

A Proposed Architecture to Index Courses on Website and Analyse the Student Follow Up

Michel Sala, Pierre Pompidor, Danièle Hérin, Gaël Isoird

► **To cite this version:**

Michel Sala, Pierre Pompidor, Danièle Hérin, Gaël Isoird. A Proposed Architecture to Index Courses on Website and Analyse the Student Follow Up. DEXA'04: 15th International Conference on Database and Expert Systems Applications, Aug 2004, Zaragoza, Spain. pp.208-212, 2004, <10.1109/DEXA.2004.1333475>. <lirmm-00108775>

HAL Id: lirmm-00108775

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00108775>

Submitted on 23 Oct 2006

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.

2.2. Educational information

We assume that the teacher does not possess any educational platform but that he provides a set of teaching aids on his course web site. These resources can be of various types such as: bibliographical references, videos, links to web sites or supplementary course material. Thus, it is necessary to define these resources which form the foundation for the course.

By using metadata [4][5][6], we can create an effective system for describing the resources. But, while a student is working with the system, access to all the information might not be useful. Thus, it is necessary to decompose the information.

2.3. Educational modules

First, one needs to obtain the non-decomposable minimal educational module. The teacher must select exactly the part of the resource that is crucial for the course. The less information that each educational module contains, the easier it is to follow the student's learning path. We can even define levels which allow us to more easily analyze the students' learning styles and customize the course. Thus, we can thus create an educational module in the shape of an n-tuple:

{id-module, id-resource, section, level}

where id-module and id-resource represent respectively the primary id of the educational module and the referenced educational resource. The "section" is a part of the useful resource for the course and the "level" represents all the levels to which this educational module can be attributed. However, a teacher may wish to impose a logical order of educational modules for the same concept. This is the basis for composing educational modules (complex modules).

A complex educational module can be represented by a diagram whose nodes are simple educational modules with arrows indicating the order in which nodes should be accessed. The only constraint is to compose complex educational modules with simple modules on similar levels.

The chains can be expressed as follows:

- Orderly sequence: $A_i -> A_j$ the students must learn educational modules A_i and A_j in sequence beginning with A_i and can access A_j only if A_i is mastered.
- Parallel : A_i and (*) A_j the students have to learn both educational modules but without notion of order.
- Alternative: $A_i \mid A_j$ the students must learn at least one of the two educational modules.

For example, a teacher provides educational modules $A_1 \dots A_8$ and decides to order them as represented in figure 2.

Thus, the teacher wants the student to work first on A_1 then on A_2 and A_3 (one can work on A_2 and A_3 in either order) then on A_4 . After A_4 activating either A_5 or A_6 then A_7 and A_8 .

The teacher then creates the complex educational module $GC = (A_1 -> (A_2 * A_3) -> A_4 -> (A_5 \mid A_6) -> A_7 -> A_8)$

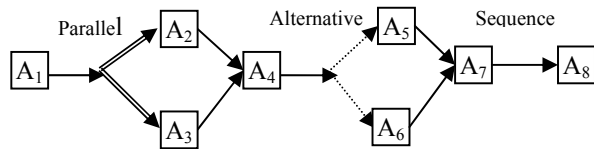


Figure 2. Example of composition of modules

To create the complex educational modules, we have to modify the structure. An educational module is then represented by the n-tuple:

{id-module, levels, type, sequence }

where the "type" indicates whether the educational module is simple or complex, and the "sequence" is a series of doublets (id resource, section) linked with the operators $->$, $*$ (and) or \mid .

For a simple educational module, the sequence contains only two parameters (id resource, section).

2.4. The notion of global ontology

The scientific community has shown interest in the concept of ontologies for representing knowledge as well as making their sharing and their reusing easier. We refer to global ontology when we reference the knowledge proposed for the learning within the course structure. The global ontology provides a semantic representation of all the concepts and their relations. This global ontology is represented by a diagram of the concepts $G = \langle C, A \rangle$ where C represents concepts and A their relations. Ontology plays a key role in the representation and the use of knowledge. It provides a definition of the vocabulary used to represent knowledge, but doesn't limit itself to a simple list of terms. It also must provide the semantic interpretation of these terms.

2.5. The graphical interface of the site

In order to follow a student's progress, it is necessary to know the site map. This can be represented by a directed graph in which pages are represented by nodes and links by arrows.

The graphical interface of the site is stored in a text file in the form: $\{a, b\}$ where "a" represents the URL of origin and "b" the destination URL.

3. Indexing courses on Website

Once the basic building blocks of our architecture are defined, we assign the educational modules and web pages to the different nodes. For this task we have chosen XML. These assignments lead to modular global ontologies and labelled global ontologies. Once these assignments are made, we focus on the problem of categorization.

3.1. The graphical interface of the site

At the present time, many sites are still written in HTML. This "language" doesn't have any data type definition and doesn't permit the description of a web page from a semantic viewpoint. For this reason, we capture the information contained in the web site, then describe it using an XML formalism [7]. Also the description of the document structure must be valid. An ontology can easily be defined using XML, with standardization achieved using the w3c (World Wide Web Consortium) guidelines.

3.2. Creation of the Labelled Global Ontology

The creation of the labelled ontology compares the students' steps in two stages with the global ontology.

The first consists of assigning the educational modules and classifying them by levels. The second phase is linking the global ontology in the site's graphical interface

3.2.1. Modular Global Ontology

The first phase to determine the modular global ontology, each concept is defined according to its hierarchy. Adding a new level requires a more precise analysis of the user's path through the course. We can define, for any given concept and any given level, the set of educational modules (simple or complex) linked to this concept at this level. With these guidelines it is possible to determine if a student at level X, wanting to study concept Y, has actually studied all X level modules classified in concept Y.

This structure can be defined in XML by the code below:

```
<concept label= «C»>
  <level1>
    <module id=«1» levels=«1,2» type= «simple»
      sequence = «sequence 1» >
    </module>
    <module id=«2» levels= «1» type= «complex»
      sequence = «sequence 2» >
    </module>
    <module id= «3» levels = «1» type= «simple»
      sequence = «sequence 3» >
    </module>
  </level1>
</level2>
```

```
...
</level2>
...
```

3.2.2. Labelled Global Ontology

The labeling phase of the global ontology is essential to our project; as it establishes the link between the semantics and the graphical interface. We attribute a tag <URL>URLi</URL> to each concept level contained in the URLi page.

```
<concept label= «Association»>
  <level1>
    <module id= «1» levels = «1,2» type= «simple»
      sequence = «sequence 1» >
    </module>
    <module id= «2» levels = «1» type= «complex»
      sequence = «sequence 2» >
    </module>
    <module id= «3» levels = «1» type= «simple»
      sequence = «sequence 3» >
    </module>
    <URL>URL1</URL>
    <URL>URL2</URL>
  </level1>
  <level2>
    ...
  </level2>
  ...
```

3.3. Management of Categories

In order to follow the educational progress of all the students and to classify them by category, it is necessary to divide the students using common criteria. This categorization is essential since it establishes a reference point that enables us to discover some potential course design problems. In this article, we do not discuss the ways the categorizations are calculated since projects already exist in this domain such as: [8], [9].

Categorization is achieved by comparing the data dynamically collected from the students' logs as they proceed through the course; possibly allowing them to move to another level based on test results.

We first highlight the browsing patterns. The browsing patterns represent the reference users' site use. The paths correspond to links of pages through the set of sessions. The recurrent use of pages show successions of pages frequently appearing in the session group. We use an Apriori type algorithm to determine the frequent sets of pages and an AprioriAll type algorithm to extract the frequent sequences [10][11]. Then, we compose the session groups related to these browsing patterns.

In the second stage we form groups of users based on these session groups. Once this stage is completed, we obtain groups of users whose common point is that they have the same browsing pattern; these two stages are represented in figure 3.

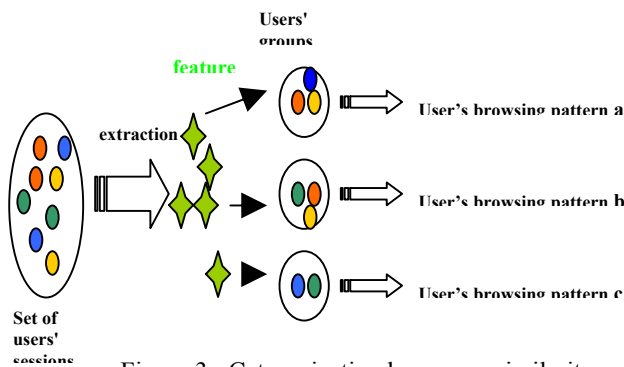


Figure 3 : Categorization by course similarity

3.4. Extraction of local ontologies

3.4.1. Difference between local and contextual ontologies

Both local and contextual ontologies are sub-types of the global ontology. Therefore they have a similar structure but they differ in the analysis of the information. When studying a local ontology, we analyze a single user's path, which provides particular information about the student. Conversely, when studying a contextual ontology, we consider the path of a group. The data gathered is useful for the teacher. It enables him to determine if all the students have understood the module well enough or if some modifications are necessary.

3.4.2. Structure of Local and contextual ontologies

As we have just discussed, local and contextual ontologies have similar structures. They are extracted from the global ontology according to the students' paths. For that purpose, we use a contextual ontology extraction algorithm which occurs in two stages :

The first stage consists in browsing the global ontology tree in a post-fixed traversal and checking for each node if a concept is visualized, articulated, or neither. The LGO visualized concepts are those which are present in the pages browsed by the category of users.

Articulation is a concept that is not visualized and where the N number of descendant nodes (including itself) is greater than the N number of each of its descendants. Every time a concept is considered we insert it into a list. The last element of the list will be the contextual ontology root (cf figure 6).

The second stage consists in creating the minimal tree that covers both the visualized and articulation concepts. For each node of the previously constituted list, we browse the tree via the father-son link and stop as soon as we find a node present in the list. Each node browsed will be part of the contextual parallel

ontology. The algorithm stops when we reach the root. At the end of this stage, we re-initialize the global ontology tree (cf figure 4).

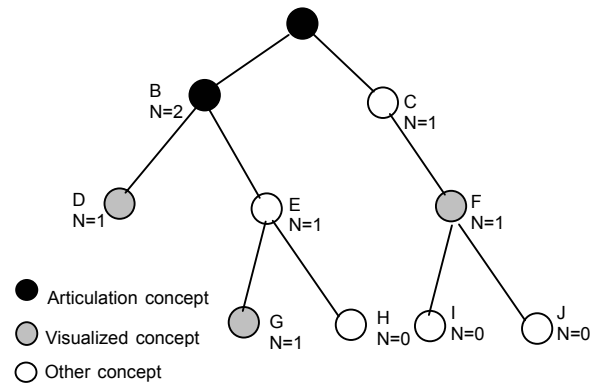


Figure 4 Local ontology design

3.4.3. Adding information to the local ontology

If we obtain additional information (such as grades for instance), we can add this to the algorithm previously described. The aim of this phase is to include data summarizing the students' success on the different concepts in these ontologies. This new information is added as attributes of local or contextual ontology. We update an "average" attribute for the concept as well as for each level (cf part 4).

4. Students' follow-up

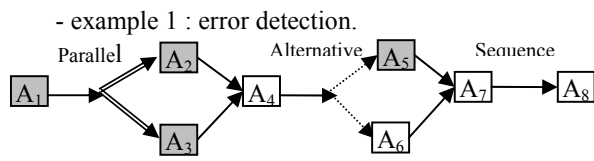
4.1 The mechanism of knowledge extraction

As each student browses the web a sequence of web pages is dedicated to his study path. The server's logs will allow us to determine a unique sequence for every student or group. With our labelled global ontology structure, we are able to determine the level and concept chosen. Based on that, we can extract a local ontology (for a student) or a contextual ontology (for a group). If we can retrieve extra data, such as grades for instance, we can work out average grades by concept and by level.

4.2. Individual follow-up

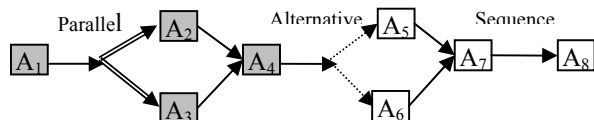
For any student, we know the concepts he must learn as well as his level of knowledge. Using the educational sequence for this course we can determine the educational modules he needs to learn as well as the constraints on them. With this information we can chronologically check the student's progress

The diagrams below refer to the example in figure 2 (educational modules accessed by the student are shown in grey) :



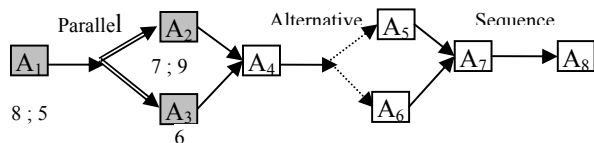
The student has reached A5 without having previously worked on A4.

- example 2 : tracing the student's progress.



The student has followed the constraints of the course. He now must work on A5 or A6.

- example 3 : follow-up, using grades.



The student has successfully followed the course but he doesn't have a sufficient grade average in A1. Nevertheless the student kept on working in A2 and A3 hence an insufficient average mark was obtained for these two educational modules.

4.3. Follow-up of groups of students

Using this method we can classify groups of users who have followed the same browsing pattern. A set of students' categories can be determined by a groups' level. Each group is based on similar student levels. For each group, we can extract the progress in the course as well as the average grade for every educational module. It is also possible to determine, on any level, the set of statistical information per educational module (average grade, number of occurrences, standard deviation...)

5. Conclusion

We believe that our system enables a teacher, having neither an educational platform nor aids for educational tool design, to keep better track of how students are doing in their course(s). We are currently testing simple interfaces with the objective that every teacher can use this approach.

We have already developed a set of building blocks to test our model. The description of both the modular global ontology and the labelled global ontology has

been implemented together with the definition of the categorization tools. Thus, we can determine from a student's learning path his local ontology together with the sequence of simple educational modules studied, and consequently the concepts learned. Also, we can categorize groups of students and obtain a Contextual Local Ontology, from which we can extract the similar concepts studied by the group of students.

Our next task is to focus on the management of complex educational modules with a grade component for each simple educational module. We are currently designing a grading procedure in order to refine these results.

6. References

- [1] W. Gerhard and M. Specht, "User modelling and Adaptative Navigation", Support in WWW-Based Tutoring, User Modelling, proceedings Of the sixth International Conference, UM97, Vienna, *Springer Wien New York*, 1997.
- [2] D. Hérin, M. Sala, P. Pompidor, "Evaluating and revising browsing from web resources educational", ITS 2002, *Springer-Verlag LNCS*, juin 2002, pp 208-218.
- [3] M. Sala, P. Pompidor, D. Hérin, "Aid to the semantic maintenance of the web site", IADIS WWW-Internet 03, Algarbe Portugal, November 03, 9 pages
- [4] <http://www.dublincore.org/documents/education-namespaces/>
- [5] IEEE Learning Technology Standards Committee (LTSSC), Learning Object Metadata (LOM), Draft Document 2001, IEEE P1484.12
- [6] http://www.adlnet.org/Scorm/scorm_index.cfm
- [7] www.w3.org/XML
- [8] P. Baudracco, A-L. Beylot, C. Fleury, S. Monnier, M. Becker, "Performance Measurement of the web server, Design of a first model". Research in official Statistics. Volume 1, number 1, 1998
- [9] H-K. Choi, O. Limb, "A behavioural Model of Web Traffic". International Conference on network Protocole, 1999
- [10] R. Srikant and R. Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements". In Proceedings of the 5th International Conference on Extending Database Technology (EDBT'96), Avignon, France, Septembre 1996, pp 3-17
- [11] R. Agawal, R. Srikant, "Fast Algorithms For Associations Rules"