Learning analytics in the LMS: Using browser extensions to embed visualizations into a Learning Management System

Derick Leony, Abelardo Pardo, Luis de la Fuente Valentín, Iago Quiñones, and Carlos Delgado Kloos

Telematics Engineering Department,
University Carlos III of Madrid,
Av. Universidad 30, Leganés, Spain
{dleony,abel,lfuente}@it.uc3m.es,iagoqo@inv.it.uc3m.es,cdk@it.uc3m.es
http://gradient.it.uc3m.es

Abstract. The use of analytics has gained a relevant level of importance in current learning environments. Due to its life-cycle of observation, analysis and intervention, it is important to make learning analytics available to all the participants of a learning process. A Learning Management System (LMS) is a very suitable platform to present results in the form of visualizations, but the development of custom modules usually requires programming and administrative work. This paper presents a system architecture to embed learning analytics visualizations without having to develop a custom LMS module. Our approach is based on the modification of the LMS' web pages through a browser extension. The details of the implementation are presented, as well as issues that might arise and how to address them.

Keywords: learning analytics, browser extensions, LMS augmentation

1 Introduction

The life-cycle of a learning analytics process can be described in three stages: observation, analysis and intervention [1]. During the observation phase, actions performed by the learner are collected and stored; this data is filtered and processed. During the analysis phase, the data is analyzed to detect patterns that help define the current situation of each learner. Finally, the learner is intervened in order to improve her learning performance given her current situation. This paper focuses on the intervention stage of the cycle.

A common approach taken to intervene is the provision of awareness by displaying visualizations to the participants of the learning activity. Learners are provided with self-awareness of their performance and comparisons with the group performance. Meanwhile, instructors are provided with awareness of the learners performance at individual and collective levels. Among the examples of application of visualizations in learning contexts we can find CAMera [2], the goal-oriented visualizations of activity tool tracking [3], and CourseViz [4].

A Learning Management Systems (LMS) is a common channel of communication to all the participants of the course; this is a main reason to use them as the channel to display visualizations. The most common approach to achieve this is through the development of a custom module to be installed in the LMS. However, given that the LMS is commonly managed at an institutional level and it must support several courses, installing a customized module becomes a complicated procedure both technically and administratively.

In this paper, we propose an architecture to embed learning analytics visualizations and reports into a LMS without having to install modules in the LMS. This solution follows a client-server pattern and is based on a learning analytics service that provides the visualizations in combination with a web browser extension used by the participants of the course. Thus, we simplify the task of providing visualizations to participants of the course, valuable for the execution pilot studies or the evaluation of visualizations.

The rest of the article is structured as follows, Section 2 describes the architecture, its components and the communication among them. Section 3 explains an implementation based on this architecture. Finally, section 4 presents some points of discussion such as problems to address and future work regarding the architecture.

2 Description of the architecture

The architecture proposed involves two main components: a visualization that is dynamically generated by a learning analytics service and a web browser extension on the client side. Both components and the method to interconnect them are described as follows.

2.1 Learning analytics service

The analytics service is in charge of generating the visualizations that will be embedded in the LMS web page by the browser extension. The visualizations can be generated as a single image or as an independent web page depending on whether the visualization will be interactive.

A visualization might present different information to the same participant, depending on the context where the visualization is being shown. This requires the service to accept parameters that indicate the current context of the viewer. For instance, the visualization of a learner performance could be displayed to both instructor and learner, but the latter could be provided with a comparison of her performance and the performance of the whole class group. In other situations, such as a general dashboard, the learning analytics service provides information about the individual user only when the visualization is seen by a learner but not when seen by an instructor.

2.2 Behavior of the browser extension

The browser extension includes a special type of script known as a *content script*, which consists of a JavaScript code executed in the context of a page whose URL matches a given pattern. It must be noted that the URL pattern does not accept a query string; thus, the verification of the query string must be done within the logic of the content script.

In the architecture, there are two sites where the content script must be run: the course home page and the course participant list. The content script modifies the home page and the page that lists the course members. The structure and design of the elements that are included into these pages follow the style of the complete LMS in order to maintain it consistent.

The modification on the course home page consists of the inclusion of an element commonly known as widget, module or portlet. This element is shown to every participant of the course, be it a learner or an instructor. The content presented is a visualization obtained from the analytics service which will be personalized for each viewer. Because of this personalization, the content script must obtain a user identifier within the content of the page, and usually available at explicit links referring to the user profile or session status.

In the list of participants the content script adds a visualization element next to each learner's portrait or full name. In order to maintain the privacy of the learners' academic information, the list of participants should be modified only if the viewer is an instructor. Thus, the content script needs to be aware of the role that the viewer plays within the course which can be achieved through mechanisms like analyzing the options available on the web page or fetching the content of the viewer's profile and analyzing the user description. As in the home page modification, the user identifier for each learner can be obtained from a link pointing to her user profile in the LMS.

2.3 Communication between extension and service

Visualizations must be available for the browser extension to embed them into the web page of the LMS when seen by a course participant. This is usually achieved by offering the visualizations through a web server and requiring authentication in order to respect the privacy of the learner data.

The communication process follows a client-server architecture, where the browser extension plays the role of the client and the visualization service is the server. The extension initiates the communication process by requesting the appropriate visualization given the current context in the LMS, which includes the web page and the user's current profile. The request includes information about the identity of the user and current context.

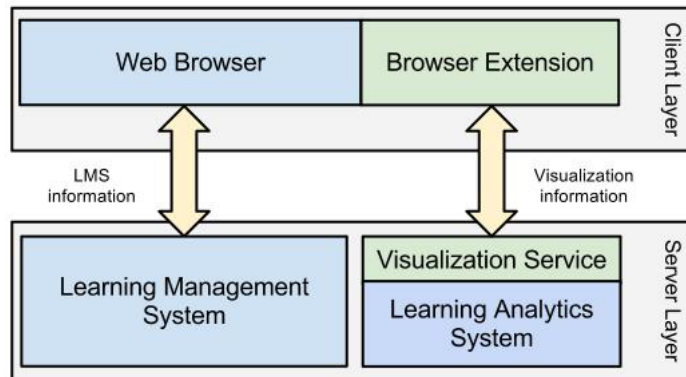The proposed architecture and the exchange of information among its components are represented in Fig. 1.

Fig. 1. Architecture for embedding visualizations into a Learning Management System through a browser extension. The elements we introduce are colored in green.

3 Implementing the architecture

In our scenario, the LMS where the visualizations are embedded is Moodle and the course participants use Google Chrome and Mozzilla Firefox as their web browsers. The visualization service is provided by LearnGLASS [5], which is a learning analytics system that supports the development of visualizations on top of a common learning dataset. LearnGLASS follows a modular architecture, it is in a specific visualization module that the visualization service was developed and published. This specific part of the module does not require to have initiated a session in GLASS, since it will be accessible from Moodle.

Regarding the browser extensions, we have used Kango Extensions [6] to develop extensions for both Firefox and Chrome. This framework allows the generation of browser-specific extensions based on a common base code.

Although the web page that generates the visualization does not require a session, it does require an authentication token as a parameter. The token is generated by a hash function of the user identifier, the current timestamp and a password known by both client and server. Fig. 2 presents a screen capture of the home page of a Moodle course as seen by a student; the red selection shows the visualization that has been inserted.

4 Discussion

Using a decentralized architecture also implies the appearance of problems that need to be addressed such as a fragmentation of client software, dependency on structure, account association and security.

User identity issues include the unification of user accounts in the two systems, the LMS and the visualization service. These problems are explained with

Fig. 2. Home page of a Moodle course as seen by a learner. The visualization embedded by our solution is highlighted in red.

more detail in [7] where the possible solutions are also included. For this implementation we have previously imported the Moodle user identifiers into GLASS database and mapped each Moodle account to its corresponding GLASS account.

A second possible issue to take into account is that by using a client-server architecture, users might have different versions of the client. This problem is approached in our implementation through the provision of an update URL. This URL is accessed by the web browser to check whether there is a new version of the extension and, when this is the case, the user has the option to upgrade.

By modifying the structure of the LMS' web pages, the correct behaviour of the presented architecture depends on keeping the same structure and visual design of the web pages. This issue is addressed through a constant monitoring of the LMS look-and-feel and the appropriate modification of the extension to adapt to change. Another mechanism to consider is the development of an alert system in the extension that informs the visualization service about any mismatch in the structure of the page.

One of the main disadvantages of the development of a client application with JavaScript is the availability and openness of its source code. Thus, the development of an authentication protocol that is password-protected becomes difficult. A possible way to overcome this problem is through the implementation of the OAuth protocol to allow the browser extension to fetch information from the visualization service. This option presents the disadvantage of requiring an initial intervention by the user.

Regarding the proposed future work, we are currently working on the deployment of a learning analytics scenario that is supported by the presented architecture. This deployment has many objectives: evaluate its viability, analyze the reaction of users to the presentation of analytics, and analyze whether

the effects of embedded visualizations in the LMS during the learning process of the participants.

Finally, another line of technical work is the generalization of the browser extension. Currently, the extension is dependant on the structure of the pages in the LMS and on the visualization service specification. By abstracting both of these characteristics, we would improve the modularity and re-usability of the extension.

Acknowledgment

References

1. Elias, T.: Learning Analytics : Definitions, Processes and Potential. http://learninganalytics.net/LearningAnalyticsDefinitionsProcessesPotential.pdf (2011) Last accessed: August 2011.
2. Schmitz, H., Scheffel, M., Friedrich, M., Jahn, M., Niemann, K., Wolpers, M.: Camera for ple. Learning in the Synergy of Multiple Disciplines (2009) 507–520
3. Santos, J., Govaerts, S., Verbert, K., Duval, E.: Goal-oriented visualizations of activity tracking: a case study with engineering students. In: Proceedings of the 2nd International Conference on Learning Analytics and Knowledge, ACM (2012) 143–152
4. Mazza, R., Dimitrova, V.: CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses. International Journal of Human-Computer Studies **65**(2) (February 2007) 125–139
5. Leony, D., Pardo, A., de la Fuente Valentín, L., Sánchez de Castro, D., Delgado Kloos, C.: GLASS: a learning analytics visualization tool. In: Proceedings of the 2nd International Conference on Learning Analytics and Knowledge, ACM (2012) 162–163
6. KangoExtensions: Kango Cross-browser extension framework. http://www.kangoextensions.com (2012) Last accessed: July 2012.
7. de la Fuente Valentín, L., Leony, D., Pardo, A., Kloos, C.: User identity issues in mashups for learning experiences using ims learning design. International Journal of Technology Enhanced Learning **3**(1) (2011) 80–92