An infrastructure for creating graphical indicators of the learner profile by mashing up different sources

Luca Mazzola
USI - Università della Svizzera italiana
ITC - Inst. for Communication Technologies
Via Buffi 13, CH 6900 Lugano, Switzerland
luca.mazzola@usi.ch

Riccardo Mazza
USI - Università della Svizzera italiana
ITC - Inst. for Communication Technologies
Via Buffi 13, CH 6900 Lugano, Switzerland
riccardo.mazza@usi.ch

ABSTRACT

The procedures to collect information about users are well known in computer science till long time. They range from getting explicit information from users, required in order to enable some functionalities, to the gathering of user behaviors, collected as log files generated by software applications. In the field of Technology Enhanced Learning the creation of a user profile is necessary in order to fulfill some didactical tasks, such as measuring the degree of participation to a course, or the performance on quizzes and assignments. The task of collecting students' data is normally performed by Learning Management Systems, which also provide with a way to explore this data. Our approach extends the information that models user profiles in Learning Management Systems with data coming from other online resources, such as social network websites. In this paper, we describe our idea of opening the learners' profile, eventually for integrating it with external on-line resources. The ultimate goal of our work is to create graphical indicators for the profile of the learners that take into account internal and external user data, in order to have a more complete and comprehensive view of user behaviors in Learning Management Systems.

Categories and Subject Descriptors

H.5.m [**INFORMATION INTERFACES AND PRESENTATION**]: Miscellaneous—*Graphical user interfaces*

General Terms

Algorithms, Design, Experimentation, Human Factors

Keywords

Open Learner Models, MashUp, Life Long Learning

1. INTRODUCTION

The procedures to collect information about users and their interaction with the software are well known in computer science till long time [1]. They were originally designed to support developers in discovering bugs or analyzing user interactions with the system, in order to improve the quality of the produced software. They range from getting explicit information from users, required in order to enable some functionality, to the gathering of user behaviors, collected as log files generated by software applications. This approach has recently become less spread with the diffusion of the personal computer, due to the difficulties of collecting user data from the personal user. Nowadays, with the wide spread of broadband, always on, Internet connections, collecting user data is again considered as an interesting process. So, almost all Web servers implement functionalities to collect user's navigation footprint [2]. This is particularly important in relation to the fact that nowadays most Learning Management Systems are web-based [3]. In the domain of Technology Enhanced Learning, tracking user behaviors in form of log data from Web applications, assumes a very important role to support both learner and teachers, working as data source for monitoring tools [4].

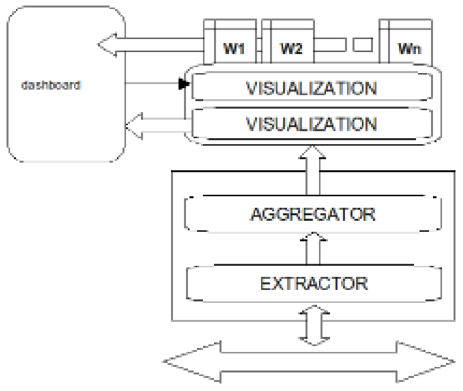


Figure 1: The suggested architecture. The three main logical blocks are visible.

A monitoring tool is a component in charge of providing relevant information for the current activity: an example is the set of led lamps on a computer case, in charge of giving an indication of the state of hardware components to the user (such as the usage of hard-disk, the network activity, the power state, and so on). Logs are normally used to create an internal profile (also called model) for the user, through a process of aggregation of specific events which allows to recognize a complex human activity or task. These models are normally stored internally without the possibility for the user to scrutinize his personal profile. The literature

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
AVI - Advanced Visual Interfaces International '10 Rome, IT
Copyright 2010 ACM 978-1-4503-0076-6 ...$10.00.

reports that opening profiles to user inspection (with the so called "Open Learner Model" approach [5]) could help in explaining personal states to the learners. Also other critical aspects could be supported by opening the personal profile, like helping in identifying problems or lacks of precision in the profile itself.

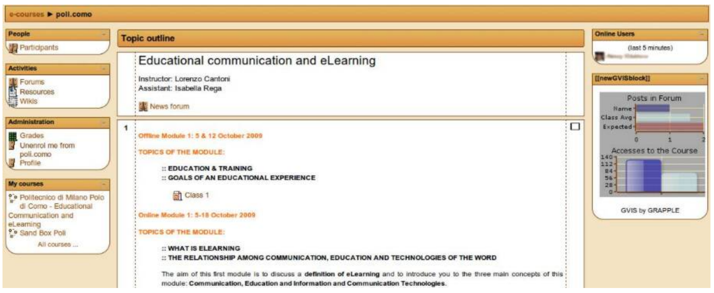
2. THE *GVIS* INFRASTRUCTURE

Providing a way to open the profile to user inspection is important in the domain of Life Long Learning: the presentation of personal information as indicator of the learning process is widely accepted as one of the key points to improve participation and increase the satisfaction of participants [6]. Although many Learning Management Systems already provide the possibility to explore user tracking data, in some cases the visual presentation of information is not well suited to the human perceptive system. In other cases, the presentation of data is limited to a subset of it or is predefined by developers and fixed [7]. We want to provide an easy way to create an effective graphical presentation of arbitrary fragments of data. We propose a three-tier architecture composed by a data extractor, a data aggregator, and a visualization layer (see Fig. 1). All the levels rely on an XML configuration file that the administrator can modify or expand in order to create graphical indicators - in form of widgets - of one or more interesting characteristics of the user profile. Our infrastructure is able to potentially connect to any data source, regardless the different connection types (databases, Web services, connection bus, ...). This is simply achieved by writing a small piece of adapter code, in order to make the protocol and data format compatible with our internal structures. In the following subsections a description of every module shown in Fig. 1 is presented, both in term of functionalities provided and in term of XML configurations. The small pieces presented and explained in the next sections are used to create the widgets shown in Fig. 3 and Fig. 4.

2.1 The Extractor

The extractor represents the lowest level of our application and is in charge of retrieving data from the sources. This piece of software takes care of making a syntactical and semantic translation of data received from a particular source to the internal format. Both the pull and the push approach can be implemented to retrieve and collect data. To achieve this objective it relies on a small amount of code that describes the data structure used by a particular source. The following excerpt of a configuration file is useful to explain some peculiarities of the module:

```
[1]<source name="MoodleEvaluationGlobal">
[1.1]   <accessinfo>
[1.1.1]     <accesstype>DB</accesstype>
[1.1.2]     <accesspoint>**IP**</accesspoint>
[1.1.3]     <accessmode>mysql</accessmode>
[1.1.4]     <accesssource>**DB_name**</accesssource>
[1.1.5]     <username>**UserID**</username>
[1.1.6]     <password>**PWD**</password>
[1.1.7]     <lifetime>30</lifetime>
        </accessinfo>
[1.2]   <query>
[1.2.1]       <sql>
select GI.userid AS name, GG.finalgrade AS value
```


e-courses ▶ poli.como

Topic outline

Educational communication and eLearning

Instructor: Lorenzo Cantoni
Assistant: Isabella Rega

News forum

1

Offline Module 1: 5 & 12 October 2009

TOPICS OF THE MODULE:

:: EDUCATION & TRAINING
:: GOALS OF AN EDUCATIONAL EXPERIENCE

Class 1

Online Module 1: 5-18 October 2009

TOPICS OF THE MODULE:

:: WHAT IS ELEARNING
:: THE RELATIONSHIP AMONG COMMUNICATION, EDUCATION AND TECHNOLOGIES OF THE WORD

The aim of this first module is to discuss a **definition of eLearning** and to introduce you to the three main concepts of this module: **Communication, Education and Information and Communication Technologies.**

Figure 2: The output of GVIS module inserted into a Moodle course.

```
FROM mdl_grade_items AS GI JOIN mdl_grade_grades AS GG
ON GI.id=GG.itemid WHERE ... AND GI.courseid=?
ORDER BY finalgrade DESC
            </sql>
[1.2.2]      <parameters>
[1.2.2.1]        <param>course.id</param>
             </parameters>
[1.2.3]      <resulttype>listofrecords</resulttype>
        </query>
     ...
  </source>
[2]<source name="MoodleEvaluationSingle">
[2.1]   <accessinfo>
            ...
[2.1.7]     <lifetime>0</lifetime>
        </accessinfo>
[2.2]   <query>
[2.2.1]     <sql>
              ...
            </sql>
[2.2.2]     <parameters>
[2.2.2.1]       <param>course.id</param>
[2.2.2.2]       <param>user.id</param>
            </parameters>
[2.2.3]     <resulttype>numeric</resulttype>
        </query>
      ...
   </source>
```

In section *[1.1]* all the parameters for the connection with the data source are included, in *[1.1.1]* the type of adapter class is declared, together with *[1.1.3]*, that refines the previous indication. Section *[1.1.7]* defines the buffer lifetime for the extracted information: in the specific case the value of 30 means that the system will bufferize and reuse the data for all the following requests that will occurr within a timeframe of 30 secs. This could be useful for data sources having a slow response time. If this functionality is not needed, a 0 value could be used like in fragment *[2.1.7]*. In the second half of the source configuration a specific query is inserted (like in *[1.2.1]*), with one or more parameters (see *[1.2.2.1]* and *[2.2.2.1]*, *[2.2.2.2]*). At the end there is a declaration of the expected output type, whose range could be one of the following: ***numeric*** *[2.2.3]*, ***record***, ***list*** or ***listofrecords*** *[1.2.3]*.

2.1.1 The Adapter

As already stated, the Extractor is able to connect to different data sources. This approach allows our solution to be seamlessly extended with different and heterogeneous data providers. When a new data provider is added to the infrastructure, a new mapping for the provider also has to be provided. This could be done either of writing a new adapter class or reusing an existing one. Right now, we have implemented specific classes for MySQL (called DB, due to the fact that it could accept almost every source conforming to SQL protocol), a specific WebService interface and a SPARQL endpoint interface.

2.2 The Aggregator

The aggregator is in charge of filtering raw data collected by the extractor and to apply some operations to aggregate data. This aggregation is based on the model that the teacher or instructional designer will provide; it represents the useful information for learner and is strictly related to the pedagogical approach provided in the learning experience. This profile externalization - achieved through information representation - could play an inportant role in supporting the learning process. With such an architecture we expect to offer a customized tool adaptable to the specific didactic design model. The use of models (based on XML syntax with an associated NameSpace) provides a formal way for designing the expected behavior of the aggregator module.

2.2.1 Didactic models

Didactic models are defined by means of configuration files that describe how source data is aggregated in order to build up meaningful and useful indicators. Here we show some XML fragments of this configuration that primarily describe which data is expected as an input (like *[3.1.1]* and *[3.1.2]* or *[4.1.1]*) and which type of information will be produced as an output (as in *[3.3]* or in *[4.2]*). The transformation process from input to output is also described in the form of a pipeline of operations: the output of a step could be used as input on a following one, like for the average operation in *[3.2.2.1]*, whose parameter compute is set to true (see *[3.2.2.2.1]*). Operations (like in *[3.2.1.1]* and *[3.2.2.1]*) are provided by internal classes that could be extended, when required. The class name - like *ExtractCol* or *average* in the example - defines the operation performed and implements an abstract model, that defines the properties and methos expected.

```
[3]<source name="getMooEvalAvg">
[3.1]  <extraction>
[3.1.1]    <toextract>MEvalGlobal</toextract>
[3.1.2]    <toextract fix="true">1</toextract>
       </extraction>
[3.2]  <computation>
[3.2.1]    <tocompute>
[3.2.1.1]      <operation>ExtractCol</operation>
[3.2.1.2]      <parameters>
[3.2.1.2.1]        <param>0</param>
[3.2.1.2.2]        <param>1</param>
               </parameters>
 [3.2.1.3]     <resulttype>list</resulttype>
           </tocompute>
[3.2.2]    <tocompute>
[3.2.2.1]      <operation>average</operation>
[3.2.2.2]      <parameters>
[3.2.2.2.1]        <param computed="true">0</param>
               </parameters>
[3.2.2.3]      <resulttype>numeric</resulttype>
           </tocompute>
       </computation>
[3.3]  <resulttype>numeric</resulttype>
   </source>

[4]<source name="getMooEvalSingle">
[4.1]  <extraction>
[4.1.1]    <toextract>MooEvalSingle</toextract>
[4.2]  </extraction>
       <resulttype>numeric</resulttype>
   </source>
```

2.3 The Visualization module

The Visualization module is the part that produces the actual visualization. It is divided in two components: the initial container, called dashboard, and the actual contents, represented by some graphical widgets that map information into the final indicator graphical form. The configuration of the dashboard can be personalized based on some parameters set at a system level, like in the following XML fragment, that defines the type (*[5.1]*) and the data sources (*[5.1.1.1]* and *[5.2.1.1]*) to be used by the actual widget:

```
[5]<widget name="Note">
     ...
[5.1]  <chart type="hbar">
[5.1.1]    <chartsource>
[5.1.1.1]      <data>getMooEvalSingle</data>
               ...
           </chartsource>
[5.2.1]    <chartsource>
[2.2.1.1]      <data>getMooEvalAvg</data>
               ...
           </chartsource>
      </chart>
   </widget>
```

The generated widget is in the form of a horizontal bar-chart (*[5.1]*) and exposes two pieces of information: the evaluation for the student (*[5.1.1.1]*) compared with the class average (*[5.2.1.1]*).

2.3.1 The Dashboard and the Widget generator

The dashboard is instantiated once for every client, when the service is started. It provides two kinds of functionalities: it is a container for all the widgets and it collects all the user interactions and feedbacks, such as data filtering or widget visibility change. As a background, it provides a common place for different widgets, each of which specialized in representing a specific aspect: in this way, it acts as a control panel of the learner situation. The interaction functionality is important because this is the only level at which the final user (learner) can express preferences or partially change the behavior of the whole system. The widgets are dynamically created on the top of the dashboard: every widget, based on a graphical template that defines its main aspects, represents an encoding of a single indicator for one or more characteristics of the learner profile. The widget is the final output of the application.

3. A FIRST APPLICATION

A first application was developed in the course "Educational Communication and eLearning" held by prof. Can-

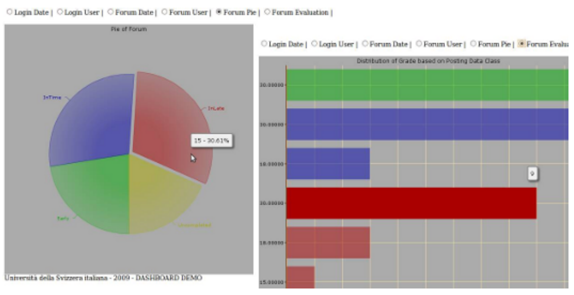


Figure 3: The GVIS module for teacher. Here a temporal classification of the posts with the relative evaluation is presented.

toni at Politecnico di Milano in Italy (see Fig.2). The GVIS application has been integrated with the Learning Management System Moodle. In this case, we have not mixed together different data sources, but our visualization infrastructure helped to mix together different data, and graphically represent contextual information about the course and the learners. Fig. 3 shows a mix of accesses to course/resources done by students and forum posts, that are the activities considered important by the Instructional designer who developed the online part of the course. The interesting part resides in the graphical comparisons between the learner's specific information and the average value achieved by the class, that can work as a contextual reference for the progress of the user. This specific widget is in charge of representing the number of logins and the contributes posted in a forum. Some functionalities for the teacher have been implemented in order to support the tutoring, as in Fig. 3, a collage of two of the widget we provide.

Specifically, in the pie chart, group of posts posted during the same temporal interval, based on different deadlines, is depicted using different colors. In the bar chart, the relations between the evaluation and the identified classes (indicated by means of the same color) are presented.

3.1 The "grade" widget

In the online course there are different activities that have be performed by the learners during specific temporal intervals and these have to be graded by tutor during the semester. For this reason we developed a specific widget for presenting the individual state, as the average grade reached by a student till now. This information, specific for every learner, is compared with the overall average grade of the class. This is particularly important because the student can check his average grade at any time, not only at the end of the course. This can be considered as an awareness tool [8]. In Fig. 4 two screenshots for different learners in distinct classes are presented, in which they can realize if they are aligned with their colleagues.

4. CONCLUSIONS

Our tool allows to aggregate information coming from different sources and to create graphical representations of these data in order to support their interpretation by means of the visual human system. In the context of Life Long Learning the presentation of this contextual information to the learners and to the teacher or tutors becomes important in order to support better awareness of the learning situation

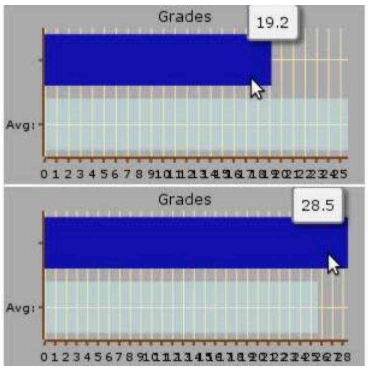


Figure 4: The graph of grades for two different students, from two distinct classes.

and to promote participation. The encoding and graphical presentation [9] [10] of this information is also relevant to make them useful for the learning process.

4.1 Acknowledgments

We want to thanks prof. Lorenzo Cantoni for letting us use the course example and the student Luca Marchesi for having provided us with so valuable ideas and for the help in implementing the infrastructure. A big thanks to our colleague Davide Eynard for all the suggestions and feedbacks. First author thanks people involved in RedInk PhD School.

5. REFERENCES

[1] Jeff, T., 2005. Software quality engineering: testing, quality assurance, and quantifiable Improvements. John Wiley

[2] Hoppe, U., Ogata, H., and Soller, A., 2007. The Role of Technology in Cscl: Studies in Technology Enhanced Collaborative Learning. Springer

[3] Mazza, R., Botturi, L., Tardini, S., 2006. FOSLET 2006. Proceeding of Workshop on Free and Open Source Learning Environments and Tool. Como, Italy

[4] Mazzola, L., Mazza, R., 2009. Supporting learners in Adaptive Learning Environments through the enhancement of the Student Model. In Proceeding of Human Computer Interaction International 2009. San Diego, CA

[5] Bull, S. ,Kay, J., 2007. Student Models that Invite the Learner In: The SMILI:() Open Learner Modelling Framework. Int. J. Artif. Intell. Ed. 17, 2. pp. 89-120

[6] Shneiderman, B., Plaisant, C., 2004.Designing the User Interface: Strategies for Effective Human Computer Interaction. 4th ed. Addison Wesley

[7] Dimitrova, V., 2003. STyLE-OLM: Interactive Open Learner Modelling. Int. J. Artif. Intell. Ed. 13, 1

[8] Romero, M., Tricot, A., and Mariné, C., 2009. Effects of a context awareness tool on students' cognition of their team-mates learning time in a distance learning project activity. Proceedings of the 9th international conference on CSCL - Volume 1 - Rhodes, Greece

[9] Shahrour, G., Bull, S., 2008. Does 'Notice' Prompt Noticing? Raising Awareness in Language Learning with an Open Learner Model. AH2008 - Springer.

[10] Spence, R., 2007. Information Visualization: design for interaction, 2nd ed. Pearson Education/Prentice Hall.