A GENERIC MODELING OF INDICATOR WITH UTL
The Collaborative Action Function Example

Sébastien Iksal, Christophe Choquet and Diem Pham Thi Ngoc
Computer Science Laboratory, University of Maine, Le Mans, France

Keywords: TEL Engineering, Activity Tracks Modeling, Modeling Language, Capitalisation, Reusing.

Abstract: Capitalising the know-how of educational data analysis techniques in order to share and re-use them is one of the major stakes today in the specific Technology Enhanced Learning (TEL) field. We have proposed a Usage Tracking Language (UTL) to formalize data analysis techniques with a goal of capitalisation and reusability. In this paper, we present an example of use on the collaborative action function which is an indicator proposed in the ICALTS Project. After a short presentation of UTL and its conceptual model, we introduce the declarative Data Combination Language for UTL called DCL4UTL and we detail the Collaborative Action (CA) indicator modeling.

1 INTRODUCTION

Educational Data Mining (EDM) techniques are often complex. Their implementation and use for a specific Technology Enhanced Learning (TEL) system is time spending. Thus, as many authors claims (see for example (Mostow, 2004)), one of the major bottlenecks in this field is to find solutions for easily sharing and reusing such techniques. With this in mind, the EU research community has driven five significant research actions for exploring the European know-how in this field (“ICALTS, 2004), (“Interaction Analysis”, 2005), (“TRAILS”, 2004) and for proposing solutions for the capitalisation and the reuse of EDM and data analysis techniques (“CAViCoLA”, 2006), (“DPULS”, 2005).

With the help of these projects results, we have proposed the Usage Tracking Language (UTL) for formalising data mining and data analysis techniques and for capitalising them as Design and Analysis Patterns (Choquet and Iksal, 2007). Recently, we have proposed a new version of this language where the automatic methods for collecting data from a TEL system and for elaborating significant indicators (in the meaning of (Harrer, Martinez-Mones and Dimitracopoulou, 2009)) for the users (researchers, TEL designers, teachers, students) could be expressed in a declarative and generic way, and could be interpreted for mining tracks from different TEL systems (Pham Thi Ngoc, Iksal and Choquet, 2009).

This paper will present how an indicator could be modelled and performed with UTL. Our aim is to demonstrate that the modeling of the indicator and its calculation method is (1) generic and reusable, and (2) understandable by humans who want to use it in a TEL system and computational, that is to say, understandable by computers. We have chosen a well-known indicator of collaboration in a synchronous Computer-Supported Collaborative Learning (CSCL) system to exemplify the use of UTL: the Collaborative Action function Indicator (CA) which was initially defined and implemented in the MODELLINGSPACE platform (Avouris, Dimitracopoulou, Komis and Margaritis, 2003).

The calculation of this indicator is based on two others, the Agent and the Interaction indicators. As synthesised below, the Agent indicator computes in a CSCL system the number of users (learners, teachers) who have used at least one collaboration or interaction tool during a collaborative session. The Interaction indicator expresses the amount of interactions that have implemented through a communication channel (e.g. a chat or any collaborative workplace) during a given period. Finally, the Collaborative Action Function indicator combines these indicators for providing a quantitative evaluation of the collaboration inside a group of users during a given period.

Consider a collaboration session interval [t_0–t_m]. This time interval is quantified using a parameter n: $t_i = t_0 + i*d$, where $d = (t_m - t_0)/n$. Then CA is expres-

Iksal S., Choquet C. and Pham Thi Ngoc D. (2010).
A GENERIC MODELING OF INDICATOR WITH UTL - The Collaborative Action Function Example.
In *Proceedings of the 2nd International Conference on Computer Supported Education*, pages 114-119

sed as follow.

$$CA(t_i) = \sum_{k=1}^{k_{max}} A(k, t_i) * I(k, t_i) \qquad (1)$$

The next section of this paper presents the conceptual model of UTL. The third section describes the modeling of the Collaborative Action Function indicator with UTL. We concludes this paper by dealing with the benefits and limitations of the use of UTL regarding the capitalization, the sharing and the reuse of indicators.

2 THE USAGE TRACKING LANGUAGE: UTL

As we said before, UTL was proposed in order to formalise data mining and data analysis techniques in a declarative way for capitalising and re-using them. We first present the DGU Model on which UTL is based for its descriptions, next, we introduce the conceptual model of our language and we finish with the combination language which is used to describe the data transformation and calculation methods.

2.1 Three Viewpoints on Tracks: The DGU Model

In our tracks' reengineering framework, we stress the need for track modeling before the learning session, and to consider tracks as a pedagogical object, like any other learning object, such as resources or scenarios for instance. If this is frequently the case in existing systems, when the tracking purpose is to provide learners and/or the system with useful information, it is more unusual for supporting the tutor, and it's rare for providing feedback to designers.

With this in mind, we could say that, as far as it could be possible for him, the designer who is engaged in the engineering process of a TEL system, should model the tracking needs of the learning session. Then, the tracking needs should be instantiated on the educational environment in order to model the effective tracking means. Finally, one should also model the expected uses of these tracks, in terms of building the descriptive scenario of the learning session for analysing the usage of the TEL system. This is the way we have defined the three facets for the tracks modeling (see Figure 1):

- the Defining (D) facet which models the tracks needs;
- the Getting (G) facet which models the tracks means;
- the Using (U) facet, which models the tracks uses purpose.

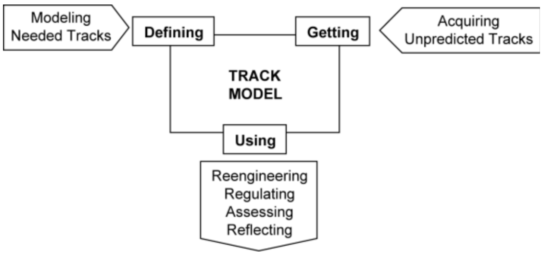

Figure 1: The DGU model of a Track.

2.2 The Conceptual Model

UTL Conceptual Model falls into three parts (cf. Figure 2), but we have restricted the focus of this paper on its re-usable part. This part of the conceptual model is called UTL/P (for UTL/Pattern), and corresponds to the capitalisable/reusable part of the language, details all transformations and data needed to build an indicator. We have identified two main data types for tracks: the derived-datum and the primary-datum.

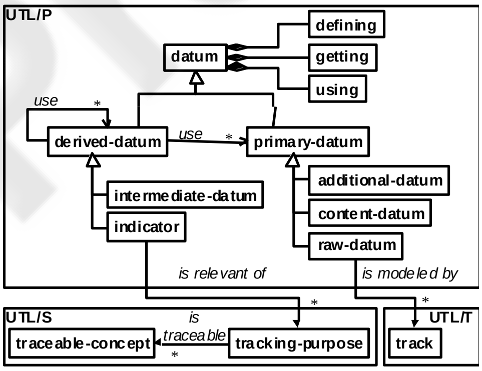

Figure 2: UTL Conceptual Model.

The primary data are not calculated or elaborated with the help of other data. They consist of raw-datum, content-datum and additional-datum. The raw-datum could be recorded before, during or after the learning session by the learning environment. The content-datum concerns to the outcomes provided by the learning session's actors. The additional-datum is linked to the learning situation and could be involved in the usage analysis.

The derived data are calculated or inferred from primary data and/or other derived data. The indicator is a derived-datum which has a pedagogical significance and is always relevant to a pedagogical context. It is always defined for, at least, one exploitation purpose (for instance, in an instructional design context, we have proposed to link each indicator to at least one concept of the learning scenario). A derived-datum which has to be calculated but which has no pedagogical significance is qualified as an intermediate-datum. All the information models of these data types have been published in (Choquet and Iksal, 2007).

2.3 The Data Combination Language

In order to build derived data and indicators we have proposed a data combination language extension called DCL4UTL (Pham Thi Ngoc, Iksal and Choquet, 2009). The main aim of this language is to be used by the analyst as a query language to describe the way to combine data for producing a new one. This language is designed to be declarative and to be processed automatically by the system. Another research goal concerns the capitalisation of these combination methods for the re-usability of them. Our proposal is based on classical languages, integrated in UTL and works on the Using section of UTL data. All Using sections have a format field and a data field. The format describes how the value has to be represented, and the data field concerns the result of the combination. The grammar of the language is too long to be published here, thus we focus on a UTL example of use.

3 AN EXAMPLE: THE "COLLABORATIVE ACTION FUNCTION" INDICATOR

To elaborate compatible raw data with the calculation of the CA indicator, we assume the TEL system is able to provide logs with the following information:

```
{logfile}:{time stamp, actionCode,
user, resourceID}
```

Where:

- *actionCode* corresponds to the interaction performed,
- *user* corresponds to the agent involved,
- *resourceID* corresponds to the interaction channel used.

We compute CA(ti) by parsing that kind of log files after or during the learning session.

3.1 Some Explanations on the DCL4UTL Syntax

DCL4UTL is based on classical operators:

- Logical operators : and, or, not, …
- Comparison operators : <=, >=, =, …

It is possible to use some pre-defined functions, in this example, we need the following:

- 'first' and 'end' to get the first and the last elements of a set of data.
- 'count' and 'countDis' to get the number of elements and the number of distinct elements of a set of data.
- 'sort' to organise a set of data according to a criterion.
- 'filter' to extract a subset of data according to a criterion.

In order to organise the combination, we have the following keywords:

- 'cal' to declare the combination
- 'as' to declare the specific organisation of data (filter, sort, …)
- 'for each' is a control for a loop.
- 'where' allows aliases for data.

3.2 Modeling of the Collaborative Action Function Indicator

This section presents the Defining (Table 1) and the Using (Table 2) parts of the UTL modeling. They correspond to descriptions that are independent from the calculus which is detailed in the next section.

Table 1: Defining Section of CA Indicator.

Defining	Title	Collaborative Action Function (CA)
	Cardinality	N
	Description	Indicates the degree of collaborative action […].

Table 2: Using Section of CA Indicator.

Using	Data	Not used in modeling

	Format	`<indicator type="I-CA">`
		`    <ca       ti="time">`
		`float`
		`    </ca>`
		`</indicator>`

3.3 Calculation Method of the Indicator

In this section, we will not present all data modeling, but the format for primary data and for derived data, we will give the format and the getting formula.

1. Firstly, we create the RD-AUR raw datum (in the UTL meaning) to model the log file's content.

```
<rawDatum type="RD-AUR">
   <timeStamp> time </timeStamp>
   <actionCode> string </actionCode>
   <user> string </user>
   <resourceID> string </resourceID>
</rawDatum>
```

2. Secondly, we describe the additional data (in the UTL meaning) corresponding to the number of intervals and the list of interaction channels:

   a) The AD-Parameter additional datum stands for the parameter n, which is the number of intervals.

```
<additionalDatum type="AD-Parameter">
   <n> integer </n>
</additionalDatum>
```

   b) The AD-Channel additional datum stands for the interaction channels' set.

```
<additionalDatum type="AD-Channel">
   <channel id="integer"> string
   </channel>
</additionalDatum>
```

3. To compute $CA(t_i)$, $I(k,t_i)$ and $A(k,t_i)$, we need to define and calculate (in the UTL meaning) the following intermediate data.

   a) The ID-BET intermediate datum stands for the beginning and the ending time of the session.

```
<intermediateDatum type="ID-BET">
   <t0> time </t0>
   <tm> time </tm>
</intermediateDatum>
```

The calculation expression is:

```
cal    ID.t0 = first(R.timeStamp);
       ID.tm = end(R.timeStamp)
as     sort(R.timeStamp)
where  ID = ID.BET.using.data ┐
           .intermediateDatum,
       R  = RD-AUR.using.data.rawDatum
```

The ID-D intermediate datum stands for $d=(t_m - t_0)/n$.

```
<intermediateDatum type="ID-D">
   <d> float </d>
</intermediateDatum>
```

The calculation expression is:

```
cal    ID1.d = (ID2.tm - ID2.t0) / AD.n
where  ID1 = ID-D.using.data ┐
           .intermediateDatum,
       ID2 = ID-BET.using.data ┐
           .intermediateDatum,
       AD = AD-Parameter.using.data ┐
           .additionalDatum
```

4. Then, we compute two indicators: I-Interactions and I-Agents.

   a) The I-Interactions indicator stands for the amount of interactions that have been implemented via channel k during (t_{i-1}-ti].

```
<indicator type="I-Interactions">
   <interactions ti="time"
     channel="integer">
      integer
   </interactions>
</indicator>
```

The calculation expression for all interaction channels is as follows:

```
cal   for each $k in AD.channel[id] do
       for each $value in [1;A.n-1] do
        $ti1 = ID1.t0 + $value * ID2.d;
        $ti0 = ID1.t0 + ($value-1) *
           ID2.d;
        I.interactions[ti=$ti1]
           [channel=$k] =
           count(R.actionCode)
      as filter(R.timeStamp <= $ti1 and
           R.timeStamp > $ti0 and
           R.resourceID == $k)
where I = I-Interactions.using.data ┐
           .indicator,
      A  = AD-Parameter.using.data ┐
           .additionalDatum,
      AD = AD-Channel.using.data ┐
           .additionalDatum,
      ID1 = ID-BET.using.data ┐
           .intermediateDatum, ID2 =
ID-D.using.data ┐
           .intermediateDatum,
      R = RD-AUR.using.data.rawDatum
```

   b) The I-Agents indicator stands for the number of agents that have posted at least one message through channel k during (t_{i-1}-ti] interval.

```
<indicator type="I-Agents">
   <nbagents ti="time"
     channel="integer">
     integer
   </nbagents>
</indicator>
```

The calculation expression for all interaction channels is as follows:

```
cal   for each $k in AD.channel[id] do
```

Table 3: Getting section of the CA Indicator.

Getting	Title	Calculation of the Collaborative Action Function			
	Component	Primary-datum	AD-Channel		
	Component	Derived-datum	ID-D		
	Component	Derived-datum	ID-BET		
	Component	Derived-datum	I-Agents		
	Component	Derived-datum	I-Interactions		
	Method	Type	Automatic		
		Tool	Title	Declarative combination function based on agents and interactions data	
			Description	Text	CA(t_i)=Sum[k=1;kmax](Agents(k,t_i)*Interactions(k,t_i))
				DCL4UTL-formula	Cf. formula located after this table

```
             for each $value in [1;A.n-1] do
               $ti1 = ID1.t0 + $value * ID2.d;
               $ti0 = ID1.t0 + ($value-1) *
                 ID2.d;
               I.nbagents[ti=$ti1][channel=$k]
                 = countDis(R.user)
           as filter (R.timeStamp <= $ti1
                 and R.timeStamp > $ti0 and
                 R.resourceID==$k)
where I = I-Interactions.using.data ¬
                .indicator,
      A = AD-Parameter.using.data ¬
                .additionalDatum,
      AD = AD-Channel.using.data ¬
                .additionalDatum,
      ID1 = ID-BET.using.data ¬
                .intermediateDatum,
      ID2 = ID-D.using.data ¬
                .intermediateDatum,
      R = RD-AUR.using.data.rawDatum
```

5. Finally, we compute CA(t_i) by defining indicator I-CA. The Table 3 represents the UTL modeling of the Getting part.

```
cal     for each distinct $ti in
          I1.interactions[ti] do
            I.ca[ti=$ti] = 0;
        for each $k in AD.channel[id] do
            I.ca[ti=$ti] = I.ca[ti==$ti]
            + (I1.interactions[ti==$ti]
            [channel==$k]*I2.nbagents
            [ti==$ti][channel==$k])
where   I = I-CA.using.data.indicator,
        I1 = I-Interactions.using.data ¬
            .indicator,
        I2 = I-Agents.using.data ¬
            .indicator,
        ID1 = ID-BET.using.data ¬
            .intermediateDatum,
        ID2 = ID-D.using.data ¬
            .intermediateDatum,
        AD  = AD-Channel.using.data ¬
            .additionalDatum
```

4 CONCLUSIONS

With this paper, we have shown how UTL and its extension DCL4UTL could be coupled for expressing and modeling indicators in a generic and reusable way. Each datum required for modeling an indicator and each calculation method could be expressed in a generic way. Part of their description targets the human (and more specifically the teacher, as the designer or as the tutor), as *getting.example* or *getting.method.tool.description.text* fields, for facilitating the capitalisation and the sharing of an indicator, expressed as a Pattern. Some other fields targets the computer for allowing the reusability and the interoperability, as *using.format* or *getting.method.description.DCL4UTL-formula*.

The only TEL system specifics are the raw data extraction methods which have to be coded according to the specificity of the log files.

Even if the major part of UTL is understandable by a teacher, it is not the case of DCL4UTL extension we propose: using this language requires competencies of a data analyst and/or a software engineer. This is one the limitation of our approach: actually, a teacher could not be independent during the indicators modeling task, he should be helped by specialists. Nevertheless, we think that this limitation could be bypassed at middle-term: more indicators will be modelled and capitalised with UTL, more they can be shared in the community of practice and reused by end-users, e.g. teachers.

We have actually developed the interpreter of DCL4UTL, and improved it on existing indicators we found in the literature and/or we need for our existing TEL systems (see for instance (Pham Thi Ngoc et al., 2009)). We have planned also to develop a user-friendly editor for facilitating the modeling with this language. We have also proposed a Web server architecture for storing and sharing indicators and other data modelled with UTL.

UTL could be tailored for different uses. As an example, we have defined add-ons to UTL specifically for Learning Management Systems (LMS) such as the Moodle platform where the learning activities are designed by using an Educational Modelling Language (Rawlings, van Rosmalen, Koper, Rodriguez-Artacho and Lefrere, 2002). Basically, one add-on, the UTL-Track (see Figure 2) allows to model raw-data produced by the LMS. The other one, the UTL-Scenario allows to model links between indicators and the pedagogical scenario. In this way, an indicator could be put in the context of a specific concept of a scenario (as a learner role, or a specific resource) and/or limited to a specific passage of the learning activity.

We actually work on a new add-on tailored for adaptable TEL systems. In this kind of system, the user could take decisions of adaptation which impact on the configuration of the TEL system and on the pedagogical scenario itself. The add-on will allow to model the context of such a decision, the modeling phase could be held during or after the learning session. This functionality will be useful for analysing the pertinence of an adaptation and its impact, for reengineering purposes.

ACKNOWLEDGEMENTS

We would like to thank all of the contributors of CAViCoLA, DPULS, IA, ICALTS and TRAILS European projects for their results reused here as basis or examples.

REFERENCES

Avouris, N., Dimitracopoulou, A., Komis, V., Margaritis, M., 2003. Participatory Analysis of Synchronous Collaborative Problem Solving using the OCAF Methodology and Tools. In *Proceedings of CSCL2003 Community events*. p. 232-234.

CAViCoLA, 2006. Computer-based Analysis and Visualization of Collaborative Learning Activities, *European Network of Excellence Kaleidoscop*e. At: http://www.noe-kaleidoscope.org (01/10/09).

Choquet, C., Iksal, S., 2007. Modeling Tracks for the Model Driven Reengineering of a TEL System. In *Journal of Interactive Learning Research (JILR), Special Issue Usage Analysis in Learning Systems: Existing Approaches and Scientific Issues,* Vol. 18 n°2. p. 161-184.

DPULS, 2005. Design Patterns for Recording and Analysing Usage of Learning Systems, *European Network of Excellence Kaleidoscop*e. At: http://www.noe-kaleidoscope.org (01/10/09).

Harrer, H., Martinez-Mones A., Dimitracopoulou A., 2009. Users' data: Collaborative and social Analysis, In *Technology-Enhanced Learning – Principles and Products*, Eds: Nicolas Balacheff, Sten Ludvigsen, Ton de Jong, Ard Lazonder, Sally Barnes, Lydia Montandon – Dordrecht : Springer Verlag, Chapter 11, p. 175-193.

Interaction Analysis, 2005. Interaction Analysis – Supporting participants in technology based learning activities, *European Network of Excellence Kaleidoscope*. At: http://www.noe-kaleidoscope.org (01/10/09).

ICALTS, 2004. Interaction & Collaboration Analysis' supporting Teachers & Students' Self-regulation, *European Network of Excellence Kaleidoscop*e. At http://www.noe-kaleidoscope (01/10/09).

Mostow, J., 2004. Some useful design tactics for mining ITS data, In *Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes, International Conference on Intelligent Tutoring Systems (ITS'04)*. p. 20-28.

Pham Thi Ngoc, D., Iksal, S., Choquet, C., Klinger, E., 2009. UTL-CL: A Declarative Calculation Language Proposal for a Learning Tracks Analysis Process, In *The 9th IEEE International Conference on Advanced Learning Technologies (ICALT' 2009)*.

Rawlings, A., van Rosmalen, P., Koper, R., Rodriguez-Artacho, M., Lefrere, P., 2002. Survey of educational modelling languages (emls). Technical report, September 19, 2002.

TRAILS, 2004. Personalised and Collaborative Trails of Digital and Non-Digital Learning Objects. *European Network of Excellence Kaleidoscop*e. At: http://www.noe-kaleidoscope.org (01/10/09).