also reported more curiosity while writing pro-choice essays. Importantly, boredom, boredom, engagement, and curiosity were the affective states that predicted essay quality.

Keywords: affect, writing, cognition, boredom, engagement, ITSs.

1 Introduction

Intelligent tutoring system (ITS) researchers have developed effective educational technologies to improve writing skills and proficiency [1-2]. However, the focus of these systems is on the cognitive and motivational aspects of writing, at the expense of the emotional aspects of the writing process. Although considerable research has focused on understanding the role of emotions in learning, there is little research investigating the emotion-cognition link within the context of writing. To investigate this gap in the literature, the present focus was on uncovering how emotions are influenced by writers’ positions on the topic of a written assignment. More specifically, how does the alignment or misalignment between personal beliefs and assigned essay position impact writers’ emotions and the quality of writing?

2 Methods

The participants were 42 undergraduates from an urban U.S. university who participated for course credit. Participants wrote two essays, one supporting pro-choice and one supporting the pro-life perspective on abortion. Participants provided self-judgments of their affective states (14 affective states plus neutral) immediately after the writing session via a retrospective affect judgment procedure by viewing a video of their face along with the screen capture video of their writing session. Essay quality was scored on a modified version of a standardized rubric similar to the one used for

scoring the SAT [3]. The judge was blind to participants' actual positions on abortion. Reliability ($r = .906$) was obtained in a previous study with similar essays.

3 Results and Discussion

Separate mixed effects binary logistic regression models were constructed for the six most frequent states (anxiety, boredom, engagement, curiosity, confusion, frustration) to investigate whether instructed essay position and actual position on abortion influenced the reported affective states. The results indicated that participants were significantly more likely to experience curiosity when asked to write a pro-choice essay compared to a pro-life essay, *irrespective* of their actual positions on abortion. The *instructed position* \times *actual position* interaction was significant for boredom and engagement, suggesting that the (mis)alignment of instructed position and actual position impacted boredom and engagement levels. Boredom was more likely to occur during misalignment but engagement was higher during alignment of positions.

A mixed effects linear regression model also revealed that boredom, engagement, and curiosity were significant predictors of essay quality. Boredom negatively predicted essay scores ($B = -.118$), whereas engagement/flow ($B = .111$) and curiosity ($B = .152$) positively predicted essay quality.

This paper offers a fine-grained investigation of affect during writing, a topic that is much neglected in the educational, ITS, and writing community. We have shown that (mis)alignment between the instructed position and writer's actual position on abortion impact boredom and engagement, which, along with curiosity, predict writing outcomes in expected directions. An ITS with boredom-alleviation and engagement-inducing capabilities has considerable potential for helping writers develop and increase proficiency.

Acknowledgments. This research was supported by the National Science Foundation (NSF) (ITR 0325428, HCC 0834847, DRL 1108845). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

References

1. Wade-Stein, D., Kintsch, E.: Summary Street: Interactive Computer Support for Writing. *Cognition and Instruction* 22, 333–362 (2004)
2. McNamara, D.S., Raine, R., Roscoe, R., Crossley, S., Jackson, G.T., Dai, J., et al.: The Writing-Pal: Natural language algorithms to support intelligent tutoring on writing strategies. In: McCarthy, P.M., Boonthum, C. (eds.) *Applied Natural Language Processing and Content Analysis: Identification, Investigation, and Resolution*, pp. 298–311. IGI Global, Hershey (2012)
3. McNamara, D.S., Crossley, S.A., McCarthy, P.M.: Linguistic Features of Writing Quality. *Written Communication* 27, 57–86 (2010)

A Multiagent-Based ITS Using Multiple Viewpoints for Propositional Logic

Evandro Costa, Priscylla Silva, Marlos Silva, Emanuele Silva,
and Anderson Santos

Federal University of Alagoas, Computer Institute
Campus A. C. Simões - Av. Lourival Melo Mota, s/n, Maceió - AL - Brazil
{ebc,pmss}@ic.ufal.br, {marlos.tacio,manutunan,andersonfarin}@gmail.com

Abstract. This paper reports on preliminary efforts to develop a multiagent-based Intelligent Tutoring System for teaching propositional logic using two integrated approaches with focus on checking the validation of a given argument. One motivation for this integration comes from the importance to involve students in two complementary viewpoints, permitting students to make connections between the two viewpoints involving different strategies, used in problem solving situations.

Keywords: Multiple Representations, Intelligent Tutoring.

1 Introduction

An important question in ITS research is on how to support multiple viewpoints or even multiple representations on a given domain knowledge. This paper addresses this question, focusing on multiple representations with different viewpoints of propositional logic domain. We have accomplished this knowledge domain modeling through a multiagent system. This paper reports on preliminary efforts to develop a multiagent-based ITS for teaching propositional logic using two integrated approaches with focus on checking the validation of a given argument. One of them is a Natural Deduction system as a proof method and the other is Semantic system by using two methods with and without truth table.

A small number of related studies have been found in the literature. For instances, the works in [2] and in [3] are closely related to the present one. The work proposed by Leana and Yacef [2] provides an interesting and well-experimented intelligent tutoring system for the teaching of logic proof using inference rules. The evaluation of this ITS indicates that the work presents gains for the students, showing an improvement in their performance. It works in just one representation of PL domain with one inference method. By the contrast our approach uses multiple representations with different inference methods.

2 An Overview of the Proposed System

Our system follows the architecture of an ITS based on the conceptual model MATHEMA [1] consists basically of three modules: the society of artificial tutoring agents (SATA), the learner interface and the authoring interface. The

interface provides access to the system through any Web browser. The authoring interface module allows the definition of the course structure and contents. Finally, the SATA consists of a multiagent system where each agent, besides communication and social capabilities, contains a tutoring system module focused on some defined part of the target domain. The fact that the system consists of multiagent society allows the distribution of domain contents and learner modeling data among the several agents that cooperate in the tutoring task. The interactions between two agents from different viewpoints take place by using JADE framework.

The MATHEMA conceptual model [1] provides a partitioning scheme, called viewpoints, after leading to sub-domains definitions. This partitioning scheme is based on epistemological assumptions about the domain knowledge. The knowledge associated with each sub-domain is structured into one or more curricula. Each curriculum consists of a set of pedagogical units and each pedagogical unit is associated to a set of problems. To each problem type is associated a body of support knowledge.

From this knowledge domain model with its parts (represented by pedagogical units), we follow with a mapping to indicate the agents in SATA. Each tutoring agent in SATA consists of three modules: The tutoring system, the social system and the distribution system. Tutoring system consists of three main components, but where the Expert Module is one of them. This module is responsible for problem solving in a given subdomain, comprising three modules: problem solver, evaluator (evaluate solutions proposed by learners), diagnose module (diagnosis mistakes of the learners).

3 Final Remarks

The preliminary results using this ITS have been positive, mainly with regard to its feasibility and usefulness. One conclusion is that our work is, to the best of our knowledge, the first in the literature to present a concrete and large in scope solution for multiple representation with multi-strategies in classical logic, specifically Propositional Logic with focus on argument validation.

References

1. de Barros Costa, E., Perkusich, A., Ferneda, E.: From a Tridimensional View of Domain Knowledge to Multi-agent Tutoring System. In: de Oliveira, F.M. (ed.) SBIA 1998. LNCS (LNAI), vol. 1515, pp. 61–72. Springer, Heidelberg (1998)
2. Lesta, L., Yacef, K.: An Intelligent Teaching Assistant System for Logic. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363, pp. 421–431. Springer, Heidelberg (2002)
3. Lukins, S., Levicki, A., Burg, J.: A tutorial program for propositional logic with human/computer interactive learning. SIGCSE Bulletin 34(1), 381–385 (2002)

Simulation-Based Training of Ill-Defined Social Domains: The Complex Environment Assessment and Tutoring System (CEATS)

Benjamin D. Nye, Gnana K. Bharathy, Barry G. Silverman, and Ceyhun Eksin

University of Pennsylvania
Ackoff Center for Advancement of Systems Approaches
120B Hayden Hall, 3320 Smith Walk
Philadelphia, PA 19104

Keywords: Hybrid Tutoring, Simulation-Based Learning, Assessment, Military.

Socio-cultural problems have special challenges that complicate training design. Problems in these domains have been called “wicked problems” due to their intractability [4]. Such problems are ill-defined: characterized by conflicting stakeholder values, disagreements over solutions, and interconnectedness between problems. Simulation-based learning can be used to explore these problems, but assessment is a bottleneck for training ill-defined domains.

Problems in ill-defined domains are heterogeneous: some problems have clear right and wrong answers, but others are subjective, context-dependent, or emergent. A possible solution is hybrid tutoring, which combines multiple tutoring approaches [2]. A hybrid tutor could match different pedagogical interventions for different types of problems. However, hybrid tutoring lacks established design principles for matching domain problems with suitable interventions.

The Complex Environment Assessment and Tutoring System (CEATS) follows two principles to support hybrid tutoring. First, semantic interfaces are used to decouple components, transforming the simulation environment into meaningful metrics. Assessments use metrics as evidence to calculate measures about domain concept qualities. The second principle is to support families of assessments. Together, this design decouples assessments from the simulation and embeds meta-data to make them meaningful for reporting and tutoring modules.

The Complex Environment Assessment and Tutoring System uses metrics as a semantic API for the learning environment. This allows different environments (e.g. simulation vs database) to share the same metric specifications, but implement their own function and query implementations. A metrics engine currently exists for use with a real-time simulation (described below) and a second metrics engine is being added to support metrics on a database of simulation runs.

In CEATS, assessments are implemented as relationships between metrics and domain knowledge. Assessments include meta-data on the objectivity, usage, frame-of-reference, assessment type (qualifier), and domain knowledge associated with the measurement. Assessment qualifiers determine the basic meaning of the assessment, such as different types of attitudes (e.g. like/dislike) or learning about concepts (e.g. mastery level). They also support assessments that designate when an opportunity to demonstrate learning or preferences has occurred. Objective vs. subjective specifies whether the assessment measures an objective truth (e.g. math problem answer) or a subjective quality (e.g. favorite math operator). Frame of reference refers to what the measurement is compared against, which can be fixed criterion (e.g. standards-based), normed (e.g. compared to peers), or ipsative (e.g. compared against self, at other times or tasks). Usage refers to the intended usage of the assessment. Formative assessment is valid during a task and tends to focus on process, while summative assessment occurs after task completion and focuses on outcomes.

The tutoring engine is currently under active development, targeting a hybrid design driven by assessment meta-data. At present, tutoring engine development is focusing on three complementary types of interventions: Error Feedback, Comparative Feedback, and Reflective Prompts. Error feedback will be driven by objective, criterion-based assessments. Comparative feedback will be employed where ipsative or normed assessments are available, such as comparing user performance against prior performance or comparing skills. When only subjective criteria are available, the system will fall back on questions that help the user reflect on their actions. This design is novel because hybrid tutoring will be driven by assessment meta-data, instead of ad-hoc pairing of pedagogy to problems.

CEATS has been integrated with the StateSim simulation environment to support AtN counter-insurgency strategy training. The Department of Defense is currently supporting the “Attack the Network” (AtN) paradigm, which outlines strategies for kinetic and non-kinetic engagement of insurgent networks that finance, develop, and deploy improvised explosive devices [3]. Users implement courses of action in StateSim, an agent-based simulation focusing on interacting factions [5]. StateSim competed in the DARPA Integrated Crisis Early Warning System (ICEWS) project and forecasted measures of state and regional instability with over 80% accuracy [1]. Currently, the CEATS engine provides metrics and assessment capabilities for a StateSim Afghanistan AtN training scenario. Future work on CEATS will complete the tutoring engine, supporting training of the ill-defined domain of counter-insurgency, and add authoring tools for assessments.

References

1. Bharathy, G.K., Silverman, B.G.: Validating agent based social systems models. In: Winter Simulation Conference (WSC 2010), pp. 441–453. IEEE (2010)

2. Fournier-Viger, P., Nkambou, R., Nguifo, E.M.: Building Intelligent Tutoring Systems for Ill-Defined Domains. In: Nkambou, R., Bourdeau, J., Mizoguchi, R. (eds.) *Advances in Intelligent Tutoring Systems*. SCI, vol. 308, pp. 81–101. Springer, Heidelberg (2010)
3. NTC Operations Group: *Attack the Network Handbook* (May 2010)
4. Rittel, H.W.J., Webber, M.M.: Dilemmas in a general theory of planning. *Policy Sciences* 4(2), 155–169 (1973)
5. Silverman, B.G., Bharathy, G.K., Nye, B.D., Kim, G.J., Roddy, M., Poe, M.: M&S methodologies: A systems approach to the social sciences. In: Sokolowski, J.A., Banks, C.M. (eds.) *Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains*, pp. 227–270. Wiley & Sons, Hoboken (2010)

Empirical Investigation on Self Fading as Adaptive Behavior of Hint Seeking

Kazuhisa Miwa¹, Hitoshi Terai¹, Nana Kanzaki¹, and Ryuichi Nakaike²

¹ Nagoya University, Nagoya, 464-8601, Japan
miwa@is.nagoya-u.ac.jp

² Kyoto University, Kyoto, 606-8501, Japan

Abstract. We investigated whether students behave adaptively in hint-seeking from the viewpoint of self-fading. To let students effectively learn, scaffolding should be eliminated gradually with the progress of learning. We define self-fading as fading behavior lowering the levels of support by students themselves. We investigated the relation between such metacognitive behavior and learning effects through two experiments in a laboratory setting and in actual class activities. The results showed that our participants successfully faded help supports, and also confirmed that those who lowered the levels of support and learned with their own efforts gained larger learning effects.

Keywords: hint seeking, self fading, scaffolding, metacognition.

1 Introduction

Students themselves have to manage their help-seeking behavior to maximize learning effects. However, many previous studies have demonstrated that students' help-seeking behavior does not follow rational principles [3]. Hint abuse is a representative irrational behavior that appears in hint-seeking where students tend to seek the most specific hints to find answers rather than seeking understanding [1]. In this paper, we investigate whether students behave adaptively in hint-seeking from the viewpoint of self-fading, which is defined with scaffolding as one central concept for providing effective learning. We define self-fading as behavior during which the students themselves lowered their levels of support. We tested if students actually faded their help support during learning and also investigated the relation between such metacognitive behavior and learning effects through two experiments in a laboratory setting and actual class activities.

2 Learning System and Task

We investigated participants' help-seeking behavior using a relatively complex learning task in which they learned natural deduction (ND). Natural deduction is a kind of proof calculus in which logical reasoning is expressed by inference rules closely related to a natural way of reasoning. Participants, e. g., for inducing

a proposition $\neg Q \rightarrow \neg P$ from a premise $P \rightarrow Q$, learned inference rules and strategies for applying the rules. Our tutoring system was developed for teaching ND to university undergraduates. It was established based on a server-client framework. Miwa, et al. (2009) [2] developed a web-based production system architecture called DoCoPro that enables such a system design to be established. The scaffolding levels can be controlled from two viewpoints: rule selection and application.

3 Experimental Results

Experiment 1 was preliminarily performed. The participants solved one problem twice using our tutoring system in a laboratory setting. Experiment 2 was performed in an actual class setting and conducted more detailed analysis. It investigated three objectives: to replicate the finding of Experiment 1, to confirm the participants' adaptive behavior for controlling LOSs based on the degree of the problem difficulties, and to confirm the relation between LOSs and learning effects.

The overall results are summarized as follows:

- The participants lowered their LOSs from the first to second trials both in Experiment 1 in a laboratory setting and in Experiment 2 in actual class activities.
- The participants adaptively lowered the levels of support when facing easier problems than when facing difficult problems.
- The participants who got higher scores in the posttest learned with lower levels of support than the students with lower scores. On the other hand, we did not observe such a tendency in the relation between the pretest scores and the support levels.
- However, in the correlation between the gains from the pre to post scores and LOSs in the learning phase, a statistically significant relation was not confirmed.

References

1. Alevan, V., Koedinger, K.R.: Limitations of Student Control: Do Students Know When They Need Help? In: Gauthier, G., VanLehn, K., Frasson, C. (eds.) ITS 2000. LNCS, vol. 1839, pp. 292–303. Springer, Heidelberg (2000)
2. Miwa, K., Nakaike, R., Morita, J., Terai, H.: Development of production system for anywhere and class practice. In: Proceedings of the 14th International Conference of Artificial Intelligence in Education, pp. 91–99 (2009)
3. Wood, H., Wood, D.: Help seeking, learning and contingent tutoring. *Computers and Education* 33, 153–169 (1999)

Scripting Discussions for Elaborative, Critical Interactions

Oliver Scheuer¹, Bruce M. McLaren^{1,2}, Armin Weinberger¹, and Sabine Niebuhr³

¹ Saarland University, Saarbrücken, Germany

² Carnegie Mellon University, Pittsburgh, PA, U.S.A.

³ Clausthal University of Technology, Clausthal, Germany
o.scheuer@mx.uni-saarland.de

Abstract. Scripting collaborative argumentation can be effective in helping students understand multiple perspectives in complex, ill-defined domains. We have developed a web-based collaborative learning environment and a collaboration script to support students in discussing and analyzing controversial texts. We present a study in which we varied one element of the script to support critical, elaborative interactions, namely whether or not students take a proponent and/or critic role. Our results suggest that roles have a positive effect on the extent of knowledge elaboration in student discussions.

Keywords: computer-supported collaborative learning, collaboration scripts, argumentation, argument mapping.

1 Introduction

It is widely recognized that critical thinking skills play an important role in today's information societies. During the past two decades many computer-based tools have been developed to support the acquisition of argumentation skills [2]. We introduce a web-based collaborative learning environment that supports students in creating and discussing argument diagrams, and a collaboration script to support students in using this environment to discuss conflicting texts. We present a study that investigates whether an additional script component, in which students take "proponent" and "critic" roles, could improve the quality of student discussions in terms of critical, elaborative interactions.

2 Learning Environment and Collaboration Script

LASAD is a highly configurable, web-based argument-diagramming environment that allows groups of students to represent arguments graphically in the form of box-and-arrow diagrams. Boxes represent statements and links represent argumentative and rhetorical relations of different types (e.g., "support", "opposition", "related to"). Besides a shared diagramming workspace students can use a chat to communicate with one another.

The FACT-2 collaboration script ("Fostering Argumentation Through Conflicting Texts") has been developed to support critical, elaborative discussion in student dyads

based on conflicting texts. It is based on distributed resources, that is, each student has exclusive access to one of two texts. Student activities are structured in four different phases: Students (1) model their texts in LASAD individually, (2) discuss, based on the diagram, aspects of each text with the partner, (3) discuss connections between the two texts, and (4) agree on a joint position and compose a joint reasoned conclusion. A previous version of the script is described in [3].

3 Study

A quasi-experimental study using a pretest-intervention-posttest design has been conducted. Both conditions used LASAD and the collaboration script described above. Opposing texts regarding climate change were used (thesis: “Developed countries have to cut their carbon emissions drastically”). For Treatment dyads an additional role script component was administered, in which students were instructed to act as proponent of their text and a constructive critic of their partner’s text. A sentence opener interface [4] was used to provide support for the proponent and constructive critic roles. The Comparison group used a standard chat instead.

Participants were students at Saarland University and received a participation fee. The sample comprises 12 Treatment dyads (i.e., with role script) and 10 Comparison dyads (i.e., no role script). The overall study took about 3 hours; 1.5 hours of which were spent on the actual task. An analysis of questionnaire data indicated that the conditions did not differ significantly in terms of relevant entry characteristics.

We report on an analysis of chat protocols. Based on the Rainbow coding framework [1] we developed and validated a coding scheme with satisfactory result ($\kappa = .76$). We distinguished three levels of argumentative elaboration. To assess the quality of each protocol we analyzed the amount of “High” elaboration moves (i.e., ones that cite, elaborate, question or criticize relevant contents). We found a non-significant trend ($p = .07$) with large effect size ($d = 0.82$) in terms of “High” codes in favor of the Treatment group, a result in accordance with our hypothesis.

Acknowledgements. We would like to thank Christoph Fehige for advice and support. The German Research Foundation (DFG) provided funding for this research.

References

1. Baker, M., Andriessen, J., Lund, K., van Amelsvoort, M., Quignard, M.: Rainbow: A Framework for Analyzing Computer-Mediated Pedagogical Debates. *IJCSCL* 2(2-3), 247–272 (2007)
2. Scheuer, O., Loll, F., Pinkwart, N., McLaren, B.M.: Computer-Supported Argumentation: A Review of the State of the Art. *IJCSCL* 5(1), 43–102 (2010)
3. Scheuer, O., McLaren, B.M., Harrell, M., Weinberger, A.: Scripting Collaboration: What Affects Does it Have on Student Argumentation? In: Hirashima, T., et al. (eds.) *Proc. of ICCE 2011*, pp. 181–188. Asia-Pacific Soc. for Computers in Education (2011)
4. Soller, A.: Supporting social interaction in an intelligent collaborative learning system. *IJAIED* 12, 40–62 (2001)

Design Requirements of a Virtual Learning Environment for Resource Sharing

Nikos Barbalios¹, Irene Ioannidou³, Panagiotis Tzionas²,
and Stefanos Paraskeuopoulos¹

¹ Dept. of Special Education, University of Thessaly

² Dept. of Automation, TEI Thessalonikis

³ Dept. of Early Childhood Care and Education, TEI Thessalonikis

Abstract. This study presents the evaluation of the design requirements of a novel model-supported virtual environment appropriate for environmental education, where the simulation process is controlled by a novel multi-agent model. The virtual environment was qualitatively evaluated by 14 students, that provided feedback about the accuracy of the graphical representations, the usability and interaction of the interface and the comprehension of the underlying process. Students suggestions were taken under consideration, modifying the virtual environment to its final form.

Keywords: virtual environment, multi-agent model, environmental education.

1 Introduction

This study introduces a novel virtual environment simulating the exploitation of a lake by a community of farmers. It consists of two elements: a multi-agent simulation (MAS) model of an ecosystem (that controls the simulation), and a virtual world that makes possible the visualization of the elements and procedures of the MAS model in a comprehensible manner. In this sense, the realism and scientific accuracy of the simulation is guaranteed, while the complexity of the ecosystem dynamics are abstracted from the students.

The MAS model presented in [1] was utilized, that simulates the exploitation of a lake by a community of farmers under various farmer behaviours and encapsulates a machine learning algorithm that can be optionally used as a water regulatory policy to reveal optimal resource allocation schemes. This model was chosen because it delivers realistic simulations, as it's agents mimic actual farmer behaviours encapsulating the growing economic pressure exerted on farmers. The MAS model was calibrated according to real data from the lake Koronia ecosystem in Greece, that was nearly depleted in 2002 due to overexploitation [1]. Additionally, the MAS model provides means of estimating the impact of the farmers' behaviour to the environment and to their potential profit.

The virtual world is implemented in the VRML programming language and entails high-detailed realistic 3-D models. It consists of a lake and 10 agricultural

fields, each one entailing an animated crop (that slowly grows as the simulation advances), water pumps and sprinklers for the fields' irrigation and an animated farmer performing some typical agricultural labour. The number of sprinklers in each field is representative of the corresponding farmer behaviour. Moreover, the environment entails 3-D entities that are usually met in agricultural landscapes (i.e. tractors, trees, birds). Students may explore the virtual world like in a first person game using the keyboard, or predefined viewpoints that focus on important aspects of the virtual world. Based on the MAS model, at the end of each simulation, the learner is informed about the outcome of the farmer's behaviour using audio and visual cues (i.e. classical music and emoticons).

2 Experiments

The virtual environment was qualitative evaluated by 7 two-person groups of elementary school students, that provided feedback regarding a) the accuracy of the graphical representations, b) the usability and interaction with the graphical interface of the virtual world and c) the comprehension of the simulation procedure. Regarding the clarity of the representations, students successfully identified all the basic elements of the model (i.e. lake, farmers, fields, irrigation method), commenting on their realistic appearance and spending a significant amount of time exploring the virtual world. Suggestions were made however to increase the number of fishes in the lake. Most of them had no problem navigating through the virtual world, however 2 of them encountered difficulties using the keyboard. Thus it was chosen to enhance the virtual environment with joystick navigation capabilities. Regarding the comprehension of the simulation procedure, students acknowledged both the behaviour of the farmers as well as the natural process that took place (i.e. water draining for irrigation purposes). In overall, the virtual environment was easily accepted by all the students, claiming that experimentation was a pleasant experience and commenting on the realism and the details of the graphical representations.

3 Conclusions

Students exhibited a positive reaction to the virtual environment, commenting on its realistic appearance, and the experimental results verified the design requirements with respect to the accuracy of the representations, the usability and the interactions with the interface and the comprehension of the simulation procedure. Students suggestions were taken under consideration, modifying the virtual environment to its final form that will be used for further research.

Reference

1. Barbalios, N., Ioannidou, I., Tzionas, P., Paraskeuopoulos, S.: A constrained multi-agent model for studying natural resource sharing. In: Proc. of 3rd Int. Conf. on Environmental Management, Engineering, Planning and Economics, Skiathos, Greece, pp. 185–191 (2011) ISBN 978-960-6865-43-5

The Effectiveness of a Pedagogical Agent's Immediate Feedback on Learners' Metacognitive Judgments during Learning with MetaTutor

Reza Feyzi-Behnagh and Roger Azevedo

McGill University, Dept. of Educational and Counselling Psychology, Montreal, Canada
reza.feyzibehnagh@mail.mcgill.ca, roger.azevedo@mcgill.ca

Abstract. Using pedagogical agents (PAs) in hypermedia learning environments have been found to be an effective way to scaffold students and provide tailored feedback to enhance learning outcomes. In this study, we investigated the effectiveness of immediate feedback provided by PA embedded in MetaTutor [1] (a multi-agent, adaptive hypermedia learning system) on learners' metacognitive calibration and bias of Feelings of Knowing (FOK) and Judgments of Learning (JOL), and accuracy of Content Evaluations (CE) made during a 2-hour learning session with the system. Seventy ($N = 70$) undergraduate students were randomly assigned to one of two instructional conditions: Immediate Feedback (IF) or the Control Group, where they were asked to learn about the circulatory system with the environment. Overall, pretest-posttest learning outcome data revealed that participants in the IF condition significantly outperformed those in the Control condition. Additionally, participants who received immediate feedback from the PA were more accurate and calibrated in their metacognitive judgments than those in the Control condition. An overall bias was found toward overconfidence in JOLs and FOKs for participants in both conditions. These findings have significant relevance for the understanding of metacognitive monitoring and regulation during complex learning with multi-agent systems and for designing metacognitive-responsive PAs capable of co-adapting to learners' cognitive and metacognitive regulatory processes.

Keywords: Metacognitive judgments, calibration; bias, pedagogical agents, immediate feedback, multi-agent systems, empirical studies.

Learners' metacognitive judgments are critical in determining both the selection and use of strategies during learning with multi-agent systems, but it has been found in previous studies that students are not usually calibrated in their metacognitive judgments. One approach to address this issue is developing multi-agent adaptive learning environments that embody artificial pedagogical agents (PAs) that are designed to model, trace, foster and scaffold students' metacognitive processes during learning (see [1]). One of the areas where PAs can assist learners is in monitoring metacognitive processes and making accurate metacognitive judgments. In this study, we examined the effects of immediate feedback on the accuracy, bias, and discrimination of learners' metacognitive judgments

(JOLs, FOKs, and CEs) during their learning about the circulatory system. These three metacognitive judgments have been selected because inaccurate judgments might either lead the participant into spending too much time on content already learned or proceed to another topic without having learned enough about the current content. 70 undergraduate students (60% females, mean age 22) participated in this study. They were randomly assigned to either an experimental (immediate feedback, IF) or control condition. Participants in the IF condition received timely prompts from the PA in MetaTutor to deploy different SRL processes, and received appropriate immediate feedback regarding their performance on the use of those strategies. On the other hand, participants in the Control group did not receive any SRL prompts and feedback from the PAs in MetaTutor. MetaTutor [1] is a multi-agent intelligent hypermedia learning environment, which contains 41 pages of text and diagrams about the human circulatory system, designed to detect, model, trace, and foster students' self-regulated learning about complex science topics. Data for this study was obtained from system-generated log-files. FOK and JOL ratings were made on a Likert scale and were converted to either positive or negative valence (1-3 to negative and 4-6 to positive valence). Three measures of Goodman-Kruskal Gamma correlation (G), bias, and discrimination scores were calculated to describe how the JOLs and FOKs correlated with participants' performance. The cognitive gain results indicated that students in the IF condition outperformed those in the control condition in the post-test they took on the circulatory system, $F(1, 68) = 33.037, p < .01, \eta^2 = .327$. Gamma values for JOLs and FOKs across the two conditions indicate that participants in the IF condition were more accurate than those in the Control condition. With regards to bias, the IF condition participants did not have any bias toward over- or under-confidence (bias = 0). However, participants in the Control condition were slightly overconfident in their JOL ratings (bias = 0.2). In terms of discrimination scores, the IF group participants had more confidence on correct than incorrect JOL and FOK judgments, and their discrimination score was more than the one for the Control condition. These findings indicate that receiving timely prompts and appropriate feedback from the PA during learning about complex topics in a multi-agent hypermedia learning environment leads to higher accuracy in metacognitive judgments made by participants. Analyses of the accuracy of CEs indicated no significant difference between the two conditions, $F(1, 258) = .826, p > .05$. As a summary, we found that receiving immediate feedback from a PA in MetaTutor increased participants' accuracy of JOLs and FOKs, decreased their bias, and improved their discrimination.

Acknowledgments. The research presented in this study has been supported by funding from the National Science Foundation (DRL 0633918) awarded to the second author and (DRL 1008282) awarded to Ronald Landis.

Reference

1. Azevedo, R., Feyzi-Behnagh, R., Harley, J., Trevors, G., Duffy, M., Bouchet, F.: MetaTutor: A learning environment for the detection, tracking, modeling, and fostering self-regulated learning. In: Azevedo, R., Aleven, V. (eds.) *International Handbook of Metacognition and Learning Technologies*. Springer, Amsterdam (in press)

Supporting Students in the Analysis of Case Studies for Professional Ethics Education

Mayya Sharipova and Gordon McCalla

University of Saskatchewan, Department of Computer Science

Abstract. Analyzing case studies in a professional ethics course can be quite challenging for students. To help, we have developed a system called Umka that supports students in this analysis by: (i) directly critiquing the arguments of a student; and (ii) supporting collaboration among multiple students. Umka achieves this support through the interaction of a well-structured interface and the use of latent semantic analysis (LSA). We conducted an experiment, which demonstrated that this cross-interaction of the interface and LSA could be a promising way to "diagnose" a student's argument even without full natural processing.

Keywords: ethics education, ill-defined domain, latent semantic analysis, supporting learner argumentation and discussion.

Introduction. Analysis of case studies is a common method in the teaching of professional ethics. Students are given a certain case study representing some ethical dilemma (the case description in Fig. 1), and several propositions - possible ways to resolve the dilemma of the case study (e.g. "Make a copy", "Don't make a copy" in Fig. 1). A student's task is to provide arguments for and against propositions, and then to synthesize the arguments to find the best proposition.

The main challenge for an ITS supporting students in the analysis of case studies is to "diagnose" arguments of students. Latent semantic analysis (LSA) is one of the techniques used to evaluate students' answers against predefined system answers [1]. But LSA works as a "bag of words" model without deep analysis of sentence structure, and therefore is not very effective in distinguishing negative arguments from positive ones which is crucial for an ethics ITS. LSA also usually assumes larger bodies of text than paragraph or sentence long arguments. We show promising results for how the interface can help constrain the student enough that LSA can work "in the small" to analyze arguments.

System's Description and Experimental Results. To distinguish whether a student makes an argument for or against a certain proposition, we incorporated a FOR and AGAINST distinction in the interface. We asked students to provide arguments FOR in the left column of the table, and arguments AGAINST in the right column (Figure 1). To give feedback to a student, the system using LSA compares the student's arguments from the FOR column against the system predefined arguments FOR, and the student's arguments from the AGAINST column against the system predefined arguments AGAINST.

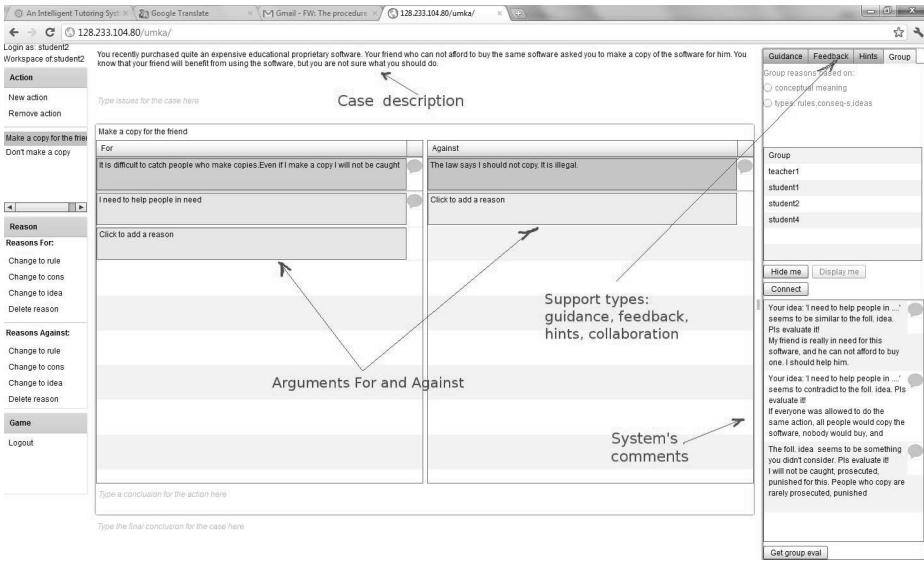


Fig. 1. A screenshot of the Umka system

To evaluate the effectiveness of the interaction of LSA and the interface in finding good matches, we conducted an experiment with 23 students. The students were asked to provide arguments related to a case study, and evaluate the relevance of the system feedback given in response to their arguments. We found that the interaction of LSA and the interface proved to be fairly effective, with an average precision of LSA 0.5 when compared to human expert judgement, which was around 4 times as high as the precision of random matching, and higher than keyword search precision. Moreover, students found 62% of feedback messages relevant.

Conclusion. Our results look promising, but still leave the room for the improvement of the diagnosis of students' arguments. Thus, future steps include enhancing LSA with other methods such as textual entailment, and using students' arguments to tune the matching algorithm by finding different ways students can phrase the same argument or even adding new students' arguments to the system. Improved diagnosis will allow more relevant feedback, and thus more effectively support students in the analysis of case studies.

Acknowledgements. The authors wish to thank the Natural Sciences and Engineering Research Council of Canada for their funding of this research project.

Reference

1. Graesser, A., Lu, S., Jackson, G., Mitchell, H., Ventura, M., Olney, A., Louwerse, M.: Autotutor: A tutor with dialogue in natural language. *Behavior Research Methods* 36(2), 180–192 (2004)

Evaluating the Automatic Extraction of Learning Objects from Electronic Textbooks Using ErauzOnt

Mikel Larrañaga, Ángel Conde, Iñaki Calvo,
Ana Arruarte, and Jon A. Elorriaga

University of the Basque Country
mikel.larranaga@ehu.es

Abstract. Content reuse is one of the major concerns in the Technology Enhanced Learning community. *ErauzOnt* is a system that uses Natural Language Processing techniques, heuristic reasoning, and ontologies to generate Learning Objects from textbooks. It has been tested with several textbooks written in the Basque language in order to evaluate the automatic construction of Learning Objects.

Keywords: Learning Objects, Content Authoring.

1 Introduction

Technology Supported Learning Systems (TSLs) require an appropriate representation of the knowledge to be learnt, that is the *Domain Module*. Content authoring is known to be an effort and time consuming task and, therefore, knowledge reuse and semi-automatic content authoring should be promoted. Textbooks, one of the traditional knowledge preserving and transferring means, can be used to gather the knowledge required to build *Domain Modules*.

Content reuse has been addressed by several projects with the aim of facilitating the development of TSLs or learning material [1,3]. *ErauzOnt* was developed to facilitate the construction of Learning Objects (LOs) from textbooks. The generation of LOs for the domain topics is accomplished through the identification and extraction of Didactic Resources (DRs), i.e., consistent fragments of the document related to one or more topics with a educational purpose. *ErauzOnt* [2] identifies and extracts these pieces using ontologies, heuristic reasoning, and Natural Language Processing (NLP) techniques.

2 Experiment

ErauzOnt has been tested over 4 textbooks on Nature Sciences written in Basque with the aim of validating the extraction of LOs from electronic textbooks. The experiment for evaluating *ErauzOnt* was carried out in the following way: four

instructional designers manually analysed the electronic textbooks, and collaboratively defined the Learning Domain Ontology (LDO) that describes the learning domain for each document. The LDOs describes the topics to be mastered and the pedagogical relationships among the topics. After that, they were requested to identify and classify the fragments of the documents related to the domain topics included in the LDOs with educational value. The instructional designers collaboratively identified *definitions*, *principle statements*, *examples*, *problem statements*, and *combined resources*, i.e., resources that combine more than one kind of DR. The identified set of DRs was then used as the reference for the evaluation of the performance of *ErauzOnt*, which relied on the LDOs defined by the instructional designers for gathering the LOs from the textbooks.

The evaluation of the gathered LOs was carried out comparing the manually identified DRs with the automatically gathered ones. Many of the manually identified DRs also were composite fragments that contain finer grain resources. An aspect to be considered to evaluate the gathered LOs is that while a LO might be more appropriate in a particular context, one of its components or a composite LO that comprises it might fit better and, therefore be more reusable, in other situations. Table 1 summarises the results of the experiment, where the LO acquisition process achieved a 70.31% *recall*, a 91.88% *precision* and a 79.66% *f-measure*. LO acquisition achieved satisfying results, although the *definitions*, *principle statements* and the *composite LOs* are more difficult to identify.

Table 1. Statistics on the LO Acquisition

	Definitions	Princ. statements	Examples	Prob. statements	Composite LOs	Total
Recall (%)	59.70	50.00	87.50	81.90	59.46	70.31
Precision (%)	91.14	96.30	100.00	88.55	97.84	91.88
F-measure (%)	72.14	65.82	93.33	85.10	73.97	79.66

Acknowledgements. This work is supported by the University of The Basque Country (UPV/EHU09/09), the Spanish Ministry of Education (TIN2009-14380), and the Basque Government (IT421-10).

References

1. de Hoog, R., Barnard, Y., Wielinga, B.J.: IMAT: Re-using multi-media electronic technical documentation for training. In: Roger, J.Y., Stanford-Smith, B., Kidd, P.T. (eds.) *Business and Work in the Information Society: New Technologies and Applications*, pp. 415–421. IOS Press (1999)
2. Larrañaga, M., Calvo, I., Elorriaga, J.A., Arruarte, A., Verbert, K., Duval, E.: *ErauzOnt: A Framework for Gathering Learning Objects from Electronic Documents*. In: *Proceedings of the ICALT 2011*, pp. 656–658. IEEE Computer Society (2011)
3. Verbert, K., Ochoa, X., Duval, E.: *The ALOCOM Framework: Towards Scalable Content Reuse*. *Journal of Digital Information* 9(1) (2008)

A Cognition-Based Game Platform and its Authoring Environment for Learning Chinese Characters

Chao-Lin Liu¹, Chia-Ying Lee², Wei-Jie Huang³, Yu-Lin Tzeng⁴, and Chia-Ru Chou⁵

^{1,3} National Chengchi University, Taiwan,
^{2,4,5} Sinica Academia, Taiwan
chaolin@nccu.edu.tw, chiaying@sinica.edu.tw

Abstract. We present integrated services for playing and building games for learning Chinese characters. This work is unique on two aspects: (1) students play games that are designed based on psycholinguistic principles and (2) teachers compile the games with software tools that are supported by sublexical information in Chinese. Players of the games experience and learn the grapheme-morpheme relationships underlying the writings and pronunciations of Chinese characters. Both visual and audio stimuli are employed to enhance the learning effects in the games. The software tools, utilizing structural knowledge about Chinese characters, offer instrumental information to facilitate the compilation of games. Preliminary studies with 116 participating students, in an elementary school in Taipei, showed that students who were given a one-month period to play the games improved their response time in naming tasks for reading Chinese characters. In addition, evaluation of the authoring tools by 20 native speakers of Chinese indicated that using the tools significantly improved the efficiency of preparing the games and the quality of the resulting games.

Keywords: grapheme-phoneme conversion, phonological components, serious games, language-dependent authoring tools, visually similar Chinese characters.

Phono-semantic characters (PSCs, henceforth) constitute more than 60% of Chinese characters in everyday lives. The writing of a PSC carries phonological and semantic information with its phonological and semantic parts, respectively. For instance, “讀”(du2), “瀆”(du2), “黷”(du2), “黷”(du2) share the same phonological components (PCs, henceforth), and contain different semantic parts. The PC, “賣”(mai4) on the right sides, provide hints about the pronunciations of these characters, and the influence of “賣” is consistent. A PC may and may not be a stand-alone character. The characters “檢”(jian3), “檢”(jian3), and “儉”(jian3) share their PCs on their right sides, but that PC is not a standalone Chinese character. A PC, when it is a stand-alone character, may and may not be pronounced the same as those characters that contain the PC. In the above examples, the pronunciations of “賣” and “讀” are different. In contrast, “洵”(tou2) is a stand-alone character, and has the same pronunciation as “洵”, “陶”, and “啣”. Despite these subtleties, learning the systematic influences of the PCs on their carrying characters significantly reduces the burden to remember the pronunciations of individual characters separately [1].

With the assistance of our software tools, teachers can compile games in the form that is illustrated on this page. Players see the *target PC* shown on the top of the screen, “里” (li3) in this game, and characters, “狸” (li2) in this snapshot, will randomly pop up from any of the six holes. Players will hear the pronunciations of the characters (the sound is played automatically to strengthen the connection between the pronunciation and writing of the character), and they have to judge within a time limit whether or not the character contains the target PC. If yes, as shown in this snapshot, the player has to hit the monster with a mouse click (or touch it on a flat panel computer). If no, the player does not have to do anything. A sequence of 10 characters will be presented to the players in a single game. Credits of players will be increased or decreased upon correct or incorrect hits (or touches), respectively. If the players collect sufficient credits, s/he will be allowed to play advanced games in which s/he learns how the characters are used in normal Chinese text.

116 students in an elementary school in Taipei participated in an evaluation of the games. Pretests and posttests were administered with (1) the Chinese Character Recognition Test (CCRT) and (2) the Rapid Automatized Naming Task (RAN). In CCRT, participants needed to write the pronunciations in Jhuyin, which is a phonetic system used in Taiwan, for 200 Chinese characters. The number of correctly written Jhuyins for the characters was recorded. In RAN, participants read 20 Chinese characters as fast as they could, and their speeds and accuracies were recorded. Experimental results show that performance of the students, in the experimental group, improved significantly in RAN speed (p -values < 0.02) but remained almost the same in RAN and CCRT accuracies.

The content of a game includes characters that do and do not contain the target PC. To have a way to control the challenge levels of the games, we require characters that do not contain the target PC to exhibit varying attractiveness. Attractive distracters make the game more challenging than those obviously unattractive ones. A character that contains components that look like the target PC is such an attractive distracter. Consider the game illustrated on the previous page. The character “理”(li3) is a correct character to click, while the character “狸”(li2) is not. We consider “狸” a challenging distracter because it looks like “理”.

Listing sufficient characters that contain the target PC demands very impressive memory about the writings of thousands of Chinese characters. It turns out that providing lists of attractive distracters are even more challenging. Experimental results showed that even native speakers of Chinese cannot perform well in these tasks.

When authoring a game, the teacher chooses the correct characters for the game, than s/he has to provide the attracters. Applying the techniques that we reported in [2], we were able to assist teachers in both tasks. Consider the target PC “里” again. Our authoring tools can provide teachers the list “鋸裡埋埋哩哩里理裡裡鐘童狸哩量” to use in the game as correct characters. Note that these characters belong to different radicals and have different pronunciations. Consequently, there is no easy way to find them all with just a dictionary, and our software tools are crucial. Moreover, we can recommend characters that look like the correct characters, e.g., “鈿鉀鍾” for “鋸”, “裸袖嘿” for “裡”, “湮湮涓” for “哩”, “狎猥猥狙” for “狸”, and “黑墨” for “里”.



We evaluated the authoring tools with 20 native speakers, each authoring games for a list of 5 target PCs. The group that used our tools was able to finish the jobs two times (on average) faster than those who did not, and the quality of the resulting games were 50% better at the same time (p-values < 0.01).

Acknowledgment. This work was supported in part by NSC-100-2221-E-004-014 and NSC-98-2517-S-004-001-MY3 projects of the National Science Council, Taiwan.

References

1. Lee, C.-Y., et al.: Consistency, regularity, and frequency effects in naming Chinese characters. *Language and Linguistics* 6(1), 75–107 (2005)
2. Liu, C.-L., et al.: Visually and phonologically similar characters in incorrect Chinese words: Analyses, identification, and applications. *ACM TALIP* 10(2), 10:1–10:39 (2011)

Effects of Text and Visual Element Integration Schemes on Online Reading Behaviors of Typical and Struggling Readers

Robert P. Dolan and Sonya Powers

Pearson, Assessment & Information, 400 Center Ridge Rd., Austin, TX 78753 USA
{bob.dolan, sonya.powers}@pearson.com

Abstract. This study evaluates the effect of different design schemes for integrating text and visual elements on student reading behaviors. Sixty three fourth and seventh grade students were eye tracked during online reading of middle school science passages embedded with pedagogically relevant visuals in the form of diagrams and photographs. Results show that students' viewing of visuals can be influenced by both how the visuals are positioned and referred to in text, and that this effect is most pronounced for struggling readers. These results have strong implications for the design of online learning materials for diverse students, and in informing adaptive approaches toward multimedia presentation.

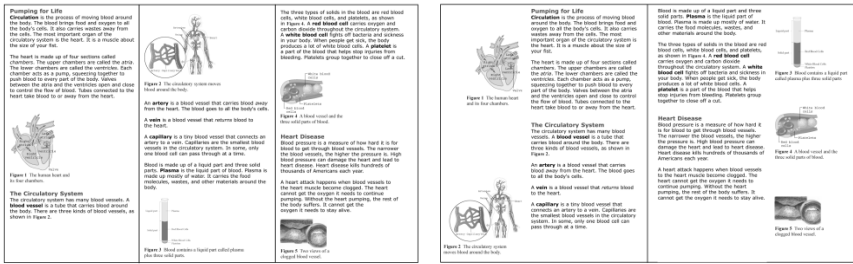
Keywords: multimedia learning, eye tracking, reading behavior, instructional design, adaptive learning.

1 Introduction

Struggling readers challenged by text decoding and/or comprehension have the most to gain through effective use of visuals. The current study evaluated the impact of different design schemes for integrating text and visual elements in multimedia learning on student reading behaviors, to understand how to design online learning system layouts that are appropriately matched to individual students' ongoing needs.

2 Methods

Seventh grade students were selected and designated as struggling or typical readers based upon WIAT-II® reading scores. Typical fourth grade were similarly selected. Students were shown five stimulus passages (one practice, four experimental). Stimulus materials consisted of middle school general science content covering circulation, senses, solar system, and storms. Three two-level stimulus factors were manipulated across the four passages: reading level (at vs. ~3-4 grades above student's reading level, based on Lexiles™), layout linearity (visuals inline with text vs. sidelined), and visual cueing (explicit vs. implicit; see Figure 1). Reading level and layout linearity varied across passages, and visual cueing was varied across visuals within passages. A randomized and counterbalanced stimulus display protocol was implemented.



Inline Stimulus

Sidelined Stimulus

Fig. 1. Two sample stimuli showing inline visuals (left) versus sidelined visuals (right). In each stimulus, visuals are alternately implicitly and explicitly cued.

Data analysis consisted of several mixed-model analyses of variance used to compute main effects and interactions of student group (between-subjects factor) with reading level, cuing, and layout linearity (within-subjects factors). Follow-up tests were used to investigate the source of differences when overall variance ratios were significant (alpha=0.05).

3 Results

Complete, interpretable data were obtained for 23 typical fourth grade readers, 15 struggling seventh grade readers, and 25 typical seventh grade readers. Unless otherwise noted, the following findings were significant at p<.05. Students spent 14.5% of passage reading time viewing inline visuals and only 8.8% viewing sidelined visuals (ES=.91σ). Struggling seventh grade readers were most impacted by inline integration of visuals, with their viewing doubling from 6.5% to 13.9% (ES=1.31σ). These students spent less time reading sidelined visuals than typical seventh grade readers (6.5% vs. 10.8%; ES=.81σ). Students spent 12.0% of passage reading time viewing explicitly cued visuals and only 11.3% viewing implicitly cued visuals (p=.057; ES=.12σ). Passage reading level had no significant effect on the percentage of time students viewed visuals, but did affect total passage reading time (250s for above level, 222s for on level; ES=.34σ).

4 Conclusion

How visual elements are integrated with text in multimedia presentations can significantly impact their use, especially by struggling readers who in general appear to make less intentional use of visuals. These results suggest that adaptive tutoring systems might be enhanced by tracking and responding to student reading challenges, for example by manipulating the placement and cueing of content and supports to effectively steer students toward their use and thus promote comprehension and learning.

Fadable Scaffolding with Cognitive Tool

Akihiro Kashihara and Makoto Ito

Graduate School of Informatics and Engineering,
The University of Electro-Communications, Japan
akihiro.kashihara@inf.uec.ac.jp

Abstract. The main issue addressed in this paper is how to accumulate cognitive experiences of learning with cognitive tool to develop learning skill. Cognitive tool gives learners a scaffold for learning process, which allows them to externalize/visualize the process or results of learning and to reify the learning process. Such reification enables the learners to gain cognitive experience of learning. This paper discusses a fadable scaffolding method for accumulating such experiences, in which functions available on the tool are fadable in a learner-adaptable way.

Keywords: Fadable scaffolding, cognitive tool, skill development.

1 Introduction

Cognitive tool encourages learners to externalize/visualize the process or results of learning to articulate their learning process in their mind. In general, cognitive tool is designed by following a model of learning representing how to learn. It could accordingly provide a scaffold for accomplishing the learning process as modeled. In particular, the cognitive tool allows learners to reify the learning process by making representation externalized/visualized on the tool operable and controllable. Such reification enables them to gain cognitive experience of learning process modeled.

We also expect cognitive tool allows the learners to accumulate the cognitive experiences to develop skill in accomplishing the learning process as modeled. On the other hand, how to accumulate such experiences is an important issue towards developing the learning skill. One common method to resolving it is to induce learners to continuously use cognitive tool. However, there is strong question as to whether the continuous use allows them to develop their learning skill in a fruitful way.

In this paper, we present a fadable scaffolding method, in which functions available on the tool can be faded in a learner-adaptable way.

2 Learning Skill Development with Cognitive Tool

The fadable scaffolding with cognitive tool is expected to produce the following two performance effects on learning process. First, the learners could accomplish the learning process in their mind without the cognitive tool. Second, they could become

more skillful in using the tool, which would promote learning how to learn. We have ascertained that fadable scaffolding induces learners to accomplish the learning process without cognitive tools [1,2]. In this paper, we focus on how the fadable scaffolding method allows learners to become more skillful in using cognitive tool.

In the fadable scaffolding method, functions available on cognitive tool can be faded according to learning skill. When learners get stuck in operating the tool without the functions faded, these functions are once again available. In fading the functions, the learners would carry on cognitive load of executing the learning process without the aid of the functions faded, which would induce them to have more chances to think of the learning process in their mind. Such cognitive load would produce the effect that they could accomplish the learning process without the tool.

In addition, the learners are expected to reconfirm significance and values of the faded functions when they get stuck in the learning process in their mind. Such reconfirmation would give them a deeper understanding of the functions, which enables them to become more skillful in operating the cognitive tool.

We have conducted a case study over 4 weeks with Interactive History [3] (IH for short) that is a cognitive tool helping learners accomplish navigational learning with hypertext-based resources. The results suggest the possibility that the fadable scaffolding induces learners to fade the functions of IH in a reasonable way and to become more skillful in operating IH. In particular, the learners who are less skillful in operating IH before using the fadable scaffolding could obtain more benefits to improve the quality of navigational learning process.

3 Conclusion

This paper has presented fadable scaffolding with cognitive tool, which induces learners to become more skillful in operating the tool to improve the quality of learning process. In future, we will conduct more detailed evaluation to refine the fadable scaffolding method, and address adaptation issues in fadable scaffolding.

Acknowledgments. The work is supported in part by Grant-in-Aid for Scientific Research (B) (No. 23300297) from the Ministry of Education, Science, and Culture of Japan.

References

1. Kashihara, A., Sawazaki, K., Shinya, M.: Learner-Adaptable Scaffolding with Cognitive Tool for Developing Self-Regulation Skill. In: Proceedings of 16th International Conference on Computers in Education, pp. 133–140 (2008)
2. Kashihara, A., Taira, K.: Developing Navigation Planning Skill with Learner-Adaptable Scaffolding. In: Proceedings of 14th International Conference on Artificial Intelligence in Education, pp. 433–440 (2009)
3. Kashihara, A., Hasegawa, S.: A Model of Meta-Learning for Web-based Navigational Learning. *Int. J. Advanced Technology for Learning* 2(4), 198–206 (2005)

Mediating Intelligence through Observation, Dependency and Agency in Making Construals of Malaria

Meurig Beynon¹ and Will Beynon²

¹ Computer Science, University of Warwick, Coventry CV4 7AL, UK
wmb@dcs.warwick.ac.uk

² University Hospitals Leicester, Leicester, UK
willmeurig@gmail.com

Abstract. Achieving co-adaptation in building an Intelligent Tutoring System (ITS) involves integrating machine and human perspectives on ‘knowledge’ and ‘intelligence’. We address this integration by using Empirical Modelling (EM) principles to make *construals*: interactive environments in which human agents acting as model-builders can explore the observation, dependency and agency that underpins their understanding of the subject domain. This approach is well-suited to domains such as medicine where reasoning draws both on scientific knowledge and evolving human experience and judgement. We illustrate this by developing construals of malaria using a web-based variant of the principal EM tool that enables many agents to participate in the process of adaptation.

1 Introduction

In supporting co-adaptation in an Intelligent Tutoring System (ITS), the conceptual framework surrounding ‘knowledge’ and ‘intelligence’ has a critical role. The well-known problems of adapting software to meet new requirements suggest that something richer than the conventional conceptual framework for computing applications is appropriate. Where a traditional computing system is conceived as a ‘program’ that reflects a paradigm of ‘computational thinking’, Empirical Modelling (EM) [1] proposes a broader perspective on computing based on the more primitive notion of ‘construal’. A *construal* is an interactive environment in which a human interpreter can experience metaphorical counterparts of the different states and transitions between states they encounter in a phenomenon they wish to understand.

In this paper, we briefly discuss the use of construals to support learning activities in which adaptation to new information or agent behaviours is essential, with specific reference to medicine. A conventional ITS can give excellent support for medical education where learning terminology, factual information and standard protocols is concerned. But clinicians and researchers can also benefit from complementary learning resources that help to develop informal and tacit knowledge that may guide their judgement. A construal fulfils this role by inviting engagement from learners with different goals, expertise, and experience. Rather than supplying definitive answers, this activity stimulates questions, and forces the learner to reflect upon their knowledge and experience. This is vital in medical education, where educators must learn to cope with emerging science, evolving practice, and ever-changing contexts.

2 Construals of Malaria

A potential application to medical education has been illustrated by developing online construals of human malaria infection [2]. More details of the principles behind this development, the construction process, and the relationship between construals and traditional computer models are given in the full version of this poster paper [1].

Observables of many different kinds are associated with understanding malaria. These can be classified according to the role of the modeller (e.g. clinician, medical researcher, malaria patient), the nature of their observation, and the other agents relevant to the modelling context (e.g. Plasmodium parasites and associated hosts). Two principal construals for malaria infection have been developed: one specific for *P. Vivax* and the other made generic. In making these construals, the modeller potentially has the current status of many relevant observables in mind. These range from the patient's current temperature or likely haemoglobin to the current activity in the blood and the biochemical interactions that are as-of-now occurring. Though it is plausible that observables of all these kinds inform the modeller's mental model, the modeller cannot actually apprehend them all in one and the same state.

The *constructed* nature of the modeller's perception of state highlights the fundamental character of a construal, and its potential educational role. The correspondence between the modeller's experience of the construal and its referent is always necessarily incomplete and in some aspects uncertain. This uncertainty is a positive rather than an undesirable feature in the learning context. The modeller can maintain the construal through carrying out exploratory interactions and rehearsing interpretations. This activity provokes questions, and may lead them to qualify their mental model in different ways, whether consolidating, refining, or softening the perceived connection between the virtual and real world. Of their nature, construals also echo the learning activity underlying the actual historical development of understanding about malaria.

3 Conclusion

EM construals for medical education are still at an early stage of development. They lack the computational ingenuity of techniques to support adaptation in conventional ITSs, but have rich promise for adaptation and co-adaptation from the *human* perspective. Suitably developed, EM tools may be a useful collaborative vehicle for helping the many participants in the learning context to communicate, critique, refine and share their mental models. They will also promote that blending of virtual and real experience, of technical problem-solving with creative problematisation, and of scientific and human perspectives that is vital to fields such as medicine.

References

1. Beynon, M., Beynon, W.: Construals as a Complement to Intelligent Tutoring Systems in Medical Education, Computer Science RR #449, University of Warwick (2012)
2. (March 14, 2012) ,
<http://go.warwick.ac.uk/em/publications/papers/119>

Supporting Social Deliberative Skills in Online Classroom Dialogues: Preliminary Results Using Automated Text Analysis*

Tom Murray, Beverly Park Woolf,
Xiaoxi Xu, Stefanie Shipe, Scott Howard, and Leah Wing

University of Massachusetts, Amherst, MA
tmurray@cs.umass.edu

Abstract. We describe a study in which we tested features of online dialogue software meant to scaffold "social deliberative skills." In addition to hand coding of the dialogue text we are exploring the use of automated text analysis tools (LIWC and Coh-Metrix) to identify relevant features, and to be used in a Facilitator Dashboard tool in development.

Keywords: online deliberation, social deliberative skills, text analysis.

Social Deliberative Skills. The capacity to deliberate with others about complex issues where interlocutors have differing viewpoints is paramount for so many life contexts, including citizen engagement, collaborative problem solving, knowledge building, and negotiating needs in personal relationships. We use the term "social deliberative skills" to point to a set of skills that are important to success in such deliberative contexts. Social deliberative skills include the skills of perspective-taking, social inquiry (perspective-seeking), meta-dialog, and reflecting on how one's biases and emotions are impacting a deliberative process. Our research is looking into how to support higher quality deliberations in online contexts by supporting such skills. We are investigating a number of deliberative contexts, including online dispute resolution (for e-commerce, divorce settlements, and workplace disputes), online civic engagement, and online discussion forums on topics of importance to participants (including college students).

We are interested in supporting higher quality deliberations in both facilitated (with mediators, arbitrators, moderators, etc.) and non-facilitated dialogues. For facilitated dialogues we are designing a Facilitator's Dashboard that will allow a facilitator to get a birds-eye-view of one or more dialogues, and monitor key indicators to help decide when and where to make useful interventions.

A key technology in our research is automated text analysis to characterize participant posts along a number of relevant dimensions, such as emotional tone, self-reflection, topic abstraction, etc. We are investigating whether text analysis methods developed by Pennabaker et al. (2007) and Graesser et al. (2011) can measure

* An extended version of the paper can be found at www.tommurray.us/socialdeliberativeskills.

characteristics relevant to supporting quality deliberation, so that this automated analysis can be used to provide real-time assessment of online dialogue.

Method. Forty college students in students in an Alternative Dispute Mediation courses were assigned a series of discussions to be had online. Students engaged in a sequence of three online dialogues, one per week, over three weeks on topics that they proposed as being controversial and interesting: marijuana legalization, sexual choices, and capital punishment. For the online discussions we used the Mediem software created by Idealogue Inc., which has a discussion forum format with a number of features to support deeper engagement and reflective dialogue.

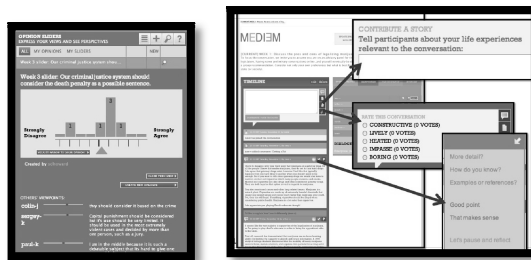


Fig. 1. A, B: Mediem Sliders and Reflective Tools

Figure 1a shows the detailed view of the Opinion Slider feature, which gives a summary view of where participants stand on an issue. Figure 2b illustrates the Story feature, which gives participants a special place to say how the issue at hand relates to them personally; the Conversation Thermometer, a meta-dialogue tool that allows participants to rate (vote on) the quality of the conversation at any time, and the Contribution Tag feature, which allows participants to give brief comments on other's contributions. There were 3 experimental conditions: Condition V used the "vanilla" version of the software with no reflective features; condition S used the Slider feature, and condition R used the other three Reflective features (but not the sliders). Data sources included a post-survey and records of text and tool use from the software.

Results. Data is still being processed and will be reported at the conference. We are analyzing social network connectivity, post-survey data, human coding of the dialogue text, and automatic coding of the dialogue text. Initial analysis using ANOVA to measure differences between the control and experimental groups looked promising but re-analysis using mixed effects methods (including hierarchical linear modeling) negated some of the significance findings. Further inspection of the data showed that students did not use the special features of the software as much as was hoped, and we are planning an additional study this spring to remedy that. Automatic text analysis, especially that produced by the LIWC system, showed promise for computational identification of features of the dialogue that would be of interest to facilitators in the Facilitator Dashboard.

References

- Graesser, A.C., McNamara, D.S., Kulikowich, J.: Coh-Metrix: Providing multilevel analyses of text characteristics. *Educational Researcher* 40(5), 223–234 (2011)
- Pennebaker, J.W., Chung, C.K., Ireland, M., Gonzales, A.L., Booth, R.J.: The development and psychometric properties of LIWC 2007, Austin, TX (2007), <http://www.LIWC.net>

Using Time Pressure to Promote Mathematical Fluency

Steve Ritter¹, Tristan Nixon¹, Derek Lomas², John Stamper², and Dixie Ching³

¹Carnegie Learning

{sritter, tnixon}@carnegielearning.com

²Carnegie Mellon University

dereklomas@gmail.com, john@stamper.org

³New York University

dixie@nyu.edu

Abstract. Time pressure helps students practice efficient strategies. We report strong effects from using games to promote fluency in mathematics.

Keywords: mathematics, evaluation, educational games, fluency, retention, number sense.

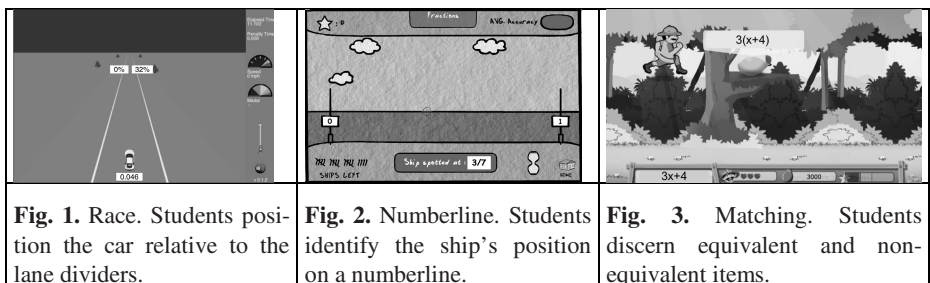
1 Introduction

There is a misperception that building fluency requires rote practice. Often, fluency represents the ability to rapidly recognize and apply an appropriate strategy for solving a problem [1]. For example, when deciding whether $\frac{1}{3}$ is greater than $\frac{1}{17}$, it is more efficient to picture pies than to form common denominators. The number sense depends on developing these kinds of reasoning abilities.

Some students possess the appropriate mathematical knowledge to solve mathematical problems but are relatively inflexible in their strategy selection and so pick inefficient (but correct) strategies [2]. The imposition of time pressure can force students to consider alternative strategies [3].

2 Game Frameworks

Our experiments incorporate three game “frameworks,” used for relative comparisons (race), absolute magnitude (numberline) and equivalence (matching).



3 Evaluation

3.1 Participants and Method

Our evaluation was conducted at a small Catholic liberal arts university that focuses is on women’s education. Sixteen of the 18 participants were women.

In each of five weeks, students played games for approximately one-half hour. They took short paper-and-pencil pre- and post-tests before and after playing, as well as a delayed test one week later. Tests were timed and designed to contain more questions than students could answer, so our main outcome is the number of questions answered correctly.

3.2 Results and Discussion

Improvements are shown in Table 1. Effect sizes are quite large, ranging from 0.4 to 2.4, indicating that these results are not only significant but substantial.

Table 1. Mean (standard deviation) correct on immediate and delayed post tests. Pretest for delayed post includes only students who took the delayed posttest. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

	<i>Ex. 1</i>		<i>Ex. 2</i>		<i>Ex. 3</i>		<i>Ex. 4</i>		<i>Ex. 5</i>	
	<i>Pre</i>	<i>Post</i>	<i>Pre</i>	<i>Post</i>	<i>Pre</i>	<i>Post</i>	<i>Pre</i>	<i>Post</i>	<i>Pre</i>	<i>Post</i>
Immediate	13.6 (5.9)	23.5*** (5.8)	20.3 (4.8)	30.2*** (7.7)	13.3 (4.6)	15.3 (3.2)	17.6 (7.6)	22.2** (8.0)	6.3 (2.7)	7.4** (2.5)
Delayed	12.4 (3.8)	20.7** (5.7)	19.3 (5.3)	34.0*** (6.7)	13.2 (4.7)	14.2 (4.1)	17.3 (8.1)	22.4* (9.3)	6.4 (2.8)	9.4 (2.8)

We may see such strong results because students are often not learning new strategies. Instead, they are practicing ways of thinking about numbers that they already possess but which have been infrequently accessed. Time pressure imposed in a game context has promise to be a highly effective method for encouraging such practice.

References

1. Rittle-Johnson, B., Star, J.R.: Does comparing solution methods facilitate conceptual and procedural knowledge? An experimental study on learning to solve equations. *Journal of Educational Psychology* 99, 561–574 (2007)
2. Siegler, R.S.: Individual differences in strategy choices: Good Students, no-so-good students, and perfectionists. *Child Development* 59, 833–851 (1988)
3. Siegler, R.S., Lemaire, P.: Older and younger adults’ strategy choices in multiplication: Testing predictions of ASCM using the choice/no choice method. *Journal of Experimental Psychology: General* 126, 71–92 (1997)

Interoperability for ITS: An Ontology of Learning Style Models

Judi McCuaig and Robert Gauthier

University of Guelph,
Guelph, Ontario, Canada

Abstract. Intelligent Tutoring Systems (ITS) often use information about student learning style to inform the decision about what activity or information to present to the learner. However, learning style models are quite different from one another and are not interchangeable. This paper discusses an ontology of learning style models that enables tutoring systems to take advantage of learning styles information from multiple learning style models.

Many different learning style models (LSM), each with a different specialization, are regularly used as components of intelligent tutoring systems. The large number of models creates confusion over the definitions and results in a lack of consistency [5, 1]. The design of the learning styles component of a tutoring system tends to be customized to suit the learning style model chosen by developers, but a better scenario is that the learning style model is chosen to suit the learning situation. However, LSMs are not modular because the relationships between models are complex and there are inconsistencies in definitions [5, 1]. An ontology that defines the generalized elements of learning style models can be used by an automated system to select the most suitable LSM based on learning context, giving the intelligent system more flexibility in learner modelling.

The ontology was created by examining three learning style models: Felder and Silvermans Learning Style Model[2], Kolbs Learning Style Index[4], and the VARK Model[3]. Four common components were found: the learning style model, the learning style dimension, the learning style stereotype, and the learner's learning style which lead to four primary classes in the ontology: Learning Style Models, Detection Methods, User Models, and Adaptations. A learner's learning style is a set of assignments that show where a learner lies on the continuum of each dimension with respect to the stereotypes.

The ontology was populated with the three example learning style models, example adaptation methods and the relationships between them. The ontology can be used to differentiate between suitable adaptations by using multiple learning style models. For example, the Felder and Silverman LSM suggests that both audio and text are useful to a stereotypical verbal learner while the VARK model suggests that audio is strongly connected to auditory learners and text is strongly connected to reading/writing learners. A tutoring system that could use both LSM models could easily make a choice between audio and text versions

of content given the stereotype score for a specific learner from both VARK and Felder-Silverman.

Experimental participants were shown a learning preference profile for a hypothetical student along with an instructional scenario. Two possible adaptations were illustrated, with one indicated as the preferred solution. The participant was asked to assess the validity of the preferred adaptation, given his/her understanding of the hypothetical learner's learning preferences and the instructional context. It was expected that participants would indicate that the ontology-suggested adaptations as being valid more often than they would indicate the other, randomly selected, adaptations were valid.

Precision scores were calculated in the traditional information retrieval fashion by dividing the number of items by the total number available; in this case by dividing the number of 'valids' by the total number of trials. The expectation was that the precision for the ontology-selected suggestions would be higher than the precision for the randomly-selected suggestions.

Table 1 shows the results. With one exception, each participant's precision was greater for the ontology-based suggestions and the median precision score is quite a bit larger for the ontology suggestions (.65 for ontology, .40 for random). These results suggest that the ontology is able to facilitate good decision making about adaptations when learning styles information comes from different learning style models and when some of the LSM information is incomplete.

Table 1. Precision Values for Random and Ontology Suggestions

	Random Ontology		Random		Ontology	
Participant 1	0.75	1.00	Participant 5	0.38	0.67	
Participant 2	0.43	0.50	Participant 6	0.33	0.63	
Participant 3	0.80	0.50	Participant 7	0.20	0.33	
Participant 4	0.33	1.00	Participant 8	0.67	0.80	

References

- [1] Coffield, F., Moseley, D., Hall, E., Ecclestone, K.: Learning styles and pedagogy in post-16 learning (2004)
- [2] Felder, R.M., Silverman, L.K.: Learning and teaching styles in engineering education. *Engineering Education* 78(7), 674–681 (1988)
- [3] Fleming, N.: VARK – a guide to learning styles (2010), <http://www.vark-learn.com/english/index.asp>
- [4] Kolb, A.Y., Kolb, D.A.: The kolb learning style Inventory Version 3.1 2005 technical specifications (2005)
- [5] Reynolds, M.: Learning styles: A critique. *Management Learning* 28(2), 115–133 (1997)

Skill Diaries: Can Periodic Self-assessment Improve Students' Learning with an Intelligent Tutoring System?

Yanjin Long and Vincent Aleven

Human Computer Interaction Institute, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
ylong@cs.cmu.edu, aleven@cs.cmu.edu

Abstract. Metacognitive theories point out that self-assessment can facilitate in-depth reflection and help direct effective self-regulated learning. Yet, not much work has investigated the relationship between students' self-assessment and learning outcomes in Intelligent Tutoring Systems (ITSs). This paper investigates this relationship with classrooms using the Geometry Cognitive Tutor. We designed a skill diary that helps students take advantage of the tutor's open learner model to periodically self-assess their geometry skill. We investigated whether it can support students' self-assessment and learning. In an experiment with 122 high school students, students in the experimental group were prompted periodically to fill out the skill diaries, whereas the control group answered general questions that did not involve active self-assessment. The experimental group performed better on a post-test of geometry skill. Further, the skill diaries helped lower-performing students to significantly improve their self-assessment accuracy and learning outcomes. This paper helps establish the important role of self-assessment in enhancing students' domain-level learning in ITSs.

Keywords: Skill diaries, periodic self-assessment, open learner model.

Self-assessment refers to students' ability to evaluate their learning status (how well they are learning/have learned). This paper investigates the relationship between self-assessment and learning in an ITS in a classroom context in which students learn with a Cognitive Tutor. Recently, many researchers have recognized the potential of Open Learner Models (OLMs) to support students' self-assessment and reflection [1]. Cognitive Tutor has its own built-in OLM, called the "Skillometer," which displays students' learning status in the form of skill bars. A previous study [2] illustrated that simply presenting an OLM by itself may not be an effective way to support self-assessment, and additional scaffolding may be necessary. Therefore, we created a structured skill diary that prompts students to self-assess and reflect (aided by the Skillometer) while they are learning in the tutor. We conducted an experiment to test the hypothesis that periodically using the skill diaries can enhance both students' self-assessment accuracy and their learning of problem solving tasks in the ITS.

A total of 122 students participated and were randomly assigned to two conditions (experimental vs. control) in the study. The experimental group periodically filled out skill diaries during learning while the control group periodically answered general

questions about the section of the tutor curriculum they were working on. Students worked through four such sections in three class periods. Pre- and post-tests were given to the students before and after the tutor sessions, each with two parts. In part I, the to-be-solved problems were shown to the students, and the students were asked to indicate on a 7-point Likert scale: “How confident are you that you can solve this problem”. In part II, students actually solved the problems. We gathered complete data for 95 students, and analyzed students’ pre-test and post-test performance, Cognitive Tutor log data and self-assessment accuracy.

Overall, both groups improved significantly from pre- to post-test (repeated measures ANOVA, $F(1, 93) = 13.103, p = .000$), but the two groups did not differ significantly on the pre-test nor the post-test. We then divided the test items into two categories: reproduction (isomorphic to the problems in the tutor) and transfer problems. We found that the experimental group did significantly better than the control group on the reproduction problems on post-test ($F(1, 93) = 3.861, p = .052$), but we found no significant difference between two groups on transfer problems ($F(1, 93) = .056, p = .814$)¹. We also divided students into higher/lower performing groups based on their pre-test performance. We found that the lower-performing students’ self-assessment accuracy and their test performance on reproduction problems improved significantly from pre- to post-test. Further, analysis of Cognitive Tutor log data revealed that the experimental group students asked for fewer hints but spent more time on each hint. They also made fewer incorrect attempts and spent less time on each step in more difficult tutor sections.

The results suggest that the skill diary can help break students’ illusion of knowing, and bringing students’ attention to unlearned content leads to more deliberate use of help from the tutor. The skill diary also may help keep students alert and motivated to focus on learning. This work helps to empirically establish the important role of self-assessment in enhancing students’ learning from problem-solving tasks in ITSs, and suggests an effective way of increasing students’ self-assessment accuracy.

Acknowledgments. We would like to thank the participating teachers and students. This work was funded by an NSF grant to the Pittsburgh Science of Learning Center (NSF Award SBE0354420).

References

1. Bull, S.: Supporting Learning with Open Learner Models. In: 4th Hellenic Conference: Information and Communication Technologies in Education, Athens (2004)
2. Long, Y., Aleven, V.: Students’ Understanding of Their Student Model. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS, vol. 6738, pp. 179–186. Springer, Heidelberg (2011)

¹ When pre-test score was used as co-variate, the difference between two groups on reproduction problems was on the borderline of significance ($F(1, 92) = 2.747, p = .101$).

An Optimal Assessment of Natural Language Student Input Using Word-to-Word Similarity Metrics

Vasile Rus and Mihai Lintean

Department of Computer Science, The University of Memphis
Memphis, TN, 38152, USA
{vrus,mclinten}@memphis.edu

Abstract. We address in this paper the important task of assessing natural language student input in dialogue-based intelligent tutoring systems. Student input, in the form of dialogue turns called contributions must be understood in order to build an accurate student model which in turn is important for providing adequate feedback and scaffolding. We present a novel, optimal semantic similarity approach based on word-to-word similarity metrics and compare it with a greedy method as well as with a baseline method on one data set from the intelligent tutoring system, AutoTutor.

Keywords: intelligent tutoring, optimal lexico-semantic matching.

1 Background and Results

We model the problem of assessing natural language student input in tutoring systems as a paraphrase identification problem. That is, we have to decide whether a student input has the same meaning as an expert answer. The student input assessment problem has been also modeled as a textual entailment task in the past [1].

Our novel method to assess a student contribution against an expert-generated answer relies on the compositionality principle and the sailor assignment algorithm that was proposed to solve the assignment problem, a well-known combinatorial optimization problem [2]. The sailor assignment algorithm optimally assigns sailors to ships based on the fitness of the sailors' skills to the ships' needs. In our case, we would like to optimally match words in the student input (the sailors) to words in the expert-generated answer (the ships) based on how well the words in student input (the sailors) fit the words in the expert answer (the ships). The fitness between the words is nothing else but their similarity according to some metric of word similarity.

The methods proposed so far that rely on the principle of compositionality to compute the semantic similarity of longer texts have been primarily greedy methods. To the best of our knowledge, nobody proposed an optimal solution based on the principle of compositionality and word-to-word similarity metrics (from the WordNet Similarity package) for the student input assessment problem. It should be noted that the optimal method is generally applicable to compute the similarity of any two texts.

The AutoTutor dataset we used contains 125 student contribution – expert answer pairs and the correct paraphrase judgment, TRUE or FALSE, as assigned by human

experts. The target domain is conceptual physics. The dataset contains 36 FALSE and 89 TRUE entailment pairs, i.e. a 28.8% versus 71.2% split (see [1] for details).

To evaluate the performance of our methods, we compare the methods' judgments with the expert judgments. The percentage of matching judgments provides the *accuracy* of the run, i.e. the fraction of correct responses. We also report kappa statistics which indicate agreement between our methods' output and the human-expert judgments for each instance while taking into account chance agreement.

Tables 1 summarizes the results on the original AutoTutor data (from Rus & Graesser, 2006; Table 1). Since the AutoTutor dataset is small, we only report results on it as a whole, i.e. only training. We also report a baseline method of guessing all the time the dominant class in the dataset (which is TRUE paraphrase for all three datasets) and a pure greedy method (*Greedy* label in the first column of the tables).

Overall, the optimum method offered better performance in terms of accuracy and kappa statistics. One reason for why they are so closed is that in optimum matching we have one-to-one word matches while in the greedy matching many-to-one matches are possible. Another reason for why the raw scores are close for greedy and optimum is the fact that student input and expert answers in both the AutoTutor and ULPC corpora are sharing many words in common (>.50). This is the case because the dialogue is highly contextualized around a given, e.g. physics, problem. In the answer, both students and experts refer to the entities and interactions in the problem statement which leads to high identical word overlap. Identical words lead to perfect word-to-word similarity scores (=1.00) increasing the overall similarity score of the two sentences in both the greedy and optimum method.

Table 1. Accuracy/kappa on AutoTutor data (* indicates statistical significance over the baseline method at $p < 0.005$ level)

ID	RES	LCH	JCN	LSA	Path	Lin	WUP
<i>Baseline</i>	.712	.712	.712	.712	.712	.712	.712
Greedy	.736/.153	.752/.204	.760/.298	.744/.365	.752/.221	.744/.354	.760/.298
Optimal	.744/.236	.752/.204	.760/.298	.744/.221	.752/.334	.752/.204	.784*/.409*

Acknowledgments. This research was supported in part by the Institute for Education Sciences under award R305A100875.

References

1. Rus, V., Graesser, A.C.: Deeper Natural Language Processing for Evaluating Student Answers in Intelligent Tutoring Systems. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence, AAAI 2006 (2006)
2. Kuhn, H.W.: The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly* 2, 83–97 (1955); Kuhn's original publication
3. Graesser, A., Olney, A., Hayes, B.C., Chipman, P.: Autotutor: A cognitive system that simulates a tutor that facilitates learning through mixed-initiative dialogue. In: *Cognitive Systems: Human Cognitive Models in System Design*. Erlbaum, Mahwah (2005)

Facilitating Co-adaptation of Technology and Education through the Creation of an Open-Source Repository of Interoperable Code

Philip I. Pavlik Jr., Jaclyn Maass, Vasile Rus, and Andrew M. Olney

University Of Memphis, Memphis, Tennessee, USA
{ppavlik, jkmaass, vrus, aolney}@memphis.edu

Abstract. Co-adaptation of technology and education is a daunting challenge because of the limited resources available to individual researchers. Without keeping ITS systems runnable on the latest computing platforms, the educational technology we develop will become obsolete by default. Without keeping ITS systems current with the latest advances in experimentation and educational research, our educational technology will become obsolete by design. To simultaneously address these challenges, we propose the creation of an open-source research consortium focused on educational code-sharing that will speed the co-adaptation of technology and education, allow for more cumulative progress in our discipline, and result in faster progress for individual projects and researchers.

Keywords: co-adaptation, intelligent tutoring systems, e-learning, instructional design.

1 Introduction

The evolution of ITS work hinges on the co-adaptive relationship between education and technology. Educational theories are tested and validated through experimentation, and are manifested in some form of technology or task that provides the examples of that theory in action. Once a technology is validated by this sort of grounded experimentation, it can be applied within the classroom. However, if that technology is to have a broad effect in various educational environments, it must be generalized by providing multiple examples of its instantiation in different contexts. It is this progressive cycle of the experimental validation and adoption of new technologies for learning that drives ITS research. Unfortunately, because of the difficulty of generalizing the technological products, this co-adaptation of education and technology can move very slowly.

As early as 1998, Ritter and Blessing identified a key way to speed up this process, which was to design educational systems with a component-architecture according to unified standards that would allow these systems to *work together*. In their vision, one way to do this was with off-the-shelf components integrated through translation tools such as those presented in their paper [1]. Their vision is very similar to what we propose: *to form specific standards for communication by bringing together*

researchers from around the world in a consortium and then provide a community infrastructure to support these standards. This Consortium for Open-source Development of Educational Software (CODES) will create a repository of interoperable/standardized code to foster communication and cooperation between sub-disciplines within the educational software research community.

We are not the first to consider the need for this community portal for sharing and finding educational software [2], but we are the first to attempt to create a consortium with the combination of the three following critical features. First, our proposed solution will create a new repository of standardized educational software code in order to enhance and accelerate research. A second important merit of the overall proposal will be the unification of the scientific effort (a function of the standardization) within the educational software community, which will provide similar benefits to those discussed by Allen Newell [3] in the context of unified models of cognition. Third, our proposal will, to enhance adoption, use a beginner-friendly application development approach instantiated by a final product that offers the researcher both individual components and full running examples of educational systems which the researcher can modify, improve, and dissect rather than having to start from scratch. Overall, the sharing, unification, and beginner friendly approach fostered by this proposal will enable faster development, allow for more constrained validation of theories, and encourage participation in our field.

2 Conclusion

We advocate a radical solution to the challenges of co-adaptation of education and technology. We have proposed that the field as a whole should move away from fully independent bodies of research towards the goals of greater technological and theoretical unification. Our aim is to engage the community in creating the standards necessary for such unification, as well as helping to build, share, and continuously improve educational software through the open-source projects advocated by this proposal. Such an integrated effort specifically fosters co-adaptation of education and technology by grounding theoretical educational developments in specific shared technologies, and conversely, by bootstrapping the adoption of new technologies in education with much greater speed than has previously been possible.

References

1. Ritter, S., Blessing, S.B.: Authoring Tools for Component-Based Learning Environments. *The Journal of the Learning Sciences* 7, 107–132 (1998)
2. Holton, D.L.: Toward a Nation of Educoders: A Roadmap for Sustainably Broadening and Improving Open Source Educational Software. In: Burton, B.J. (ed.) *Open-Source Solutions in Education: Theory and Practice*, pp. 47–61. Informing Science Press, Santa Rosa (2010)
3. Newell, A.: *Unified Theories of Cognition*. Harvard University Press, Cambridge (1990)

A Low-Cost Scalable Solution for Monitoring Affective State of Students in E-learning Environment Using Mouse and Keystroke Data

Po-Ming Lee¹, Wei-Hsuan Tsui², and Tzu-Chien Hsiao^{1,2,*}

¹ Institute of Computer Science and Engineering,
National Chiao Tung University, Taiwan (R.O.C.)

² Institute of Biomedical Engineering, College of Computer Science,
National Chiao Tung University, Taiwan (R.O.C.)
labview@cs.nctu.edu.tw

Abstract. This study proposed a user-independent intelligent system that reports the affective state of students in a non-intrusive and low-cost manner by utilizing mouse record and keystroke data collected in dynamic world. A scalable client-server architecture for student affective state monitoring in e-learning environment is also demonstrated.

Keywords: E-learning, Affect Detection, Keystroke, Mouse Record, Client-Server Architecture.

A Low-Cost and Scalable System for Affect Monitoring (LSAM) is proposed based on a recent proposed affect recognition technique [1]. We designed a scenario that in an e-learning environment, an eTutor/eLecturer teaching, for example, C# programming, can use the computer to give the lecture, and also inquire the emotional status of students which stayed at home simultaneously. The affective information of students, whenever feeling bored, being frustrated, or being excited, can be resolved and transmitted to the lecturer without bothering the students in changing the manner of using the ordinary devices, or bothering on remembering to setup and turn on additional devices. Based on the provided information, the lecturer can control the challenge level of materials by for example, decreasing the speed in teaching, or giving more examples for the described concept. By using LSAM, the maintenance on optimal experience of learning of students become feasible [2]. The Figure 1 illustrates the user interface that was displayed on the screen used by the lecturer. The content used for presentation is displayed in the middle, and the affective information of students is displayed in message boxes. The onset of the message boxes was configured to notify the lecturer by displaying the affective status of student and also a video capture from video camera (if available) only when extreme emotional responses occur.

The Figure 1 also illustrated the system architecture of LSAM. The client in LSAM can be a personal computer or a laptop, the LSAM client side software for Tablet PCs may also be implemented because of the similar function provides by touch panel and

* Corresponding Author.

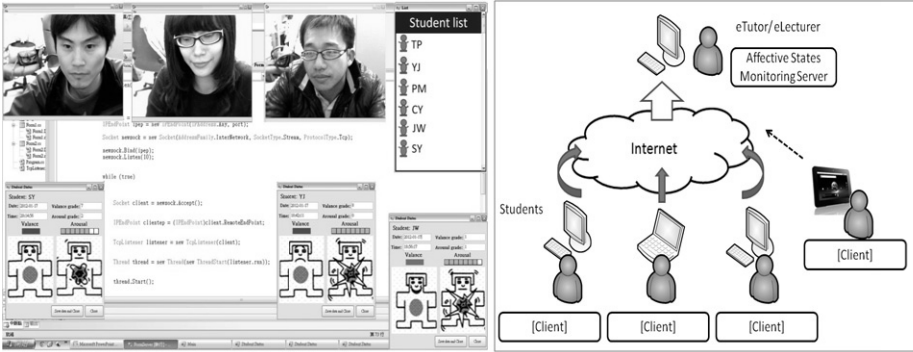


Fig. 1. The user interface the lecturer sees on the monitor and the system architecture of LSAM

mouse; the data collected from touch panel contains the affective information as well as mouse movement data does. The software installed for LSAM client is run in the background and starts every time in the beginning of the startup of the operating system, the keystroke and mouse movement data is collected all the time in a time resolution of 100 nanoseconds.

Acknowledgements. This work was fully supported by Taiwan National Science Council under Grant Number: NSC-100-2220-E-009-041 and NSC-100-2627-E-010-001. This work was also supported in part by the UST-UCSD International Center of Excellence in Advanced Bioengineering sponsored by the Taiwan National Science Council I-RiCE Program under Grant Number: NSC-100-2911-I-009-101.

References

1. Epp, C., Lippold, M., Mandryk, R.L.: Identifying emotional states using keystroke dynamics. In: Proceedings of the 2011 Annual Conference on Human factors in Computing Systems, pp. 715–724. ACM, Vancouver (2011)
2. Csikszentmihalyi, M.: Flow: The Psychology of Optimal Experience. Harper Perennial Modern Classics (2008)

Impact of an Adaptive Tutorial on Student Learning

Fethi A. Inan¹, Fatih Ari¹, Raymond Flores², Amani Zaier¹, and Ismahan Arslan-Ari¹

¹ Educational Instructional Technology, Texas Tech University, USA

² Middle/Secondary Level Mathematics Education, Wichita State University, USA
{fethi.inan, fatih.ari, amani.zaier, ismahan.arslan}@ttu.edu,
raymond.flores@wichita.edu

Abstract. In this study, we examined the effectiveness of an adaptive tutorial on college students' learning outcomes, mainly, learning performance, motivation, and study time. Two versions of the tutorial were developed; adaptive and non-adaptive. A total of 134 undergraduate students were randomly assigned to adaptive (n=74) or non-adaptive (n=60). Our results revealed that the adaptive group had a significantly higher knowledge gains than the non-adaptive group.

Keywords: Adaptive Web-Based Learning Environment, Individual Differences, Adaptive Instruction, Adaptive Hypermedia, Online Learning.

1 Purpose of Study

The purpose of this study was to investigate the effect of an adaptive web-based tutorial on students' performances, motivation and learning time. Two versions of the tutorial were developed; adaptive and non-adaptive. The main question we sought to answer was which groups (adaptive and non-adaptive) would achieve the maximum benefit from the tutorial. The independent variable was the group and the dependent measures were knowledge, motivation, and learning time.

2 Participants

A total of 134 undergraduate students (79 females and 55 males) were randomly assigned by computer to adaptive (n=74) or non-adaptive (n=60). Participants came from six sections of an undergraduate introductory technology course in a large southwestern American university. Students came from different disciplines such as Exercise and Sports Science, Sociology, Rehabilitation, etc. and their ages ranged from 18 and 40 with a median age of 19.

3 Adaptive Tutorial

A web-based adaptive tutorial on basic introductory statistics was developed by utilizing adaptive hypermedia methods with strategies proposed by instructional theory and motivation models [1]. Once students enter the tutorial, their prior knowledge and motivation levels were assessed. Based on gathered data, the adaptive tutorial auto-

matically incorporated relevant adaptive strategies and provided appropriate content and examples to corresponding clusters. Once students finish the tutorial, their knowledge and motivation levels were assessed again.

4 Data Collection and Instruments

- Achievement was measured using a locally developed 20 item multiple choice instrument over the introductory statistics topics covered in the tutorial.
- Items adapted from the Instructional Materials Motivational Scale (IMMS) were used to measure student motivation level [2]. Cronbach's alpha for the IMMS ranged from .61 to .81 [2].
- System logs were used to analyze the time spent on task.

5 Results

A series of ANOVAs and ANCOVAs were conducted to answer the research questions. Results revealed that there was a significant difference between adaptive ($M=6.63$; $SD=2.50$) and non-adaptive ($M=5.20$; $SD=2.51$) groups in terms of knowledge differences, $F(1,131)=10.299$, $p=.002$. However, there was no significant difference in terms of student post motivation scores. In addition, the students in the adaptive group spent significantly more time on the tutorial than the students in the non-adaptive group, $F(1,132)=4.249$, $p=.041$.

Acknowledgements. This research study was funded by grants from Texas Tech University College of Education and EDUCAUSE through the Next Generation Learning Challenges.

References

1. Inan, F.A., Flores, R., Ari, F., Arslan-Ari, I.: Towards Individualized Online Learning: The Design and Development of an Adaptive Web Based Learning Environment. *Journal of Interactive Learning Research* 12(4), 467–489 (2011)
2. Keller, J.M.: Development and use of the ARCS model of instructional design. *Journal of Instructional Development* 10(3), 2–10 (1987)

Technology Enhanced Learning Program That Makes Thinking the Outside to Train Meta-cognitive Skill through Knowledge Co-creation Discussion

Kazuhiisa Seta¹, Liang Cui², Mitsuru Ikeda², and Noriyuki Matsuda³

¹ 1-1, Naka-ku, Gakuen-cho, Sakai, Osaka, 599-8531, Japan

² 1-1, Asahi-dai, Nomi, Ishikawa, 923-1292, Japan

³ Sakaedani 930, Wakayama-city 640-8510, Japan

seta@mi.s.osakafu-u.ac.jp, {cui-liang, ikeda}@jaist.ac.jp,
matsuda@sys.wakayama-u.ac.jp

Abstract. We aim at developing a semester course to train thinking skills for 1st year bachelor students who just entered a university. Our goal in the educational program is to let learners perceive the *isomorphism* between thinking process for self-dialogue and one for discussion, and motivate them to learn the logical structure of self-dialogue through workshop and to acquire meta-cognition abilities through leading the discussion. We give knowledge co-creation task with a case that requires them to perform their meta-cognitive activities.

Keywords: meta-cognition, make thinking outside, knowledge co-creation discussion, isomorphism between thinking process for self-dialogue and one for discussion, collaborative learning.

1 Introduction

In our research, we aim at developing a semester lecture course to train thinking skills for 1st year bachelor students who just entered a university. Training of meta-cognitive skill is not embedded into the current school curriculum, in general, although it is trained by displaying the skill: they do not tend to be aware of the necessity to consciously monitor and control their own thinking processes. Needless to say, the earlier learners are able to be aware of the necessity of conducting “thinking about thinking,” the more they will learn knowledge in depth in the university: it changes their learning processes and/or study habit [1, 2].

We give knowledge co-creation task with a case that requires them to perform their meta-cognitive activities. The learning materials in this educational program are: An educational software to support learners to examine their thinking processes for self-dialogue: Sizhi, textbook to explain the details of the workshop, and a collection of reports/questionnaires that stimulate learners’ meta-level thinking on their thinking process.

2 Knowledge Co-creation Workshop Using Sizhi in the Lecture Course

The Sizhi (Fig. 1) is a learning environment designed for developing the learner’s ability to conduct logical thinking for self-dialogue and to appropriately reflect on ones’ thinking process by oneself. We measured the learners’ self-evaluation of how efficiently they have been using thinking skills, and how motivated they learn thinking skills are (cognition of importance). The learners answered about the Target which is related with knowledge building, and the Distracter (finding flaws in oneself). As a result, we found the Target became higher, and the Distracter became lower as the program progressed.

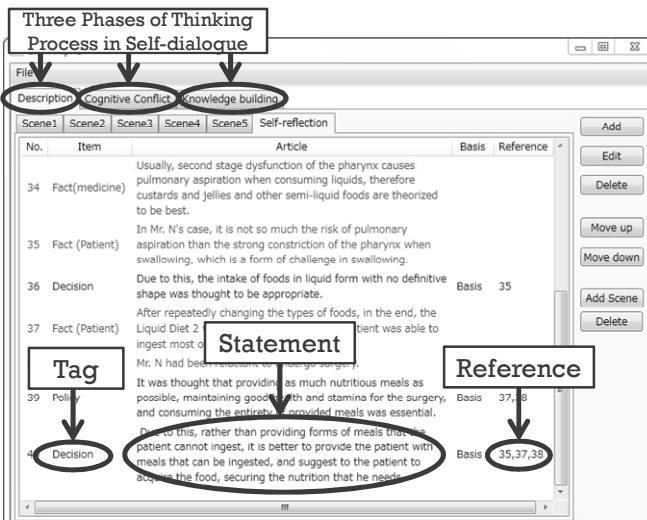


Fig. 1. A Screen Image of Sizhi (Description Phase)

3 Concluding Remarks

We overviewed a thinking skill development program using Sizhi. We gained the favorable data that reveals the intended educational benefits.

References

1. Brown, A.L., Bransford, J.D., Ferrara, R.A., Campione, J.C.: Learning, Remembering, and Understanding. In: Markman, E.M., Flavell, J.H. (eds.) Handbook of Child Psychology, Cognitive Development, vol. 3, pp. 515–529. Wiley, New York (1983)
2. Flavell, J.H.: Metacognitive aspects of problem solving. In: Resnick, L. (ed.) The Nature of Intelligence, pp. 231–235. Lawrence Erlbaum Associates, Hillsdale (1976)

Open Student Models to Enhance Blended-Learning

Maite Martín¹, Ainhoa Álvarez¹, David Reina¹, Isabel Fernández-Castro¹,
Maite Urretavizcaya¹, and Susan Bull²

¹ Department of Languages and Computer Systems
University of the Basque Country, UPV/EHU, Spain
maite.martinr@ehu.es

² Elec., Elec. and Computer Engineering, Univ. of Birmingham, UK
s.bull@bham.ac.uk

Abstract. This work describes some experiences developed with the aim of evaluating the usefulness of opening student models in blended learning contexts. We have enriched the Student Model of the MAgAdI learning environment and opened it to both students and teachers. The preliminary results show that the former improve their reflection ability while the latter receive support to plan and monitor the students' learning process.

Keywords: open student model, user-dependent model views, blended model.

1 Interactions in Student Model for Feedback

Current educational trends increasingly involve the blended use of technological/social environments (Blended-Learning) to improve learning. Our hypothesis is that the more rich, fluent and accessible the interaction information is for students and teachers, the better the students' learning results are likely to be. Therefore, we propose to enrich the explicit student model of the computer environment with interaction information [1]; and then opening/showing it to both teachers and students in order to *improve students' learning* and the *adaptation of teachers* to students' needs. Thus, we have provided a visual representation for the student model to cover four learning objectives [2]: encourage *reflection*, *plan & monitor* the learning process, enable students to *navigate the learning system* and increase the *accuracy of the Student Model* data.

Some experiences have been developed about extending the MAgAdI environment [3], so that teachers and students can use editing/visualization tools to access its Open Student Model (OSM). The model contains personal characteristics, learning preferences and a layered overlay model with knowledge levels and properties of the acquisition process of each element. The opening mechanism consists of graphical organization tools, temporal measurements, color codes, representative icons, skill meters and visibility tools. Additionally, as described in [4] and to answer students' requests, the OSM has been enriched with students' answers to evaluation resources.

OSM can be accessed directly by students, and also can provide personalized contextual information to enrich the student learning environment. In the *direct* use, students follow a topic tree structure to access information of specific courses

(*reflection* and *plan&monitor*). However, the contextual information provided depends on the learning situation. So, when a task is scheduled for the student, the definition of the task is shown together with his/her knowledge level of the task's requisites (*navigation* and *plan&monitor*). Otherwise, when the student is working freely, the colors on the topic tree show the student's knowledge level of every topic and the state of every evaluation resource: passed, failed or not tried (*navigation*).

Concerning teachers, the aim of OSM is to produce a strong cohesion among learning strategies, be these promoted by teachers or by MAgAdI. To facilitate this, OSM allows teachers inspection (*reflection*) and provides updating mechanisms (*accuracy*). Besides, OSM helps teachers to plan following face-to-face (F2F) lessons and to monitor and guide the students' learning process (*plan & monitor*).

2 Evaluation and Conclusions

OSM evaluation experiences started in December 2010, in the "Data Base Development" course (Univ. of the Basque Country) with 18 students and their teacher. Those experiences tried to discover main *usefulness* and *usability* deficiencies; in addition, they explored the effects on teacher lectures' *adaptation*. Students attended to regular F2F lectures and used MAgAdI as a support for course revision prior to the first semester exam.

A students' survey and an interview with the teacher were carried out to obtain opinions about both the information accessible and the usability of the interface. The results were promising with both types of user; overall, the teacher found it interesting and very useful to access the OSM. However work remains on issues such as group information, shared information and privacy. Most of the students (94%) found that the tool was *easy to use* and that the functionality of *seeing the learning activities together with the results obtained* was very interesting. They stated also that *including the performance feedback gives them trust in the OSM*. Finally, students *requested for information about the class and peers' performance*. Therefore, a Student Group Model design is included in our research agenda.

Acknowledgments. This work has been partially supported by Spanish MEC TIN2009-14380 and the Basque Government IT421-10.

References

1. Martín, M., Álvarez, A., Fernández-Castro, I., Reina, D., Urretavizcaya, M.: Experiences in Visualizing the Analysis of Blended-Learning Interactions to Support Teachers. In: Int. Conf. on Advanced Learning Technologies (ICALT 2011), pp. 265–266 (2011)
2. Bull, S., Kay, J.: Student Models that Invite the Learner. In: The SMILI Open Learner Modelling Framework. Int. Journal of Artificial Intelligence in Education, vol. 17, pp. 89–120 (2007)
3. Álvarez, A., Ruíz, S., Martín, M., Fernández-Castro, I., Urretavizcaya, M.: MAGADI: a Blended-Learning Framework for Overall Learning. In: Int. Conf. on Artificial Intelligence in Education (AIED 2009), pp. 557–564. IOS Press (2009)
4. Lazarinis, F., Retalis, S.: Analyze Me: Open Learner Model in an Adaptive Web Testing System. Int. J. Artif. Intell. Ed. 17, 255 (2007)

ZooQuest: A Mobile Game-Based Learning Application for Fifth Graders

Gerard Veenhof, Jacobijn Sandberg, and Marinus Maris

University of Amsterdam, Informatics Institute
gerard.veenhof@gmail.com

Abstract. This study examined ZooQuest, a mobile game that supported fifth graders in the process of learning English as a second language. ZooQuest embedded the Mobile English Learning (MEL) application and was compared to MEL as a stand-alone application. Two groups were compared in a quasi-experimental pre- and posttest design. Fifth graders that used the ZooQuest application spent more time on learning at home than fifth graders that used the MEL application and obtained significant better learning results on the posttest than they did on the pretest. The ZooQuest application demonstrated its benefits in the practice of language learning outside school.

Keywords: mobile learning, serious games, motivation, informal learning.

1 Introduction

The study *Mobile English Learning (MEL)* by Sandberg et al. (2011) demonstrated that fifth graders are motivated to learn with a mobile application on voluntary basis. Sandberg et al. (2011) concluded that formal learning at school can be enhanced by informal mobile learning, outside school. Although using the MEL application resulted in significant learning gains, within 15 days of monitoring, the number of students using the application decreased, as well as the average playtime per student. A mobile application in the style of a serious game can offer learning opportunities to keep the learner engaged and motivated. This study investigated the added value of embedding the original MEL application in a game called ZooQuest.

2 Method

A game-based learning environment depends on its structural design that is formed by different game characteristics. Garris, Ahlers & Driskell (2002) categorized various game characteristics in terms of 6 broad dimensions: fantasy, rules/goals, sensory stimuli, challenge, mystery and control. The dimensions were incorporated in ZooQuest, which added a surrounding game layer to the MEL application. The question central to this study was whether the ZooQuest application lead to better learning results and more motivation compared to usage of the MEL application.

A quasi-experimental pre- and posttest design was adopted. Two groups of students were compared. Each group represented a specific condition. The first

condition, a control condition, was derived from the initial MEL experiment. Condition 1 consisted of students that 1) took English lessons at school and 2) used the MEL application at the zoo and 3) used the MEL application at home for 2 weeks. Condition 2 consisted of students that solely used the ZooQuest application at home for 2 weeks. The study involved a total of 43 fifth graders (27 boys, 22 girls) from 2 different primary schools. Ages ranged from 8-10. The pre- and posttest consisted an English vocabulary test that measured passive and active English word knowledge of the students. The test relied on 50 target words divided over 6 categories: animal names, animal habitat, animal, animal characteristics, animal behavior and 'abstract' words.

During the 2 week learning phase, subjects in both conditions received a personal smartphone with either the MEL application or the ZooQuest application.

3 Results

A dependent samples *t*-test explained that both groups scored significantly higher on the posttest than they did on the pretest (for both passive and active world knowledge). Condition 1 (MEL) outperformed condition 2 (ZooQuest) in active word knowledge, however, there was no significant effect between the conditions in comparing the results from the passive word knowledge test. The students from condition 1 spent additional time at school during which active use of English words was practiced. This may explain the effect found for active word knowledge. Students in condition 2 spent more time in the MEL environment at home than students in condition 1.

4 Conclusion

The ZooQuest environment motivated the subjects to spend more time in the MEL-environment, however, students that used the ZooQuest application finished the ZooQuest game too soon (within 1 week). In subsequent research, the ZooQuest game should be adjusted and extended in order to sustain student motivation. Overall, the ZooQuest application demonstrated its benefits in the practice of language learning outside school.¹

References

1. Sandberg, J., Maris, M., De Geus, K.: Mobile English learning: An evidence-based study with fifth graders. *Computers & Education* 57(1), 1334–1347 (2011)
2. Garris, R., Ahlers, R., Driskell, J.E.: Games, Motivation, and Learning: A Research and Practice Model. *Simulation & Gaming* 33(4), 441–467 (2002)

¹ A full version of this paper can be found at www.gerardveenhof.nl

Drawing-Based Modeling for Early Science Education

Wouter R. van Joolingen, Lars Bollen, Frank Leenaars, and Hannie Gijlers

University of Twente
w.r.vanjoelingen@utwente.nl

Abstract. Creating models is at the heart of any scientific endeavor and therefore should have a place in science curricula. We present three approaches, a collaborative drawing tool to support scientific dialogue, a domain specific tool providing intelligent support for learning about gear systems as well as a free-hand drawing tool to support learner created animation.

Keywords: simulation, modeling, sketch recognition, exercise selection.

The creation, modification and evaluation of models are core ingredients of a scientific world view [1]. Trying to grasp phenomena by modeling them and then investigating those models through reasoning and simulation is an important way of building scientific knowledge. Representations are the mediating link between mental models of the learners and real world systems. Consequently, an effective representation for modeling is one in which the properties of a phenomena and their relationships are made explicit and visible for learners. In the current poster we explore the benefits of *drawing* for modeling, in order to support learners in expressing their models and engaging in a realistic cycle of representing, executing and evaluating models. We present systems for collaborative drawing in a pre-modeling stage, for domain specific drawing-based modeling and a system in which the drawing “talks back” when the learner specifies drawing elements, their properties and relations.

As drawing facilitates idea sharing, disambiguation of conceptual understanding, and assists students in attaining a shared focus [2, 3], there is benefit in creating collaborative drawing environments. To fully benefit from *collaborative drawing*, it is important that students engage in task-focused [4] and elaborated meaning making activities. Two possible means of supporting the drawing and collaborative processes were investigated [5]: awareness support and scripting. In the first case the learners were prompted on missing elements in their drawings, in the second, the script made learners create individual drawings first, to serve as input for a joint drawing. Study findings indicate that students in the scripted condition perform significantly better on the concept recognition test and drawing quality than their peers in the control group.

With *GearSketch* [6] young learners can explore the domain of gears and chains by creating simulations. As such simulations require precision drawings, learners are assisted by converting circles to gears as well as automatic snapping of gears and shrinking of chains. *GearSketch* has an internal representation of gears and chains to compute turning speeds and directions of gears and chains. *GearSketch* offers learners

integrated instructions, questions to answer and puzzles to solve. These offer students guidance in their exploration of the gears domain. Puzzle selection is done based on a Bayesian learner model. In a study with 78 fifth grade students, the effectiveness of a version of GearSketch with simulation-based support was compared to a version without this support. These results show that simulation-based support in a digital drawing environment can lead to higher learning gains.

Our drawing and modeling tool *SimSketch* bridges the gap between informal, sketch-based representations and formal, executable models. To this purpose, SimSketch can be used to draw strokes to externalize their learners' models of a phenomenon. Learners can then place "stickers" on their drawing, each representing a behavioral primitive, such as movements, reproduction, avoidance etc. The model that has been created by combining the learner's drawing and the behavioral annotations can be executed and simulated. SimSketch is targeted at learners in primary and secondary education and is suitable for numerous educational domains, since the behavioral primitives are highly generic and applicable to various phenomena, such as the movement of celestial bodies, predator-prey systems, swarming behavior, traffic systems and many more.

The three examples of the drawing-based approach to modeling presented here, illustrate both the potential and the research agenda in modeling research: the support for learners to create high quality drawings, creating challenging tasks within reach of the learner and providing smooth represent-run-revise cycles based on drawings. The presented tools are available from <http://modeldrawing.eu>.

References

1. Louca, L.T., Zacharia, Z.C.: Modeling-based learning in science education: cognitive, metacognitive, social, material and epistemological contributions. *Educational Review*, 1–22 (2011)
2. Brooks, M.: Drawing, Visualisation and Young Children's Exploration of "Big Ideas". *International Journal of Science Education* 31, 319–341 (2009)
3. Ainsworth, S., Prain, V., Tytler, R.: Drawing to Learn in Science. *Science* 333, 1096–1097 (2011)
4. Anjewierden, A., Gijlers, H., Kolloffel, B., Saab, N., de Hoog, R.: Examining the relation between domain-related communication and collaborative inquiry learning. *Computers & Education* 57, 1741–1748 (2011)
5. Gijlers, H., van Dijk, A.M., Weinberger, A.: How can Scripts and Awareness Tools Orchestrate Individual and Collaborative Drawing of Elementary Students for Learning Science? In: CSCL 2011. ISLS (2011)
6. Leenaars, F., van Joolingen, W.R., Gijlers, H., Bollen, L.: Drawing-Based Simulation for Primary School Science Education: An experimental study of the GearSketch learning environment. In: DIGITEL (2012)

An OWL Ontology for IEEE-LOM and OBAA Metadata

João Carlos Gluz¹ and Rosa M. Vicari²

¹Post-Graduation Program in Applied Computer Science (PIPICA) – UNISINOS – Brazil
jcgluz@unisinios.br

²Interdisciplinary Center for Educational Technologies (CINTED) – UFRGS – Brazil
rosa@inf.ufrgs.br

Abstract. This work introduces the OBAA metadata ontology, which is an OWL ontology created to represent all metadata from IEEE-LOM standard and OBAA metadata proposal. This ontology provides the basic vocabulary of linguistic terms that agents can use to query and manipulate metadata information. The work presents main structure of the ontology, and shows a concrete example of how to represent learning objects with this ontology.

Keywords: LOM Metadata Ontology, IEEE-LOM, OBAA, OWL.

1 The OBAA Metadata Ontology

A major problem existent in the current literature on ontologies applied to educational technologies is the lack of an established, and public OWL ontology, which specifies the properties of all the IEEE-LOM [1] Learning Objects (LO) metadata. The OBAA metadata ontology was defined in order to remedy this problem. The OBAA (learning Objects Assisted by Agents) metadata proposal [3] is an extension of the IEEE-LOM, including support for (a) adaptability and interoperability of LO on digital platforms such as Web, Digital TV (DTV) and mobile, (b) compatibility with international standards, (c) accessibility of LO by all citizens, and (d) independence, and flexibility of the technologies. Figure 1 shows the general structure of OBAA ontology.

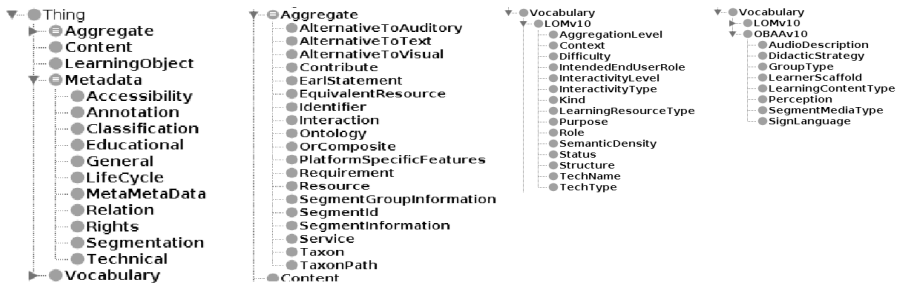


Fig. 1. The OBAA ontology

This ontology completely covers all IEEE-LOM metadata, and the new metadata defined in OBAA proposal. All IEEE-LOM or OBAA simple data types were mapped to XSD, or RDF data types. Values from these simple data type are associated to metadata instances via OWL data properties, named according to the format:

its <simple metadata element name> **Is**

where the <simple metadata element name> placeholder is retrieved from the original specification (IEEE-LOM or OBAA), with the first letter capitalized. Following the subject-predicate-object structure of RDF triples, it is possible to define triples like:

```

mdata1   itsTitleIs      "Test Object 1"@en.
mdata2   itsFormatIs    "text/html".
mdata3   itsHasVisualIs true.
    
```

Aggregate data elements are represented by individuals of *Aggregate* subclass. These elements are associated to metadata instances through OWL object properties, named according the format:

has <complex metadata element name>

Using this format, one can form RDF triples with the following structure:

```

mdata4 hasIdentifier id1.
id1   itsCatalogIs  "OBAA Test Objects".
id1   itsEntryIs    "obj1".
    
```

Every relationship *has*<complex metadata element name> from OBAA ontology, have an inverse relationship, *is*<complex metadata element name>*Of*.

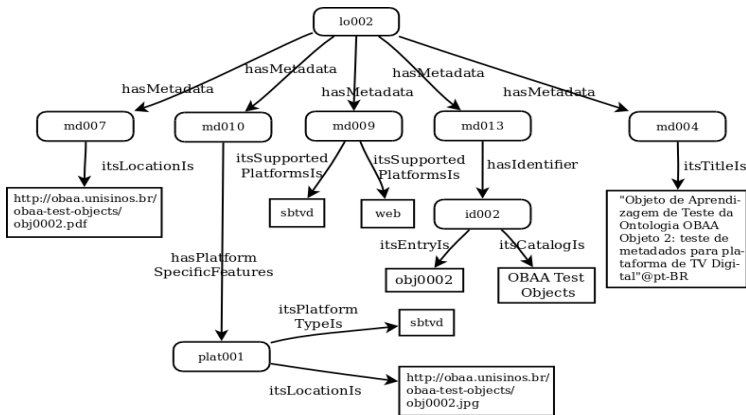


Fig. 2. A learning object example showing DTV/Web inter-operation metadata

The current version of the OBAA ontology is located in <<http://obaa.unisinos.br/obaa22.owl>>. This ontology meets the OWL DL profile, equivalent to a Description

Logic with structure *ALCIF[d]*. This site contains a test base ontology populated with several LO, located in <http://obaa.unisinos.br/obaa22-test-objects.owl>. This test ontology can be queried through a web interface available at <http://obaa.unisinos.br/MILOS-QU/>, with SPARQL/TERP [2].

References

1. IEEE-LTSC. Std 1484.12.1 - IEEE Learning Technology Standard Committee (LTSC) - Standard for Learning Object Metadata (LOM). IEEE (2002)
2. Sirin, E., Bulka, B., Smith, M.: Terp: Syntax for OWL-friendly SPARQL Queries. In: 7th OWL Experiences and Directions Workshop, San Francisco (2010)
3. Viccari, R., Gluz, J., Passerino, L., et al.: The OBAA Proposal for Learning Objects Supported by Agents. In: Procs. of MASEIE Workshop – AAMAS 2010, Toronto, Canada (2010)

Classifying Topics of Video Lecture Contents Using Speech Recognition Technology

Jun Park^{1,2} and Jihie Kim¹

¹ Information Sciences Institute/USC, Marina Del Ray, U.S.A.

² Electronics and Telecommunications Research Institute, Daejeon, Korea
{junpark, jihie}@isi.edu

Abstract. We explore a speech-based topic classification approach. We generate the transcript of input video lecture based on speech recognition technology and identify the topic by comparing its term-based vector with topic models. The preliminary experiment result shows that the speech-based topic classification works well, with its performance comparable to one that directly uses manual transcripts. The approach also shows robustness against speech recognition errors up to 40.6%.

Keywords: Topic classification, Topic modeling, Speech recognition, Tf-idf.

1 Topic Classification Based on Speech Recognition

The speech recognition based topic classification procedure is divided into two parts: topic model training and topic classification of input videos using the topic model. Topics are represented as a vector space model, in which tf-idf[1] weights become the vector elements. We first build the topic model (a vector of size N) for each topic by computing tf-idf (term frequency – inverse document frequency) weights using the document in the training corpus and selecting top N terms that have highest weights.

To classify the topic of the given input video lecture, we generate its transcription text by using speech recognition technology, and build a tf-idf vector from the transcription text. Then, the best matching topic is selected as the topic of the lecture based on the cosine similarity.

The key component of this approach is 'speech recognition'. Recently, Google made its speech recognition service available to the public through its applications including Chrome browser, Youtube, etc. Thus, we make use of the Google's speech recognition functionality.

2 Experiment and Discussion

We selected C++ programming learning as a task domain due to its popularity in many undergraduate engineering programs. As the training corpus, we chose the textbook of 'Thinking in C++'[2], the text of which is available on the web. We built 27 topic models, one for each chapter as described in Section 1.

As the test data, we select five C++ programming video lectures, provided by the Missouri University of Science and Technology [3]. This lecture series is available on

the Youtube site, where we obtained both the original transcripts and the speech recognition result by the 'Closed Caption(CC)' menu.

Table 1 shows the list of the test video lectures and the test results with the tf-idf vector size 1000. The last column shows the speech recognition performance in terms of the word error rate (WER): $WER(\%) = 100 * (1 - (C-I)/N)$, where N, C and I are the total, the correctly recognized and the inserted number of words, respectively.

Table 1. Test result for 5 lecture video contents

Test data				Rank (among 27 topics)		WER (%)
Lec.No.	Lecture title	Length (mm:ss)	#Words	for CC	for SR	
8.5	Function Overloading	02:13	383	2	5	28.7
8.7	Inline Functions	02:23	375	1	1	29.6
13.2	String & Character Manipul.	17:40	2661	1	1	39.7
15.7.0	Overloading Operators	11:45	1815	1	1	33.3
15.10	Template Classes	16:02	2256	1	1	40.6

Using the closed caption (CC; i.e. perfect recognition), the topic model ranks the correct topic as the first for four lectures. For the remaining one, the correct topic is ranked as the second. Although the topic model was generated from *written* textbook corpus, its classification of *speech* transcripts is reasonably accurate. This indicates that the written document topic models can be effectively used in modeling speech data. The classification of speech recognition (SR) output presents similar results as shown in the sixth column. Note that the SR's WER ranges from 28.7% to 40.6%. This indicates that the speech recognition technology can be a useful tool in processing educational contents, even with a considerable amount of recognition error.

3 Summary

To assess the feasibility of applying speech recognition technology to educational contents, we explored a new speech-based topic classification approach. Our preliminary results show that text generated from videos using speech recognition can be effectively used for classifying topics of video lectures. It is notable that the approach shows robustness against varying levels of speech recognition errors. We also observe that written text models work well for speech data.

Acknowledgement. This is supported by the National Science Foundation, REESEE program (award #1008747).

References

1. Maning, D.C., Raghavan, P., Schutze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
2. Eckel, B.: Thinking in C++: Introduction to Standard C++, vol. I, II. Prentice Hall (2000)
3. Missouri S&T Courses site:
<http://www.youtube.com/user/MissouriSandTCourses>

An Agent-Based Infrastructure for the Support of Learning Objects Life-Cycle

João Carlos Gluz¹, Rosa M. Vicari², and Liliana M. Passerino²

¹ Post-Graduation Program in Applied Computer Science (PIPICA) – UNISINOS – Brazil
jcgluz@unisinos.br

² Interdisciplinary Center for Educational Technologies (CINTED) – UFRGS – Brazil
rosa@inf.ufrgs.br, liliana@cinted.ufrgs.br

Abstract. This work presents the MILOS infrastructure. This infrastructure will implement the functionalities needed to create, manage, search, use and publish learning objects compatible with OBAA metadata proposal. MILOS project starts from several innovative assumptions, integrating agent and ontology technologies to support the adaptability, interoperability and accessibility requirements specified by OBAA. This work shows the assumptions of MILOS project, and the main elements of its architecture.

Keywords: Learning Objects, Pedagogical Agents, Learning Systems Architecture, Ontology Engineering, Agent Oriented Software Engineering, Multi-agent Systems.

1 The MILOS Infrastructure

The OBAA metadata standard proposal [2] was defined in an open and flexible way, being compatible with the current scenario of educational and multimedia standards. It is expected that this proposal enables the interoperability of Learning Objects (LO) in Web, Digital TV (DTV), and mobile platforms. The MILOS (Multiagent Infrastructure for Learning Object Support) infrastructure is based on a multi-agent architecture that implements the functionality needed to support all activities involved in the lifecycle of some LO, including activities like authoring, management, search, and educational use of the LO. The basic expectation about the services provided by MILOS is that users can only say what should be done with the OA, without going into details of how this should be done. To do so, the MILOS infrastructure project assumes an innovative epistemic premise, upheld by recent technological advances, which offers a way to design, and build the infrastructure:

(1) A learning object is essentially a Knowledge Object (KO) able to be distributed in educational technology systems.

The vision of a LO as a KO is consistent with the LO's goals in teaching contexts. The real problem is that the technology to support this vision is not yet available, especially when one takes into account the diversity of possible formats for educational contents. However, the usual division of LO in two levels of abstraction: 1)

educational contents level, and 2) metadata level, enables an important initial step on the path that leads to a treatment of LO as a KO:

(II) Without loss of generality or applicability, it is possible to consider LO metadata as symbolic structures that can be subject of the current techniques of knowledge representation, and manipulation.

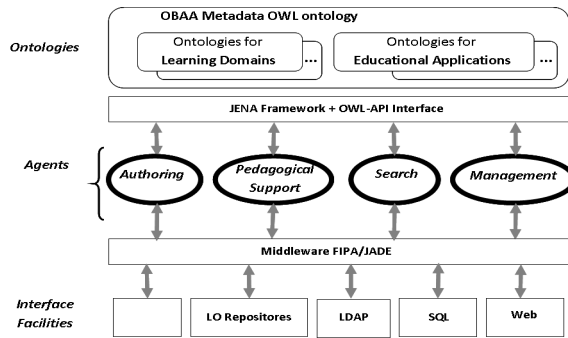


Fig. 1. Overview of MILOS infrastructure architecture

Figure 1 presents the general architecture of MILOS infrastructure. This architecture was divided in three main layers:

Ontology layer: responsible for the specification of knowledge that will be shared among infrastructure agents. The OBAA metadata ontology [1] provides the basis for the ontology layer. In addition to this ontology, it should be specified in this layer all learning domain, and educational application ontologies to be used by MILOS. All of these ontologies must be defined in OWL as derivations of OBAA metadata ontology, i.e., they must include OBAA metadata ontology, and regard its metadata definitions as the common terms in all MILOS applications. LO metadata define the terminology adopted by the main agents of MILOS. These metadata elements corresponds to the attributes, properties and relationships of the terminology, while the metadata values correspond to the terms of terminology.

Agents layer: responsible for implementing the requirements foreseen in OBAA proposal. MILOS agents incorporate knowledge that allow their users to perform activities over LO based on their professional knowledge, and skills, but without requiring technical knowledge about LO. These activities encompass the entire life cycle of a LO, distributed in four large multi-agent systems: (a) Search System: supports search, and retrieval of LO; Pedagogical Support System: supports the pedagogical use of LO; Authoring System: supports LO authoring activities; Management System: supports LO storing, managing, and publishing.

Interface facilities layer: responsible by the communication of MILOS agents with web servers, external web learnings environments, LO repositories, databases, directory services, and other legacy educational applications.

The MILOS project started in the second quarter of 2011, and is planned to run for three years. The development methodology follows a spiral cycle, thus the initial subsystems prototypes will be the base for posterior developments, until the full

functional support of OBAA requirements be achieved. The initial results are available at the MILOS portal located in <http://obaa.unisinos.br/>.

References

1. Gluz, J., Vicari, R.: Uma Ontologia OWL para Metadados IEEE-LOM, Dublin-Core e OBAA. Anais do XXII SBIE, Aracaju (2011)
2. Vicari, R., Gluz, J., Passerino, L.M., Santos, E., Primo, T., Rossi, L., Bordignon, A., Behar, P., Filho, R., Roesler, V.: The OBAA Proposal for Learning Objects Supported by Agents. In: Procs. MASEIE Workshop – AAMAS 2010, Toronto, Canada (2010)

Cluster Based Feedback Provision Strategies in Intelligent Tutoring Systems

Sebastian Gross¹, Xibin Zhu², Barbara Hammer², and Niels Pinkwart¹

¹ Clausthal University of Technology, Germany
{sebastian.gross,niels.pinkwart}@tu-clausthal.de
² Bielefeld University, Germany
{xzhu,bhammer}@techfak.uni-bielefeld.de

Abstract. In this paper, we propose the use of machine learning techniques operating on sets of student solutions in order to automatically infer structure on these spaces. Feedback opportunities can then be derived from the clustered data. A validation of the approach based on data from a programming course confirmed the feasibility of the approach.

Keywords: intelligent tutoring systems, ill-defined domains, machine learning.

1 Introduction

In many domains such as law, argumentation or art, most problems are ill-defined and have ambiguous solutions that can be argued for (and against!) but that are impossible to verify formally [4]. If ITSs cannot rely on explicit models, it may be possible to acquire information about the domain in terms of examples given by students or experts. Since these learnt models are widely data driven, machine learning techniques such as clustering constitute a key technology to infer meaningful information from given examples. The approach presented in this paper is based on clusters of student solutions where the solutions within each cluster might have a different quality but are structurally similar.

2 Clustered Solution Spaces: Feedback Strategies

In this section, we discuss two cases of how feedback based on clustered sets of student solutions can be given in the absence of formal domain models.

In the first case, we assume that grades for most of the student solutions in the data set are available (e.g., via assessments by human tutors). Every class of the solution space can then be represented by one student solution which has a high structural similarity to the other student solutions in the class (i.e., it is near the center of the class), and has a high grade (i.e., it is a good solution).

These representative solutions can then be used to give feedback to students who submit a new solution. A newly submitted (potentially erroneous) student

solution will then be analyzed (in terms of which class it belongs to) and compared structurally to the representative solution within this class. The result of this comparison can be fed back to students in various forms, including (i) a direct comparison showing the student's solution and the representative solution, highlighting differences between both, or (ii) the highlighting of potentially erroneous parts in the student's solution (i.e., the parts where it differs from the representative solution) without explicitly showing the representative solution.

In the second case, we assume that reliable scores for solutions are not available. As such, representative good solutions as previously defined cannot be computed. Here, one option is to use peer reviewing among the group of students. Another way of providing feedback is peer tutoring [3] in which a reviewing student is tutoring another student. The peer tutor can give hints about evident mistakes and can ask questions about potential mistakes. In this second case, the clustering can be helpful for selecting appropriate peer reviewers or tutors.

3 Validating and Discussing the Approach: a Case Study

To validate our approach and investigate whether our method for feedback provision is practically applicable, we conducted a case study. We used a data set from a Java programming class. For this data, scores assigned by human experts were available for every student solution. The solution clusters were computed using affinity propagation (AP) [2]. Similarities between solutions were computed based on Plaggie [1], a plagiarism detection algorithm that calculates a simple structural comparison of two programs. This way, we represented the space of all solution structures by means of a small number of prototypical correct solutions (case 1 from above). In general, this test confirmed our expectations: the resulting clusters were relatively clear, and overall the ways of feedback provision as suggested above made sense for this data set.

However, there were also some limitations. For very poor student solutions that lack any structure, the methods still have drawbacks – these were added to one of the clusters, but the ways of feedback provision did not make much sense. As long as the structural similarity between solutions is high, our methods for feedback provision make sense – less so if elements within a cluster are dissimilar.

References

- [1] Ahtiainen, A., Surakka, S., Rahikainen, M.: Plaggie: Gnu-licensed source code plagiarism detection engine for java exercises. In: Proceedings of the 6th Baltic Sea Conference on Computing Education Research: Koli Calling 2006, Baltic Sea 2006, pp. 141–142. ACM, New York (2006)
- [2] Frey, B.J., Dueck, D.: Clustering by passing messages between data points. *Science* 315, 972–976 (2007)
- [3] Goodlad, S., Hirst, B.: Peer tutoring: a guide to learning by teaching. Kogan Page (1989)
- [4] Lynch, C., Ashley, K.D., Pinkwart, N., Aleven, V.: Concepts, structures, and goals: Redefining ill-definedness. *International Journal of Artificial Intelligence in Education* 19(3), 253–266 (2010)

A Web Comic Strip Creator for Educational Comics with Assessable Learning Objectives

Fotis Lazarinis and Elaine Pearson

School of Computing, Teesside University, UK
f.lazarinis@scm.tees.ac.uk,
e.pearson@tees.ac.uk

Abstract. In this paper, we present an application which enables educators to edit comic strips and associate educational goals to each comic strip. Post-comic activities are used to measure the success of the objectives. IEEE LOM metadata are associated to each comic which can be packaged using SCORM.

Keywords: Educational comics, Assessment, Edutainment, Hypermedia.

1 Introduction

Comics can improve the motivation of poor readers to try harder in order to understand the story [1] and have been used to teach both science and art topics. Children exposed to science comics were able to give scientific explanations [2]. Anatomy comic strips were designed to help students learn the complexities of anatomy in a straightforward and humorous way [3]. The presented application enables educators to edit comic strips and associate specific learning objectives. Through post-comic activities, such as tests, the success of the objectives is measured.

2 System Description

The authoring environment supports the creation of sequences of images through the selection of scenes, human and animal characters, text captions and other objects (e.g. arrows, stars, boxes, trees) (see figure 1). Each comic is associated with one or more topics and one or more learning objectives. Topics are hierarchies of concepts, coded using Topic Maps (www.topicmaps.org). The educational goals can be high level, e.g. “understand concept X” or specific objectives “to achieve specific score in the post-activity”. Educators can set their own educational goals by adapting one of the existing predefined rules. In general there are three categories of customizable goals supported by the system: *(i)* related to knowledge (e.g. understand concept X), *(ii)* related to skills (e.g. be able to perform or to adjust) and *(iii)* related to attitudes (e.g. justify or defend a specific case). Test creators associate one or more categories of goals to each comic and for each goal one or more specific objectives should be declared. For example, the general goal could be “Understand concept Mathematics/Addition” and the specific objectives are “Achieve score at least 90% in the post-comic test” and “Answer correctly all questions of difficulty level 1”.

Educators are able to form a post-comic activity (see Figure 2) and connect it to the objectives. They are also able to provide keywords and short textual descriptions for each comic which are encoded in IEEE LOM (ltsc.ieee.org/wg12/). These metadata facilitate the reusability of comic strips. The data can be exported as a SCORM (www.adlnet.gov/capabilities/scorm) learning object to be reused in other learning platforms. The resultant sequences of PNG images are embedded in HTML files and displayed through a regular Web browser with the aid of JavaScript and are thus available to every environment running Web pages.



Fig. 1. EduComicStrip Comic Editor

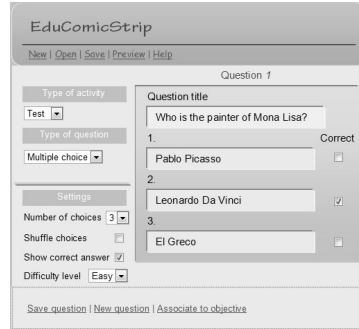


Fig. 2. Post-comic activity

3 Conclusions

The prototype of the system was evaluated with the aid of 5 educators and 5 students. The educators were primary school teachers and the students were 10 year-old children who attend the 4th-grade class of the primary school. Overall, the initial evaluation showed that the system is easy to use and useful for educational purposes and could be employed in a number of learning actions to enhance the understanding of the students and improve their engagement to the activity. However, more assessment experiments are needed before it could be used in real world learning activities.

References

- [1] Hutchinson, K.: An experiment in the use of comics as instructional material. *Journal of Educational Sociology* 23(4), 236–245 (1949)
- [2] Weitkamp, E., Burnet, F.: The Chemedian brings laughter to the chemistry classroom. *Int. J. Sci. Educ.* 29, 1911–1929 (2007)
- [3] Park, J.S., KiM, D.H., Chung, M.S.: Anatomy comic strips. *Anatomical Sciences Education* 4(5), 275–279 (2011), doi:10.1002/ase.224

A Layered Architecture for Online Lab-works: Experimentation in the Computer Science Education

Bouabid Mohamed El Amine, Philippe Vidal, and Julien Broisin

118 Route de Narbonne, 31062 Toulouse, France, +33 561 557 402
{bouabid,vidal,broisin}@irit.fr

Abstract. Practical competencies are key components of any computing education curriculum. Today, several computer experiment tools exist, however, these tools are originally intended to experts, and do not integrate very well into the existing online learning environments, in particular, they lack efficient support for teamwork, tutoring and instructional design. In this paper we introduce a model-driven engineering approach to transparently integrate remote computer experiments into distant learning curriculums. The originality of this framework stands on two key components: a middleware layer that acts as glue between existing Learning Management Systems and remote laboratories and a set of standard unifying and extensible models representing the whole system including its lab components, the versatile experiments and the actors' actions.

Keywords: Technology Enhanced Learning, Remote Lab-Works, Computing Experiments, Distributed Architecture, Model-Driven Approach.

1 Introduction

In this paper, we present a model driven engineering approach, independent from any specific tool, allowing teachers to design and transparently deploy computer experiments on a remote lab. Learners are then able to interact with the remote experiment and to benefit from pedagogical services such as teamwork and tutoring.

Our approach consists in reusing the Web-Based Enterprise Management (WBEM) initiative [3] in order to unify management of computing experiments which stands on both a Common Information Model (CIM) to represent managed elements, and a support architecture to facilitate control and administration of the managed resources.

CIM exploits object concepts to model and manage systems, networks and applications [3], which can be handled by acting on the matching CIM objects' attributes and methods. In [1] we presented details of the models describing both experiments' components and users' activities on these experiments.

2 A Layered Architecture to Actively Manage Remote Experiments

The global framework supporting our approach is a three-tier architecture: The Upper layer is composed of a web-based LMS (Moodle) integrating specific GUIs. In [2] we presented the details of these GUIs dedicated to operate remote lab experiments

The Integration layer acts as a bridge between the upper and lower layers by exposing some services to the learning layer handling users' requests. They are then translated to actions performed on the remote experiments, and return the matching results. At this level, there is a CIM repository, a kind of object database which store the matching instances representing the states of the remote experiments but also the explicit records of the activities performed by human actors. The testbed layer we built using an existing testing tool (MLN) is integrated to the overall system through dedicated WBEM adapters called providers [3].

In order to test our framework, a usability testing has been conducted with two teams of learners, Testers were asked to perform the lab-work and to answer a usability survey. Although this early test has only affected few learners, the results allowed us to have a positive evaluation of our approach but some drawbacks emerged regarding some missed capabilities and bugs as well as some desired features.

3 Conclusion

In this paper, we presented an approach to facilitate the design and management of online computer lab-works based on (1) a set of CIM models to represent the multiple entities implied into a remote lab-work activity, (2) a layered architecture, and (3) a GUI dedicated to computing experiments. Teamwork and tutoring tasks are supported by specific GUI providing each actor with the status of the experiments he is working on and the awareness of all operations performed by other users on them. An early usability testing confirmed the relevance of our approach.

References

1. Bouabid, A., Vidal, P., Broisin, J.: Integrating Learning Management Systems and Practical Learning Activities: the case of Computer and Network Experiments. In: The 9th IEEE International Conference on Advanced Learning Technologies, Riga, Latvia, July 14-18, pp. 398–402. IEEE Computer Society (2009)
2. Bouabid, M.A., Vidal, P., Broisin, J.: A Web Application Dedicated to Online Practical Activities: the Case of System and Network Experiments. In: The 11th IEEE International Conference on Advanced Learning Technologies, Athens, Georgia, USA, July 06-08. IEEE Computer Society (2011)
3. Distributed Management Task Force, CIM Tutorial (2003), <http://www.wbemsolutions.com/tutorials/DMTF/dmtftutorial.pdf>

A Serious Game for Teaching Conflict Resolution to Children

Joana Campos, Henrique Campos, Carlos Martinho, and Ana Paiva

INESC-ID and Instituto Superior Técnico – Technical University of Lisbon,
Av. Prof. Cavaco Silva, Taguspark 2744-016, Porto Salvo, Portugal
{joana.campos,henrique.t.campos,carlos.martinho}@ist.utl.pt,
ana.paiva@inesc-id.pt

Abstract. To learn how to manage conflict situations is essential for a healthier society. In this paper, we present a serious game scenario that aims at reinforcing this pro-social behaviour in children by using emotional agents as NPCs.

Keywords: Conflict, Virtual agents, Serious Game.

1 Introduction

Conflict is pervasive in our society and is often related to critical aspects of our human nature. However, despite conflicts being associated with negative feelings and destructive behaviours, conflict situations may lead to positive outcomes, as they offer opportunities for growth and improvement [4]. Yet conflict, when not managed adequately, can have some negative and dramatic consequences in our society. For that reason, being able to cope with conflict situations and handle different kind of conflict scenarios is something that one should learn how to master and this pro-social behaviour should be fostered since early stages in life.

Educational interventions in schools have taken different forms (eg. peer mediation programs or drama workshops) and have proven to have a positive impact on students behaviour. However, these classroom settings are static and promote in-class learning and most of the times are not adapted to one individuals specific needs. To address this issue and to go beyond impersonal learning, games have been object of research as a tool to immerse people in a powerful environment that allows users to learn new skills, knowledge and attitudes [3].

In this paper, we propose an educational game - *My Dream Theatre* that intends to prepare children to manage conflict more effectively and independently. *My Dream Theatre* prototype is integrated within the SIREN¹ project, which aims to develop an adaptive serious game for teaching conflict resolution to children.

¹ <http://sirenproject.eu/>

2 Emergent Conflicts out of Emotional Agents

Social conflict is a dyadic process which encapsulates perception, emotions, behaviours of both parties and consequences as a result of such interaction [1]. In this research, we aim at addressing this phenomena by creating groups of agents, that engage in natural situations of conflict, in the environment of an educative game for children.

My Dream Theatre is an educational game that aims at teaching 9 to 11-year-old children, some conflict resolution skills. The game setting is a school theatre club directed by the user. The user has to select an adequate cast for each performance. Each cast member (non-player character - NPC) has a set of characteristics, such as preferences for roles, a level of proficiency, interests and personality. As the user assigns roles to each one of the characters, conflict situations may emerge due to the agents' conflicting goals and their choice of actions to handle the situation. When conflict arises, the user intervention is required to balance the agent's proficiency and cooperativeness (which may have to be mediated by the user) and assure a good performance in the end.

Conflict in real life is highly dependent on emotional responses reflected on one's actions. Hence, we integrated a model of emotions in this prototype – FAtiMAs agent model [2], which steams from OCC cognitive theory of emotions and is the base of the agents decision making process. We consider that such emotional processes in the agents' minds is essential to capture the essence of the real conflicts found in the description of real world situations. The escalation process will therefore be an effect of each character appraisal of the situation and it evolves as a result of the interplay between the agents.

This initial prototype intends to address some elements of the deep structure of conflict, as for example, one's emotions and others' points of view, which are important variables to understand how one should cope in a certain situation. By having an environment that suits the users needs and experiences, we expect to promote transferable knowledge to real life situations.

Acknowledgements. The research leading to these results has received funding from European Community's FP7 ICT under grant agreement n° 258453, FCT (INESC-ID multiannual funding) through the PIDDAC Program funds and a PhD scholarship (SFRH/BD/75342/2010) granted by FCT.

References

1. Coombs, C.H., Avrunin, G.S.: *The Structure of Conflict*. Psychology Press (1988)
2. Dias, J., Paiva, A.: *Feeling and Reasoning: A Computational Model for Emotional Characters*. In: Bento, C., Cardoso, A., Dias, G. (eds.) *EPIA 2005*. LNCS (LNAI), vol. 3808, pp. 127–140. Springer, Heidelberg (2005)
3. Lieberman, D.A.: *What can we learn from playing interactive games?* In: *Playing Video Games: Motives, Responses, and Consequences*, pp. 379–397 (2006)
4. Tessier, C., Müller, H.J., Fiorino, H., Chaudron, L.: *Agent's Conflicts: New Issues*, pp. 2–20. Springer (2001)

Towards Social Mobile Blended Learning

Amr Abozeid¹, Mohammed Abdel Razek^{1,2}, and Claude Frasson³

¹ Azhar University, Faculty of Science,

Math. & Computer Science Depart. Naser City, Cairo, Egypt

² Deanship of Distance Learning, King Abdulaziz University, Kingdom of Saudi Arabia

³ Computer Science Department, University of Montréal, CP 6128 succ. Montréal, QC, Canada

amrapozaid@gmail.com, maabdulrazek1@kau.edu.sa,

frasson@iro.umontreal.ca

Abstract. Mobile technologies and Web 2.0 have led to explosions of communication, with a resulting increased need for people to process and utilize that new communication. This paper presents a new approach to create an integrated approach for learners by bring traditional education, mobile learning, along with social network into one adaptive blended learning environment. This approach introduces an adaptation mechanism to adapt learning objects to meet learner characteristics and their mobile capabilities.

Keywords: Social Network, Web 2.0, Mobile Learning, Adaptive Learning.

1 Introduction

Nowadays, the growth of online education encourages teachers in physical classrooms to use Internet-based content and resources. Meanwhile, mobile learning continues to challenge the boundaries imposed by traditional classroom learning to improve education and exploit technology in furthering that aim. The need of (just in time - just enough) learning and the diversity of learners' characteristics as well as mobile technologies requires adaption for different cases [4].

This paper presents a framework for system architecture to combines the best elements of online, social network and face-to-face learning via mobile. It tends not only to create social blended learning environment, but also to adapt course's learning object. Our adaptive learning platform aims to build learning environment and provide learning objects based on learner's individualized information usage behavior, habits, preferences and etc. The adaptation individualize the content based on learner's Level of knowledge (beginner, immediate, professional), learning styles (active – reflective, sensing – intuitive, visual – verbal, sequential – global), location (on campus, off campus), time (allowed time).

In this paper, we display the architecture of Social Mobile Interactive Blended Learning System (SMIBLS). SMIBLS uses mobile and Bluetooth technologies to increase interactions and communications between instructors and students during classroom on campus, while, it uses Web 2.0 to enrich the communication between instructors and students during classroom off campus. MIBLS contains learning

activities that leads to enhance learning process during classroom. As shown in Figure 1, SMIBLS consists of two basic user interfaces: one devoted to the instructor and the other for the student.

These interfaces can be accessed from Mobile and PC desktop as well. In order to provide adaptation, there are two modules: Learner Adaptation Module (LAM) , and Instructor Adaptation Module (IAM). These modules use an adaptation engine to adapt the educational activity according to the context. LAM is the process of automatically adjusting learning contents based on learner’s needs. The adaptation measures the needs based on the contexts which are considered by learners context (e.g. preferences, knowledge level and style), the educational context (e.g. requirements, pedagogical theory, achievements and results), the infrastructures context (e.g. networks, devices) and the environments context (e.g. neighbors, weather and noise level) [7]. Meanwhile, LAM allows user to collaborate via Web 2.0 tools like Face book and Twitter to establish alive communication between instructor in the classroom and his students outside the classroom.

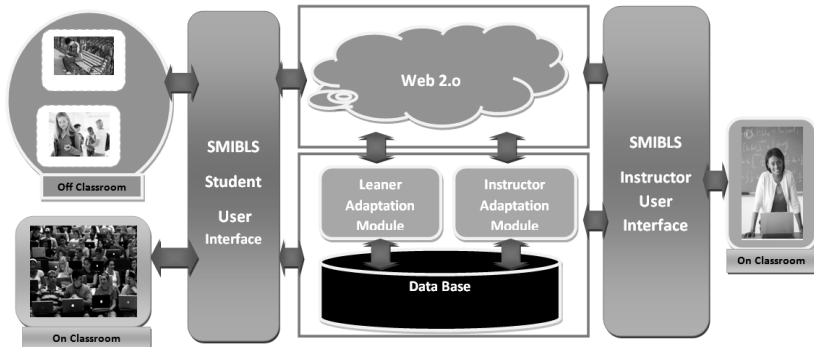


Fig. 1. SMIBLS Architecture

IAM allows instructor to formulate a quiz or a question and send it to students in the classroom or to the student outside the classroom. IAM lets multimedia learning objects with different format such as swf, mp3 and mp4. Each LO is a file contains the learning content that cover part from the chapter objectives. The LO should be small and meaningful in order to enable students browsing it in their free time or bus traveling. From the education perspective, we suggested that LO should contain definitions, theories, remarks, and important parts covered in chapter or lecture notes.

References

1. Xie, P., et al.: Research on the method of recomposing learning objects and tools in adaptive learning platform. In: Digital Techniques and Systems Entertainment for Education, pp. 326–336 (2010)

2. Abozeid, A., Razeq, M.A., El-Sofany, H.F., Ghaleb, F.F.M.: Mobile Interactive Blended Learning System. *IEEE Multidisciplinary Engineering Education Magazine* 5 (2010)
3. Guzmán, E., Conejo, R., Pérez-De-La-Cruz, J.-L.: Adaptive testing for hierarchical student models. *User Modeling and User-Adapted Interaction*, 119–157 (2007)
4. Drira, R., et al.: What can we adapt in a Mobile Learning Systems? In: *The Proceedings of the Second International Conference on Interactive Mobile and Computer Aided Learning (IMCL 2007) in Collaboration with IEEE, Amman, Jordan, April 18-20*, pp. 19–21 (2006)

Learning Looping: From Natural Language to Worked Examples

Leigh Ann Sudol-DeLyser, Mark Stehlik, and Sharon Carver

Carnegie Mellon University

1 Introduction

One important introductory concept in many CS courses is repetition (looping), the automated repeating of individual commands. In this work, we present results from a study of college undergraduates' naive conceptions of repetition, their difficulties with learning to construct valid repetition statements, and their abilities to apply what they have learned to new problem solving situations. Although computer programming is a new topic when high school or college students encounter it for the first time, students can draw upon their previous life experiences when solving problems. Those conceptions that align with CS topics [2,3] have been shown to be influenced by students' prior experiences. Alignment through analogies can be helpful [1] although where the scientific concept differs, common knowledge can hinder learning [4].

For many students, the topic of looping is their first encounter with non-linearity in their programs. Until this point, each line of code is executed once, and then control moves to the next line of code. Such linearity makes reasoning about the programs straightforward. With the addition of looping, in the code you will need to evaluate a termination condition and then either repeat prior lines of code, or move to the next statement after the loop. While returning to a previous command or location is not unusual in everyday life and natural language, it is an important change in the way that novices see their code.

2 Methods

Fifty three (53) participants with no CS experience were recruited for this study from Carnegie Mellon University, The University of Pittsburgh, and Allegheny College. Participants took a paper and pencil pretest asking them to write directions to help a robot solve three problems involving repetition. Upon completing the pretest, participants completed the tutoring sequence for one looping construct (for or while) and then an assessment asking them to code two loops similar to the ones in the tutor. After that assessment, participants completed the tutoring sequence for the second looping construct and a second assessment.¹ Students then completed a self-efficacy survey and a transfer test.

¹ Students were randomly assigned to see tutoring for the construction of “for” loops or “while” loops first, and in a within-subjects design saw the other looping construct.

The online tutor was constructed to take advantage of parallel worked examples and problems in order to demonstrate the three components of a loop: initialization, termination, and update. Each concept was described separately and students viewed a worked example and then completed two similar examples, constructing code only for the component that was highlighted on the page. After completing the page for each component, students were then asked to combine the components, writing the entire loop for each example on separate tutoring screens. The tutor did not use a compiler, instead employing regular expressions for matching the text based responses.

3 Results

Results of the pretest indicate that participants, even when directed to focus on the three components of a loop, are likely to either omit or implicitly provide directions when writing repetition directions in natural language. Participants were most likely to make implicit references in the termination or update conditions, often asking the robot to “repeat” without quantifying how many times, or exactly which commands were to be repeated.

In the tutoring sequence, participants answered correctly 80% of the time and had the most difficulty with the initialization and update statements. Participants were mostly accurate on the assessments, scoring an average of 9.1 points out of 10 on both assessments. Participants had the most difficulty with correctly ordering the loop statements and writing an accurate update statement.

Results of the transfer test indicated that participants still had a tendency to use implicit language when describing code in natural language. When asked to solve a problem containing nested loops, the termination and update conditions again caused students the most trouble, and many students did not construct the secondary loop correctly. Recommendations are to offer more practice assembling complete looping structures and additional problems for students who struggle.²

References

1. Mayer, R.: The psychology of how novices learn computer programming. *ACM Comput. Surv.* 13(1), 121–141 (1981)
2. Simon, B., Bouvier, D., Chen, T., Lewandowski, G., McCartney, R., Sanders, K.: Common sense computing(episode 4): Debugging. *Computer Science Education* 18, 117–133 (2008)
3. Sudol, L.A., Stehlik, M., Carver, S.: Mental models of data. In: *Proceedings of the 9th Koli Calling International Conference on Computing Education Research* (2009)
4. Vosniadou, S.: Mental models in conceptual development. *Model Based Reasoning: Science, Technology, Values* 1 (2002)

² This work was supported through the Program for Interdisciplinary Education Research (PIER) at CMU, funded through Grant R305B040063 to CMU, from the Institute for Education Sciences. The opinions expressed are those of the authors and do not represent the views of the Institute or the US Department of Education.

A Basic Model of Metacognition: A Repository to Trigger Reflection

Alejandro Peña Ayala^{1,2,3}, Rafael Dominguez de Leon², and Riichiro Mizoguchi³

¹WOLNM, ²ESIME-Z-National Polytechnic Institute, ³ISIR-Osaka University,
¹31 Julio 1859 # 1099-B, Leyes Reforma, DF, 09310, Mexico
apenaa@ipn.mx, rdominguez55@gmail.com
<http://www.wolnm.org/apa>

³ Institute of Scientific and Industrial Research (ISIR), Osaka University

Abstract. In this work, we model a couple of basic metacognitive skills: knowledge and regulation. The aim is depicting underlying concepts of knowledge and regulation domains. We promote reflection on learners once they access their respective model.

Keywords: Metacognition, awareness, knowledge, regulation, reflection.

1 Introduction

Metacognition means: "...the active monitoring and consequent regulation and orchestration of cognitive transactions in relation to the cognitive objects or data on which they bear, usually in service of some concrete goal ..." [4].

Concerning the metacognitive knowledge (MK), Gama asserts: "It consists primarily of knowledge or beliefs about what factors or variables act and interact in what ways to affect the course and outcome of cognitive enterprises" [6]. As regards with metacognitive regulation (MR), it refers to processes that coordinate cognition [3].

With the aim to progressively model learners' metacognitive skills, we tailor the first version of our metacognitive model with two key domains: MK and MR. Likewise, we design and develop a trial to elicit responses of a group of college students about their beliefs, habits, and likings at learning. Those answers are raw information about some items of MK and MR. As a result, we find out some interesting highlights to be presented in this work.

2 Experiment

We chose the Metacognitive Awareness Inventory (MAI) designed by Schraw and Dennison [18]. MAI elicits information about learner's beliefs, habits, and preferences. It holds a questionnaire of 52 questions (Q) to be answered as true or false. Based on the learner's responses, MAI estimates her level of metacognitive awareness. But, they are split into two supersets of 17 and 35 questions to measure the level of MK and metacognitive regulation MR.

These sets are respectively computed with $QMK = 17$ and $QMR = 35$. Likewise, both supersets are respectively organized into three and five sets. So MK owns declarative, procedural and conditional knowledge sets of concepts and MR contains five basic sets of concepts: planning, information management strategies, comprehension monitoring, debugging strategies, evaluation.

The MAI questionnaire was applied to a sample of college students, who pursue a bachelor degree in Information Technologies in Mexico. Volunteers are studying the fifth semester of a program of eight semesters. The size of the sample (n) was 25.

Our metacognitive model contains three concepts about the level of metacognitive awareness, knowledge, and regulation. It also holds the prior stated eight concepts.

3 Conclusions

The statistics of true responses for the QMK questions of MK show: Based on n , the mean is 16.6 and the median is 17. The range is 15 from 9 to 24 that respectively correspond to the 36% and 96% of n . It reveals: The likings are known by at least a third of the sample (e.g., #16: "I know what the teacher expects me to learn") and there are some habits well know by nearly all the members of the sample (e.g., #46: "I learn more when I am interested in the topic").

The results of the MR domain show: The sample scored 538 truth answers, the 61% of a maximum of 875 (i.e., $n * QMR$). Based on QMR, the mean is 21.5 positive answers per subject and the median is 21. The range was 20 from a 10 to 30 that respectively correspond to 29% and 86% of QMR. Thus, 57% is the difference between subjects with the least and the highest metacognitive regulation!

As a future work we plan: to add other components to the metacognitive model, such as: monitoring, reflection, and control. We are going to refine the questionnaire and make new trials. We will also author content to enhance students' metacognition.

Acknowledgments. First author gives testimony of the strength given by his Father, Brother Jesus and Helper, as part of the research projects of World Outreach Light to the Nations Ministries (WOLNM). This work is supported by: CONACYT 118862, CONACYT-SNI-36453, CONACYT 118962-162727, SeAca/COTEPABE/144/11, SIP-20120266, IPN-SIP-EDI: SIP/DI/DOPI/EDI-0505/11, IPN-COFAA-SIBE.

Reference

1. Flavell, J.H.: Metacognitive aspects of problem solving. In: Resnick, L.B. (ed.) *The Nature of Intelligence*, pp. 231–236. Erlbaum, Hillsdale (1976)
2. Gama, C.A.: *Integrating Metacognition Instruction in Interactive Learning Environments*. PhD Thesis, University of Sussex. Sussex, UK (2004)
3. Fernandez-Duque, D., Baird, J.A., Posner, M.I.: Executive attention and metacognitive regulation. *Consciousness and Cognition* 10, 288–307 (2000)
4. Schraw, G., Dennison, R.S.: Assessing metacognitive awareness. *Contemporary Educational Psychology* 19, 460–475 (1994)

Analyzing Affective Constructs: Emotions ‘n Attitudes

Ivon Arroyo¹, David Shanabrook¹, Winslow Burleson², and Beverly Park Woolf²

¹Department of Computer Science, University of Massachusetts Amherst
{ivon, dhshanab, bev}@cs.umass.edu

²School of Computing, Informatics and Engineering, Arizona State University
winslow.burleson@asu.edu

Abstract. We analyze the relationship between a variety of affective constructs that have been researched, as it is not clear what is the breadth of affective variables to model -- which constructs are equivalent, related, or unrelated.

1 Introduction

Much research has attempted to model and recognize student affect in Interactive Learning Environments, and started exploring mechanisms to repair or cope with negative emotions. However, there are many different theories and constructs for student affect, so many that it is hard to compare these constructs and approaches. This article presents the results of a correlation research study to establish the relationship between three sets of affective constructs, in the hope to eliminate redundant items and make richer constructs, when they are not equivalent. We compare items from the **control-value theory of emotions** [1], which describes several emotions related to achievement in learning situations (in Figure 1, they correspond to variables ending in `_P`, e.g. an item of the `PRIDE_P` construct is “I am proud of my contributions to math class”). The **emotion constructs** by Arroyo et al. [2] were engendered from hundreds of students in real classrooms to classify their frustration, interest, etc. These can be considered *emotions* when asked inside of the tutoring system (e.g. “how frustrated do you feel?”) and *affective predispositions* when asked in a pre/post survey (“how frustrated do you get when solving math problems?”). These are variables ending in `_A`, e.g. an item of the `PRIDE_A` construct is “Do you feel proud when solving math problems?”). Last, the **attitude constructs** by Eccles [3] try to understand students’ concept of themselves as capable to carry out the task, These are variables end in `_E`, e.g. an item of the self-confidence `SC_E` construct is “How good would you be at learning something new in math?”). We present the result of a subset of these correlations, corresponding to the set of *control-oriented* affective constructs, composed of `PRIDE_P`, `PRIDE_A`, `ANGER_P`, `ANXIETY_P`, `ANX_A`, `SHAME_P`, `SHAME_A`, `HOPL_P`, `FRUS_A`, `HOPL_A`, `SC_E`, `CON_A` in the next section.

2 Results

Two hundred and forty middle and high school students (N=240) took a survey before using a mathematics tutoring system. We establish that if the correlation between two

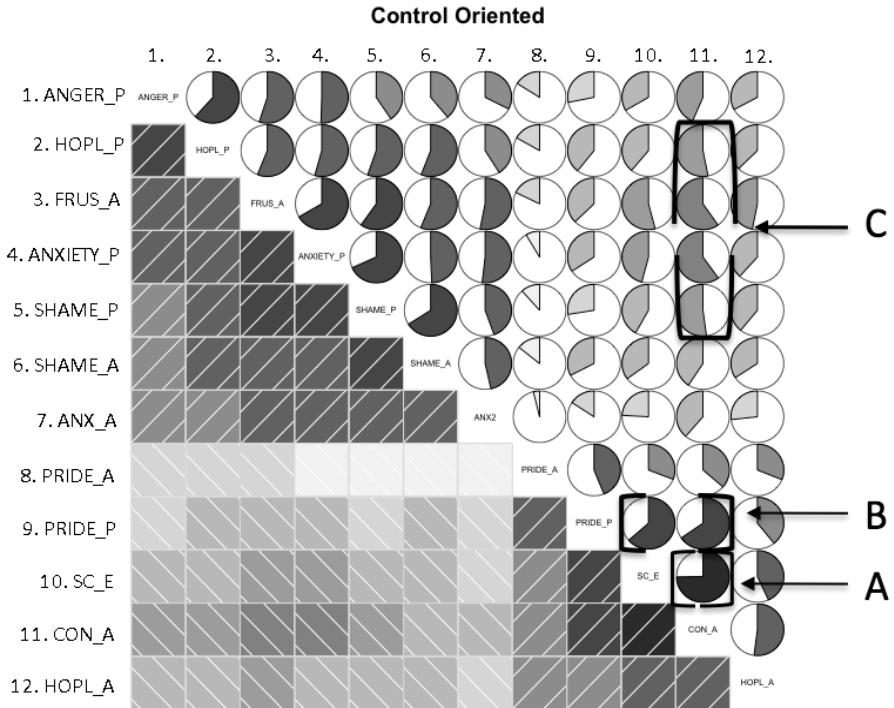


Fig. 1. R values for correlations between affective constructs. A → Eccle's SC_E (Math Self-Concept) and Arroyo's CON_A ("How confident do you feel...?") are *equivalent*, redundant; B → PRIDE_P is *highly related* to both SC_E and CON_A, they should be combined; C → CON_A is *highly related* (negatively) to ANXIETY_P, and also FRUS_A, HOPL_P, HOPL_A, SHAME_P. This means that talking with students about their "confidence" we are talking about a complex combination of emotional experiences related to hope, anxiety, frustration, shame and pride.

items is $R \geq 0.75$, then the constructs are basically *equivalent*, and one of them can be omitted in any further assessment. We consider two constructs to be *highly related* when $R \geq 0.5$ and $R < 0.75$, and will be combined as they refer to the same construct. We consider two constructs to be *moderately related* when $R \geq 0.25$ and $R < 0.5$. The last possibility is that the constructs are *unrelated* ($R < 0.25$), just different.

References

1. Pekrun, R., Goetz, T., Frenzel, A.C.: Academic Emotions Questionnaire—Mathematics (AEQ-M): User's manual. University of Munich, Department of Psychology (2005)
2. Arroyo, I., Cooper, D.G., Burseson, W., Woolf, B.P., Muldner, K., Christopherson, R.: Emotion Sensors Go To School. In: Proceedings of AIED 2009, pp. 17–24. IOS Press (2009)
3. Eccles, J., Wigfield, A., Harold, R.D., Blumenfeld, P.: Age and gender differences in children's self and task perceptions during elementary school. *Child Develop.* 64, 830–847 (1993)

Interactive Virtual Representations, Fractions, and Formative Feedback

Maria Mendiburo, Brian Sulcer, Gautam Biswas, and Ted Hasselbring

Vanderbilt University, Nashville TN 37235, USA

{maria.mendiburo,brian.sulcer,gautam.biswas,t.hasselbring}@vanderbilt.edu

Abstract. In this study, we explore the potential benefits of formative feedback when students are constructing virtual representations of fractions. We find greater effects for feedback about the accuracy of students' responses to the questions they answer using the representations than for feedback about the accuracy of the representations themselves. The results suggest further study of the timing when students receive feedback about their representations and the ways to adapt feedback for different learners.

Keywords: fractions, virtual representations, feedback.

1 Introduction

Virtual manipulatives offer many potential benefits, including the ability to associate active experience with manipulatives to symbolic notation through feedback [1]. In an extensive review of the literature on feedback, Shute [2] determined that formative feedback comes in a variety of types and can be administered at various times during the learning process, but research shows the most effective forms of feedback are nonevaluative, supportive, timely, and specific. In this study, we explore the benefits of delivering immediate feedback when students construct virtual representations of fractions and then use the virtual representations to compare symbolically represented fractions.

The system we designed for students to create virtual representations of fractions delivers two different kinds of feedback to students: 1) feedback about the accuracy and correctness of the virtual representations students create with suggestions of how to correct mistakes, and 2) feedback about the correctness of students' responses to questions that ask students to compare symbolically represented fractions. We assigned 37 students drawn from three intact, sixth-grade mathematics classes at a charter middle school in Middle Tennessee to four treatment conditions in which they received different combinations of these two types of feedback and used both quantitative and qualitative methods of analysis to examine student outcomes.

2 Results and Discussion

The students in all four treatment groups made statistically significant gains from pre-test to post-test, which indicates the computer system and the

instructional activities we designed for the experiment help students learn how to compare the relative size of fractions. However, we found no statistically significant treatment effect for these gains. The lack of a treatment effect in the gains between pre-test and post-test contrast the results from the two practice activities. The results from the practice activities showed a statistically significant treatment effect for the response feedback factor on response correctness in both practice activities and a trend in the scores for the response feedback factor on model correctness in one practice activity. The fact that the effect of the response feedback appeared fairly strong during both activities but disappeared on the post-test introduces the possibility that students need to receive feedback from the computer system for a longer period of time before they will be able to achieve at similar levels without feedback. In addition, it introduces the possibility that the feedback needs to be removed in a scaffolded rather than an “all or nothing” format in order to maintain a more consistent level of student performance. The trend for the response feedback factor on model correctness suggests that students may be more motivated to utilize feedback about model correctness if they already know their answer to a question is wrong, which is a hypothesis we intend to test in future research.

Visual analysis of the detail reports rendered about each student by the computer system and subsequent post-hoc analysis suggest the feedback students receive from the system will be more effective if we follow Shute’s framework and adapt the feedback to the cognitive and non-cognitive characteristics of the learner as well as to different types of knowledge and skills. While we can’t draw gender-based conclusions from these results given that the female and male students at the school where we conducted this study are taught in separate classes, it appears that the female students who participated in this study may benefit from feedback designed to encourage less dependence on the manipulatives, and the male students may benefit from scaffolding that allows students to skip using the manipulatives only until the student incorrectly answers several questions in a row. We also plan to test these hypotheses in future research.

Acknowledgements. The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A100110 to Vanderbilt University. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

References

1. Clements, D., McMillan, S.: Rethinking ‘concrete’ manipulatives. *Teaching Children Mathematics* 2(5), 270–279 (1996)
2. Shute, V.: Focus on formative feedback. *Review of Educational Research* 78(1), 153–189 (2008)

An Intelligent System to Support Accurate Transcription of University Lectures

Miltiades Papadopoulos and Elaine Pearson

Accessibility Research Centre, Teesside University, United Kingdom
{M.Papadopoulos, E.Pearson}@tees.ac.uk

Abstract. The performance of current Automatic Speech Recognition systems in the lecture environment is still below the level required for accurate transcription of lectures. This paper reports on the results of a study to assess the potential of the Semantic and Syntactic Transcription Analysing Tool in the production of meaningful post-lecture material with minimal investment in time and effort by academic staff.

Keywords. Accessibility, Automatic Speech Recognition (ASR).

1 Introduction

The lecture environment isolates students with hearing disabilities, while learners studying in a foreign language and those whose note taking skills are limited find lectures hard to follow, understand and recall [5]. Automatic Speech Recognition (ASR) technology can be employed to make lectures more flexible through the use of text transcripts. However, the performance of current systems in the lecture situation is still below the required levels. This paper reports on a set of experiments to test the validity of a mechanism that aims to minimise the evaluation process of imperfect transcripts and is a step forward in the production of meaningful support materials for students in a timely manner.

2 A Pragmatic Approach to Accurate Transcription

Research into the readability and usability of speech transcription has determined that an accuracy of at least 90% is required [4]. Extensive training is necessary in order for systems to achieve comparable accuracy [1], however the effort and workload required for the editing process to make ASR-generated transcripts meaningful to students is still unacceptably high [2]. Accepting that current systems are not suitable for use in the lecture theatre, we considered a different approach by bringing together research from the Natural Language Processing and Human Computer Interaction domains. The resultant mechanism, the Semantic and Syntactic Transcription Analysing Tool (SSTAT) [3] analyses transcripts, detects incorrect sentences and reports on the nature of the errors in a user-friendly interface. It also supports a targeted re-training process to improve overall efficiency for subsequent transcriptions.

3 Experiments

A study was devised to evaluate the potential of SSTAT in the lecture situation. The study was divided into three phases; reduction in editing time, improvement in accuracy rates over time and the level of acceptability for students of edited transcripts. The first experiment measured the reduction in editing time required to produce accurate transcripts using SSTAT compared to manual editing. The results revealed a decrease of 42.2% in the editing time of transcripts required to reach a transcription accuracy of approximately 88%. This demonstrates a significant potential as an automated method for supporting the transcripts' editing process. The second experiment utilised the targeted re-training feature to determine the improvement in accuracy rates of the transcripts over a lecture series on the same topic and by the same lecturer. This experiment used five lectures from five different modules and the improvement was measured on each of four passes through the system's cycle. The results revealed a mean overall increase of 5% between the first and fifth ASR-generated transcript. The third experiment examined students' perceived level of usability of transcripts to verify the tools potential in producing meaningful materials. Twenty-six students were involved and a repeated measures ANOVA was conducted to measure the variance in perceived usability across transcripts' three different accuracy levels. The results revealed that an accuracy of at least 87.5% was considered sufficient by students for the production of usable post-lecture materials. Current speech recognition systems are able to deliver in realistic conditions accuracy rates between 75-85% [1; 2]. This can be increased using SSTAT to acceptable levels.

SSTAT constitutes a unique approach to producing transcripts that reach an acceptable quality threshold for use by students with a significantly reduced investment in time and effort compared to manual transcription.

References

1. Kheir, R., Way, T.: Inclusion of Deaf Students in Computer Science Classes Using Real-Time Speech Transcription. In: 12th Annual Conference on Innovation & Technology in Computer Science Education, pp. 261–265. ACM Press, New York (2007)
2. Papadopoulos, M., Pearson, E.: An Analysing Tool to Facilitate the Evaluation Process of Automatic Lecture Transcriptions. In: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2009, pp. 2189–2198. AACE, Chesapeake (2009)
3. Papadopoulos, M., Pearson, E.: A System to Support Accurate Transcription of Information Systems Lectures for Disabled Students. In: 22nd Australasian Conference on Information Systems, Sydney, Australia (2011)
4. Stuckless, R.: Recognition Means More Than Just Getting the Words Right: Beyond Accuracy to Readability. *Speech Technology*, 30–35 (October/November)
5. Wald, M., Bain, K.: Universal Access to Communication and Learning: The Role of Automatic Speech Recognition. *J. Universal Access in the Information Society* 6(4), 435–447 (2008)

Multi-context Recommendation in Technology Enhanced Learning

Majda Maâtallah and Hassina Seridi-Bouchelaghem

LABGED Laboratory, University of Badji Mokhtar Annaba, Po-Box 12, 23000, Algeria
{maatallah, seridi}@labged.net

Abstract. Recommender Systems (RSs) have been applied recently in Technology Enhanced Learning (TEL) to let recommending relevant learning resources to teachers or learners. In this paper, we propose a novel recommendation technique that combines a fuzzy collaborative filtering algorithm with content based one to make better recommendation, using learners' preferences and importance of knowledge to recommend items with different context corresponding to their different interests and tastes. Empirical evaluations show that the proposed technique is feasible and effective.

Keywords: Technology-Enhanced Learning, Recommender Systems, Collaborative Filtering, Content Based Filtering, Learner Profile.

1 Multi-Context Recommendation Process

Recently, Recommender Systems (RSs) are applied in the e-learning field, particularly in Technology Enhanced Learning (TEL)[1][2][3], in order to personalize learning content and connect suitable learners with each other according to their individual needs, preferences, and learning goals. Learners' needs and preferences change over time, where they want learning from resources with different context. This creates the need of Adaptive RSs able to generate recommendations with different tastes depending on the learner's preferences. To this aim, we propose a new hybrid technique that combines CF and CBF to generate multi-context recommendations to lifelong learners that fit their different tastes and interests.

To enhance the accuracy of TEL recommendations, we are conducted toward hybridization between CF and CBF, with adding knowledge importance of the learner. First, we propose to construct clusters of users automatically from the evaluation matrix. Then, we propose an adjusted fuzzy neighborhood algorithm to select just fuzzy nearest neighbors belonging to fuzzy nearest clusters using the difference between membership degrees as similarity measure between learners. Then, we make CF-based prediction of the learner preferences by combining linearly prediction results of user-based and item-based algorithms. Secondly, we give scores to topics to promote courses according to the topics' frequency and evaluations made by the learner. Then, we make predictions based on taxonomic content according to similarities with nearest courses and their evaluations. To address limitations of CF and CB predictions, we have proposed to blend them linearly.

Finally, we generate a Top-K recommendation process adapted to TEL field by introducing the importance of knowledge, proposed in [3], in the calculation of courses' ranks. Then, we select Top-K items from each Top-N list, generated in single clusters, according to membership degrees of the learner to these clusters.

2 Experiment and Results

In our experiments, we have used the Moodle¹ platform integrating in it books from BX-Book-Rating database², knowledge level of the learner and our technique.

First, we evaluated the prediction performance using the novel adapted MAE [3] using variant sizes of clusters. From Fig.1, we observe that the MAE has an inverse relationship with cluster sizes, and we can notice that the new MAE, in almost all cases, is smaller than usual MAE, due to the weighting of learner's knowledge. Then, we evaluated the Top-K recommendation performance using the F1 metric. From Fig.2, we observe that the F1 metric increase with number of recommended courses. It can be seen also that the recommendation performance of the system is good.

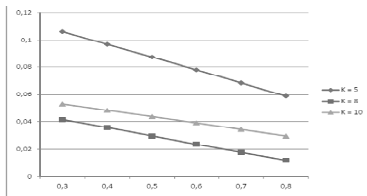


Fig. 1. MAE performance, less value means better performance

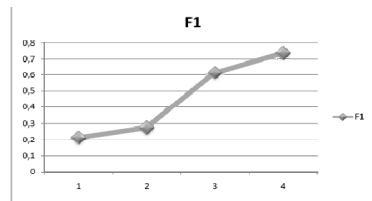


Fig. 2. F1metric evolution

Experimental results show that the proposed approach can improve the recommendation accuracy. In the future work, we will elaborate this technique to generate multi-context recommendations taking in the account implicit feedback and temporal effects.

References

1. Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H., Koper, R.: Recommender Systems in Technology Enhanced Learning. In: The1st RSs Handbook. Springer, Berlin (2010)
2. Garcia, E., Romero, C., Ventura, S., Castro, C.D.: An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. User Modeling and User-Adapted Interaction 19(1-2) (2009)
3. Bobadilla, J., Serradilla, F., Hernando, A.: Collaborative filtering adapted to recommender systems of e-learning. Journal of KBS, Knowledge-Based Systems 22(4) (2009)

¹ www.moodle.org

² www.informatik.uni-freiburg.de/~chiegler/BX/

Author Index

- Abdel Razek, Mohammed 707
Abozeid, Amr 707
Adamson, David 346, 531, 551
Ahmadi Olounabadi, Atefeh 603
Aleven, Vincent 174, 673
Álvarez, Ainhoa 685
Arealillo-Herráez, Miguel 630
Ari, Fatih 681
Arnau, David 630
Arroyo, Ivon 46, 714
Arruarte, Ana 655
Arslan-Ari, Ismahan 681
Ashwin, Emma 262
Ayesh, Aladdin 630
Azevedo, Roger 40, 59, 212, 651
- Baker, Ryan S.J.D. 434, 444, 636
Barbalios, Nikos 649
Barnes, Tiffany 304, 597, 612, 615
Baschera, Gian-Marco 389
Beaulieu, Gabriel 201
Beck, Joseph E. 90, 268
Becker, Lee 368
Beek, Wouter 292
Beheshti, Behzad 454
Belmontez, Ricardo 636
Benton, Laura 262
Beuth, Jack 551
Beynon, Meurig 664
Beynon, Will 664
Bharathy, Gnana K. 642
Biswas, Gautam 505, 716
Bittencourt, Ig Ibert 298
Blanchard, Emmanuel G. 280
Bollen, Lars 689
Bouabid, Mohamed El Amine 703
Bouchet, François 40, 212
Boyer, Kristy Elizabeth 52
Bratko, Ivan 286
Brawner, Keith W. 72, 582
Bredeweg, Bert 292
Broisin, Julien 703
Brosnan, Mark 262
Bull, Susan 411, 685
- Burleson, Winslow 46, 243, 714
Burlison, Jonathan 212
Busetto, Alberto Giovanni 389
- Cade, Whitney 256, 557
Cai, Zhiqiang 626
Calvo, Iñaki 655
Calvo, Rafael A. 78, 358
Campos, Henrique 705
Campos, Joana 705
Carlson, Ryan 11, 563
Carver, Sharon 710
Cassell, Justine 11
Castro, Maynor Jimenez 636
Cerri, Stefano A. 606
Chalfoun, Pierre 1, 84
Chaouachi, Maher 65
Chauncey Strain, Amber 59, 618
Chebil, Raoudha 606
Ching, Dixie 669
Chiru, Costin-Gabriel 330
Chou, Chia-Ru 657
Churcher, Neville 422
Cohen, William W. 185, 493
Conati, Cristina 112
Conde, Ángel 655
Conejo Muñoz, Ricardo 310
Corbett, Albert T. 444
Córdova-Sánchez, Mariheida 195
Costa, Evandro 640
Cui, Liang 683
- Dagami, Michelle Marie C. 634
Dascalu, Mihai 352
Delozanne, Elisabeth 123
Derbali, Lotfi 129
Desmarais, Michel C. 454
Després, Christophe 517
Dessus, Philippe 352
D'Mello, Sidney 59, 256, 541, 557, 576,
618, 638
Dolan, Robert P. 660
Dominguez de Leon, Rafael 712
Dragon, Toby 340

- Duan, Ying 162
 Duffy, Melissa 212
 Dyke, Gregory 531, 551

 Eagle, Michael John 304, 615
 Eksin, Ceyhun 642
 Elias, Endhe 298
 Elorriaga, Jon A. 655

 Faghihi, Usef 233
 Fernández-Castro, Isabel 685
 Ferreira, Rafael 298
 Feyzi-Behnagh, Reza 212, 651
 Finkelstein, Samantha 11
 Flores, Raymond 681
 Floryan, Mark 340
 Fontaine, Samantha 636
 Forbus, Kenneth D. 620
 Forsyth, Carol 626
 Fournier-Viger, Philippe 233
 Frasson, Claude 1, 65, 84, 129, 707

 Galvez Cordero, Jaime 310
 Gauthier, Robert 671
 Genato, Ryan 636
 George, Sébastien 632
 Germany, Mae-Lynn 626
 Gijlers, Hannie 689
 Gluz, João Carlos 691, 696
 Goel, Gagan 428
 Goldberg, Benjamin S. 72
 Goldman, Susan 274
 Gong, Yue 268
 Gonzalez, Avelino 582
 González-Calero, José Antonio 630
 Gowda, Sujith M. 434, 444
 Graesser, Arthur 162, 256, 541, 576, 626
 Grafsgaard, Joseph F. 52
 Grawemeyer, Beate 262
 Gross, Markus 389
 Gross, Melissa 618
 Gross, Sebastian 699
 Groznik, Vida 286
 Guia, Thea Faye G. 634
 Guid, Matej 286
 Guin, Nathalie 622
 Guzman De Los Riscos, Eduardo 310

 Hammer, Barbara 699
 Harley, Jason M. 40, 212

 Hasselbring, Ted 716
 Hastings, Peter 274
 Hayashi, Yugo 22
 Hays, Patrick 256
 Heffernan, Neil T. 268, 399, 405
 Hirashima, Tsukasa 620
 Hoffmann, Kristin F. 464
 Holden, Heather K. 624
 Holland, Jay 422, 499
 Hong, Yuan-Jin 511
 Horiguchi, Tomoya 620
 Hossain, Gahangir 212
 Howard, Scott 666
 Howley, Iris 531, 551
 Hsiao, Tzu-Chien 679
 Huang, Wei-Jie 657
 Hughes, Simon 274
 Hussain, Md. Sazzad 78
 Huynh-Kim-Bang, Benjamin 135

 Ikeda, Mitsuru 683
 Inan, Fethi A. 681
 Ioannidou, Irene 649
 Isotani, Seiji 298
 Ito, Makoto 662

 Jacoboni, Pierre 517
 Jaques, Patrícia 298
 Jarušek, Petr 379
 Jin, Wei 304
 Johnson, Hilary 262
 Johnson, Matthew W. 304, 597
 Jraidi, Imène 1

 Kanzaki, Nana 645
 Karlovčec, Mario 195
 Käser, Tanja 389
 Kashiwara, Akihiro 662
 Kay, Judy 482
 Keiser, Victoria 101, 563
 Keshtkar, Fazel 162
 Kim, Jihie 570, 694
 Kinnebrew, John S. 505
 Koedinger, Kenneth R. 101, 185, 222,
 493, 563, 588, 591
 Kohn, Juliane 389
 Kucian, Karin 389
 Kumar, Amruth N. 524

- Labat, Jean-Marc 135, 168
 Lajoie, Susanne 511
 Lallé, Sébastien 428, 622
 Landis, Ronald S. 212
 Larrañaga, Mikel 655
 Lawless, Kimberly 274
 Lazarinis, Fotis 701
 Le, Nguyen-Thinh 320
 Lebeau, Jean-François 201
 Lee, Chia-Ying 657
 Lee, Po-Ming 679
 Lee, Seung Y. 476
 Leenaars, Frank 689
 Lehman, Blair 256, 541, 557, 576
 Lehmann, Lorrie 304, 612
 Lejouad Chaari, Wided 606
 Lekira, Aina 517
 Lester, James C. 52, 141, 464, 470, 476
 Li, Nan 185, 493
 Lin, Bin 422
 Lintean, Mihai 675
 Liu, Chao-Lin 657
 Liu, Ming 358
 Lomas, Derek 588, 669
 Long, Yanjin 673
 Luengo, Vanda 428, 622

 Maass, Jaclyn 677
 Maâtallah, Majda 720
 Macam, Francis Jan P. 634
 Magliano, Joseph 274
 Maris, Marinus 687
 Marne, Bertrand 135
 Marsella, Stacy C. 151
 Martín, Maite 685
 Martínez Maldonado, Roberto 482
 Martinho, Carlos 705
 Mathews, Moffat 422, 499
 Matsuda, Noboru 101, 563
 Matsuda, Noriyuki 683
 Mayers, André 201, 233
 Mayfield, Elijah 551
 McCalla, Gordon 653
 McCuaig, Judi 671
 McLaren, Bruce M. 647
 Mendiburo, Maria 716
 Mephu-Nguifo, Engelbert 233
 Millis, Keith 626
 Mills, Caitlin 541, 638
 Miquilino, Dalgoberito 298

 Mitrovic, Antonija 422, 499, 579, 603,
 634
 Miwa, Kazuhisa 645
 Mizoguchi, Riichiro 712
 Monkaresi, Hamed 78
 Morgan, Brent 162
 Mott, Bradford W. 141, 470, 476
 Možina, Martin 286
 Muir, Mary 112
 Muller, Ryan 588
 Muratet, Mathieu 123, 168
 Murray, Tom 666
 Myneni, Lakshman S. 250

 Naceur, Rhouma 454
 Nakaike, Ryuichi 645
 Narayanan, N. Hari 250
 Nash, Padraig 162
 Niebuhr, Sabine 647
 Nietfeld, John L. 464
 Nixon, Tristan 669
 Nkambou, Roger 233
 Nye, Benjamin D. 642

 Ocumpaugh, Jaclyn 444
 Ogan, Amy 11, 636
 Okimoto, Tomoko 597
 Olney, Andrew M. 256, 677

 Pacampara, Nicole 212
 Paiva, Ana 705
 Palmer, Martha 368
 Papadopoulos, Miltiades 718
 Paquette, Luc 201
 Paraskeuopoulos, Stefanos 649
 Pardos, Zachary A. 195, 405, 434
 Park, Jun 694
 Passerino, Liliana M. 696
 Patel, Kishan 588
 Pavlik Jr., Philip I. 677
 Pearson, Elaine 701, 718
 Peckham, Terry 585
 Pelánek, Radek 379
 Peña Ayala, Alejandro 712
 Penstein Rosé, Carolyn 346, 531, 551,
 563
 Person, Natalie 256, 557
 Pinkwart, Niels 320, 699
 Poitras, Eric 511
 Powers, Sonya 660

- Py, Dominique 517
 Pynadath, David V. 151
- Rahman, A.K.M. Mahbubur 212
 Rai, Dovan 90
 Raizada, Rohan 101
 Rau, Martina A. 174
 Reina, David 685
 Reye, Jim 600, 628
 Ritter, Steve 669
 Rivers, Kelly 591
 Rodrigo, Ma. Mercedes T. 634, 636
 Rohrbach, Stacie 174
 Rowe, Jonathan 470
 Rummel, Nikol 174, 222
 Rus, Vasile 675, 677
- Sabourin, Jennifer 141, 470
 Sadikov, Aleksander 286
 Sandberg, Jacobijn 687
 Santos, Anderson 640
 Sárközy, Gábor N. 405
 Scheuer, Oliver 647
 Schubert, Michael 632
 Schwendimann, Beat 482
 Segedy, James R. 505
 Seridi-Bouchelaghem, Hassina 720
 Serna, Audrey 632
 Seta, Kazuhisa 683
 Shanabrook, David Hilton 46, 714
 Shareghi Najar, Amir 579
 Sharipova, Mayya 609, 653
 Shipe, Stefanie 666
 Shores, Lucy R. 141, 464
 Silva, Alan 298
 Silva, Emanuele 640
 Silva, Marlos 640
 Silva, Priscylla 640
 Silverman, Barry G. 642
 Soriano, Jose Carlo A. 636
 Sottolare, Robert 582
 Stamper, John 304, 588, 669
 Stehlik, Mark 710
 Stylianides, Gabriel 101
 Sudol-DeLyser, Leigh Ann 710
 Sugay, Jessica O. 634
 Sulcer, Brian 716
- Tanveer, M. Iftekhhar 212
 Taub, Michelle 212
 Tenório, Thyago 298
 Terai, Hitoshi 645
 Thomas, Pradeepa 168
 Torguet, Patrice 123
 Trausan-Matu, Stefan 330, 352
 Trevors, Gregory 212
 Trivedi, Shubhendu 405
 Tsui, Wei-Hsuan 679
 Tzeng, Yu-Lin 657
 Tzionas, Panagiotis 649
- Ulinski, Amy C. 594
 Urretavizcaya, Maite 685
- van Joolingen, Wouter R. 689
 van Vuuren, Sarel 368
 Veenhof, Gerard 687
 Viallet, Fabienne 123
 Vicari, Rosa M. 691, 696
 Vidal, Philippe 703
 von Aster, Michael 389
- Walker, Erin 11, 222, 243, 636
 Walker, Sean 222
 Wang, Ning 151
 Wang, Yutao 399
 Ward, Wayne 368
 Weinberger, Armin 647
 Weragama, Dinesha 600, 628
 Wiederrecht, Melissa A. 594
 Williams, Claire 256
 Wilson, Dale-Marie 612
 Wing, Leah 666
 Wisdom, John 135
 Woolf, Beverly Park 46, 340, 666, 714
- Xu, Xiaoxi 666
- Yacef, Kalina 482
 Yarzebinski, Evelyn 101
 Yeasin, Mohamed 212
 Yessad, Amel 168
 Yoo, Jaebong 570
- Zaier, Amani 681
 Zhang, Li 33
 Zhu, Xibin 699