USAGE TRACKING LANGUAGE: A META LANGUAGE FOR MODELLING TRACKS IN TEL SYSTEMS

Christophe Choquet, Sébastien Iksal
LIUM Laboratory, University of Maine, IUT de Laval, 52 rue des Docteurs Calmette et Guérin, 53020 Laval, France

Keywords: Track modelling, Technology enhanced learning, Model driven reengineering.

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 numerous vectors, such as interviews, questionnaires, videos or 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 is designed to be generic and we present an instantiation of a part of it with IMS-Learning Design, the representation model we chose for our three years of experiments.

1 INTRODUCTION

Nowadays, most of the Web interactive systems need some kind of feedback on the usage in order to improve them. In the context of distance learning, the desynchronization between teachers' 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 (Corbière, & Choquet, 2004). 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, we think they should be integrated in the analysis.

Our research contribution is fully in line with our approach to the engineering and reengineering of e-learning systems, where we particularly stress the need for a formal description of the design view, to help the analysis of observed uses and to compare them with the designer's intention (i.e., predictive scenario) (Lejeune, & Pernin, 2004), in order to enhance the quality of the learning. When designers use an Educational Modeling Language (EML) such as Learning Design (Koper, Olivier, & Anderson, 2003) proposed by IMS, 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 environment.

Our aim is to provide the actors of a Technology Enhanced Learning (TEL) System with a language dedicated to the description of the tracks and their semantics, including the definition of the needs and the acquisition's means. Our Usage Tracking Language (UTL) aims to be neutral regarding technologies, systems and EMLs. Moreover, it allows the structuring of tracks, from raw data – those acquired and provided by the educational environment during the learning session – to indicators (ICALTS, 2004) which mean something significant for its user. They usually denote a significant fact or event that happened during the learning session, on which users (e.g. designers, tutors) could base some conclusions concerning the quality of the learning, the interaction or the learning environment itself.

In the next section, we present the conceptual model of UTL and its information models. The third section illustrates how one could make an instantiation of this meta-language on both the EML used for modelling the pedagogical scenario of the learning system – here, IMS LD, and the track formats used by the TEL system – here, "Free Style Learning" system (Brocke, 2001). We conclude this part with a use case of UTL with a three years experimentation. It concerns a learning system which is composed of six activities designed for teaching network services programming skills.

Choquet C. and Iksal S. (2006).
USAGE TRACKING LANGUAGE: A META LANGUAGE FOR MODELLING TRACKS IN TEL SYSTEMS.
In *Proceedings of the First International Conference on Software and Data Technologies*, pages 133-138
DOI: 10.5220/0001312701330138

2 THE UTL MODEL

2.1 Track Conceptual Model

Some recent European works focus on the tracking problematic (i.e. track representation, acquisition and analysis). Most of these works, such as DPULS, ICALTS, IA, and TRAILS projects have taken place in the Kaleidoscope European Network of Excellence (Kaleidoscope, 2004) and each of these projects have influenced our proposal. We have identified two main data types for tracks: the *derived-datum* type and the *primary-datum* type.
The primary data are not calculated or elaborated with the help of other data or knowledge. They could be recorded before, during or after the learning session (e.g. a log file, a questionnaire). This kind of data is a *raw-datum*. The *content-datum* type concerns the outcomes provided by the learning session actors (e.g. productions of the learners, a tutor report). Both of these data have to be identified in the collection of tracks provided by the learning environment, in terms of location and format. We introduce here the *keyword* and the *value* elements for this purpose. These elements will be discussed further in the paper. The *additional-datum* type qualifies a datum which is linked to the learning situation and could be involved in the usage analysis.

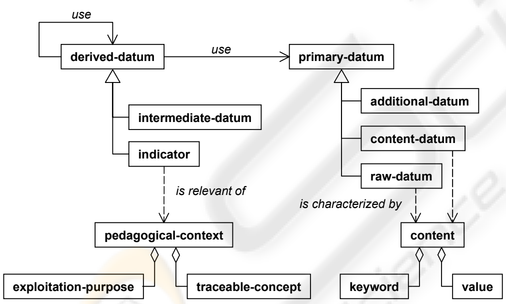

Figure 1: The conceptual model of UTL.

The derived data are calculated or inferred from primary data or other derived data. The *indicator* type qualifies derived data which have a pedagogical significance. Thus, an indicator is always relevant to a *pedagogical context*: it is always defined for, at least, one *exploitation purpose*, and linked to, at least, one concept of the scenario. We will detail this specific aspect further in the paper. A derived datum which has to be calculated but which has no pedagogical significance is an *intermediate-datum*.

We will now detail the information model of each data types. The formalism used is the IMS LD Information Model (IMS/LD, 2003) notation.

- The diagrams are tree structures, to be read from left to right. An element on the left contains the elements on the right side.
- < is an OR relationship.
- [is an AND relationship.
- *: the element occurs zero or more times.
- +: the element occurs one or more times.
- ?: the element is optional.
- No symbol: the element occurs one time.

Each data type has three facets (Defining, Getting, Using) which allow two processes for modelling a datum: the predicted one, when the designers, during the design phase, declare the datum as needed, and the unpredicted one, when the datum is collected or calculated without an explicit designer's request. In the first process, the Defining and the Using facets are filled first; then the Getting facet is discussed with developers. This is the way one could provide, for instance, examples and descriptions, rather than a specific technique or tool. In the second process, developers and/or analysts fill the Getting facet first, then the Using and Defining facets are discussed with designers.

2.2 The Raw-datum Information Model

Defining is composed by the *Title* of the datum and a *Description* could be added. *Getting* focuses on the mean for acquiring the datum. It is composed by the *Collection-type* element which could be a *Human-collection*, operated by at least one *Role* (e.g. an observer), with a specific *Collection-vector* (e.g. a video recorder), or an *Automatic-collection*. This kind of collection is characterised by the nature (e.g. log file) of the collection – the *Record-type* and the *Record-tool*. If this tool is already available in the learning environment, one could provide its *Location*; if not, one could provide the developers with a *Description* and/or some known *Examples*. *Getting* is also composed by the *Location* of the datum (e.g. URL of the file), and by the *Acquisition-time* of the datum ('Before-session', 'During-session', 'After-session'). *Using* is composed by two elements: the *Used-by* one, which exists only for commodity about the data dependencies, and the *Content* one which allows the retrieving of the datum from its source. The category of a datum’s content could be *Keyword* or *Value*. These generic concepts allow the description of multiple tracks formats from text files to databases and videos (see 2.8). This *Content* element constitutes the part of the meta-language which could be instantiated on the tracks of a specific learning environment.

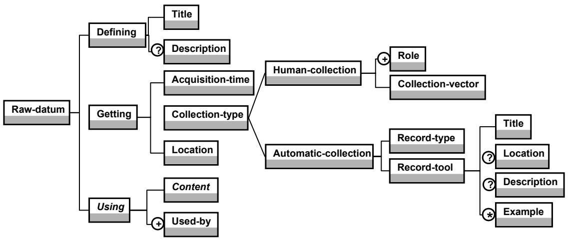

Figure 2: The raw-datum information model.

2.3 The Content-datum Information Model

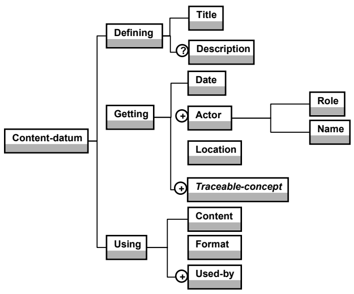

Figure 3: The content-datum information model.

As for raw data, the *Defining* is composed of a *Title* and a possible *Description*. Content data are the outcomes of a learning session. Thus, they are always well-identified. *The Getting* is then characterised by its *Location*, the *Date* of the production and the *Actor* who has produced this datum. It is also characterised by at least one *Traceable-concept* of the scenario. *Traceable-concept* constitutes the part of the meta-language which could be instantiated on a specific EML (see 2.7). The *Using* facet is composed of the *Content* of the datum, its *Format* and, as for raw-data, the list of the data which use it.

2.4 The Additional-datum Information Model

Additional data are multiples (e.g. predictive scenario, ontology); thus, the *Defining* adds the *Type* of the datum to its *Title* and its *Description*. An additional datum is well known and identified: the *Getting* refers only to its *Location*. The *Using* facet is composed of the *Content* of the datum, its *Format* and the list of the data which use it.

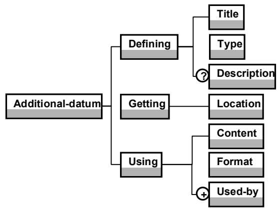

Figure 4: The additional-datum information model.

2.5 The Intermediate-datum Information Model

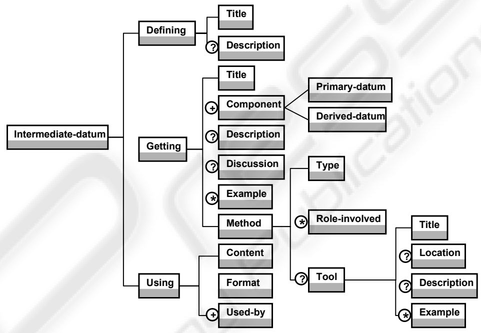

Figure 5: The intermediate-datum information model.

The *Defining* is composed of a *Title* and a possible *Description*. The *Getting* characterises the mean for establishing the datum. It is mainly composed by the *Components* element, which allows the definition of the graph of dependencies of the datum, which is always defined with the use of primary data and/or derived data, and by the *Method* element. The getting method *Type* could be 'Manual', 'Semi-automatic' or 'Automatic'. If a human intervention is required, one should define it with the help of the *Role-involved* element. If the method is semi-automatic or automatic, the support *Tool* has to be defined by its *Location*, if available, or by a *Description* and some *Examples*. We assume here that only one tool could be specified for an intermediate datum. If more than one are needed, several intermediate data have to be defined. The *Using* facet is composed with the *Content* of the datum, its *Format* and the data list which use it.

2.6 The Indicator Information Model

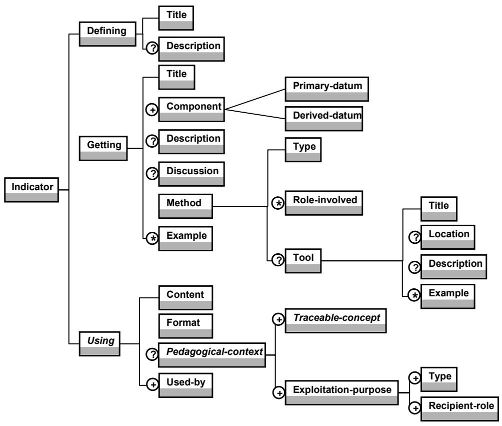

Figure 6: The indicator information model.

The *Defining* and *Getting* facets are similar to the Intermediate-datum facets. The *Using* facet is characterised by a *Pedagogical-context* element which defines the context of use and the purpose of the Indicator. This context is described by a *Traceable-concept*, as the content data, and by an *Exploitation-purpose*, performed by at least a *Recipient-role*. We have currently defined four Types for this exploitation – reengineering, regulating, assessing, reflecting – but we consider this Type element as an open list.

2.7 The Traceable-concept Information Model

This part of UTL is used to classify all concepts of the representation model used to express the pedagogical scenario that are traceable. This section has been designed to be as generic as possible, in order to be compatible with the majority of designer's models. A *Traceable-concept* is a concept of the designer's model from which it is possible to track something (e.g. an activity with its beginning, end and duration). The description of the *Traceable-concept* is composed of all relationships with other Traceable-concepts (e.g. an activity realised a resource). The *Title* of the relationship brings more semantic to the interpretation of tracks' context. This concept is for instance, in the context of an EML scenario modelled with IMS LD, an activity. But it could also be an Enterprise concept which is domain specific and could not be reified with an EML. So the *Type* attribute refers to these two values: Enterprise and Abstract-scenario. The *Observed-use* allows the description of the relationship between tracks and the traceable concept.

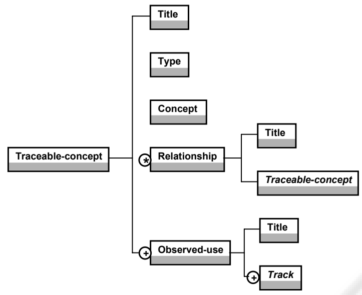

Figure 7: The traceable-concept information model.

2.8 The Track Information Model

In order to work on the track itself, we need to identify it or a part of it. Thus, we have defined the *Track* information model. This model is also generic, and we propose an implementation that could work with the majority of track formats, but we have only experimented it on log files (See 3).

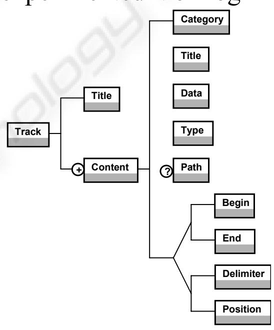

Figure 8: The track information model.

To manage each format, we have defined the *Type* field which takes values in the open list (e.g. Text, XML, Database), and an optional *Path* which contains the path to the specific data to describe (XPath, SQL). For describing the location of data inside a string, we propose the use of character positions and/or tokens. We consider two categories of content in tracks. "Keyword" is used to retrieve the track, it is a word (or a sentence) which is always present in the same kind of track. And "Value" depends on the learner, it may be the time spent to read or the name of the page read. The *Content* locations are used to specify the position inside the track of the keyword or the value. The specific attributes for the specification of the Content locations are the following: *Title* is used to name the content – to associate semantics; *Begin* gives the first character position of the content; *End* gives the last character position of the content (-1 for the end of the line); *Delimiter* sets the delimiter used to break down the track into tokens; *Position* gives the position of the token. The *Data* field is used to store the value or to indicate the keyword.

3 EXAMPLE OF INSTANTIATION

3.1 Instantiation on the Learning Design Model

We have used IMS-LD as a representation model for the designer. In order to manage tracks according to this language, the following piece of code is an extract of the instantiation concerning Activity, Role and Resource.

```
     <xsd:element name="Activity"
type="TraceableConceptType"
substitutionGroup="TraceableConcept"/>
    <xsd:element name="Role"
type="TraceableConceptType"
substitutionGroup="TraceableConcept"/>
    <xsd:element name="Resource"
type="TraceableConceptType"
substitutionGroup="TraceableConcept"/>
```

The next stage consists in instantiating the UTL-LD file with a specific scenario which we decided to analyse. This step is necessary to associate semantic to tracks, that is to say to link each track with the relevant object of the learning scenario. The following piece of code represents an excerpt of the relationships between all activities and resources of our experiment.

```
     <Activity Title="Discovering the
system" Type=”Abstract-scenario”>
       <Relationship Title="Use"
Concept="TextStudy"/>
       <Relationship Title="Use"
Concept="SlideShow"/>
       <Relationship Title="Use"
Concept="CaseStudy"/>
       <Relationship Title="Use"
Concept="LearningByDoing"/>
   </Activity>
```

3.2 Instantiation of UTL in FSL Log Format

Once the scenario's data are prepared, the tracks' format has to be described according to the deployment platform. The next piece of code is an excerpt of representation concerning the *VideoIntro* resource which is an introduction of the course. We describe here keywords that are necessary to identify the track, for instance "Intro gestartet" for the beginning of the video, and also values that have to be extracted, for instance the date of the track.

```
<Resource Title="VideoIntro">
       <ObservedUse Title="Managing">
           <Track Title="Start">
               <Content
Category="Value" Title="Date"
Type="Text" Begin="1" End="26"/>
               <Content
Category="Keyword" Title="Task"
Type="Text"  Begin="33"End="40">
FreeApp</Content>
               <Content
Category="Keyword" Title="Object"
Type="Text"  Begin="42" End="57">Intro
gestartet</Content>
           </Track>
…
       </ObservedUse>
</Resource>
```

3.3 A Use Case

We propose a use case where interpreted tracks are used to compute indicators. It is based on the "Playing Around with Learning Resources" design pattern, taken in the DPULS Project (DPULS, 2005). This pattern provides an approach to detect learner playing around at the beginning of an activity. Its solution is based on two indicators: "The characterisation of the sequence of resources" and "The characterisation of the time of an activity". The first one defines the sequence of resources attempted by a learner as "non significant" if the duration of each resource is less than a fraction (here 10%) of the Typical Learning Time defined for the relative resource. The second one defines the time of an activity as "the beginning" if the effective duration of the activity is less than a fraction (here 10%) of the Typical Learning Time of an activity. UTL is able to identify and extract the raw data. It allows the formalisation of the indicators' generation for the pedagogical designer. The additional data such as the typical learning time can be extracted from the prescribed scenario, for instance the field 5.9 of the (LOM, 2002). This is a percentage of the use time of a resource considered as a minimum time. We present the description of these data with UTL in Table 1, which presents the information table for the raw-datum called “Started time of a resource”.

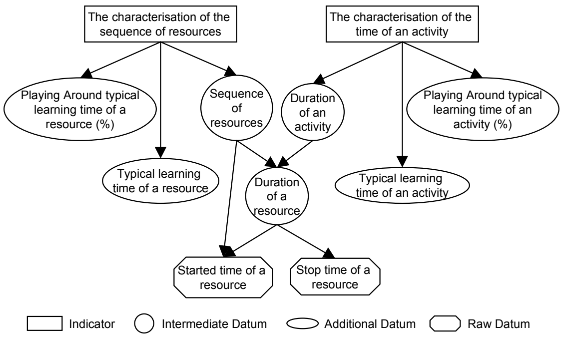

Figure 9: Maps of indicators and data used.

Table 1: Information table for a raw-datum.

D	Title	Started time of the video intro
	Description	These datum stores the time of the beginning of a video’s use.
G	Acquisition-time	During-session
	Record-type	Log-file
	Record-tool.Title	FSL methods for the generation of tracks
	Record-tool.Location	~*exp*/*StudentID*/*file*.FSL
U	Content.Keyword	-“FreeApp” from char. 33 to 40 -*“intro* gestartet” from char. 42 to 57
	Content.Value	Date from character in position 1 to 26
	Used-by	“Sequence of resources”

4 CONCLUSION

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 of his scenario, the designer could define what to track, why it should tracked, and how structuring the tracks by defining indicators and intermediate data with appropriate tools and methods. Each data can be combined with others in order to provide high level indicators for the analyst or the designer. (Seel, & Dijkstra, 1997) have shown that teachers and trainers 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 of the instructional language used by a designer in order to identify opportunities and observation possibilities (Barré, & Choquet, 2005). They reason on the structure of the instructional language and provide the designer with information on the observation‘s needs. 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.

REFERENCES

Barré, V., Choquet, C., 2005. Language Independent Rules for Suggesting and Formalising Observed Uses in a Pedagogical Reengineering Context. In *ICALT'05, IEEE International Conference on Advanced Learning Technologies* (p. 550-554). IEEE Press.

Brocke, J. V., 2001. Freestyle Learning - Concept, Platforms, and Applications for Individual Learning Scenarios. In *46th International Scientific Colloquium*. Ilmenau Technical University Press.

Corbière, A., Choquet, C., 2004. Re-engineering Method for Multimedia System in Education. In *MSE'04, IEEE Sixth International Symposium on Multimedia Software Engineering* (p. 80-87). IEEE Press.

DPULS, 2005. *Design Patterns for Recording and Analysing Usage of Learning Systems*. Consulted May, 2006, at http://www.noe-kaleidoscope.org

IMS/LD, 2003. *IMS Learning Design*. Consulted May, 2006, at http://www.imsglobal.org/learningdesign/index.html

Kaleidoscope, 2004. Consulted May, 2006, at http://www.noe-kaleidoscope.org

Koper, R., Olivier, B., Anderson, T.,2003. *IMS Learning Design Information Model (version 1.0)*. IMS Global Learning Consortium Press.

Lejeune, A., Pernin, J-P., 2004. A Taxonomy for Scenario-based Engineering. In *CELDA'04, Cognition and Exploratory Learning in Digital Age* (p.249-256). IADIS Press.

LOM, 2002. *Draft Standard for Learning Object Metadata*. IEEE Press.

Seel, N., Dijkstra, S., 1997. General Introduction. In *Instructional Design : International Perspectives. (vol. 2)* (p. 1-13). Hillsdale, NJ, Lawrence Erlbaum Associates Press.