Some applications of RCA to Software Engineering

November 2, 2012

Relational Concept Analysis

Useful features

- Multi-classification
  - extraction of groups of similar objects inside predetermined categories
  - extraction and organization of abstract concepts in the categories
- Analysis
  - extraction of implication rules based on relations
  - measures on object groups
  - extraction of patterns in data
- Visualization
  - of the multi-classification
  - of interconnections inside the multi-classification

Relational Concept Analysis

Kinds of applications in software engineering

- Reengineering of existing software, by building new software artefacts
- Classification of software artefacts
- Learning from model transformation examples, and inferring transformation rules

Reengineering of existing software, by building new software artefacts

- UML class diagram refactoring
  - *M. Dao, M. Huchard, M. Rouane Hacene, C. Roume, P. Valtchev: Improving Generalization Level in UML Models Iterative Cross Generalization in Practice. ICCS 2004: 346-360*
  - *G. Arévalo, J.-R. Falleri, M. Huchard, C. Nebut: Building Abstractions in Class Models: Formal Concept Analysis in a Model-Driven Approach. MoDELS 2006: 513-527*
- UML Use case diagram refactoring
  - *X. Dolques, M. Huchard, C. Nebut, P. Reitz. Fixing generalization defects in UML use case diagrams. Fundamenta Informaticae 2012.*

Reengineering of existing software, by building new software artefacts

- Blob design defect correction
  - *N. Moha, M. Rouane Hacene, P. Valtchev, Y.-G. Guéhéneuc: Refactorings of Design Defects Using Relational Concept Analysis. ICFCA 2008: 289-304*
- Extracting architectures in object-oriented software
  - *A.-E. El Hamdouni, A. Seriai, M. Huchard Component-based Architecture Recovery from Object-Oriented Systems via Relational Concept Analysis. CLA 2010: 259-270*

Classification of software artefacts

- Classification of web services
  - *Z. Azmeh, M. Driss, F. Hamoui, M. Huchard, N. Moha, C. Tibermacine, Selection of Composable Web Services Driven by User Requirements. Application and Experience Track of ICWS 2011*

Software analysis

- Analysis of the evolution of class diagrams
  - *A. Osman-Guédi, A. Miralles, B. Amar, M. Huchard, T. Libourel and C. Nebut. How Relational Concept Analysis can help to Observe the Evolution of Business. Concept Lattices and Applications (CLA 2012), October 11-14, 2012.*

Learning from model transformation examples, and inferring transformation rules

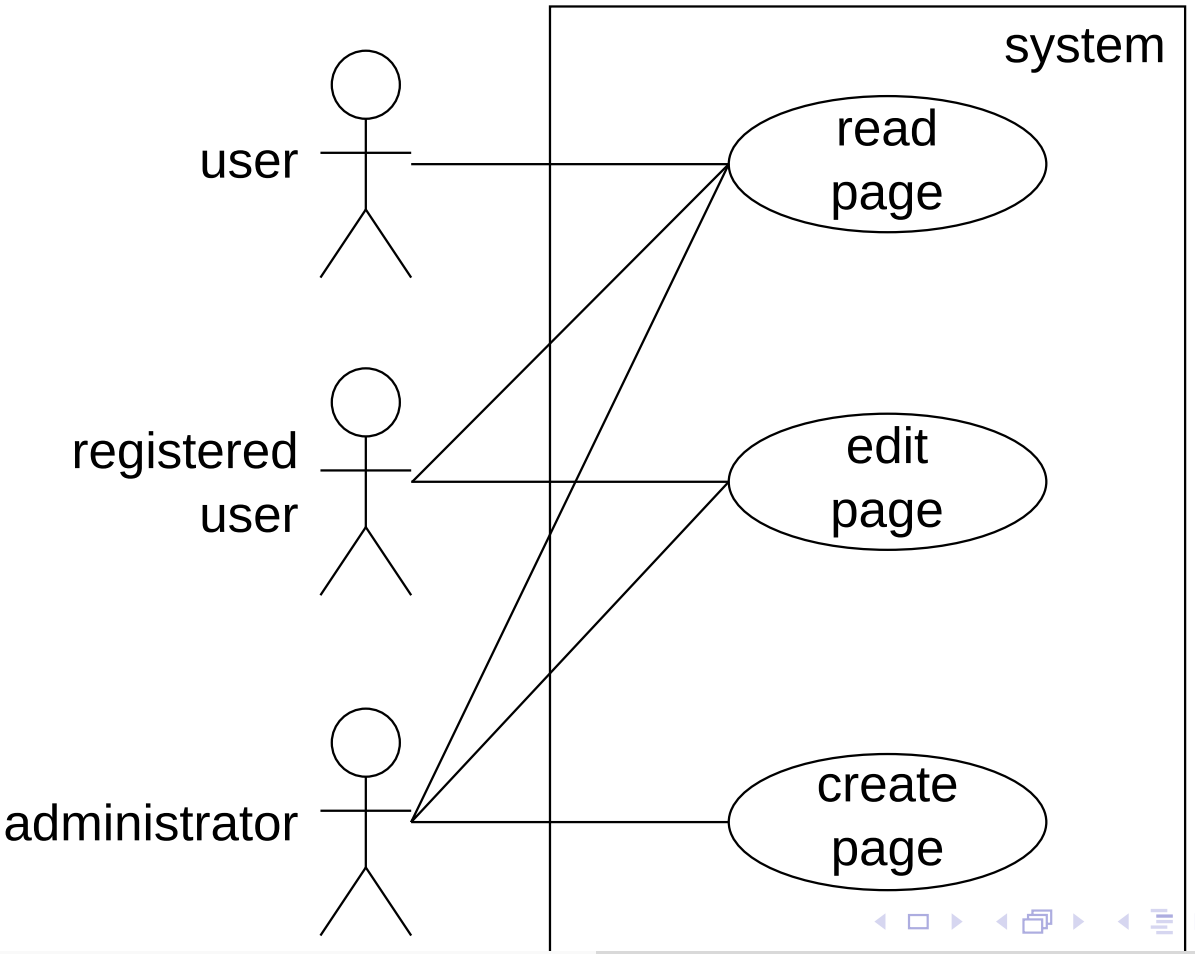
- Learning model Transformation patterns in MDE
  - *H. Saada, X. Dolques, M. Huchard, C. Nebut, H. A. Sahraoui: Generation of Operational Transformation Rules from Examples of Model Transformations. MoDELS 2012: 546-561*

Relational Concept Analysis

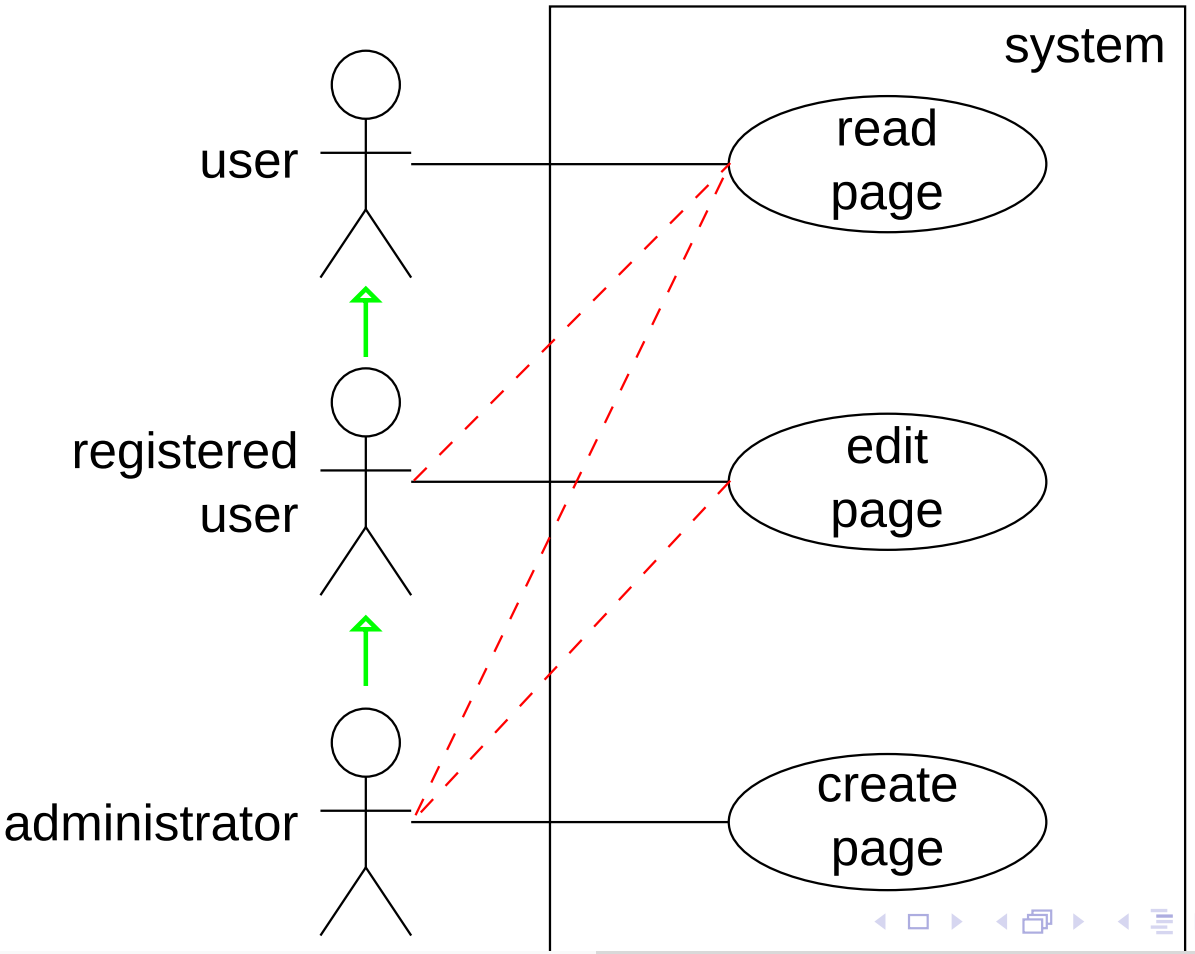
Issues for an application to a domain

- Choose relevant part of data
- Encode it in Relational Context Family: with scaling procedures, adaptations
- Run the RCA algorithm
- Filtering the obtained Concept Lattice Family
- Interpret and measure the results

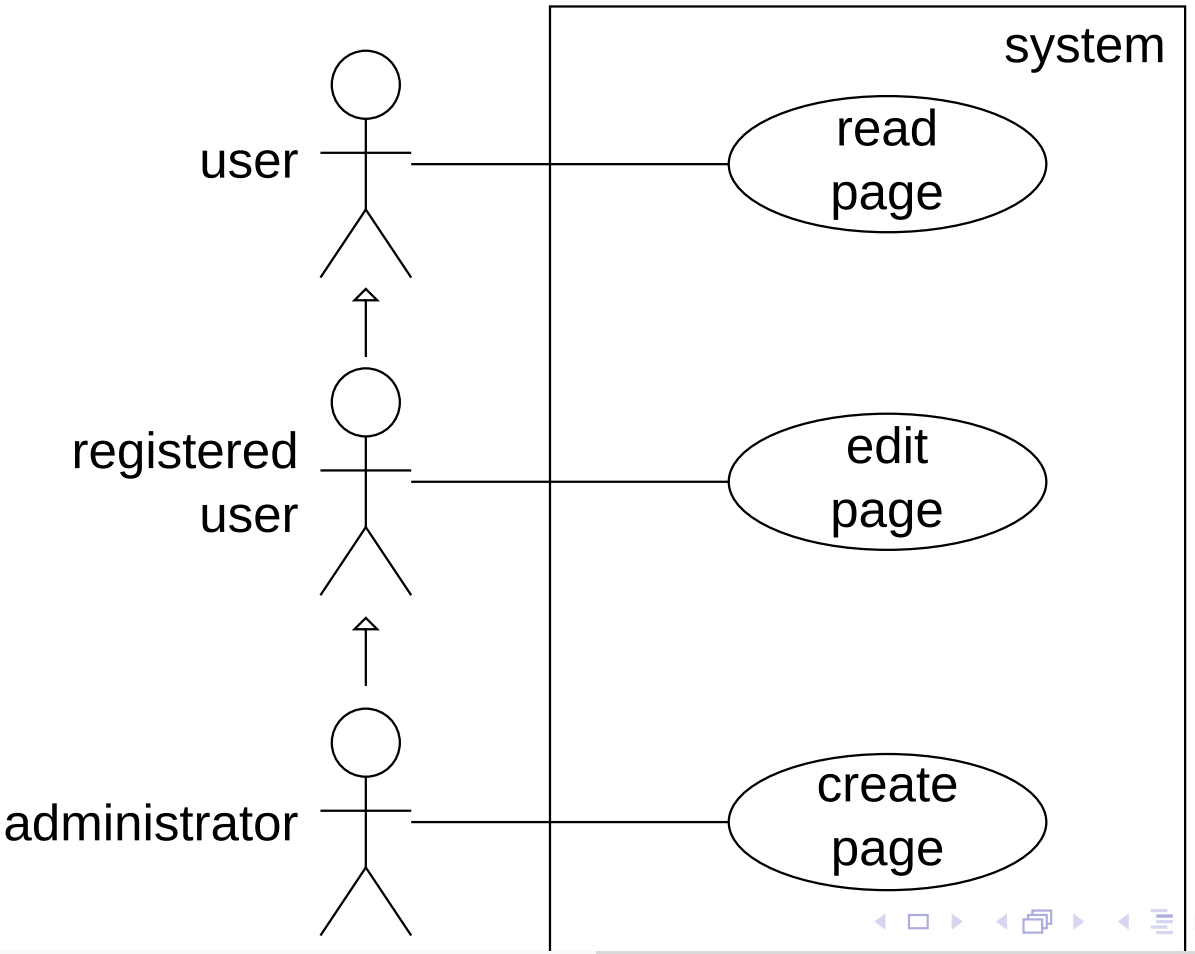
Reengineering UML use case diagrams for fixing generalization defects



Reengineering UML use case diagrams for fixing generalization defects

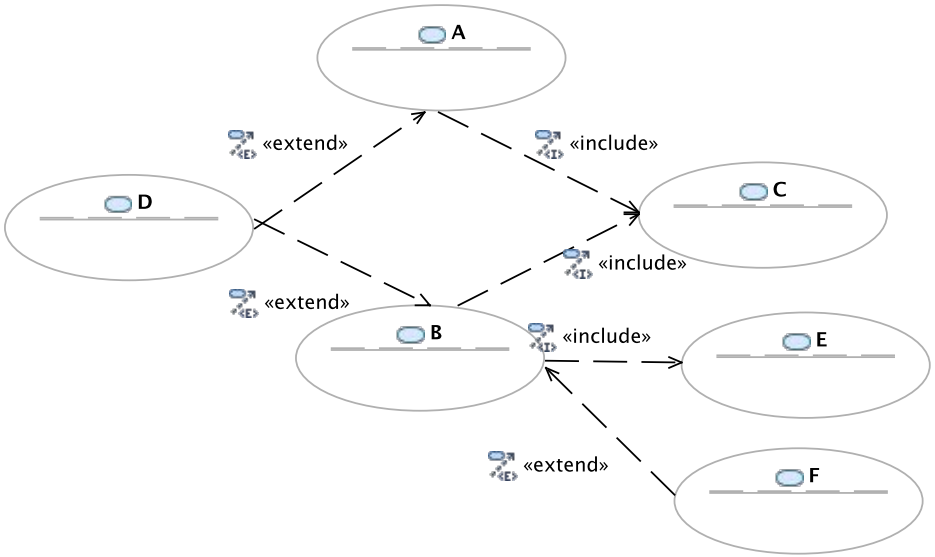


Reengineering UML use case diagrams for fixing generalization defects

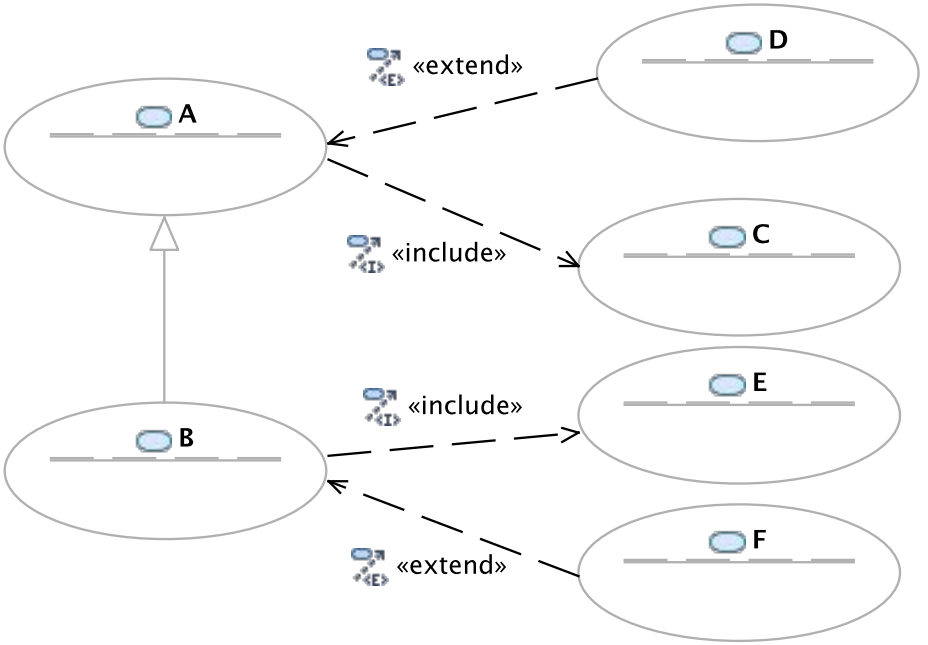


RCA: global application of refactoring Patterns

e.g. combining *include* and *extend*



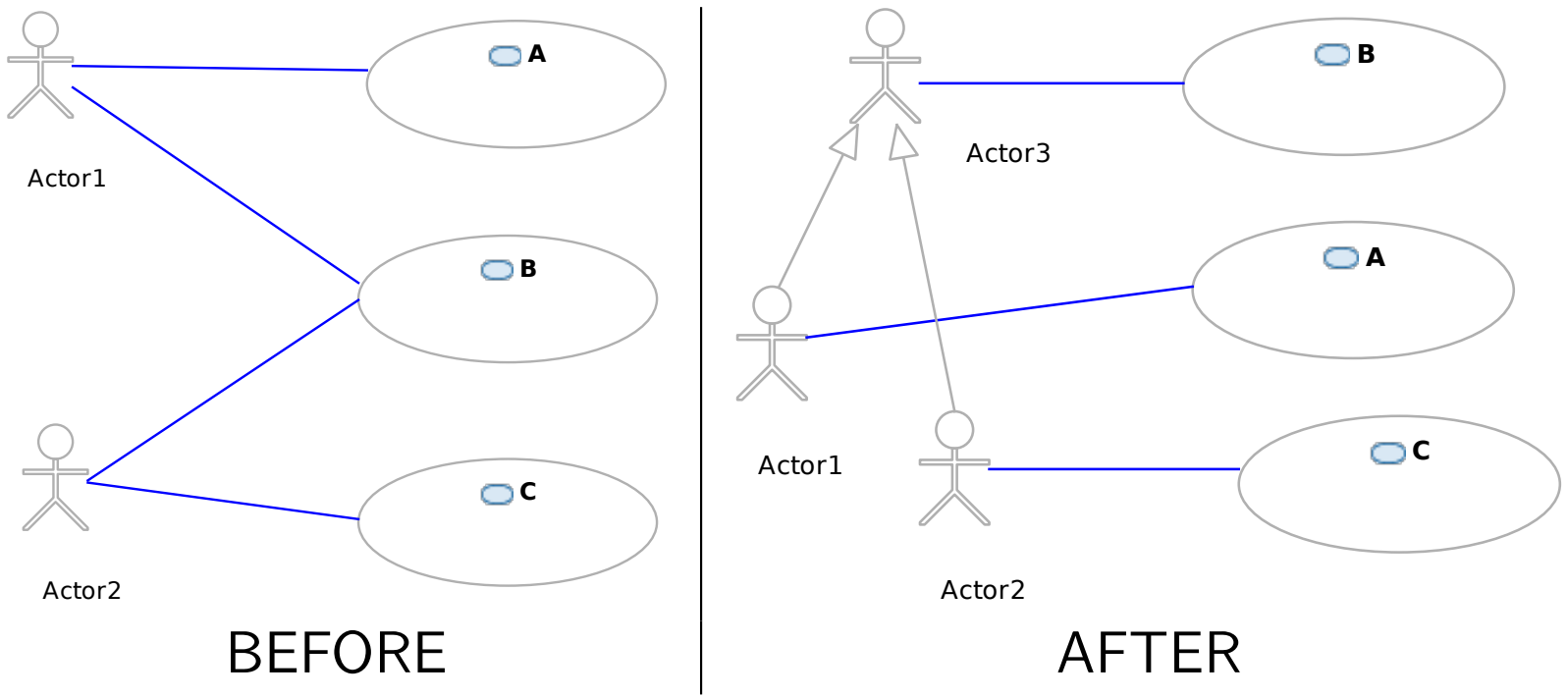
BEFORE



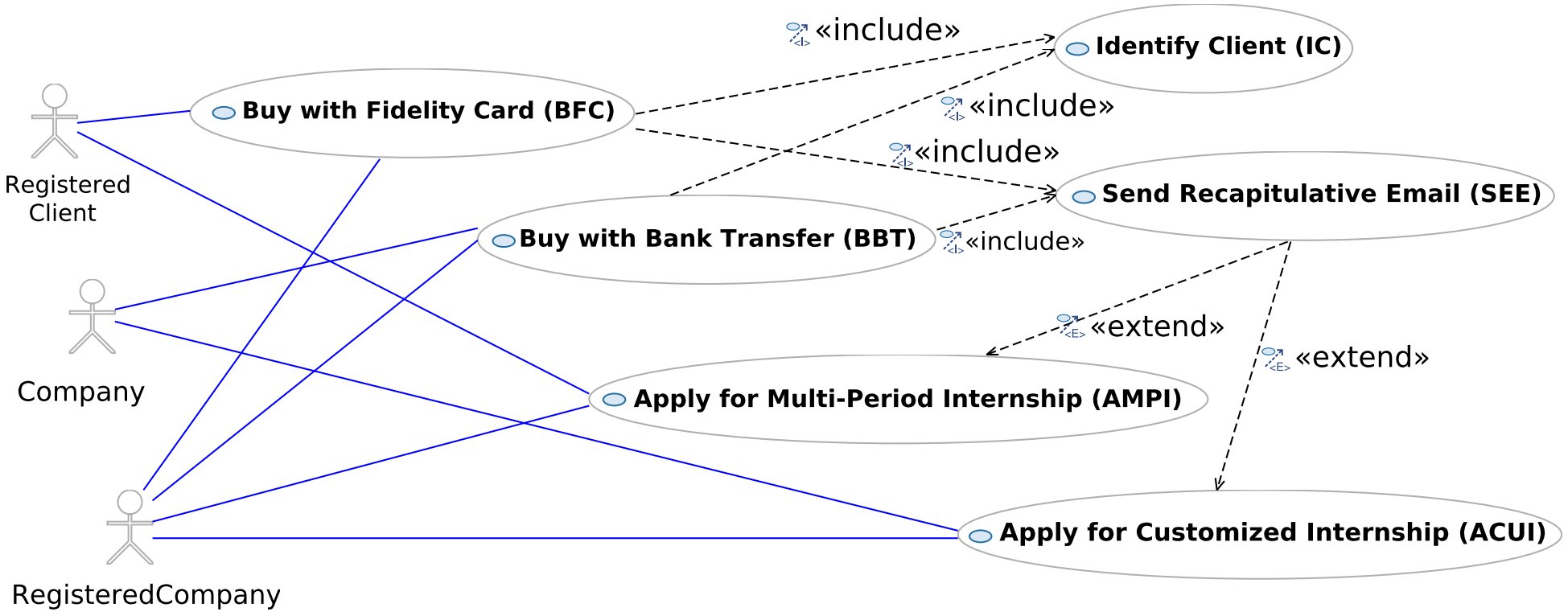
AFTER

RCA: global application of refactoring Patterns

e. g. *Actor* factorisation



Illustrative example of the Internship Subscription



Chosen relational schema

Formal Contexts

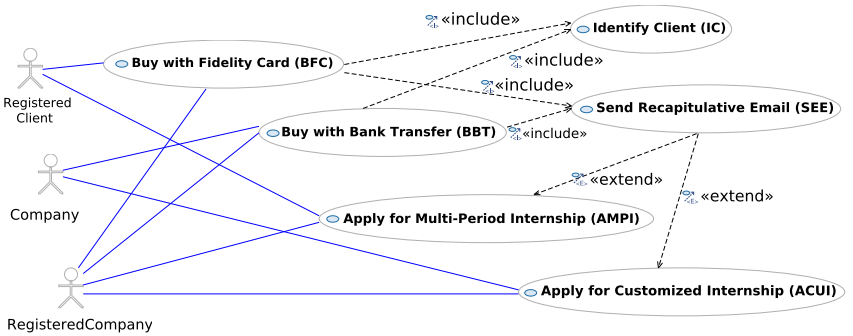


Table: Formal contexts for RCA refactoring: elements to be classified

use case	Name					
	BFC	**SEE**	**AMPI**	**BBT**	**ACUI**	**IC**
BFC	×					
SEE		×				
AMPI			×			
BBT				×		
ACUI					×	
IC						×

actor
RegisteredClient
Company
RegisteredCompany

Relational Contexts

relations between formal contexts

Table: Relational Contexts for RCA refactoring: relations from Use Cases to Use Cases

Includes	BFC	SEE	AMPI	BBT	ACUI	IC
BFC		×				×
SEE						
AMPI						
BBT		×				×
ACUI						
IC						

IsExtendedBy	BFC	SEE	AMPI	BBT	ACUI	IC
BFC						
SEE						
AMPI		×				
BBT						
ACUI		×				
IC						

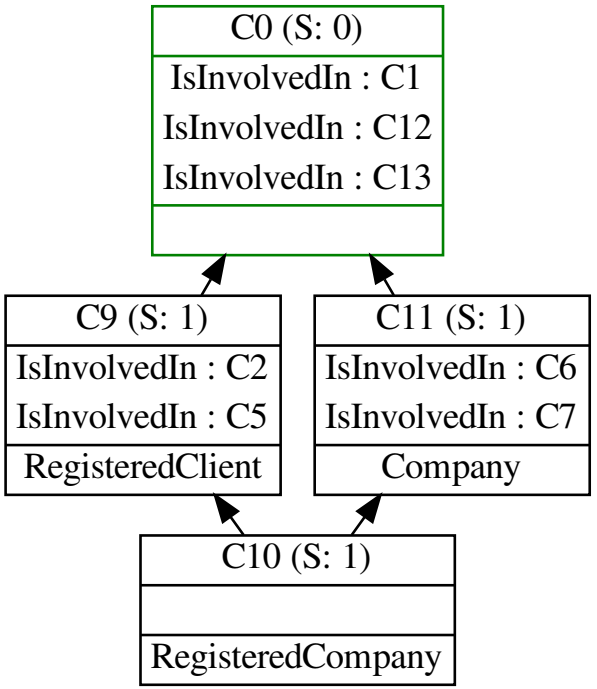
Relational Contexts

relations between formal contexts

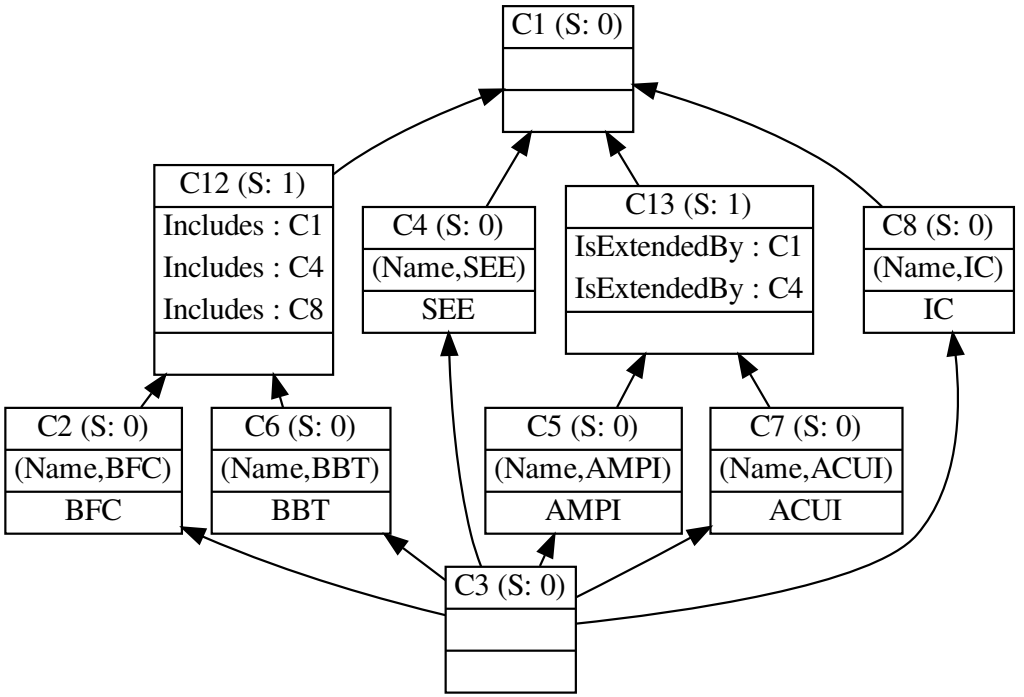
Table: Relational Contexts for RCA refactoring : relations from Actors to Use Cases

IsInvolvedIn	BFC	SEE	AMPI	BBT	ACUI	IC
RegisteredClient	×		×			
Company				×	×	
RegisteredCompany	×		×	×	×	

Concept lattice family

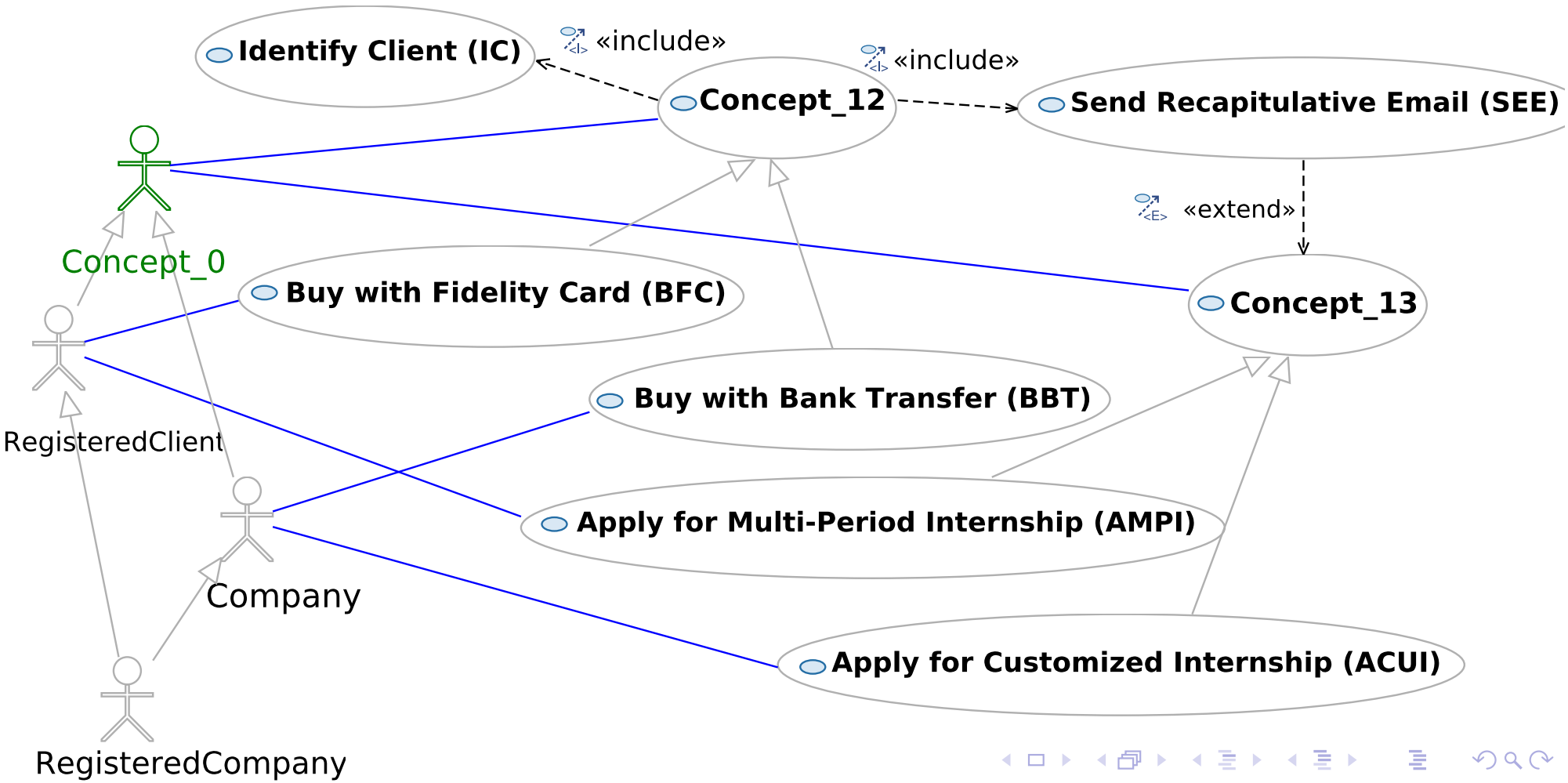


Actors lattice



Use Cases lattice

RCA Final Diagram



Obtained results

Case study

- Tests on 24 use case diagrams from different sources
- Metrics for evaluating the "simplification"
  - Density($|edges|/|nodes|^2$)
  - Degrees Average ($2|edges|/|nodes|$)
  - Maximal degree
- Metrics improved except in 2-6 diagrams

Perspectives

- Extend this work by adding other sources of information such as scenarios

Analyzing factorization evolution

Context: Environment and Territory domains (IRSTEA)

- Development of Information System involves many actors and scientists
- Actors often have different (sometimes opposed) viewpoints
- Designer has to merge various viewpoints in a global UML diagram that evolves progressively
- There is a need for support the evolution of systems

How study systems evolution ?

Classical model indicators

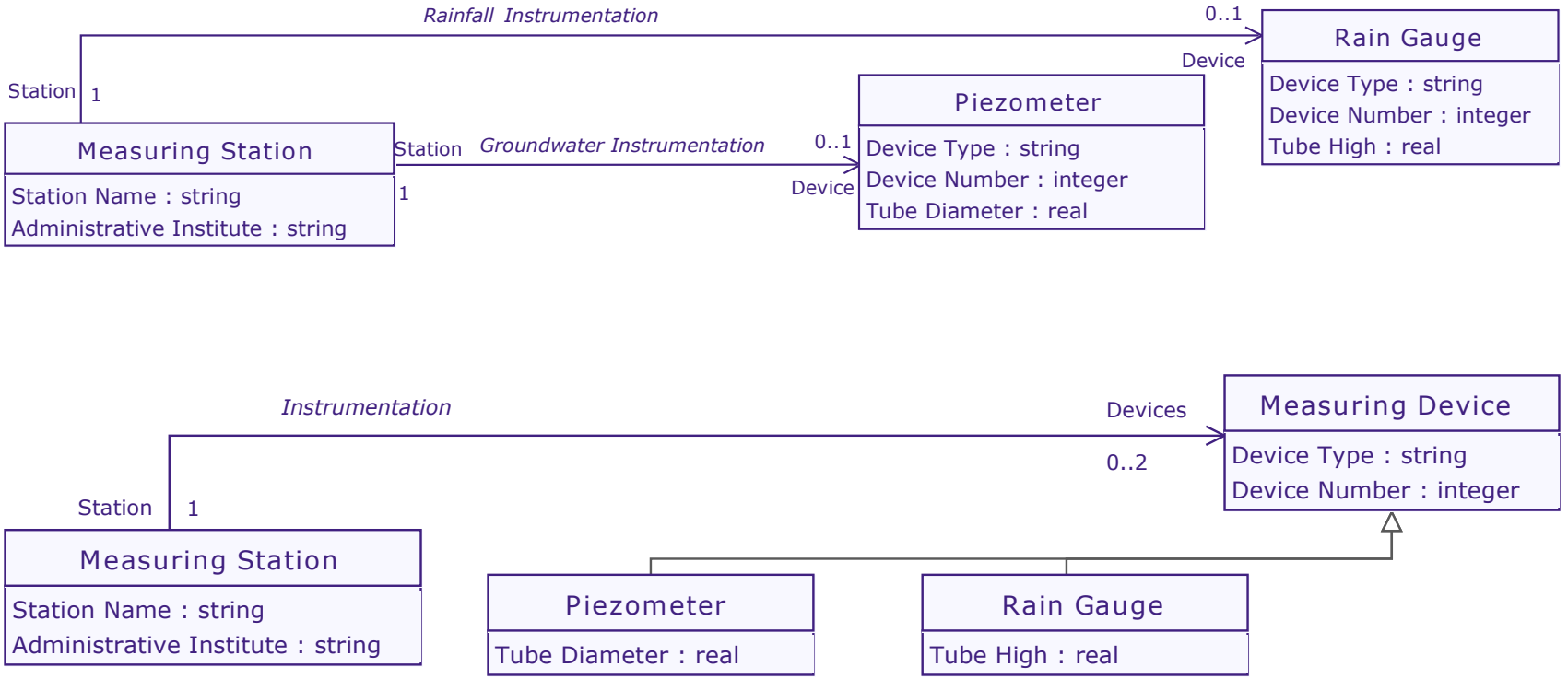
For example, number of elements of various kinds (classes, methods. . .)

- Do not reveal complex evolution :
  - precision in the description of model elements
  - level of abstraction and factoring

Proposition

Develop indicators based on the application of Relational Concept Analysis

RCA : abstractions on all UML elements

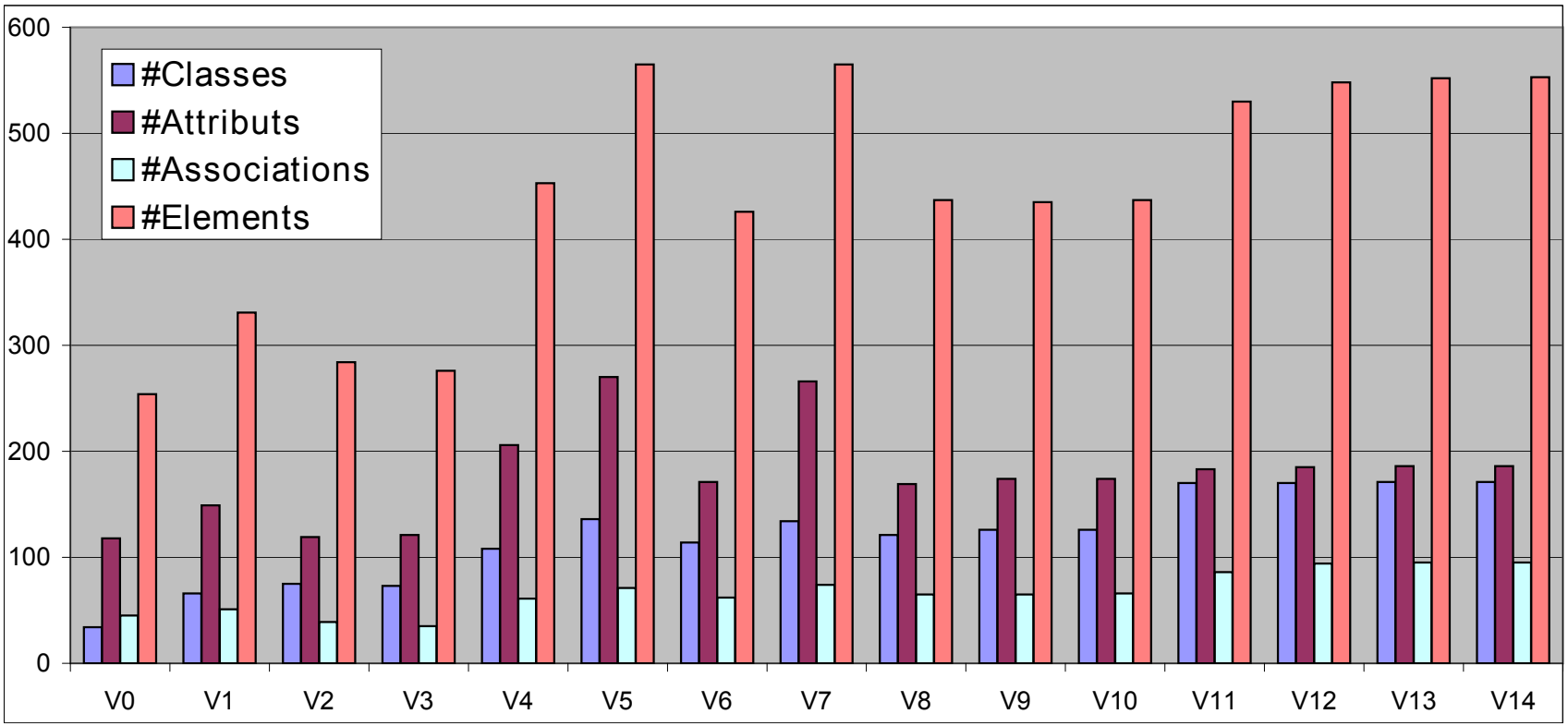


EIS-Pesticides : an Environmental Information System

- Pesticides model groups together the knowledge and information produced by :
  - the Transfer Team, which studies the transfer of pesticides to the rivers
  - the Practice Team, which studies the agricultural practice of the farmer
- UML is used to capitalize the knowledge
- Finally, the UML model will be transformed in database schema
- During the analysis phase, models are archived before each major change

A brief History of EIS-Pesticides

- 15 versions
- Evolution of the different model elements:



Studied RCA configurations

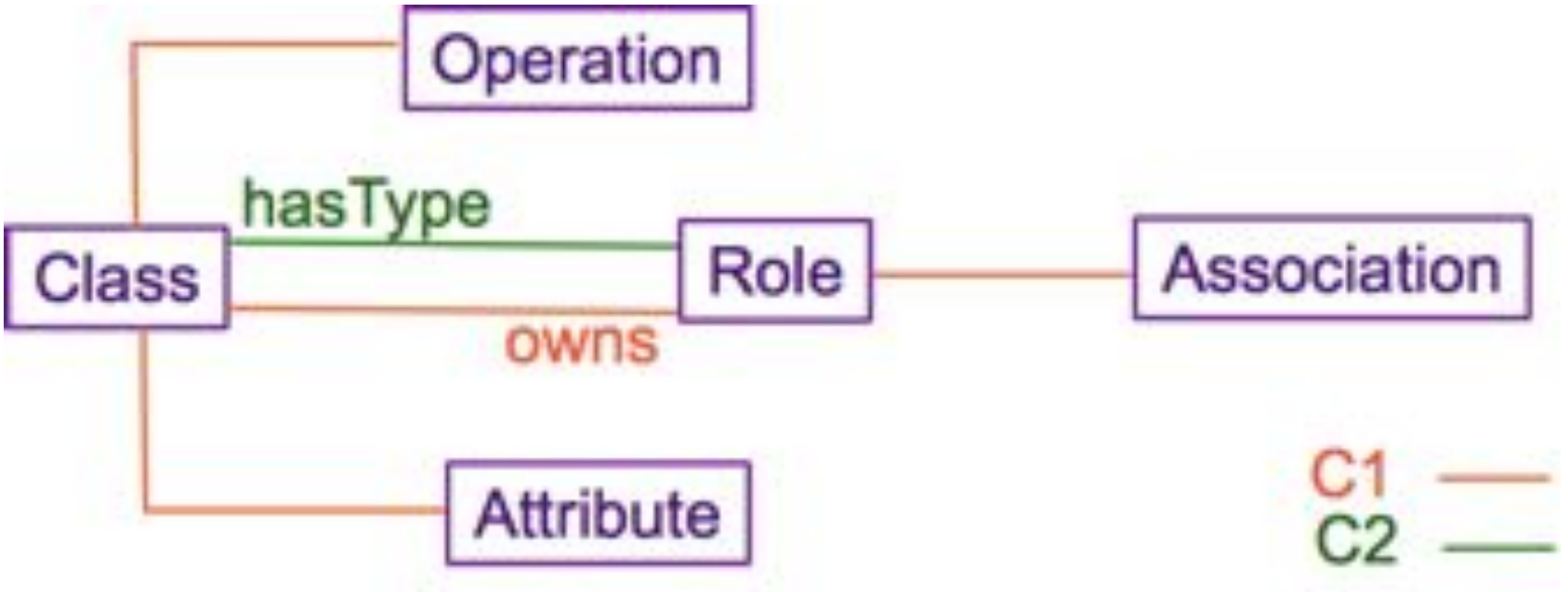
C1 – presented in this talk

- $owns_1 \subseteq classes \times attributes$
- $owns_2 \subseteq classes \times operations$
- $owns_3 \subseteq classes \times roles$
- $owns_4 \subseteq Associations \times roles$

C2

- C1
- $hasType_5 \subseteq roles \times classes$

Chosen relational schema



Lattice indicators evolution

As RCA produces an unique normal form, our metrics are based on the comparison of these normal forms

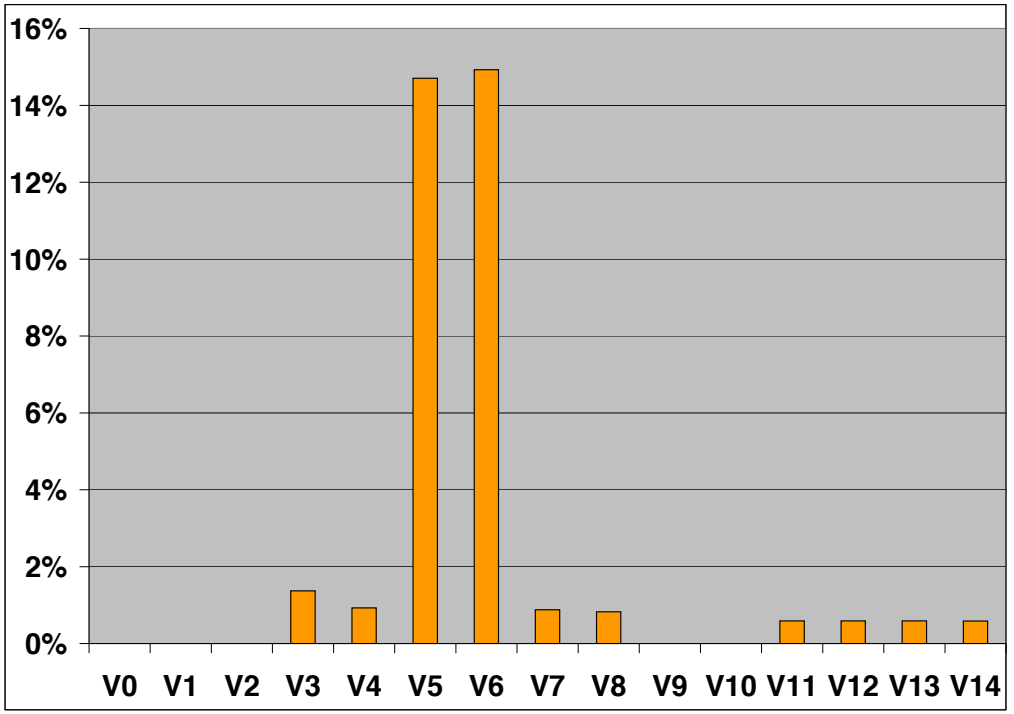
The ratio of merged concepts: #Merge / #Model Elements

- Merged Concepts have a proper extent that contains more than one element
- They merge several formal objects with the same description

The ratio of new concepts: #New / #Model Elements

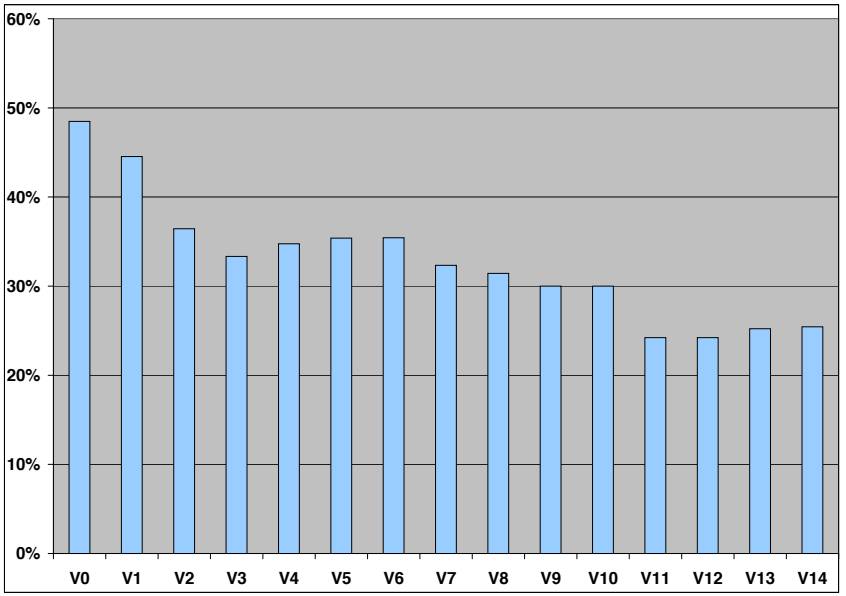
- New Concepts have an empty proper extent
- They factorize formal attributes

Indicators on Classes : Merged Classes



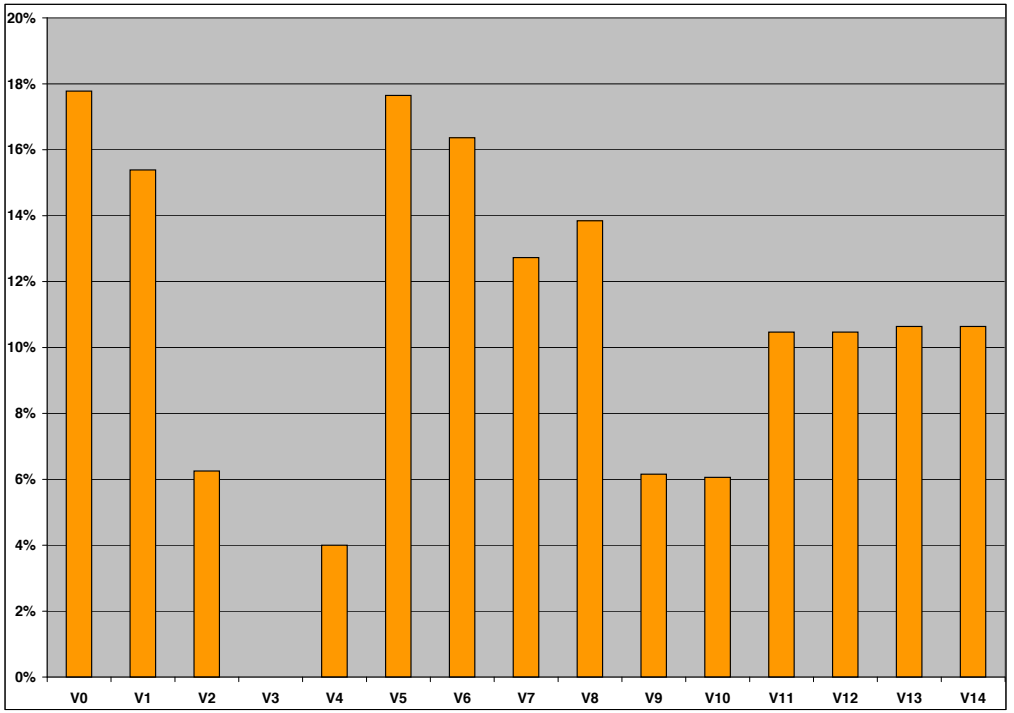
- Here the metric reveals package duplication in V5, V6
- V3, V4, V7, V8, V11-V14: a few classes with poor description

Indicators on Classes : New Classes



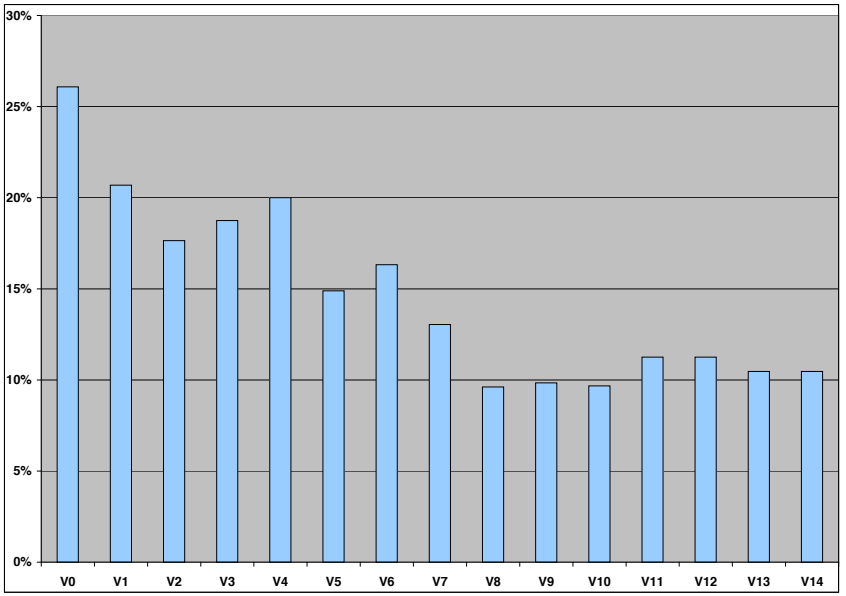
- Progressive decrease even if the number of classes increases
- The abstraction level of the model improves
- V5, V6 : the package duplication degrades the abstraction level

Indicators on Associations : Merged Associations



- Decreasing tendencies
- Very low factorization in the first versions and in the package duplication case (V5-V6)

Indicators on Associations : New Associations



- Expresses the potential lack of association abstractions
- V0 to V9 : decreasing tendencies (factorization improves)
- increases in the last versions : new added associations need factorization

Discussion

Classical metrics to analyze

- Evolution of data encapsulation ($\simeq$ number of classes)
- Evolution of the completion of the model ($\simeq$ number of attributes)
- Evolution of the relational aspect ($\simeq$ number of roles / associations)

RCA-based metrics to complete the analyze

- Evolution of the merged ratio indicates if identical or badly described described model elements are introduced
- Evolution of the new ratio indicates the level of abstraction

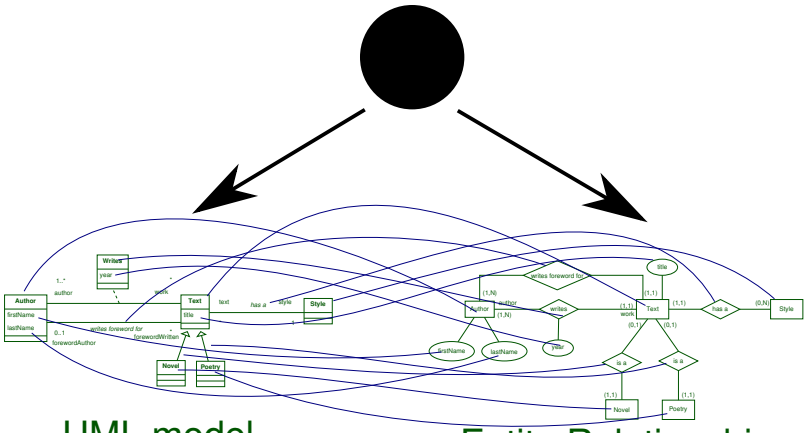
Discussion

- RCA-based indicators to observe model evolution
- Indicators are systematically computed for the 15 versions of Pesticide models (about 200 classes)
- Recommendations have been extracted from these metrics, confirmed by the story of the model

Perspectives

- Implement traceability links to better monitor model evolution and assist the designer
- Control the new concepts that emerge from RCA to reconstruct relevant normalized UML models

Learning model transformations: Input Data

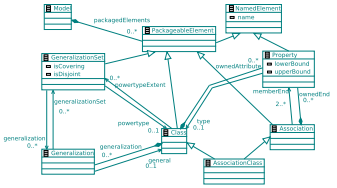


UML model

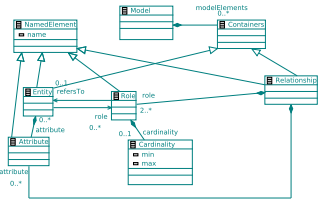
Entity-Relationship model

conforming to

conforming to

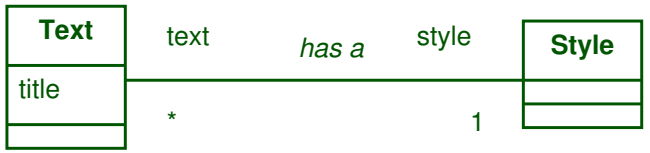


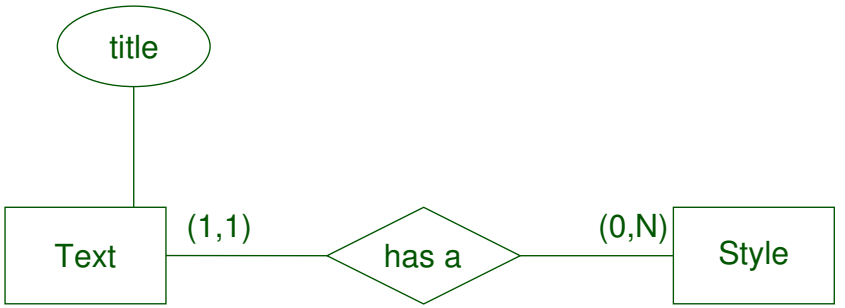
UML metamodel

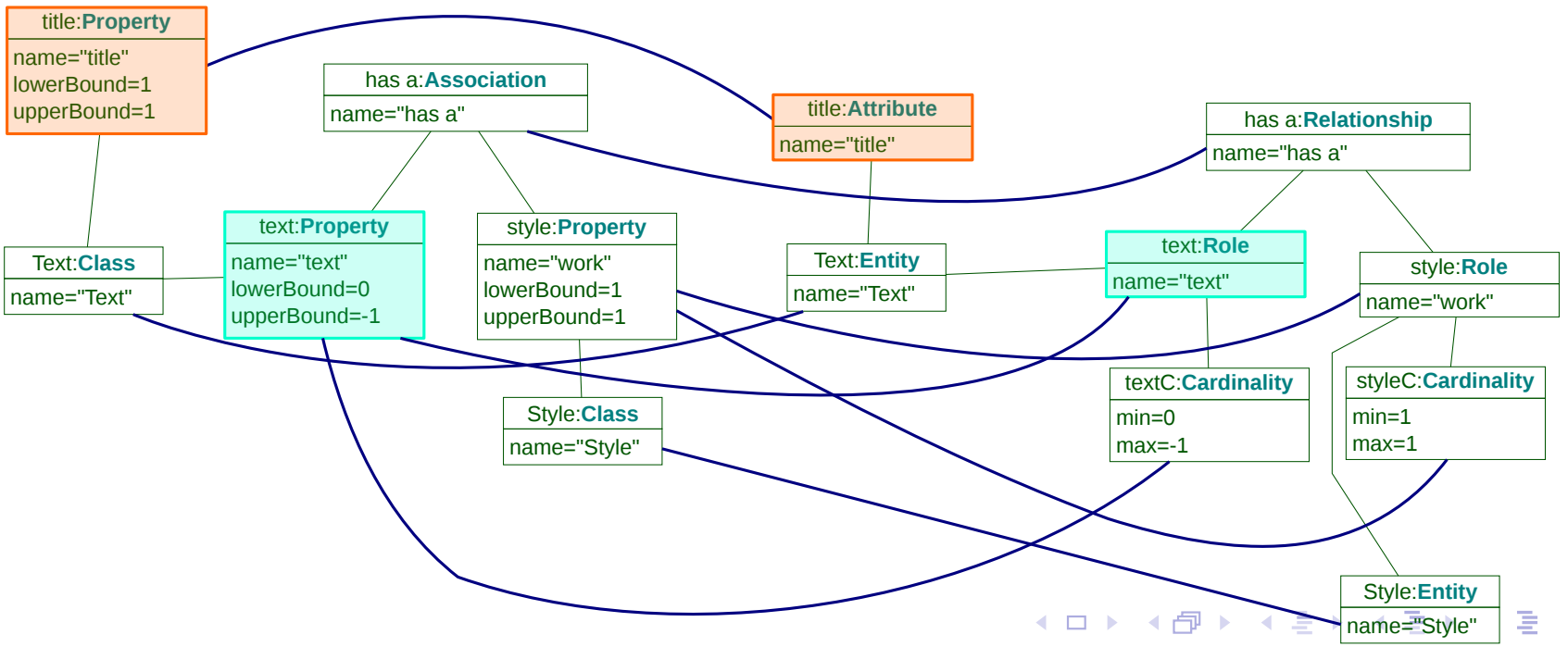


Entity-Relationship metamodel

Learning model transformations: Input Data (excerpt)







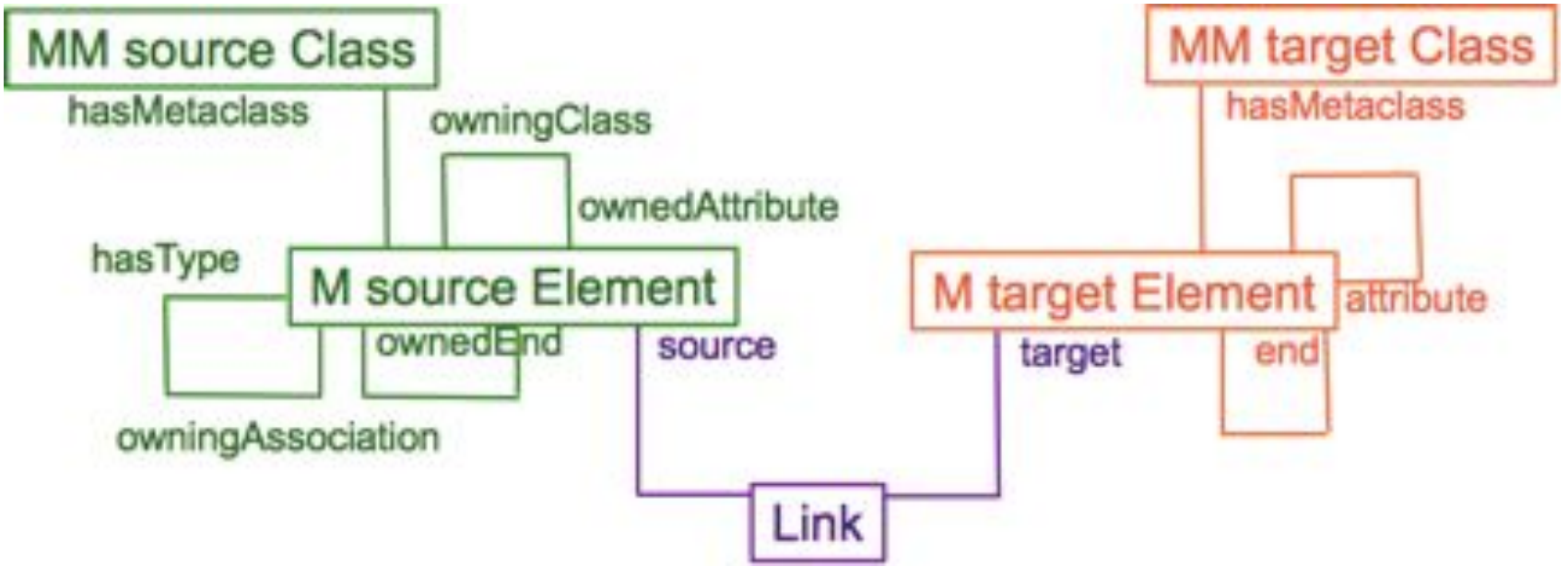
Learning model transformations: Encoding data

- Consider the properties of the model elements:
  - their meta-class
  - their relations with their neighbors
  - the properties of their neighbors
- Classify the different properties from the examples
- Classify the matching links considering the classification of the properties of their extremities

Relational Concept Family

- Metamodel (MM) elements contexts
  - Source MM context
  - Target MM context
- Model (M) elements contexts
  - Source M context
  - Target M context
- Matching links context
- Relations
  - between M source elements and their meta-class from the source MM
  - between M target elements and their meta-class from the target MM
  - between M source elements: *e.g. ownedAttribute*
  - between M target elements: *e.g. attribute*
  - between matching links and their source from the source model
  - between matching links and their target from the target model

Chosen relational schema



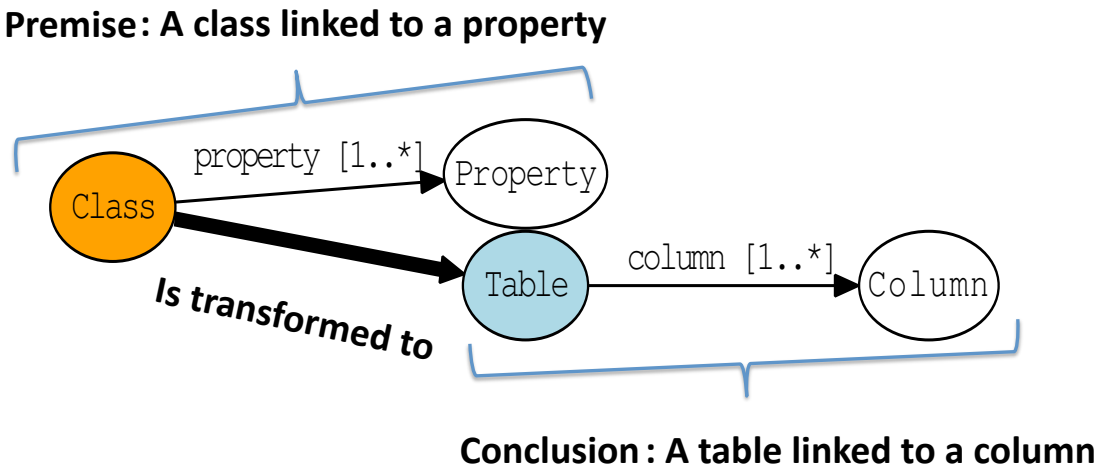
The obtained Concept Lattice Family

- 2 concept lattices of MM elements: source and target
- 2 concept lattices of M elements: source and target
- 1 concept lattice of links

From the concept lattice of links, we deduce a lattice of transformation patterns

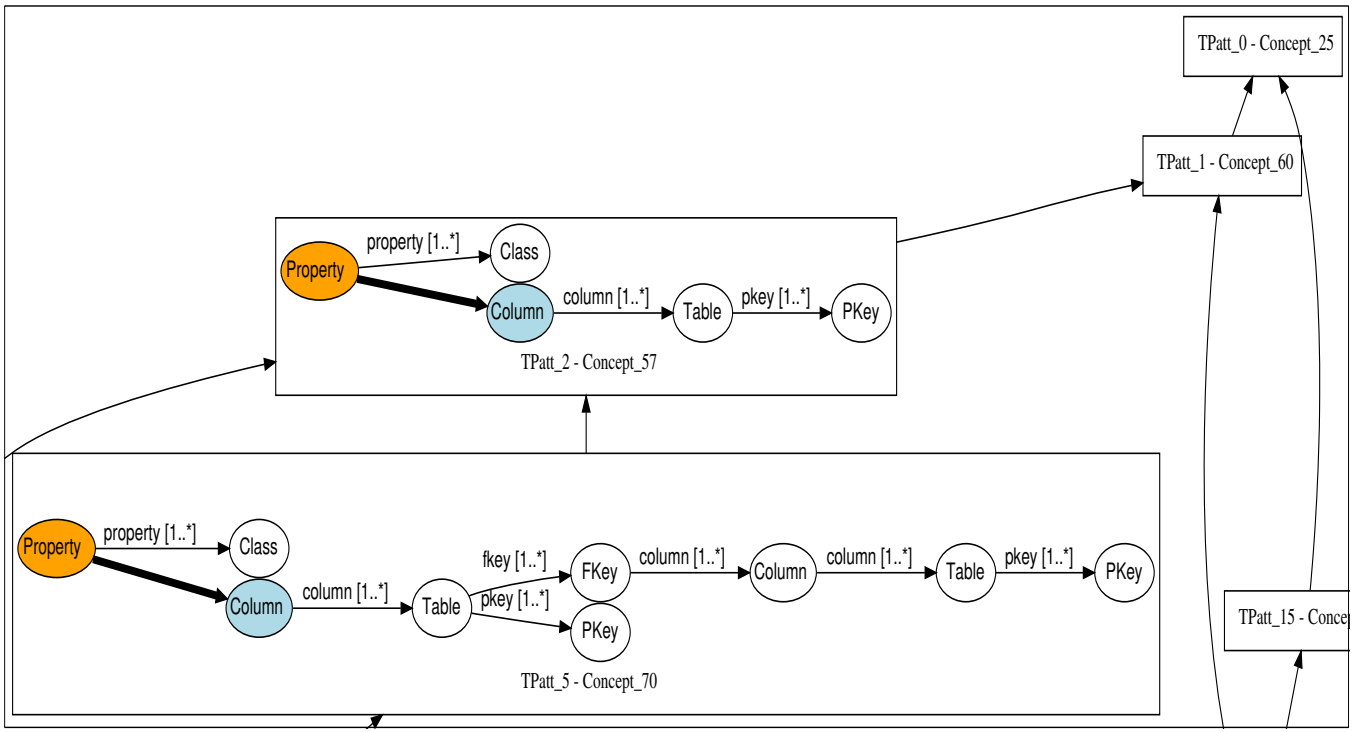
Obtained Transformation Pattern lattice

A transformation pattern



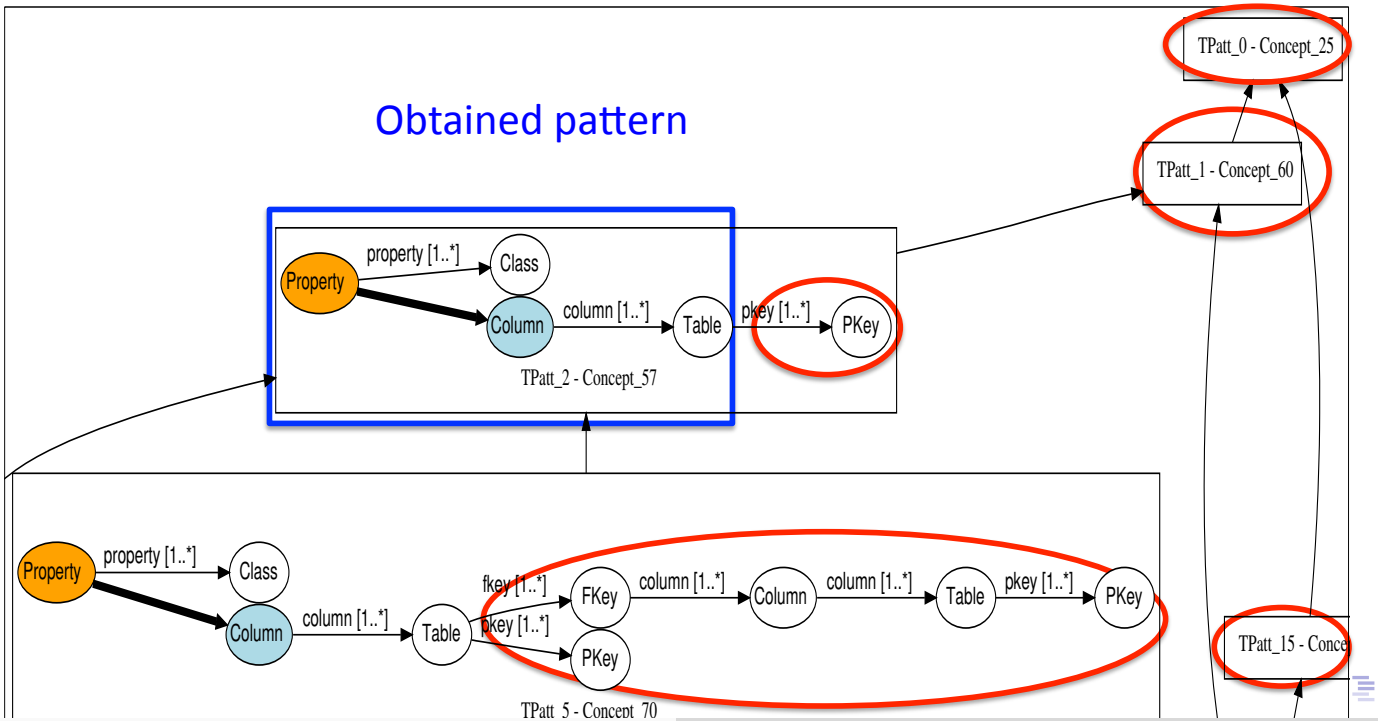
Obtained Transformation Pattern lattice

The lattice



Obtained Transformation Pattern lattice

The filtered lattice

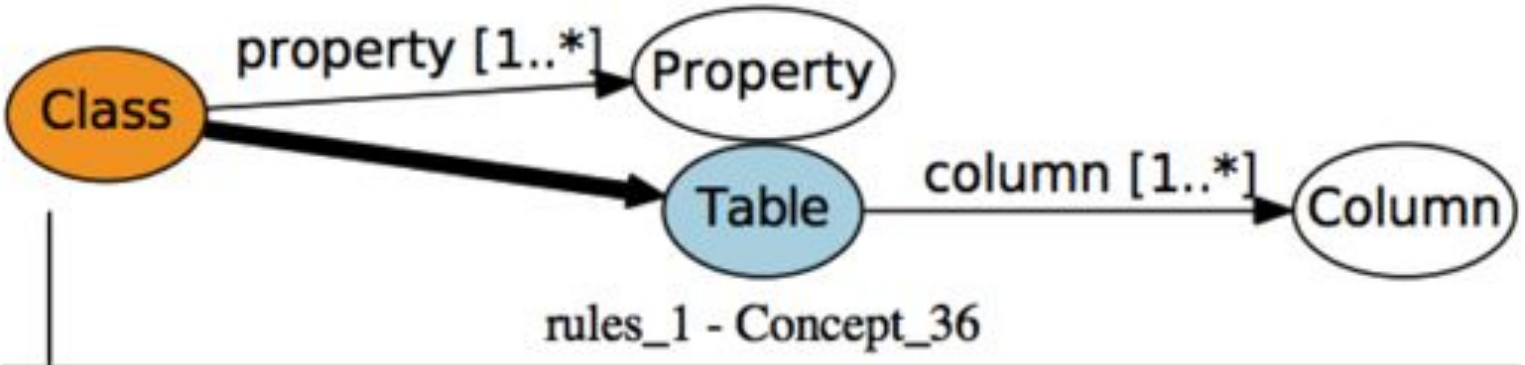


Using Jess as rule engine

Jess

- Rule engine integrated in the Java platform
- Declarative language: adapted to our patterns
- A Jess program is mainly composed of facts and rules
  - Facts: data (conform to templates)
  - Rules: conditions and actions

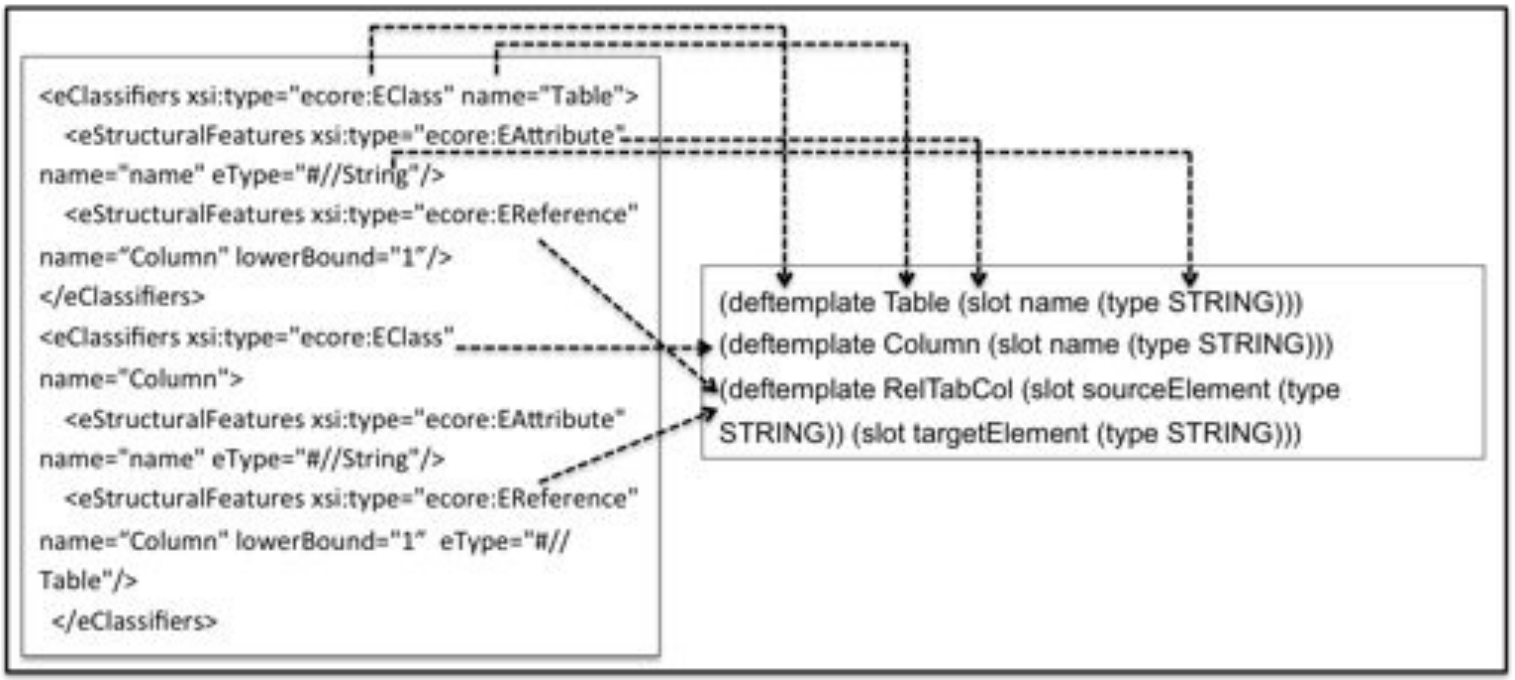
transformations rules: Jess rules



rules_1 - Concept_36

```
(defrule R1
(Class (name ?i) )
(Property (name ?j) )
(Rel_Property (sourceElement ?ref1) (targetElement ?ref2) )
(test (eq ?i ?ref1) (eq ?j ?ref2) )
=>
(assert (Table ?i) )
(assert (Column ?j) )
(assert (Rel_Column (sourceElement ?i) (targetElement ?j) ))
)
```

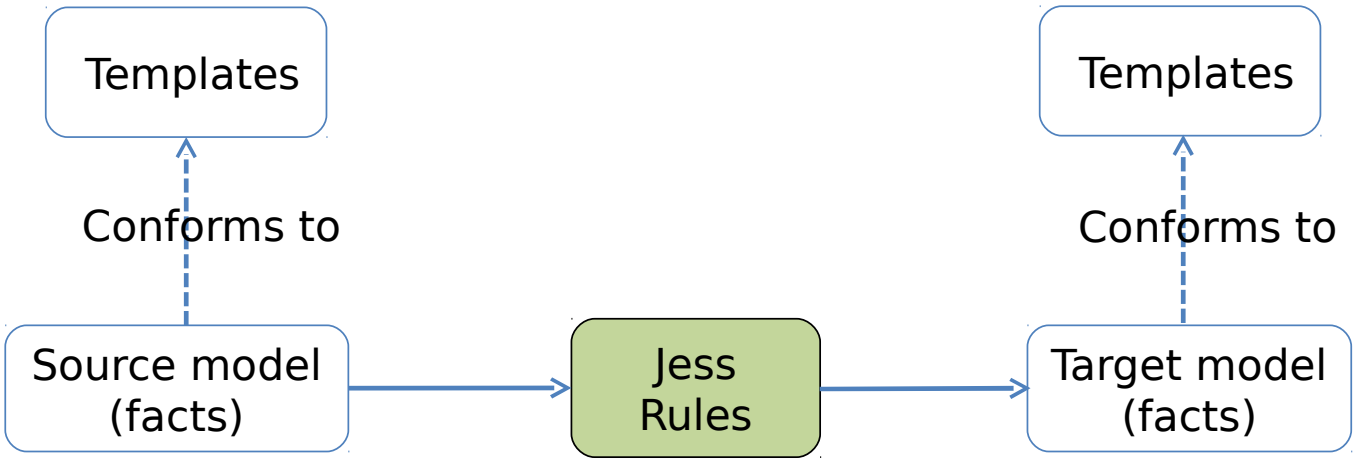
Meta-models: Jess templates



Models: Jess facts

```
1  (MAIN :: Table (name Client))
2  (MAIN :: Pkey (name Client_Nbr))
3  (MAIN :: Rel_pkey (sourceElement Client) (targetElement Client_Nbr))
4  (MAIN :: Column (name Client_Name))
5  (MAIN :: Rel_Column (sourceElement Client) (targetElement Client_Name))
6  (MAIN :: Column (name Address))
7  (MAIN :: Rel_Column (sourceElement Client) (targetElement Address))
8  (MAIN :: Table (name Reservation_Request))
9  (MAIN :: Pkey (name Reservation_Nbr))
10 (MAIN :: Rel_pkey (sourceElement Reservation_Request) (targetElement Reservation_Nbr))
11 (MAIN :: Fkey (name Client_Nbr))
12 (MAIN :: Rel_pkey (sourceElement Reservation_Request) (targetElement Reservation_Nbr))
13 (MAIN :: Column (name Start_Date))
14 (MAIN :: Rel_Column (sourceElement Reservation_Request) (targetElement Start_Date))
15 (MAIN :: Column (name End_Date)
16 (MAIN :: Rel_Column (sourceElement Reservation_Request) (targetElement End_Date))
```

Applying the obtained Jess rules



- `Fact2Model` not yet implemented

3-fold cross validation on 30 examples: Excerpt of the results

Examples	Fold 2	
	Precision Average	Recall Average
1	0.78	0.79
2	0.90	0.75
3	0.85	0.77
4	0.77	0.79
5	1	0.80
6	1	0.77
7	0.85	0.77
8	0.85	0.80
9	1	0.75
10	1	0.80

- Precision: ratio of correct transf. elements among all the computed transf. elements (average for all types)
- Recall: ratio of correctly found transf. elements among all the expected transf. elements (average for all types)

Conclusion

RCA

- A data analysis method, clustering / unsupervised learning technique
- focus on classifying objects in different interconnected classifications
- progressive concept emergence

Other applications

in KR and DM

- Ontology construction (R. Bendaoud, M. Rouane Hacene, Y. Toussaint, B. Delecroix, A. Napoli)
- Ontology restructuring (M. Rouane-Hacene, R. Nkambou and P. Valtchev)
- Analysis of course water quality (X. Dolques, F. Le Ber)
- Discovering hidden user profiles in a semantic actors-activities network (Z. Azmeh, I. Mirbel)