Usage Analysis Driven by Models in a Pedagogical Context

Sébastien Iksal, Christophe Choquet

Usage Analysis Driven by Models in a Pedagogical Context

Sébastien Iksal, Christophe Choquet
University of Maine – LIUM – CNRS FRE 2730
IUT de Laval, 52 rue des Docteurs Calmette et Guérin, 53020 Laval, France
firstname.lastname@univ-lemans.fr

Abstract. In the context of distance learning and teaching, the re-engineering process needs a feedback on the learners' usage of the learning system. The feedback is given by interviews, questionnaires, but in the majority of systems, it is given by log files. We consider that it is important to interpret tracks in order to compare the designer's intentions with the learners' activities during a session. In this paper, we present the usage tracking language – UTL. This language was designed to be generic and we present an instantiation with IMS-Learning Design, the representation model we chose for our three years of experiments. At the end of the paper, we develop an instance of a scenario for tracks analysis and we present the architecture of services around UTL.

Introduction

Nowadays, numerous interactive systems are available on the Web. Most of these systems need some kind of feedback on the usage in order to improve them. In the specific context of distance learning and teaching, the desynchronization between teachers' two major roles – instructional designer and tutor – brings about a lack of uses feedback. The software development process should explicitly integrate a usage analysis phase, which can provide designers with significant information on their systems' uses for a reengineering purpose [1].
Semantic Web aims at facilitating data management on the Web. It brings languages, standards and corresponding tools that make the sharing and building of automatic and semi-automatic programs easier [2]. Automatic usage analysis is often made by mathematicians or computer engineers. In order to facilitate the appropriation, the comprehension and the interpretation of results by instructional designers, who are the main actors of an e-learning system development process, we think they should be fully integrated in this analysis phase.

The research contribution we present in this paper is fully in line with our approach to the reengineering of e-learning systems, where we particularly stress the need for a formal description of the design view, in terms of scenarios and learning resources, to help the analysis of observed uses (i.e., descriptive scenarios) and to compare them with the designer's intention (i.e., predictive scenario) [3]. When designers use a formal language such as Learning Design [4] proposed by IMS Global Learning Consortium [5] to explicit their intention regarding the learners activities during a session, a set of observation needs are implicitly defined. Thus, one of the student data analysis difficulties resides in the correlation between these needs and the tracking means provided by the educational system, all the more in the case where courses designed will be broadcasted by a Learning Management System (LMS), which usually provides predefined tracking abilities. We propose in this paper a meta-language to describe the track semantics recorded by a LMS and to link them to observation needs defined in the predictive scenario. This meta-language could be instantiated both in the formal language used to describe the pedagogical scenario and in the track file format implemented by the LMS.

The next section of this paper presents this meta-language, called Usage Tracking Language (UTL). In a third part, we provide a use case which highlights the possibilities of this language. Finally, we present an open architecture for usage analysis based on the exploitation of UTL. All the examples cited in this article are taken from a number of tests we have made with our students over the last three years. The first one is composed of six activities designed for teaching network services programming skills. We used the "Free Style Learning" system [6], based on “Open-USS” LMS [7], in which students can navigate as they chose to between all the activities. Our designers have defined a predictive scenario and, each year, we have compared this scenario with descriptive ones by hand, for a reengineering purpose. The second experiment started last year. It aims at students learning the main notions of project management by a collaborative work around a real software development project.

1. A Meta-Language For Usage Analysis

All the systems which need to analyze the user behavior work with data-mining techniques [8] or by hand. These techniques are often used to build user models or to adapt the content or the layout to the user [9]. They are based on statistical or mathematical analyses [10]. In our case, we are interested in analyzing the user behavior in order to improve the pedagogical scenario and the learning materials. Our proposal consists of an analysis driven by models. That is to say, using a model to describe the learning scenario, and using the same model as a guideline to analyze the user behavior inside the Learning Management System. We consider that the result of an analysis will be better used if it has a meaning for the designer of the system and/or the content.

As already mentioned, our activity focuses on the re-engineering driven by models. We consider that each designer has his own representation model for the learning activity. In order to facilitate the comprehension of the analysis, the tools must take into account the designer model and provide the results using the same model. In our experiments, we focus on a standard model of representation: IMS Learning Design. But, in the future we want to refer to a meta-model in which all designer’s models may be described. XML-Schema is an interesting candidate because a number of models are based on this meta-language. We currently have a project on the collaborative design of a model of representation for learning scenario. In this project, we plan to develop a collaborative editor based on XML-Schema. So, one of the goals of this project is to design tools that may work on XML-Schema in order to interpret the designer’s models. Since the beginning of our experiments, we have used IMS-LD to describe learning scenario, and IMS-LD has its own description in XML-Schema.

1.1 Usage Tracking Language : UTL

Even if we are able to process the designer’s model, it is not sufficient for the automation of the tracks analysis. We will also propose a specific language for describing the track semantics according to the designer’s model. This language – called UTL, for Usage Tracking Language – is a meta-language which needs to be instantiated according to (i) the designer’s model and, (ii) the specific format of the logs. Because they have not been designed for this, existing representation models don't include tracking facilities, so UTL is proposed to link tracks and designers’ models through the semantic data. UTL is implemented in RDFS syntax [11]. Figure 1 describes the UTL part concerning the representation. This part of UTL is necessary to interpret some elements of the representation model which are observed. This section has been designed to be as generic as possible, because we want it to be compatible with the majority of designer's models. The term *CONCEPT* refers to all concepts that are defined by the designer, for instance in LD, we can have *Activity, learning object, role, … TRACEABLE CONCEPTS* are concepts from which it is possible to track something, for instance, a video player is a traceable concept from which we can track the start/stop events.

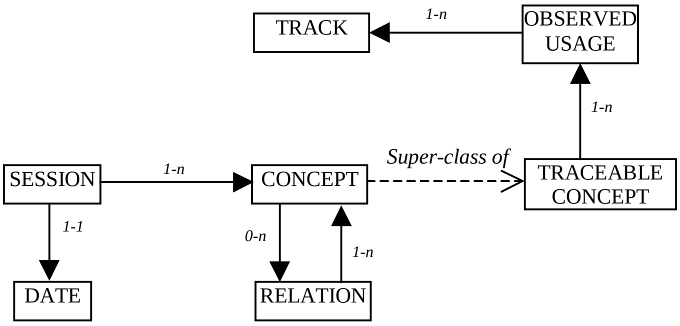

Figure 1: The meta-language UTL – Representation part

In order to work on the track itself, we need to identify it or a part of it. Thus, we have defined another section in UTL : the track representation presented in the Figure 2. The model is also generic, and we propose an implementation that should work with the majority of log formats, as the location of data may be described with a character position and/or with tokens. This section of UTL is useful for retrieving specific tracks, extracting values and bringing sense to each of them. The *KEYWORD* is used to retrieve the track, it is a word which is always present in the track. The *VALUE* depends on the learner, it may be the time spent to read, the name of the page read or the score of the evaluation exercise.

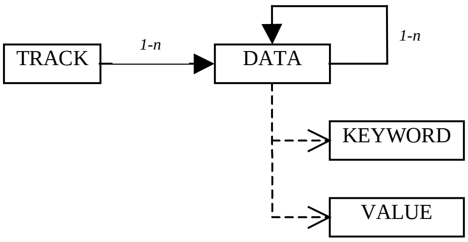

Figure 2: The meta-language UTL – Tracks part

The specific attributes for the specification of the data locations are the following :

- Type : Is used to type the data – to associate semantics.
- Begin : Gives the first character position of the data.
- End : Gives the last character position of the data.
- Delimiter : Sets the delimiter used to break down the track into tokens.
- Position : Gives the position of the token.

The data locations are used to specify the position inside the track of the keyword or the value.If we consider a prescribed scenario in IMS-LD and tracks generated by FSL, Figure 3 is an example of the instantiation of UTL. In this case, a session is identified by the student identifier, because for one session we have a set of log files which corresponds to the work of a single student. First, we describe some data that can be extracted according to these two models. In the following example, we describe a track which represent the end of a video player done by the learner.

```
<? xml version="1.0" encoding="iso-8859-1" ?>
<TRACKING
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="fslTrack.xsd"
   session="SRC1E06" date="18/12/2002">
  <ACTIVITY name="View Objectives">
    <USING>
      <LEARNINGOBJECT name="Introduction"
```

```
                    type="VideoPlayer">
         <OBSERVEDUSAGE name="functioning">
           <TRACK type="stop player">
             <VALUE type="date" begin="1" end="25"/>
             <KEYWORD type="component" begin="38"
                      end="52">FreeVideoPlayer</DATA>
             <KEYWORD type="command" begin="55"
                      end="58">stop</DATA>
             <VALUE type="duration" begin="74"
                    end="end"/>
           </TRACK>
           ...
```

Figure 3: Example of track description

This description has been used to filter the log file and to extract the following track :

```
[18/12/2002:09:45:29 +0043]           [FreeVideoPlayer] stop() currentTime=182.0s
```

We have also obtain the following data :

```
Date of the track : 18/12/2002:09:45:29  Duration of the video : 182.0s
```

In this example, we worked with a single student. In other experiments, we may have to track the activity of a group (especially in collaborative work). UTL is able to describe tracks if we have a single log file for all members – server log file –, and also if we collect a set of log files, one per member – client log file. We just have to define in the designer model the concept of “group” and “member of group”.

1.2 Instantiation of UTL in IMS-LD

In our experiments, we have used IMS-LD as a representation model for the designer. In order to manage tracks according to this language, we have instantiated a part of IMS-LD in UTL (See Figure 4).

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
         xmlns:utl="utl.rdfs">
  <rdfs:Class rdf:ID='Activity'>
    <rdfs:subClassOf rdf:resource='&utl;TraceableConcept' />
  </rdfs:Class>
  <rdfs:Class rdf:ID='Role'>
    <rdfs:subClassOf rdf:resource='&utl;TraceableConcept' />
  </rdfs:Class>
  <rdfs:Class rdf:ID='Environment'>
    <rdfs:subClassOf rdf:resource='&utl;Concept' />
  </rdfs:Class>
  <rdfs:Class rdf:ID='LearningObject'>
    <rdfs:subClassOf rdf:resource='&utl;TraceableConcept' />
    <rdfs:subClassOf rdf:resource='#Environment' />
  </rdfs:Class>
  ...
  <rdf:Property rdf:ID='performs'>
    <rdfs:subPropertyOf rdf:resource='&utl;Relation' />
    <rdfs:domain rdf:resource='#Role'/>
    <rdfs:range  rdf:resource='#Activity'/>
  </rdf:Property>
  <rdf:Property rdf:ID='using'>
    <rdfs:subPropertyOf rdf:resource='&utl;Relation' />
    <rdfs:domain rdf:resource='#Activity'/>
    <rdfs:range  rdf:resource='#Environment'/>
  </rdf:Property>
</rdf:RDF>
```

Figure 4: UTL file for IMS-LD

2. Scenario of Tracks analysis

Our first need on usage analysis is about track analysis. We have three years of logs on two different experiments. For each of these case studies, we have a prescribed scenario described in IMS Learning Design. We use our Usage tracking language to bring semantics to each track. The first step consists in the interpretation of tracks according to the designer model and the corresponding track semantic description. Next, the observed usage of the learning system are available for the analysis.

2.1 Tracks Interpretation

At the beginning, automatic track analysis needs an automatic interpretation of these tracks. UTL is designed to add semantics to the content of the log files. We use it to filter the content of the log files, that is to say, to keep only tracks that are considered relevant by the designer. A track is relevant if a description is given inside the UTL file. The second use of UTL consists of associating a specific type to each track and in extracting values that are representative of the learner's activity.

The result of this stage is a data structure which contains the interpretable tracks and which is shareable between all services of our architecture. The data structure is available also for each researcher who wishes to propose new services.

2.2 Usage Analysis

There are various ways to use the observed usage-interpreted tracks. Our first service retrieves patterns to find the resource usage, or to compare a learner scenario with the predictive scenario. Next, by means of the semantic description of tracks, it is possible to define services in a declarative way.

Examples of analysis results are the following: rate of use of a resource, performance of a student, emergence of a role (leader, …), extraction of an observed learning scenario, detection of a sequence of resource uses which have not been prescribed…

To present the usage analysis, we will focus on three cases : A statistical data, a result which has to be re-transcribed in the designer's model, and an intelligent information detection.

A statistical data. These data are, for instance, the rate of use of a resource, the average mark concerning the evaluation exercise, or the time spent on a particular activity (the shortest, the average, the longest). We have to filter the tracks according to their semantics and to make a small calculation on them. As an example, for the data (a) a first solution is to count students for whom we find at least one track about the use of the resource. In our experiments, we have observed that sometimes we have tracks about the use of the resource, but the student has spent less than 15s on the resource because he clicked everywhere during an exploration period. The solution adopted was to detect the duration of each period of use, and to count students who have spent a minimum of 15s on at least one period.

Retranscription in the designer's model. One of the main goals of the re-engineering driven by models is to use the same representation model for the description of the predictive scenario by the designer as for the observed scenario build with tracks generated by the learning system. In our first experiments, we worked with IMS-LD as a representation model. The interest in the use of a common model is the possible comparison between the different scenarios, that leads us to identify non-predicted usages of resources or incoherences in the sequence of activities. In one of our experiments (the one based on FSL), we observed that some students have used the evaluation exercise as a quiz at the beginning of the experiment, they just have navigated inside the list of questions in order to self-evaluate their knowledge (before the first activity of the learning session). That observation leads us to propose two facets on our exercise, one for evaluation and another for a quiz. We consider two kinds of retranscription of the observed scenario: the one generated from a single student tracks, and a stereotypical scenario that represent a combination of all student scenarios.

(a) Retranscription of one student observed scenario.
First, we have to read the representation model in order to identify the core concept, such as the activity for IMS-LD. Next, we filter tracks in order to represent this concept and all its components. The last step consists in organizing all instances of the core concept in a sequence which corresponds to the observed scenario.

(b) Retranscription of a stereotypical observed scenario.
A stereotypical observed scenario corresponds to the combination of all student scenarios. To build this scenario, we must have all the students' observed scenarios. Next, we compare the sequence of core elements (for instance, activities), and we compare in depth each element. We observe the percentages about the use or the position in the sequence of each element. A stereotypical scenario is a graph where each relation is qualified with the percentage of students which have chosen the corresponding direction.

3. An Open Architecture for Usage Analysis

We have observed that many researchers are interested in a collection of tools which may assist them in the "semantic" analysis of the learner's learning session. We propose an architecture that may create a kind of practice community around the usage analysis. Our approach is close to that of Web Services, that is to say each server has to declare itself to the system and it provides a set of services. Servers may be deployed everywhere. But in our case, a service is a collection of methods around a specific concept or domain. For instance a service may be the management of log files, and we may have one method per log format. The other major difference consists in the use of methods, all methods are available from all the servers, because with our architecture based on RMI, we don't have to know where the method is executed, we just need to know its head (name, parameter, output). We execute all methods from one server. In our approach, we share also a common data structure between each service to facilitate the sharing of complex data.

3.1 The Architecture of Services

The most important feature of the architecture is that each researcher in usage analysis must be able to add new services in order to share them. That is why we have chosen to propose an architecture based on the Java technology and the RMI functionality. The architecture is presented in the figure 5. It is a cluster of servers around a special one which is called the *Router*. A server provides various services and registers itself at the router. A service is a set of methods that may be executed in order to request or modify something. For instance, we can have a service for the importation of log files with one method per log format, or a service about the analysis of chat discussion. This architecture is open and distributed; that is to say, it is possible to plug new servers from everywhere in the cluster. The community of researchers may use this architecture for analyzing users' behavior, but also may propose new methods in order to have feedback on the use of these methods on various data repositories. The user just has to connect to one server and to ask for a service, he doesn't have to know which server is concerned nor where the process is run. He can simply use the services available in the architecture.

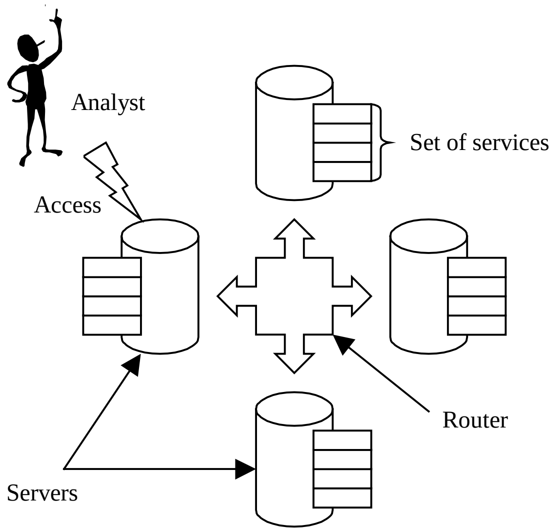

Figure 5: The architecture

3.2 Integration of Other Services

The distributed aspect of our architecture enables us to have numerous usage analysis services. We consider a data structure specific for each kind of data – tracks, chat, interviews, etc. For the moment, we are working essentially on track analysis. We choose to have some data structures instead of a database, in order to keep researchers free to use their own data management systems (databases, knowledge based systems, …) for their services.

4. Conclusion and perspectives

The meta-language presented in this paper is well suited for defining what the system has to track, based on the predictive scenario designed for a learning activity. For each traceable concept present in his scenario, the designer could define what to track (e.g. the tracking means) and why it should tracked (e.g. the semantics of the track chunk). Because of its meta-level, UTL could be also used, after usage analysis, to define and highlight semantic links between predictive and descriptive scenarios, via the association between traceable concepts and observed uses.

Works such as [12] have shown that teachers and trainers – who are the main potential designers of educational systems – have some difficulties in instructional design, especially regarding the explicitation and the technical reification of their pedagogical intentions. We are defining rules which can be inferred on the meta-model (e.g. the XML schema) of the instructional language used by a designer (for instance, Learning Design) in order to identify opportunities and observation possibilities. They reason on the structure of the instructional language (data-type, relations, etc.) and provide to the designer information on the needs of observation. These needs are relative to the concepts of the language and thus, define the traceable concepts. Using these rules with UTL could be a way to provide designers with a semi-automatic tool for decision helping purposes. Our approach of student data capture is focused on automatic techniques driven by designer prescriptions. UTL is presently without the spectrum of both existing non-automatic techniques, such as interviews for instance, and data-mining or machine learning ones. We think all these techniques, including ours, are complementary. One of our research objectives is to enlarge the spectrum and the abilities of UTL, in order to take into account results established with these other techniques. Non-automatic data capture methods are usually based on interviews and questionnaires deployed during and after the session. Questions asked to students and / or tutors are defined regarding

(i) the learning objectives and the activities proposed (e.g. the designer's intention) and (ii) the characteristics of the session (e.g. the social and technical context). All of these are known (or assumed) when the designer defines the predictive scenario. Concerning this aspect, we have started a study with researchers specialized in usage analysis (Communication Science background) of which the objective is to define when, why and how a designer has to explicit the requirements to these techniques.

References

[1] Corbière A., Choquet C., Re-engineering method for multimedia system in education, In: IEEE Sixth International Symposium on Multimedia Software Engineering (MSE), 2004, Miami (USA), p. 80-87.

[2] Berners-lee, T.. Weaving the Web. San Francisco: Harper, 1999.

[3] Lejeune A., Pernin J-P., A taxonomy for scenario-based engineering, Cognition and Exploratory Learning in Digital Age (CELDA 2004) Proceedings, p.249-256, Lisboa, Portugal, dec. 2004

[4] Koper R., Olivier B., Anderson T. eds., « IMS Learning Design Information Model », IMS Global Learning Consortium, Inc., version 1.0, 20/01/2003.

[5] http://www.imsglobal.org

[6] Brocke, J. v.. Freestyle Learning - Concept, Platforms, and Applications for Individual Learning Scenarios. 46th International Scientific Colloquium, Ilmenau Technical University, 2001.

[7] Grob, H. L., F. Bensberg, et al.. Developing, Deploying, Using and Evaluating an Open Source Learning Management System. Journal of Computing and Information Technology 12 no 2: 127-134, 2004.

[8] Mostow, J.. Some useful design tactics for mining ITS data. Proceedings of the ITS2004 Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes, Maceió, Alagoas, Brazil, 2004.

[9] Zheng C., Fan L., Huan L., Yin L., Wei-Ying M., Liu W.. User Intention Modeling in Web Applications Using Data Mining. World Wide Web: Internet and Web Information Systems, 5, 181–191, Kluwer Academic Publishers, 2002.

[10] Bazsaliscza M., Naim P. . Data Mining pour le Web, Editions Eyrolles, Paris, 2001.

[11] Brickley D., Guha R.V.. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004

[12] Seel N., Dijkstra S.. General Introduction. In Dijkstra, S., Seel, N., Schott, F. & Tennyson, R.D. (Eds.). Instructional Design : International Perspectives. (vol. 2). Hillsdale, NJ, Lawrence Erlbaum Associates, p. 1-13, 1997.