GVIS: an integrating infrastructure for adaptively mashing up user data from different sources

Luca Mazzola, Riccardo Mazza
University of Lugano, Faculty of Communication Sciences
Institute for Communication Technology
Via Buffi 13, CH-6904 Lugano, Switzerland
Phone: +41 58 666 4760 - Fax: +41 58 666 4647

Abstract

In this article we present an infrastructure for creating mash up visual representations of the user profile that combines data from different sources. We explored this approach in the context of Life Long Learning, where different platforms or services are often used to support the learning process. The system is highly configurable and adaptive: data sources, data aggregations, and visualizations can be configured on the fly by the administrative user without changing any part of the software, and have an adaptive behavior based on linear combination of conditions about user or system characteristics. The visual profiles produced can assume different graphical formats and can be bound to different data, automatically adapting to personal preferences, knowledge, and contexts. We applied our infrastructure to a set of federated Learning Management Systems, retrieving information from different sources and creating some indicators of the learning activity. The software we developed provides learners with adaptive indicators of the learning state, and allows instructors to monitor the progress of their learners.

Keywords— **TEL, HCI, Adaptive Presentations, Data Mashup.**

1 Introduction

One of the problematic aspects of Technology Enhanced Learning (TEL) is the lack of contextual information to support online learners [1]. Even if nowadays we are quite used to fulfill many tasks without direct support, or feedback from others

(such as work, search for information, or play), this could be a problem in the context of TEL [2].

As reported in literature [3], one of the key issues to solve for making TEL an effective instrument for didactics is the low level of engagement perceived by users with the online learning experience. This is particularly important in the context of Life Long Learning (LLL), in which the online learning is becoming more and more prevalent. One of the most important sources of information to provide feedback to students is the student model, created by some learning environments to keep track of the learner's knowledge and skills acquired during the learning process. In particular, adaptive learning systems maintain a student model to allow the system to perform some adaptation based on the knowledge acquired by the learner during the process. In order to increase the level of engagement of learners and stimulating the perception of user about his current status, a possible solution is to allow the students to inspect their user model, in order to encourage reflection as learning [4]. Student models are usually made available to learners in form of visual representations. In the LLL context, quite rarely all the user information is collected into a single tool, but often they are stored in a number of platforms used for different purposes. Very often companies and schools make use of different learning environments for some reasons (such as different LMS maintained by different suppliers of courses, intranet websites, custom application). For this reason, there is a need to aggregate data from different tools (student models, intranet data, LMS data, ...) and provide an uniform way to present this data to the interested users, preferably in visual format [5]. On the other side, the online user is exposed to an overabundance of data, that requires special capability to deal with [6]. Learners are being confronted with the consequent information overload problem, that becomes a real problem when it distracts from the learning activity and makes the learner confused about relevant information.

The adaptation can help in creating more comprehensible and usable indicators. For example, the adaptive dashboards used in the field of Business Intelligence [7]; in these applications, adaptation is used to build smart indicators to be placed in the dashboard and minimize the cognitive overload.

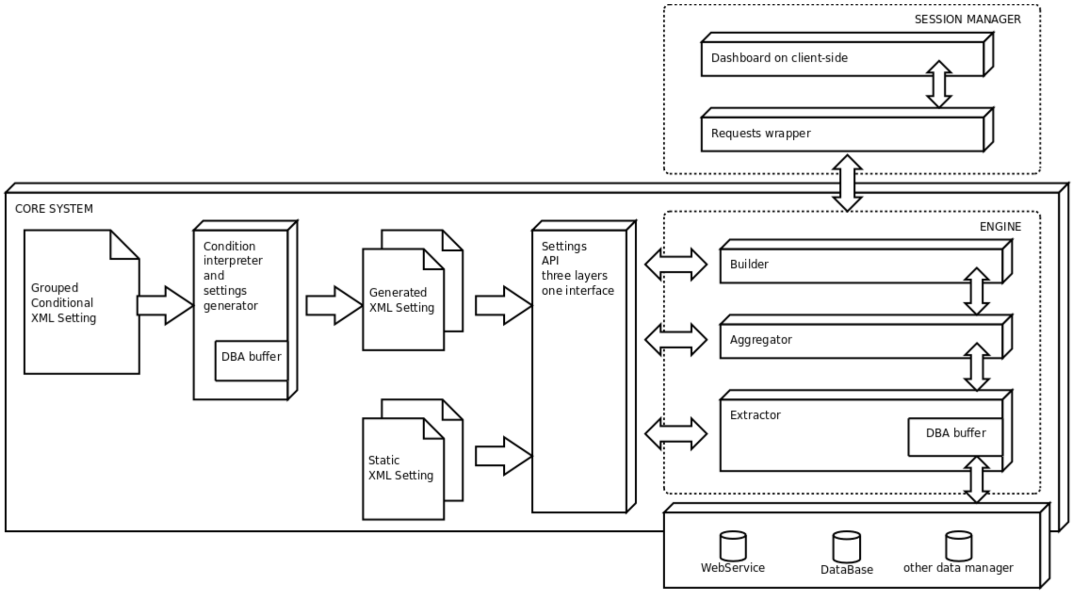


Figure 1: The adaptive architecture: the adaptation is produced in the two upper phases based on XML contextual rules driven by conditions on data already calculated.

2 The Infrastructure

Providing modalities for opening the profile to the user inspection is important in the domain of LLL: the presentation of indicators of the learning process is widely accepted as one of the key point to improve participation and increase the satisfaction of participants [9]. GVIS is an infrastructure we are developing, able to extract data from different sources and enable instructional designers to easily create adaptive indicators of the learning state for learners and tutors.

Although many Learning Management Systems already provide the possibility to explore the user tracking data, in some cases the visual presentation of the information is not well suited to the human perceptive system. In other cases, the presentation of data is limited to a subset of data or is predefined by developers and fixed. We want to provide an easy way to create effective graphical presentation of arbitrary data from different sources.

We propose a three-tier architecture composed by a data extractor, an aggregator of data, and a builder (see Fig. 1). All the levels rely on a configuration file that the administrator can change or expand in order to create graphical indicators of one or more interesting characteristics of the user profile, in form of widgets. An important aspect of this infrastructure is the possibility to connect to any data source with different connection types (databases, Web services, connection bus, ...), only by writing a small piece of adapter code. In the following subsections a description of modules is presented, with a particular attention to the adaptation capability provided.

2.1 The Extractor

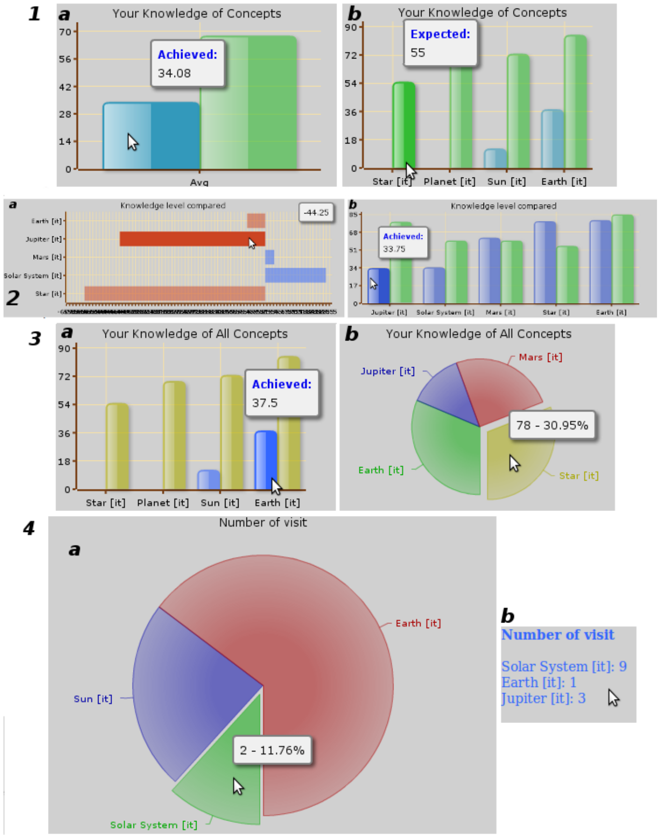
The extractor is the lower level and is in charge of retrieving data from the sources. This piece of software is in charge of making a syntactical and semantic translation of data received from a particular source to the internal format. To achieve this objective it relies on small amount of code that describes the data structure used by a particular source.

2.2 The Aggregator

The aggregator is in charge of filtering the raw data collected by the extractor and to apply some operations to create aggregated information. This aggregation of data is based on the didactical model that the teacher or instructional designer will provide, represents the most useful information for learner, and is strictly related to the pedagogical approach provided in the learning experience. The use of models (based on XML syntax) provides a formal way for designing the expected behavior of the aggregator module.

2.3 The Builder

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 the information into the final indicator graphical form. The configuration of the dashboard can be personalized based on some parameters set at a system level.



Number of visit

Solar System [it]: 9
Earth [it]: 1
Jupiter [it]: 3

Figure 2: The four widgets providing the kind of adaptations presented in this paper

2.4 Adaptivity

In the current implementation, the two upper layers (aggregator and builder) can be enhanced with adaptive features. The adaptivity is modeled in the configuration files through a simple XML schema. This schema supports the conditional construct IF: this allows the GVIZ visualization to have a different behavior with different properties. The properties can be any combination of source data values, on which a set of mathematical and logical operators can be applied. For instance, we can decide that a particular widget may show a comparison of the knowledge level of a student with the class, only if his current knowledge level is greater than a threshold value. Or we may want to show a particular widget only to the instructor of the course, not to the students. This is implemented by including conditional instructions in the XML configuration files of the aggregator and builder. To this end, the configuration files may contain variables, logical and arithmetical operators: we have implemented the common comparison operators (more than, less than, equal and different) and the logical operators AND, OR, XOR(exclusive or), NOT. The structure is composed of four main tags: *OP* contains the expression to be evaluated, *OPERANDS* contains the definition of every variable considered, *TRUE* is the branch that will be executed in case the value of the expression is true and *FALSE* branch otherwise. It is also possible to nest a new conditional tree inside one of the branches. The following example represents a condition that evaluates whether the list of concepts in course X is not empty and either the average knowledge of concept A is greater than 3 and there are no students subscribed to the course:

```
<cond>
 <op>(v1 AND ((A &gt; 3) OR !(z)))</op>
 <operands>
  <val id="v1">CourseX.Concepts.list</val>
  <val id="z">CourseX.Student.count</val>
  <val id="A">ConceptA.mean.knowledge</val>
 </operands>
 <true>...</true>
 <false>...</false>
</cond>
```

In the conditional expression we can put every variable of the user model, but also variables that represent user preferences and user device configurations.

3 A first application in TEL

We applied the software to data from different LMSes used in a controlled environment to support a LLL project. The following examples will show some possibilities of adaptive configuration. The adaptive behavior can be performed by the aggregator and the builder, and is driven by course data and/or user data. This data is collected by the extractor, or can be explicitly declared by the learner.

The first example, based on *user* data, (see Fig. 2.1), adapts at the aggregation level the number of concepts to display, in order to switch between a compact (image on the left) or detailed view (image on the right), depending on the number of concepts visited by the learner. The same for Fig. 2.2, where two views of the same information are presented: in the left part the difference between expected and achieved students' knowledge level is mapped into the length of bars, where in the right part their absolute values are presented with distinct bars. In Fig. 2.3, another example (based on *course* data) is presented, with adaptation condition included in the building layer. To optimize the readability of the widget, it shows the same information (that is concepts and their matching knowledge levels) in two different formats: a pie chart or a bar chart, depending on the number of concepts. In the last widget a builder adaptation to the type of user connection or declared hardware is provided, with a textual list well suited from mobile and handheld based platform or a graphical widget for larger displays and broadband access (see Fig. 2.4).

3.1 Expected impact

As showed by the literature, opening partially the user model to learners inspection is able to increase their trusting in the creation procedure and to leverage their spontaneous participation for the method calibration [4]. We expect, through the GVIS architecture to offer this functionality, mixing it with the possibility to write adaptive rules, in order to stress the process of adaptation, based on the didactic model used for the development of the course. As final output, we expect to offer an enhancement of the learner model [10], able to support all the exposed benefit, but reducing the cognitive overload that this new kind of information may create on users. A first evaluation, based on a questionnaire, was completed in a test case and the learners' feedbacks demonstrated that our tool could be potentially useful, even if some minor remarks about the actual implementation. In the next phases we are planning more structured evaluations, both in term of methodologies applied and of aspects investigated.

4 Conclusions

Our tool allows to aggregate information coming from different sources and to create adaptive graphical presentations of these data in order to support the visual human system [11]. In the context of Life Long Learning the presentation of these contextual information to the learners and to the teacher or tutors assumes a relevant importance in order to support better awareness of the learning situation and to promote participation [12]. The possibility to include adaptation in the generation of the widgets could help avoiding the informative overloading, that is a critical aspect in the learning context. The encoding of this information in graphical format is also important in order to make it useful for the learning process [9]. Some open issues still remain, such as providing a set of adaptation templates to be reused, and the availability of an editor for the configurations, in order to support the instructional designers' work. Also, some procedure for filtering and reordering the data through easy visual interfaces seems to be a future possible real enhancement. Last, but not least, we consider important to provide an evaluation of the impact of this approach on learning, both from the point of view of self-reflection and awareness, and of instructional effectiveness.

Acknowledgments

This work was partially supported by EU FP7 project GRAPPLE (215434)[13] The authors would like to thank all the project partners.

References

[1] McCalla, G. (2004). "The Ecological Approach to the Design of E-Learning Environments: Purpose-based Capture and Use of Information About Learners." In Journal of Interactive Media in Education, 2004 (7).

[2] Heath, T., Motta, E., Dzbor, M. (2005). "Uses of Contextual Information to Support Online Tasks." In Proceedings 1st AKT Doctoral Symposium, pages pp. 107-113, KMi, The Open University, Milton Keynes.

[3] Laurillard, D., Oliver, M., Wasson, B., Ulrich, H. (2009) "Implementing Technology-Enhanced Learning." In: Technology-Enhanced Learning: Principles and Products. SpringerLink, Netherlands, pp. 289-306.

[4] Bull, S (1997). "See Yourself Write: A Simple Student Model to Make Students Think". In Proceedings of 6th Intl Conference on User Modeling, 1997.

[5] Dror, B., Nadine, M., Mike, B., (2008). "Visualization and Analysis of Student Interaction in an Adaptive Exploratory Learning Environment". In Proceedings 1st Int. Workshop in Intelligent Support for Exploratory Environments, EC-TEL'08

[6] Chen, C. (2009). "Influence of Perceived Information Overload on Learning in Computer-Mediated Communication". In Proceedings of the 8th international Conference on Advances in Web Based Learning.

[7] Schutz, D. (2009). “Educational Intelligence: Bringing Schools into the 21st Century”. Available at SSRN: http://ssrn.com/abstract=1336243

[8] Romero, M., Tricot, A., and Marin, C. 2009. “Effects of a context awareness tool on students’ cognition of their team-mates learning time in a distance learning project activity.” In Proceedings of the 9th international Conference on CSCL

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

[10] Mazzola, L. and 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. 20-23/07/2009.

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

[12] Bull, S., Kay, J. (2008). “Metacognition and Open Learner Models.” In Metacognition Workshop, Intelligent Tutoring Systems 2008.

[13] De Bra, P., Pechenizkiy, M., van der Sluijs, K. & Smits, D. (2008). “GRAPPLE: Integrating Adaptive Learning into Learning Management Systems.” In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008 (pp. 5183-5188). Chesapeake, VA: AACE.