UTL-CL : A Declarative Calculation Language Proposal for a Learning Tracks Analysis Process

Diem Pham Thi Ngoc¹, Sébastien Iksal¹, Christophe Choquet¹, Evelyne Klinger²
¹University of Maine – LIUM
IUT de Laval – Dépt. SRC
52 rue des Docteurs Calmette et Guérin
53020 LAVAL - FRANCE
FirstName.LastName@univ-lemans.fr
²Arts et Metiers ParisTech- LAMPA
Equipe Presence et Innovation – Entité HIT
4 rue de l'Ermitage
53000 LAVAL - FRANCE
evelyne.klinger@angers.ensam.fr

Abstract

In this paper, we propose a track analysis process about the activity in learning systems that helps teachers to benefit of the experience and the knowledge of the community in the usage analysis. We also propose a language for calculation of teachers' observation needs. This language will be implemented in tracks' analysis tools, which support teachers to establish their tracking needs in order to improve the pedagogical scenario that they designed.

1. Introduction

One of the actual bottlenecks of the Technology Enhanced Learning (TEL) systems engineering is the support of tracks analysis for three purposes: (1) adaptivity, e.g. allowing the system to adapt itself dynamically, (2) adaptability, e.g. allowing the learners and the tutors to adapt the system during a learning session, and (3) reengineering, e.g. allowing the designers to modify the system after a learning session. With this in mind, the REDiM (model driven re-engineering) project adopts a model driven engineering (MDE) approach for the instructional design where the predictive pedagogical scenario is composed by two woven models: the formal description of the sequence of the activities that learners and tutors have to perform and the modeling of the observation needs, in terms of significant indicators which are able to help to understand the effective activities performed during a learning session. In this framework, REDiM has focused on the design and the reengineering of such predictive scenarios: how a designer could be helped (1) for modeling a predictive scenario, including indicators, and (2) for analyzing these indicators after a learning session, in order to modify the predictive scenario for the next session. This is the way we have proposed the Usage Tracking Language (UTL) [1], a meta-language dedicated to the modeling of indicators, woven with the activities modeling, inside the predictive pedagogical scenario. Each indicator is formally described as the result of several transformations, combinations and abstractions of raw data collected during the learning session. The purpose of this paper is to present a part of this language, called UTL-CL (UTL Calculation Language), our proposal to formalize the method of calculation of an indicator from raw data. This proposal is a step towards the definition of Design Patterns for indicators, where the problem is how to get an indicator from raw data and where the solution is the relative calculation method.

The next section of this paper shapes the context of this research. The third section defines the specific tracks analysis process we propose. The fourth section is focused on the description of UTL-CL with
an example. We conclude this paper with some immediate perspectives in terms of implementation.

2. Context of research

Results of the track analysis of a learning system are often used by teachers to anticipate the evaluation, the modification, and the improvement of the teaching and learning activities. For example, tracks' analysis can help teachers to better understand learners' activities [2], in order to improve the presentation of concepts and practical exercises [3] or to improve the teaching [4]. However, these results are not motivated from observation needs explicitly expressed by teachers. In the majority of the existing systems (for example, [3] [5] [6]) that take into account the usage analysis of learning systems, teachers are slightly or not solicited to define their observation needs. Furthermore, the competence and the knowledge of teachers in the analysis and the study of a learning session should be shared and reused in the community. The DPULS Project (Design Patterns for collecting and analyzing Usage of Learning Systems) [7] provided the design patterns allowing teachers and tutors to be able to capitalize and re-use the know-how of collecting tracks and usage analysis of a learning system towards another. However, these patterns are limited to a description and this project did not propose any reusable automatic analysis tools to execute these patterns. In order to resolve these problems, our objective is to propose methods and tools to teachers in observing and analyzing the progress of the learning situation. In other words, the purpose is to answer the following questions: (1) How to help teachers to use the experience and the knowledge of the community in the usage analysis? (2) How to help analysts to specify the observation means to establish the observation needs of teachers? (3) How to help teachers to evaluate their observation needs in order to improve their learning scenario?

3. Analysis process

The process of design and re-engineering of a learning scenario falls into three parts, the learning design, the learning session and the session analysis. The Figure 1 illustrates this process. We consider that this activity implies the collaboration among three roles. Teachers decide to provide a learning activity by means of computers. Developers are responsible for the deployment of a learning environment according to the learning scenario and needs. And analysts formalize teachers' observation needs and use these specifications to provide some feedback of the learning session.

![Figure 1. Process of Design and Re-engineering of a learning scenario](image)

Actually we decided to focus on the analysis process (cf. Figure 2) which is divided into four subprocesses. The modeling process concerns the specification of indicators from teachers' observation needs. An indicator is "a variable that describe 'something' related to: (a) the mode or the process or the 'quality' of the considered 'cognitive system' learning activity (task related process or quality, (b) the features or the quality of the interaction product and/or (c) the mode, the process or the quality of the collaboration, when acting in the frame of a social context forming via the technology based learning environment" [8]. Analysts use a particular formalism to express the characteristics of an indicator and how it is computed. After the learning session, there is a tracks' pre-processing step which transforms raw tracks into primary data (in the DPULS meaning). Primary data and indicators' definitions are used by the computation process which evaluates indicators and the corresponding derived data. Finally, the capitalization process uses indicators and derived data's definitions as patterns for others learning sessions' analysis.
4. A Calculation Language

In the field of the tracks analysis of learning systems, we can find several tools that allow the analysis of data collected by learning systems; they often exploit tracks for a particular context (chat, forum, etc.) and are limited to a closed set of operations. DIAS [5] proposed fifty-two indicators. Synergo [9] makes some statistics (the number of messages exchanged by each learner in a chat session, the number of objects carried out by each actor, etc) and SBT (French acronym for Tracks Based System) [10] proposes the following operators: selection, filtering and transformation. They have to be modified or re-engineered if the format of tracks is changed or modified.

In case of operations on tracks, some query languages (such as SQL, Xquery) can be used. For example, the LISTEN project [11] uses SQL to analyze collected data from the Reading Tutor. These languages allow many different types of operations on database (SQL) or XML data and documents (Xquery). They are often suitable for programmers to develop applications that concern the database. Simple SQL or Xquery commands are sufficient; however they are not adequate to calculate complex observation needs that comprise qualitative and quantitative needs. For example, a teacher needs to find a most popular scenario. Moreover, these languages are not easy to be used and expressed by analysts. Besides, CLIPS is also a language that is widely used for building expert systems because it is fast and efficient. It is also employed to operate on tracks. But the use of CLIPS requests definition of rules and facts. Therefore, this language is often adequate to the logical deduction and not suitable for calculating and combining tracks.

To resolve above problems, it is necessary to have analysis tools or means, which are independent of the format of tracks and that can calculate the observation needs of teachers from data collected by various learning systems.

4.1 A proposal for UTL

In the context of the REDiM project, [1] proposed UTL meta-language. UTL proposes two high-level types of data: the primary datum (PD) and the derived datum (DD). The primary datum is not calculated or established with the assistance of other data and is composed of the raw datum (RD), the content datum (CD) and the additional datum (AD). The intermediate datum (ID) and the indicator (I) constitute the type of derived datum that is calculated from primary data and/or other derived data. The Figure 3 shows the UTL conceptual model. Each UTL type of data is defined according to three facets of the DGU model: The Defining (D) facet defines
the observation needs; The Getting (G) facet describes the observation means to implement the data acquisition; and the Using (U) facet defines the data’s uses once they are calculated. Although UTL can be used to describe the necessary data for the tracks’ analysis, its description of the observation means is informal and it does not allow to generate the analysis tools. Therefore, the UTL-based Calculation Language (UTL-CL) is proposed to facilitate the generation and the re-use of analysis tools. It is a formal language based on UTL and inspired from classical query languages (SQL, Xquery, etc.).

The calculation language is composed of operators which combine the primary data and the derived data of UTL to calculate the observation needs (they can be qualitative or quantitative indicators) defined by teachers. This language helps analysts to define observation means that are used to calculate automatically the derived data. More precisely, it helps to produce Using.data element of the derived data. The result of this calculation is expressed in the format described in the Using.format element.

This language also allows analysts to represent teachers’ observation needs in the form of the calculation expressions. These expressions are composed of two components: operator and operand. Operand is the data collected by learning systems. It is one of the elements of the Using.data field of UTL data. Operator can be a symbol (for example: +, -, *, /, etc.) or a function (for example: sort, filter, compare, etc.). The main syntax of language is the following:

```
cal <Elements [ as <Conditions>] > where <Data>
```

Where:
- Elements: which are elements of Using.data field to calculate.
- Conditions: which are conditions executed before calculating.
- Data: which are necessary data for calculating

### 4.2 Usage example of UTL-CL

The following example presents a usage of UTL-CL with tracks data generated by the VAP-S (Virtual Action Planing Supermarket) [12] which was developed by Klinger and Marié for helping in the diagnosis of the dysexecutive syndrome [13][14]. The simple observation need proposed is the number of good actions that the patient performed in the virtual supermarket.

The necessary data to calculate this indicator is the raw data that represent actions of patient in the supermarket. This data have the following format:

```
<rawDatum type="VAPS-RD-Action">
  <actor>string</actor>
  <action>string</action>
  <controlPoint>time</controlPoint>
  <evaluation>string</evaluation>
</rawDatum>
```

The format of result is as follows:

```
<indicator type="VAPS-I-GA">
  <actor name="string">
    <goodActionNumber>integer</goodActionNumber>
  </actor>
</indicator>
```

The method of calculation is: (i) Select the raw data of type VAPS-RD-Action with an evaluation as good; (ii) Based on the result of (i), count the number of good actions per person. The expression for the above calculation is as follows:

```
cal I.actor[\(name = R.actor\)];
I.actor.goodActionNumber = count(R.action, R.actor) as filter(R.evaluation=="good")
where I= VAPS-I-GA.using.data.indicator, R= VAPS-RD-Action.using.data.rawDatum
```

Where:
- count: get the number of elements of a set of data.
- filter: extract a subset of data according to a criterion.

### 5. An analysis tool based on the language

The language that we presented in the previous section allows us to develop an analysis tool. The Figure 4 presents the proposed architecture for this analysis tool. It is composed of five modules:
- The interface module which helps analysts in the building and the definition of calculation method of indicators. It also helps teachers to exploit the calculation result.
- The interpretation module which interprets the calculation method defined by analysts and calls necessary services for the calculation of indicator.
- The service module which is composed of a set of analysis services.
- The transformation module whose role is to format the result according to the derived data definition (Using.format).
- The management module which manages data access in the database.

![Figure 4. Architecture of the analysis tool](image)

6. Conclusion

We present in this paper a process of the track analysis of learning systems. From this process, we suggest a language of calculation that can be used to combine data to establish teachers' observation needs. In the context of the REDiM project, we have chosen UTL for building our calculation language and based on it we also propose an architecture for the analysis tool that we are actually developing. This language is generic and based on XML. So it can be used by other analysis systems which also deal with XML data.

Our future work is to complete the realization of our tool and to apply it for calculating observation needs and to evaluate its efficiency. Furthermore, at present our language does not integrate external operators (i.e. operators of other analysis tools). Therefore, our future objectives are (i) to integrate these operators into our language and realize them with our analysis tool and (ii) to adapt UTL and its environment (tools) for allowing their use during the session and not only after it.

7. References


