A Proposed Architecture to Assist a Teacher in Course Planning

Michel Sala, Pierre Pompidor, Danièle Hérin, Gaël Isoird

A proposed architecture to assist a teacher in course planning

Michel Sala, Pierre Pompidor, Danièle Hérin and Gaël Isoird
LIRMM - 161 rue Ada - 34392 Montpellier, France
{sala, Pompidor, dh,isoird}@lirmm.fr

Abstract

In this article, we propose an architecture and an environment that enables a teacher to follow his students during their learning phase (assuming that the teacher doesn't have an educational platform at his disposal). This environment uses the web as a media and allows the teacher to assess his course at the end of the session. We envision that the student's follow-up can be achieved individually or at the level of several homogeneous groups of students. For that purpose, we model the course by using a global ontology (an overall view of the system) with limited educational resources. We then link these resources to the various concepts contained in the global ontology. In order to follow the students' improvements, we categorize and extract from the servers' logs similar groups of students using the pages which the students have accessed.

1 Introduction

In this study, we assume that the teacher doesn't have an educational platform at his disposal, but does have a set of educational resources accessible on the web and available to the students. To enable the instructor to follow his students' improvements, these educational resources are stored on the same server with the logs available to him. In order to individualize the teacher's interaction with the student, a student's profile is developed. Our focus in this paper is on the composition of a course and its assessment, not the student's model nor the users profile [1]. This paper builds on work described in an article which we presented at ITS [2] on the revision and composition of a class course from educational resources on the web. The general principle of our architecture was tested on a problem linked to the maintenance of a web site [3] for the computer editor PC-Soft (www.pcsoft.fr).

In this paper, we present the basic building blocks of our architecture which include: modelling of the educational information, definition of the global ontology, storage of the site's graphical interface and a description of the course. Next we define the interactions of our environment, i.e. the creation of the labelled global ontology and the management of the categorizations. Based on the information and the basic interactions that are defined, we discuss the dynamic aspect, i.e. the extraction of the local ontologies. Using these local ontologies, we explain the mechanisms that allows the teacher to re-evaluate his course. We conclude with a summary of our main contributions.

2 The global architecture

2.1 Structure of the global architecture

The global architecture of our system is represented in figure 1. At the bottom-left part, we represent the basic information, namely :
- the educational module representing the basic educational information,
- the global ontology representing the abstract modeling of the course,

- the site representing the web site with all pages and inter-linked pages.

The assignment of the educational modules in the global ontology, allows us to match the educational modules and the global ontology concepts (Modular Global Ontology MGO). The assignment of the web pages in the global ontology allows us to match the pages and the global ontology concepts (Labeled Global Ontology LGO).

The top-right part of the graph depicts how, based on the server logs, we determine a student's Local Ontology (LO) or a group of students (after categorization) Contextual Ontology (CO). Using LO and CO, we obtain the data which allows us to review the model (educational module and\or global ontology) or the site references.

2.2 Educational information

As the teacher designs his course, he refers to various educational resources extracting data, which constitute the basic building blocks of his course. We call these "educational modules".

2.2.1 Educational resources

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.

Using standards such as the Dublin Core [4], LOM [5] or SCORM [6] we define the educational content. On advantage of using LOM is that a teacher wants to create his own resource (or add metadata on an existing resource) is not constrained because LOM provides many definitions. This allows the designer to create categories at a suitable level of detail.

By using this metadata, 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.

2.2.2 Educational modules

2.2.2.1   The simple educational module

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).
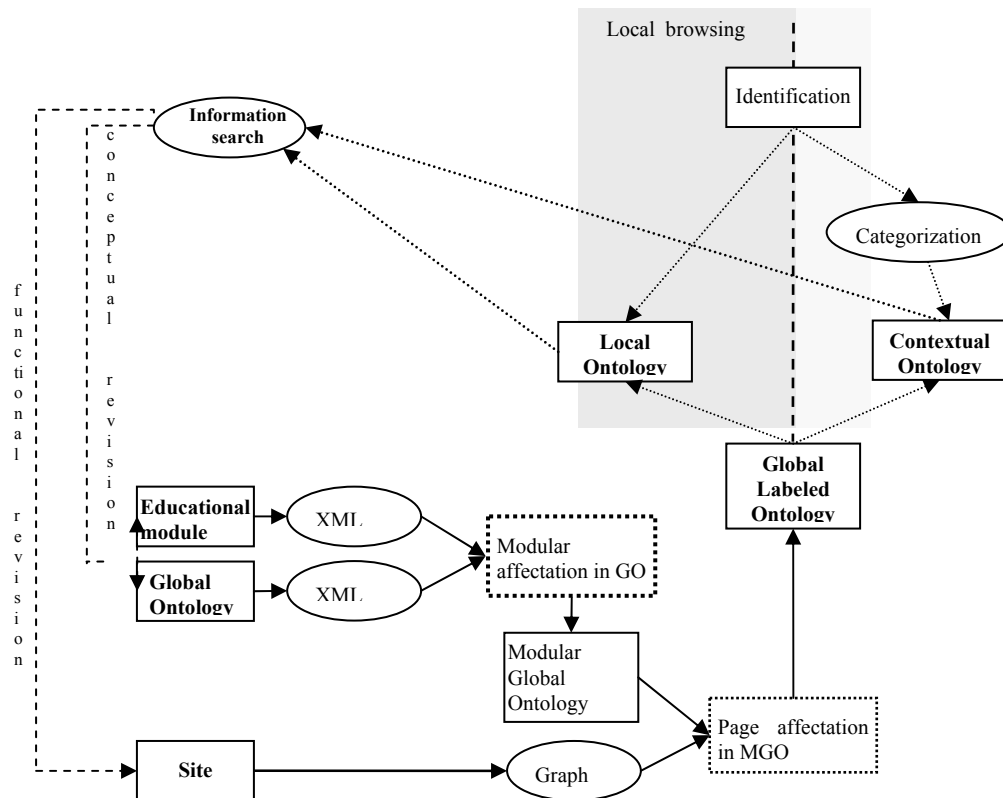
Fig. 1. Global architecture

2.2.2.2 The complex educational module

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: Gi - > Gj the students must learn educational modules Gi and Gj in sequence beginning with Gi and can access Gj only if Gi is mastered.

- Parallel : Gi and (*) Gj the students have to learn both educational modules but without notion of order.

- Alternative: Gi ¦ Gj the students must learn at least one of the two educational modules.

For example, a teacher provides educational modules G1, G2, G3, G4, G5, G6 and decides to order them as represented in figure 2.

Thus, the teacher wants the student to work first on G1 then on G2, then on G3 and G4 (one can work on G3 and G4 in either order) then finish by activating either G5 or G6.

The teacher then creates the complex educational module

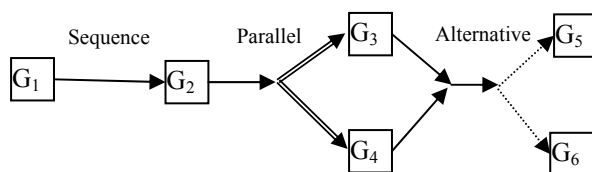GC = (G1-> G2-> (G3*G4) - > (G5|G6))

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

Referring to the example of the previous complex educational module and supposing that these modules are all of levels 1 or 2, we obtain the following structure :
{ ID-module7, {1,2}, consists of [(ID-resource1, section1) - > (ID-resource2, section2) - > ((ID-resource3, section3) and (ID-resource4, section4)) - > ((ID-resource5, section5) ¦ (ID-resource6, section6))]}
The educational module establishes the basic building block of the concept. Figure 3 summarizes the structure of our representation.
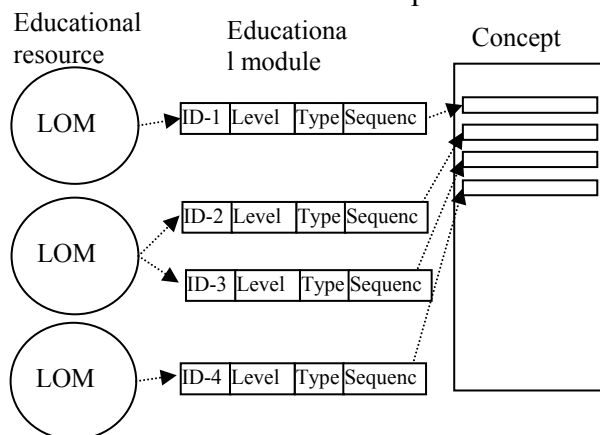

Fig. 3. Design of the educational resource

2.3 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. Among various definitions given in the literature, the most frequently quoted is one from Gruber : "An ontology is an explicit specification of a conceptualization, i.e. a description of a part of the "world" in terms of concepts and relations between these concepts " [7]. Our research has led us to the development of ontologies in various fields :
- the Chimere project [8] uses the ontologies as references for the questioning of several sources of heterogeneous data whose structure is not known a priori,
- to help establish a structure for scientific discovery in experimental sciences [9].

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 = < C, A > 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.

The development of an ontology progresses through the following phases :

     - limiting the domain of interest and the level of abstraction to describe it,

     - defining the vocabulary specific to the domain,

     - modeling the knowledge in terms of a taxonomy of concepts and in terms of the relations between them.

2.4 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. Scheduling

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 Ontology representation in XML

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 [10]. An advantage of XML is it provides a well-structured description of information on a site, contrary to HTML. The XML document shouldn't include any syntactic mistakes. 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.

Languages like DAML (Darpa Agent Markup Language, www.DAML.org) and OIL (Ontology Inference Layer) [11] have been certified by w3c. The OWL project (Ontology Web Language [12]), the fusion of the first two projects, currently is not certified.

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

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.

For example, using the notation defined in section 2.2.2.2., a structure of educational modules for the concept C is defined below as :

id	level	type	sequence
1	1	Simple	Sequence 1
2	1	Complex	Sequence 2
3	1	Simple	Sequence 3
1	2	Simple	Sequence 1
4	2	Simple	Sequence 4
4	3	Simple	Sequence 4
…			

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>
      <module id= «1» levels = «1,2» type= «simple»
      sequence = «sequence 1»>
  </module>
  <module id= «4» levels = «2,3» type= «simple»
      sequence = «sequence 4»>
  </module>
 </level2>
 <level3>
 …
 </level3>
 …
```

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. Based on the study of the students' path through the site (via the server log) and on the knowledge of his level, we can determine

whether a student has actually studied all the educational modules corresponding to his level (cf part 4).

If we consider the example of the "Association" concept of level 1 is used in URL : URL1 and URL2, and that the "Association" concept of level 2 is used in URL : URL1 and URL3, for each concept (ontological graph node), we obtain a structure such as in the following example.

```
<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>
              <module id= «1» levels = «1,2» type= «simple»
              sequence = «sequence 1» >
              </module>
              <module id= «4» levels = «2,3» type= «simple»
              sequence = «sequence 4» >
              </module>
              <URL>URL1</URL>
              <URL>URL3</URL>
   </level2>
   <level3>
…
   </level3>
…
```

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: [13], [14].

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.

The usual procedure for establishing a learning site is to create an identification phase. This enables us to gather information about the student and facilitates tracing his steps, his categorization and more generally speaking his learning. We can choose between two case studies :

        - analysis of the student's individual course,
        - analysis of the collective course (group of students)

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 [15][16]. 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 patters.

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.

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.

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. Results
4.1 Students' follow up
### 4.1.1 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.

4.1.2 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
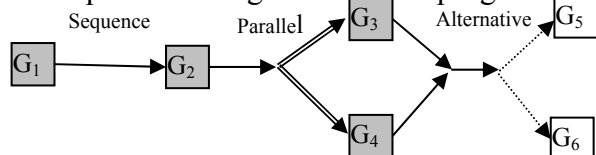
The diagrams below refer to the example in figure 2 (educational modules accessed by the student are shown in grey) :
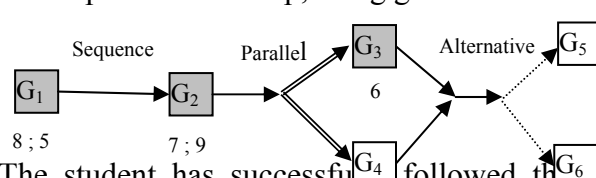
- example 1 : error detection.
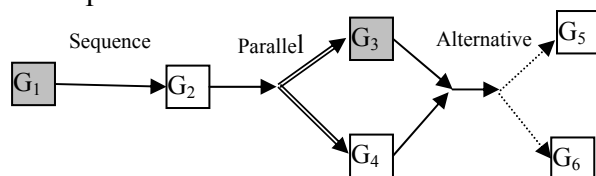
The student has reached G3 without having previously worked on G2.

- example 2 : tracing the student's progress.

The student has followed the constraints of the course. He now must work on G5 or G6.

- example 3 : follow-up, using grades.

The student has successfully followed the course but he doesn't have a sufficient grade average in G1. Nevertheless the student kept on working in G2 and G3 hence an insufficient average mark was obtained for these two educational modules.

4.1.3 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…).

4.2 Revision

Once the individual and group course is completed, the teacher can review it. This reviewing falls into two categories:
- a referential review that highlights the mistakes made in assigning educational modules,
- a conceptual review that enables the teacher to review or re-consider the design of the course.

4.2.1 Referential reviewing

The teacher makes a referential review if he has assigned an educational module to a wrong concept or to the wrong level of a concept. He will realize his mistake if students spend very little time on an educational module (they have spotted some problem between the knowledge searched and that contained in the module) or if the average mark obtained on this module is too high or too low. The teacher can then change the module in relation to the concept or to the module level for the same concept. Thus, we update the Modular Global Ontology (MGO) which updates the Local Global Ontology (LGO).

4.2.2 Conceptual reviewing

The conceptual revision will enable the teacher to determine if the course is achieving his objectives by examining the results obtained by the students. If he notices errors on a simple educational module, he can try to fix the problem. Mistakes could include the following :
- The initial educational prerequisites were not sufficient so the teacher started his course with too many difficult concepts. In this case, he must add some educational modules at the beginning of his course.
- The linking of simple educational modules does not create a good enough learning path for the student. In this case, the teacher needs to review his linking operators (->, &, and |) as well as the sequences between them.
- An incomplete assessment of the student's "presumed" knowledge leads to an inconsistency between the results expected and those obtained.
In the first two cases, the conceptual reviewing will lead to a modification of the educational modules, whereas the last case requires that the instructor more carefully check and assess the students' background before they begin the course.

5 Conclusion

In this study we 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 develop 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.

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.

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, "*Evaluing 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-namespace/

[5]   IEEE Learning Technology Standards Committee (LTSSC), Learning Object Metadate (LOM), Draft Document 2001, IEEE P1484.12

[6]   http://www.adlnet.org/Scorm/scorm_index.cfm

[7]   T. Gruber, "*A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition*", 1993, pp 19-220.

[8]   M-S Segret, P. Pompidor, D. Herin, M. Sala, "*Use of ontologies to integrate some information semi-structured exits of pages web*", INFORSID'2000, Lyon pp 37-55

[9]   M. Sala, P. Pompidor, D. Hérin, "*A framework to review complex experimental knowledge*", OOIS 2002, Springer-Verlag LNCS, septembre 2002, pp 167-172.

[10]   www.w3.org/XML

[11]   M. F. Fernandez, D. Florescu, A. Y. Levy, D. Suciu, "*Web-Site Management: The Strudel Approach. Data*", Engineering Bulletin 21(2), 1998, pp 14-20.

[12]   w3.org/TR/2002/WD-owl-ref-20021112/

[13]   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

[14]   H-K. Choi, O. Limb, "*A behavioural Model of Web Traffic*". International Conference on network Protocole, 1999

[15]   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

[16]   R. Agawal, R. Srikant, "*Fast Algorithms For Associations Rules*".