Adding mash-up based tailorability to VLEs for scripted Collaborative Learning

Juan I. Asensio-Pérez[1], Miguel L. Bote-Lorenzo[1], Guillermo Vega-Gorgojo[1], Yannis A. Dimitriadis[1], Eduardo Gómez-Sánchez[1], Eloy D. Villasclaras-Fernández[1]

[1] School of Telecom. Eng., Univ. of Valladolid, C. del Cementerio s/n, 47011 Valladolid, Spain
{juaase@tel,migbot@tel,guiveg@tel,edugom@tel,yannis@tel,evilfer@ulises.tel}.uva.es

1 Introduction

The use of Virtual Learning Environments (VLEs) is gaining momentum in education. However, most widely spread VLEs such as Moodle, Sakai or .LRN miss two important features for the support of *collaborative learning* [1] thus precluding their adoption for putting into practice this pedagogical approach that is considered to be more effective than individual and competitive learning under many circumstances. First, they are not *tailorable* [2]. In this way, educators cannot incorporate existing tools in the VLEs in order to enable their use for the support of new collaborative learning scenarios. And second, they do not support *scripting* [3]. This precludes the possibility of enhancing the effectiveness of collaborative learning by structuring the interactions between learners in the VLEs.

Gridcole [4] is a VLE proposed by the authors that overcomes these limitations by combining the IMS-LD[1] specification and service-oriented technologies. More specifically, Gridcole allows educators to easily integrate service-based tools provided by third parties for the support of scripted collaborative learning scenarios described with IMS-LD. Nevertheless, two issues have been identified that may discourage the adoption of Gridcole. One of them is that Gridcole relies on the availability of a sufficient set of tools developed by third parties and compliant with the technological requirements posed for their integration. Gridcole requires third-party tools to follow WSRF[2] standards that, unfortunately, are not yet commonly adopted for the development of tools with potential learning uses. The other issue refers to the fact that Gridcole was conceived to replace other VLEs. However, users (teachers, learners) may be reluctant to adopt a new VLE once they have got used to an existing VLE, despite expected benefits.

This work-in-progress paper proposes an extension for already-existing VLEs in order to add support for tailorability and scripting in collaborative learning. In this proposal, tailorability is achieved by mash-up integration [5] of third-party tools including web-based applications such as Google gadgets or Youtube-like video players. Those third-party tools also follow the principles of service-orientation advocated by Gridcole, as they provide functionality supported by third-party computational resources typically accessed through a custom-made lightweight client, but are based on technologies much more widely accepted. This opens the possibility of integrating in VLEs a large number of tools that are already available and that can

[1] IMS Learning Design. http://www.imsglobal.org/learningdesign/

[2] Web Services Resource Framework. http://www.oasis-open.org/committees/wsrf/

be eventually employed for the realization of many learning scenarios. In addition, the extension is aware of the activities, tools and groups defined in IMS-LD based scripts so that VLEs can rely on this information for the support of collaborative scripted learning. The aforementioned features should improve the collaborative learning support of current VLEs by enabling the integration and proper configuration of the tools required for effective interactions among learners in collaborative scenarios.

2 Proposed extension and prototype

Fig. 1 depicts the components of the proposed extension as well as the interactions that support the main use cases. Grey-filled blocks represent already existing systems (VLE, authoring tools, and third-party tools). The main element of the proposed extension is the Integration Manager that acts as an intermediary among the other blocks. As the extension intends to be valid for a wide range of existing VLEs and script authoring tools, it relies on "adaptors" that cope with the communication and data formats particularities of each of them.

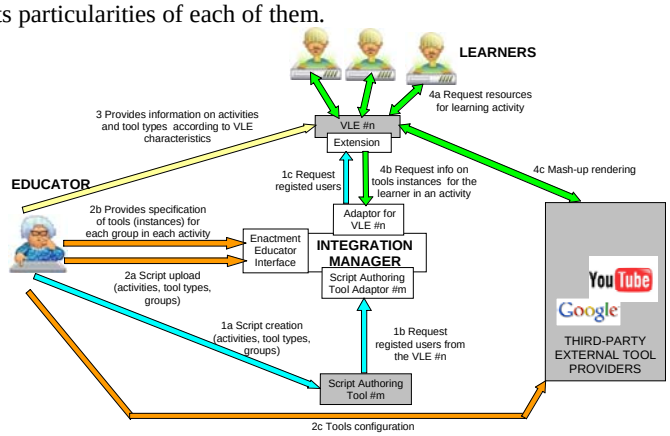

Figure 1. Blocks of the proposed extension and main interactions.

In the "Script Authoring" use case, educators describe their learning scenarios using an IMS-LD script authoring tool such as Collage[3] (interaction 1a), defining the sequence of activities to be performed, the tool types required for each of them, and the groups of learners that will participate (typically different groups for different activities). In order to facilitate this authoring process, the authoring tools may request the Integration Manager to download from the VLE data on the currently registered learners (interactions 1b and 1c).

Once the script is created, the educator performs the "Mash-up Configuration" use case. Using the Enactment Educator Interface of the Integration Manager, the educator uploads the created script (interaction 2a). Next, the educator indicates the precise tool instances that will be employed by each participant or group in each activity (interaction 2b). Additionally, the educator could need to interact with the providers of particular tool instances so as to configure them (interaction 2c).

[3] http://gsic.tel.uva.es/collage

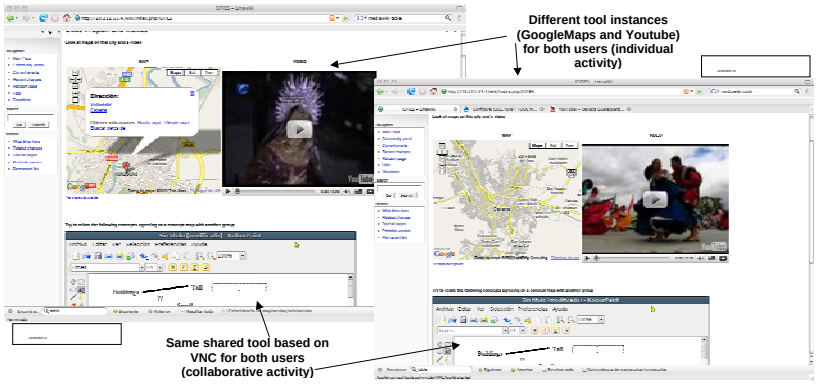

Figure 2. MediaWiki renders different mash-ups according to Integration Manager instructions.

The next use case, "VLE Configuration", would require the educator to interact with the VLE in order to provide set-up information such as which script should be requested to the Integration Manager, how to "link" VLE pages interface with script activities or how to render subsequent tools mash-ups (interaction 3). These interactions are heavily VLE-dependant and, in some cases, they would not even be needed if the VLE adaptor for the Integration Manager is capable of carrying them out automatically.

Finally, the "Enactment" use case involves participants joining the situation through the VLE (interaction 4a) and the VLE extension asking the Integration Manager to obtain the set of tool instances required for each activity (interaction 4b). This way, tailorability is achieved through mash-up integration of external tools that actually run on the providers' environment, while required client code for tool rendering is embedded in the VLE (interaction 4c).

A preliminary prototype has already been implemented to assess the feasibility of this proposal. MediaWiki, which was selected as the target VLE due to its simplicity, has been extended to communicate with the Integration Manager. The extension is based on MediaWiki "tag extensions" that define new wiki tags so that the educator can request the addition of desired tool types to wiki pages. The Integration Manager, implemented as a PHP module of the Drupal Content Management System, is capable of interpreting IMS-LD scripts created with the Collage authoring tool to extract the information on activities and associated tool types. Additionally, it can parse information on groups of learners created for each of the script activities provided by Collage. Furthermore, this prototype enables the integration of web-based tools such as Google gadgets or YouTube-like video players, as well as third-party standalone applications that can be shared by a group of participants via Virtual Network Computing (VNC) technology. For integrating those tools, the educator simply has to "copy and paste" the HTML code or URL indicated by the tool provider that enables the download of the tool client during the realization of the scenario.

To illustrate the functioning of this prototype, Fig. 2 shows the MediaWiki interfaces of two participants in a learning situation. Remarkably, MediaWiki shows a customized mash-up for each participant, based on the information obtained from the Integration Manager although the wiki page edited by the teacher is the same for all of them. The Integration Manager thus makes MediaWiki aware of groups, a crucial

aspect for the support of collaborative learning scenarios, and enables the educator to create a single mash-up template (in this case the edited wiki page) for all participants that is filled with appropriate tool instances (known by the Integration Manager) during the realization.

3 Discussion

VLE tailorability based on the integration of service-oriented tools is not a new concept. Nevertheless, this paper has explored a new way of achieving tailorability by using increasingly accepted mash-up techniques. This new scenario could benefit from the availability of a larger set of tools and implies an interesting shift from previous proposals (such as Gridcole) regarding the relationship among VLEs and tool providers. Using mash-ups, VLEs adapt themselves to the technology and capabilities, in terms of integration, of existing tools. No additional requirements are posed to the tool provider by the VLE (actually, there is no communication among the Integration Manager and tool providers in Fig. 1). But in some cases, this integration model could not be enough. For instance, [6] proposes the integration of third-party tools (widgets) by uploading then into an ad-hoc container that enables the easy creation by educators of tool instances according to the number of groups involved in a learning activity. However that approach poses new requirements to tool providers thus eventually resulting in a possible much smaller set of available third-party tools. Therefore, this trade-off between expected tools capabilities (more requirements on the tool-provider side) and the number of available tools (less requirements on the tool-provider side) is still an open and important issue. The authors consider that VLE tailorability should rely on different types of tool-providers (in terms of the number of requirements of the "contracts" among them and VLEs). But also that the proper choice of those types of tools needs further research on their implications on script design processes as well as on other Computer-Supported Collaborative Learning (CSCL) aspects not considered in this paper: interaction analysis, tool searching, flow of learning artifacts among tools, etc.

References

1. Dillenbourg, P.: Collaborative Learning: Cognitive and Computational Approaches. Elsevier Science, Oxford, UK (1999)
2. Morch, A.: Three Levels of End-User Tailoring: Customization, Integration and Extension. The 3rd Decennial Aarhus Conference, Aarhus, Denmark (1995) 41-45
3. Dillenbourg, P.: Over-Scripting CSCL: the Risks of Blending Collaborative Learning With Instructional Design. In: Kirschner, P. A. (eds.): Three Worlds of CSCL. Can We Support CSCL. Heerlen, Open Universiteit Nederland (2002) 61-91
4. Bote-Lorenzo, M.L., Gómez-Sánchez, E., Vega-Gorgojo, G., Dimitriadis, Y.A., Asensio-Pérez, J.I., Jorrín-Abellán, I.M.: Gridcole: A tailorable grid service based system that supports scripted collaborative learning. Computers & Education. 51 (1) (2008) 155-172
5. Severance, C., Hardin, J., Whyte, A.: The coming functionality mash-up in Personal Learning Environments. Interactive Learning Environments. 16 (1) (2008) 47-62
6. Wilson, S. Sharples, P., Griffiths, D.: Extending IMS Learning Design services using Widgets: Initial findings and proposed architecture. Open Workshop on Current research on IMS-LD and Lifelong Competence Development Infrastructures, Barcelona, Spain, 2007.