

GVIS: A Facility for Adaptively Mashing Up and Representing Open Learner Models

Luca Mazzola and 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
{luca.mazzola,riccardo.mazza}@usi.ch

Abstract. In this article we present an infrastructure for creating mash up and visual representations of the user profile that combine 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: data sources, data aggregations, and visualizations can be configured on the fly without changing any part of the software and have an adaptive behavior based on user's or system's characteristics. The visual profiles produced can have different graphical formats and can be bound to different data, automatically adapting to personal preferences, knowledge, and contexts. A first evaluation, conducted through a questionnaire, seems to be promising thanks to the perceived usefulness and the interest in the tool.

Keywords: Technology Enhanced Learning, Human Computer Interaction, Adaptive Presentations, Data Mashup, Open Learner Model.

1 Introduction

One of the key issues for making TEL an effective instrument for didactics is to increase the level of engagement of students with the online learning experience [1]. This is particularly important in the context of Life Long Learning (LLL), where the online learning is becoming more and more prevalent. To increase the level of engagement of learners, adaptive learning systems aim to guide learners, automatically adapting to personal preferences, prior knowledge, skills and competences, learning goals, and the personal or social context in which the experience takes place. One of the most important components of adaptive learning systems is the student model, which is in charge of keeping track of the student's knowledge and skills acquired during the learning process. In the LLL context, quite rarely learners use a single support tool, but 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 applications). Consequently, all user information is not stored only into a specific student model system, but it is often distributed in a number of platforms used for different purposes. For this reason, there is a need to

aggregate data from different tools (student models, intranet usage data, LMS data, and so on) and provide a uniform way to present these data to the interested users, preferably in visual format [2]. A specific multi-tier infrastructure was designed for this purposes. The idea was developed in the context of the GRAPPLE project [3], a research aimed at making already existing LMSes able to include adaptive contents and to share some controlled set of data about their users. Other approaches to the problem of creating personalized experience in TEL could be found under different names, such as “educational mashup” [4] or “ubiquitous and decentralized user model” [5] [6]. All these researches aim at aggregating user data from different systems, although they tackle the problem from different points of view. From the user point of view, we have to consider that a high quantity of mashed-up data might cause an overload issue [7], that becomes problematic when it distracts from the learning activity and makes the learner confused about the represented data. To limit the overload problem, the visual representations are adaptive to the role, to the context, and to the activities performed by the learner. The adaptation can help in creating more comprehensible and easier indicators. Next section briefly describes the implemented infrastructure. Then we present one adoption case, in the context of a real course and some very initial results of the evaluation we performed with learners of this course.

2 The Infrastructure

Providing modalities for opening the user profile for inspection [8] is important in the domain of LLL: the presentation of indicators of the learning process is widely accepted as one of the key points to improve participation and increase the satisfaction of participants [9]. GVIS - acronym for GRAPPLE Visualization Infrastructure Service - is an infrastructure we developed, able to extract data from different sources and enable instructional designers to easily create adaptive indicators of the learning state for learners and tutors. The user’s profile is normally created on the basis of the user activities and interactions in the learning environment. Adaptive systems maintain this information in the learner (or user) models, whilst regular (not adaptive) Learning Management Systems (LMS) keep this information in form of logs, or tracking data. In the GRAPPLE adaptive system, course contents are built with respect to conceptual domain and adaptation structures, based on *concepts*, and the learner model stores, among other things, the related *knowledge level*: the measure of learner’s knowledge on a specific concept. Although many LMS already provide the possibility to explore this user tracking data, in some cases the visual presentation of the information is not well suited to their users’ specific needs. There are notable exceptions to this situation in the field of OLM -like OLMlets [10], in which the learner can choose between seven different representations-, but they normally rely on data from a single system they were developed with. So, we want to provide an easy way to create effective graphical presentation of arbitrary data from different and heterogeneous sources.

2.1 Adaptivity

We propose a three-tier architecture composed by a data extractor, a data aggregator, and a builder. All the levels rely on a configuration file that the *instructional designer* can change or expand in order to create graphical indicators of one or more interesting characteristics of the user profile, in form of widgets. In the current implementation, the two upper layers (aggregator and builder) can be enhanced with adaptive features. The extractor layer retrieves raw data from the sources, then the aggregator is in charge of merging and filtering data to extract more refined information. This aggregation is based on the model that the instructional designer wants to provide to the learners and reflects the didactic approach adopted in the course. With such architecture, the support to the learning process, based on adaptive profile externalization, can be reached by adapting the visualization to the specific didactic model. The adaptivity is modeled in the configuration files for both cases through a simple XML schema. This schema supports the conditional construct IF: this allows the GVIS 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 learners. 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 following example represents a possible condition: *If the list of concepts in course X is not empty and either the average knowledge of concept A is greater than 3 or there are no students subscribed to the course, then display a particular widget, otherwise display another widget*. In the false case, we present another condition, related to the knowledge level of the current learner and compared with the average level for the course X. In this way, we can implement conditions as complex as needed and we can specify different behaviors at any granularity level. 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 Possible Application in TEL

We applied the software to data from different LMSes used in a controlled experiment to support a distance learning course. 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 (i.e. data not directly related to a single user, such as the number of concepts in a course) and/or user data (i.e. data about learner, like performed activities, acquired knowledge or expressed preferences). All these data are collected by the

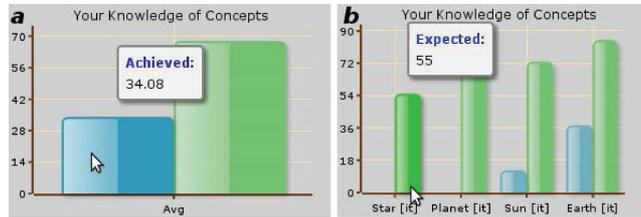


Fig. 1. Example of the produced widgets: different aggregation level of same data

extractor, or can be explicitly declared by the learner through his preferences and personal settings. The first example, based on *user* data (see Fig. 1), adapts at the aggregation level. It presents the knowledge achieved by a student over the concepts of the course. In Fig. 1 on the left there is a compact view, where the average knowledge level of the learner over the concepts of the course is compared to the average knowledge level of the class. The right side of Fig. 1, instead, presents detailed information for each concept of the course. The choice of which of these two widgets should be presented to the user is made by the GVIS engine on the basis of the number of concepts to display: if the number of concepts is too high to be represented in the detailed visualization, GVIS will present the aggregated one. It is important to notice that GVIS will only present data of concepts visited by the learner and this means that the adaptation rule can change the presented object automatically as the learner will visit more concepts of the course. The threshold according to which one of the two alternative visualizations is chosen could be either a fixed number in the XML configuration file or a value that is calculated on the basis of one or more characteristics of the class or activities done on the LMS.

Another visualization we developed, presented in Fig. 2, adapts the type of graph to the preferences of the user and to the number of concepts, in order to optimize the readability of the widget. It shows the information in two different formats: on the left, a bar chart represents the knowledge levels of a learner over the concepts of the course, comparing these values with the expected knowledge level; on the right, a pie chart gives a representation of the knowledge for every concept with their relative weight in the total knowledge achieved in the course. In Fig. 3, another example (based on *course* data) is presented, with an

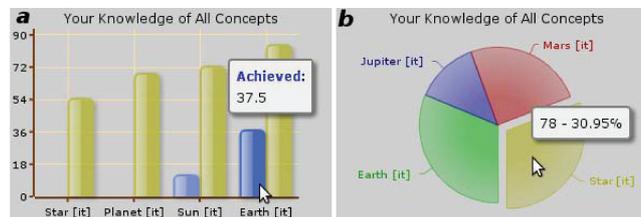


Fig. 2. Example of the produced widgets: different graphical representation of the same base data, aggregated in different way

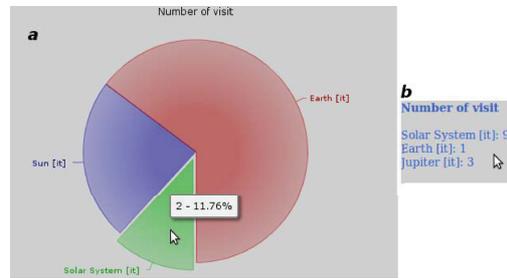


Fig. 3. Example of the produced widgets: graphical versus textual representation

adaptation condition included at the builder layer. In this case the adaptation is driven by the type of hardware used by the learner or by the connection speed. A textual list is well suited for mobile phone devices or handheld based platform, while a graphical widget (on the left) is suitable for larger displays and broadband access.

4 A First Test Case

In order to test the GVIS visualization module in a real environment, we set a testing platform with the Moodle LMS platform that includes a GVIS visualization module. We created a set of six indicators: two for the learners – very compact and seamlessly integrated into the Moodle users interface – and four for the tutor/teacher. This initial pilot phase was conducted applying the GVIS architecture to a Master blended course. Although in this test case we did not aggregate data from different sources, our visualization infrastructure helped to aggregate data coming from different tables in the Moodle database, and to visually represent contextual information about the course and the learners. We developed a specific widget in charge of representing the number of logins and the messages posted in a forum. An interesting outcome was the graphical comparison between the learner's specific information and the average value achieved by the class, that can work as a reference for the self-monitoring process of the user's progress. Some functionalities for the teacher have been implemented in order to support the tutoring, like the combination of accesses to resources of the course done by students and the number of messages posted to the forums. Others widgets were developed for this pilot study: the number of login to the course group by date and by student, and the number of forum post by date. We ran an initial evaluation that allowed learners to provide feedback on their experience with the system. A questionnaire with 16 questions on a 5 point based Likert-scale was developed and submitted at the end of the course with an online survey tool. For brevity, we do not include the questions and the analytical results, but we provide a qualitative description of the survey's results. The majority of the investigated aspects we collected reached at least an almost full satisfaction. Some aspects have different perception by the user and two of

them revealed to be problematic: the capability of the tool to leverage mental workload and the possibility that it hinders the collaboration among peers.

5 Conclusions

We designed a tool that allows to aggregate information coming from different sources and to create adaptive graphical presentations of these data in order to support the teaching and learning needs in learning environments. The possibility to include adaptation in the generation of widgets could help avoiding 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. 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. 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.

References

1. Laurillard, D., Oliver, M., Wasson, B., Ulrich, H.: Implementing Technology-Enhanced Learning. In: *Technology-Enhanced Learning: Principles and Products*, pp. 289–306. Springer, Netherlands (2009)
2. Dror, B., Nadine, M., Mike, B.: 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 2008* (2008)
3. Van Der Sluijs, K., Hover, K.: Integrating Adaptive Functionality in a LMS. *Int. Journal of Emerging Technologies in Learning (IJET)* 4(4) (2009)
4. Esposito, F., Licchelli, O., Semeraro, G.: Discovering Student Models in e-learning Systems. *Journal of Universal Computer Science* 10(1), 47–57 (2004)
5. Van Der Sluijs, K., Houben, G.: A generic component for exchanging user models between web-based systems. *International Journal of Continuing Engineering Education and Life-Long Learning* 16(1/2), 64–76 (2006)
6. Heckmann, D., Schwartz, T., Brandherm, B., Krner, A.: Decentralized User Modeling with UserML and GUMO. In: *Proceedings of the Workshop on Decentralized, Agent Based and Social Approaches to User Modelling (DASUM 2005)*, pp. 61–65 (2005)
7. Chen, C.: Influence of Perceived Information Overload on Learning in Computer-Mediated Communication. In: Spaniol, M., Li, Q., Klamma, R., Lau, R.W.H. (eds.) *ICWL 2009*. LNCS, vol. 5686, pp. 112–115. Springer, Heidelberg (2009)
8. Bull, S., Kay, J.: Metacognition and Open Learner Models. In: *Metacognition Workshop, Intelligent Tutoring Systems 2008* (2008)
9. Shahrour, G., Bull, S.: Does “Notice” Prompt Noticing? Raising Awareness in Language Learning with an Open Learner Model. In: Nejdil, W., Kay, J., Pu, P., Herder, E. (eds.) *AH 2008*. LNCS, vol. 5149, pp. 173–182. Springer, Heidelberg (2008)
10. Bull, S., Gardner, P., Ahmad, N., Ting, J., Clarke, B.: Use and Trust of Simple Independent Open Learner Models to Support Learning within and across Courses. In: Houben, G.-J., McCalla, G., Pianesi, F., Zancanaro, M. (eds.) *UMAP 2009*. LNCS, vol. 5535, pp. 42–53. Springer, Heidelberg (2009)