Integration of external tools with GLUE! in LAMS: requirements, implementation and a case study.

Carlos Alario-Hoyos, Miguel L. Bote-Lorenzo, Eduardo Gómez-Sánchez, David A. Velasco-Villanueva, Juan I. Asensio-Pérez, Guillermo Vega-Gorgojo and Adolfo Ruiz-Calleja
School of Telecommunication Engineering,
University of Valladolid, Spain

LAMS is a well-known learning platform that enables the design, enactment and realization of sequenced collaborative learning activities. However, LAMS lessons are limited by a small set of built-in tools. Few works have so far tried to add new tools to LAMS, mainly due to the high development effort required. GLUE! (Group Learning Uniform Environment) is an architecture that could overcome this limitation, since it enables the integration of multiple existing external tools in multiple existing learning platforms with a low development effort. This paper discusses the requirements and decisions taken in the design and development of a GLUE! adapter for LAMS. This adapter enables a seamless integration of third-party tools, supporting also the main LAMS distinctive features, namely: the monitoring of students' performance, the creation of learning pathways and the sharing of learning designs. Evidences of the appropriateness of the approach have been gathered by means of a case study that includes the design and enactment of a lesson that requires third-party tools. This lesson has already been tested by the authors and will be realized by real students during this semester.

Keywords: LAMS, GLUE!, integration, collaboration.

Introduction

Virtual Learning Environments (VLEs), sometimes referred as Learning Management Systems, or simply learning platforms, are widespread software systems to support teaching and learning in education (Weller 2007). Outstanding examples of VLEs are LAMS, Moodle, Blackboard or Sakai. VLEs offer a shared customizable learning space that enables the design, enactment and realization of different learning situations, which are supported by a set of resources and tools, in order to promote the communication and collaboration among participants (Catherall 2005). Nevertheless, the set of built-in tools included in VLEs is considered small and limited by many practitioners (Bower et al. 2011). In this context, the integration of third-party existing tools in VLEs aims at offering educators a larger and more flexible set of tools for the design and enactment of their learning activities (Fuente-Valentin et al. 2011).

Unfortunately, in the case of LAMS, very few external tools have been integrated so far. A lower institutional adoption, as compared to Moodle or Blackboard, and a more complex integration contract (Ghiglione et al. 2006) may be the reasons that have discouraged third-party contributions in LAMS. One of the few integrated tools found in the literature is <e-Adventure> (Del Blanco et al. 2010), an authoring platform for the design of educational games that was included in the recent LAMS 2.3.5 release. The integration of additional external tools would enable a wider range of learning activities in a LAMS lesson, which is currently limited by about twenty-five built-in tools.

In contrast, many third-party developers have integrated external tools in other VLEs like Moodle or Blackboard, by means of rather *ad hoc* extensions, called Moodle Modules and Blackboard Building Blocks, respectively. This may promote the interest of institutions and practitioners in using these two VLEs for their courses. Nevertheless, the effort required to implement each of these extensions is normally very demanding, and the code or even the acquired experience can be hardly reused when integrating the same tool in LAMS or in other VLEs, mainly due to the existing heterogeneity, regarding technologies, programming languages, or interfaces.

In the last few years several works have tackled the generic integration of third-party tools in multiple VLEs. A well-known example is the IMS Learning Tool Interoperability (http://www.imsglobal.org/lti) specification, although its tight integration model, which demanded a high development effort, discouraged most VLE and tool providers from its adoption. Three generic loosely-coupled integration approaches have been recently proposed with the aim of reducing that development effort, thus facilitating the integration of multiple tools in multiple VLEs. First, the Apache Wookie architecture (Wilson et al. 2008) enables the integration of tools developed as W3C widgets (http://www.w3.org/TR/widgets) in different learning platforms. A plug-in that integrates W3C widgets in LAMS following this architecture is reported, but it has not been included in any LAMS release. Nonetheless, the simple functionality offered by those widgets and the few number of educational tools that currently meet the W3C specification still limit the lessons that can be designed and enacted in LAMS. Second, Basic LTI (http://www.imsglobal.org/lti) is a specification that enables a simple launching of external tools that provide a compliant endpoint. Despite its relative success, Basic LTI has not been adopted by LAMS providers. Maybe Basic LTI lack of support to groupwork and collaboration, which are recurrent strategies to promote a deeper learning (Dillenbourg et al. 1995), as well as the main advantage afforded by LAMS according to teachers' perceptions in Bower's study (Bower et al. 2011), discouraged LAMS providers to meet this specification.

In order to overcome the limitations of Apache Wookie and Basic LTI, the authors proposed GLUE! (Alario et al. 2010), a loosely-coupled middleware architecture that imposes only basic requirements that most VLEs and tools already meet, and supports enough functionality for the design, enactment and realization of collaborative learning situations that require the use of external tools. The GLUE! implementation is continuously growing, and it currently integrates more than twenty external tools in Moodle and MediaWiki. GLUE! could easily enable the integration of third-party tools in LAMS by implementing a software component that adapts the technologies, interfaces and data models defined in the LAMS contract (Ghiglione et al. 2006) and in the GLUE! contract (Alario et al. 2011). This adapter would automatically enable the use of the available GLUE! external tools, such as Google Docs, Dabbleboard or Doodle, in LAMS. Nevertheless, the integration of third-party tools should be able to exploit those LAMS distinctive features that are most appreciated by educational practitioners. Apart from groupwork, these features are (Bower et al. 2011): the monitoring and intervention of educators during students' performance; and the creation of sequences of activities with learning paths and branches. Besides, the reusability and sharing of learning designs in different courses, as an outstanding LAMS principle, should be also considered.

In this context, this paper contributes with the process of designing and implementing a GLUE! adapter for LAMS, including a discussion of the significant requirements that affect the decisions taken in this process. After implementing this adapter, a collaborative learning situation that was realized in Moodle with real educators and students is enacted in LAMS, to show that GLUE! enables: the integration of third-party external tools in LAMS; and the reuse of conceptual learning designs among VLEs, supporting the positive and distinctive LAMS features regarding monitoring and learning paths, as well as group management and collaboration.

The remainder of this paper proceeds as follows. Next section briefly describes the GLUE! architecture, emphasizing the process of designing and implementing VLE adapters. After that, the requirements for the development of a LAMS adapter are identified. Significantly, some of these requirements come from the GLUE! contract, while some others are a consequence of the distinctive LAMS features. In the following section, implementation details are reported, just before the description of the enactment of the collaborative learning situation used as a case study. Finally, the conclusions of the paper are remarked.

An overview of the GLUE! Architecture

GLUE! is a three-tier loosely-coupled architecture that promotes the many-to-many integration of third-party external tools in VLEs with a low development effort, whose integration contract imposes few technical requirements to the VLEs and tools to integrate, and whose functionality supports the design, enactment and realization of collaborative learning situations within VLEs (Alario et al. 2010). GLUE! employs the adapter pattern (Gamma et al. 1995) to enable the communication between VLEs and tools through a centralized software element called GLUElet Manager, in charge of supporting the life cycle of external tools. This life cycle includes the creation and configuration, update, retrieval and deletion of tool instances, and their assignment to VLE users according to their roles and the group they belong to. The information needed for the communication with integrated tools along with administrative data is stored in the internal tool registry. It is noteworthy that, since the GLUElet Manager implements most functional commonalities required for supporting collaboration, the development of VLE and tool adapters is simplified. Figure 1 shows the current GLUE! implementation, which includes two VLE adapters for Moodle and MediaWiki, and eight tool adapters with representative examples of tools.

In order to integrate these tools in LAMS, a VLE adapter that meets both the LAMS contract and the GLUE! contract must be developed, as it is represented with the striped shape. These contracts establish the requirements, interfaces and data models that must be met by VLE adapters. In the case of GLUE!, the contract was defined with the aim of imposing few requirements (as compared to most integration approaches) based on widespread standards (Alario et al. 2011), in order to promote the integration of a wide range of tools and VLEs. Here, the more demanding challenge is the graphical integration of external tools, so that they look like built-in VLE tools. GLUE! facilitates this graphical integration through the use of URLs that can be easily embedded in the VLE web user interface. Once the VLE adapter is developed, tools registered in the internal tool registry become automatically available in the design and enactment of LAMS lessons. Next section focuses on the high level requirements that must be considered in the development of the LAMS adapter. For further details on low level requirements the LAMS and GLUE! contracts should be checked.

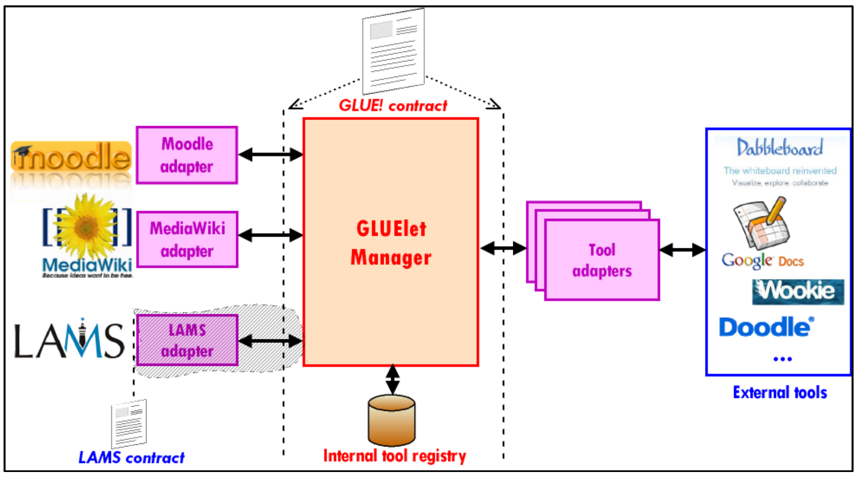


Figure 1: Current implementation of the GLUE! architecture

Requirements for the integration of external tools with GLUE! in LAMS

The high level requirements that must be taken into account when designing and implementing a GLUE! adapter for LAMS are a consequence of the functionality supported by both the LAMS and GLUE! contracts. The functionality in GLUE! is oriented to support the aforementioned tool life cycle, while the functionality in LAMS addresses its specific management of learning designs, which can be very different from that in other VLEs. For example, whereas Moodle encourages a more open, unstructured design of the courses, allowing educators to refine and iterate on a learning design as the course is being delivered, LAMS advocates for completing a sequenced lesson in the authoring environment, before its deployment and execution in the monitoring environment (Bower et al. 2011). This philosophy allows a clear division of educators' tasks in two roles, author and monitor, and thus involves a different set of steps to control the life cycle of external tools within LAMS, as further discussed in this section. Besides, this VLE is influenced by Educational Modeling Languages, and specially IMS LD (http://www.imsglobal.org/ learningdesign), thus giving great importance to sequencing activities, sharing and reusing learning designs, and tracking students' work. These are considered the main distinctive LAMS features (Bower et al. 2011), and so, they must be supported by the LAMS adapter.

The design decisions aimed at meeting these requirements are discussed next. For the sake of clarity they are presented in two sets: requirements related to the life cycle of external tools (including their setting and their location within a LAMS lesson), which are common to the implementation of any other VLE adapter; and requirements related to the main distinctive LAMS features (including the monitoring of students' work, the creation of learning sequences, and the reuse and sharing of lessons), which are specific of this VLE.

Requirements related to the life cycle of external tools

The life cycle supported by GLUE! enables the selection of external tools, as well as the creation, update and deletion of configured instances. This subsection discusses how the tool life cycle described in the GLUE! contract fits in the LAMS management of built-in tools during the design and enactment of a LAMS lesson.

Selection and configuration of tools

LAMS built-in tool are represented as activities (e.g. chat or forum) in the authoring environment. These activities allow some initial settings that can be modified by authors and monitors until the activity starts. The LAMS adapter should graphically represent GLUE! as any other LAMS activity in the authoring environment, including in its basic settings forms for the selection and the initial configuration of external tools. This configuration could be changed until learners run the activity, as it happens with built-in tools.

Creation and deletion of tool instances

LAMS built-in tool instances are created once a lesson is deployed and the groups are particularized in the monitoring environment, and are deleted when the lesson is removed from the course. Similarly, the LAMS adapter should automatically request GLUE! the creation of external instances when deploying a lesson. Besides, removing a lesson should entail the automatic deletion of its external tool instances. Remarkably, in both cases, the number of groups in each activity determines the amount of instances that must be created, since each instance is shared among the participants belonging to the same group.

Management of groups

Educators populate groups in LAMS after deploying a lesson in the monitoring environment. These groups can be later modified to include latecomers, to exclude no-shows or to balance the workload on the fly. LAMS is aware of these changes, granting access to built-in tools only to those users belonging to the same group. The GLUE! contract supports the automatic assignment of tool instances to VLE users depending on their groups, as well as later updates about the members of each group. Therefore, the LAMS adapter should send GLUE! the list of users belonging to the same group when requesting the creation of external tool instances, as well as react and inform about further changes. Besides, and as it happens in built-in tools, groups could be modified by adding new members at any moment during a lesson, even in those activities that are running or have finished

Requirements related to the main distinctive LAMS features

The requirements considered in the design and development of the LAMS adapter do not only stem from the management of the tool life cycle, but also from the specific interactions between LAMS and its built-in tools, in order to support LAMS main distinctive features. This subsection discusses how the LAMS adapter should be designed and implemented to support these features when integrating external tools through the GLUE! architecture.

Monitoring students' performance

LAMS enables educators to monitor students' work in the monitoring environment, including also some tracking features like time charts or sequence status. The GLUE! contract requires information about the VLE users sharing each tool instance, which may include the full course list or just the students of a certain group. However, educators or monitors that assess students' work can also be included. Therefore, the LAMS adapter should be responsible for including the educators of a course as extra users in every tool instance. Once instances are created, educators could visualize their content and annotate them, thus monitoring students' performance. The remaining LAMS functionality for monitoring needs no further explanation, since it can be achieved just by following the LAMS contract.

Use of learning sequences and pathways

LAMS enables the creation of sequences of activities in the authoring environment. Nevertheless, each external tool and instance is independently managed by GLUE!. The LAMS adapter should take advantage of this independence to generate isolated activities that include external tools, and that may become sequences through the LAMS own authoring logic for the management of transitions and branches. These sequences could be easily supported by GLUE! if students or educators manually decide when to move forward or backward, as it happens in the LAMS monitoring environment when students click on the "Next Activity" button, or return to previous activities using the LAMS activity diagram, or when educators control the learning flow by including stop signs. Similarly, branching could be easily supported if decisions are based on "teacher's choice" or "group-based". Nevertheless, GLUE! loosely-coupling precludes from using decisions based on "learner's output", since the architecture does not implement any mechanisms to retrieve outcomes from external tools yet.

Management of learning designs
LAMS was designed to facilitate the import/export of learning designs and their sharing among educators. Therefore, the LAMS adapter should comply with the LAMS contract to enable the reuse of the same learning design in different courses. Designs including external tools could be imported and deployed in the same LAMS installation as usual. The same applies to other LAMS installations, provided that they are connected to the same GLUElet Manager, or at least with other GLUE! installation that integrates all the external tools included in the learning design.

Implementation of a GLUE! adapter for LAMS

The authors have developed a GLUE! adapter for the LAMS 2.3 version, taking into account the requirements and design decisions discussed in the previous section. It is noteworthy that this is just a reference implementation and other alternative implementations are feasible. This adapter supports the life cycle of external tools, thus enabling the design, enactment and realization of collaborative learning situations, and the main distinctive LAMS features to monitor students' work, define learning pathways and reuse learning designs. The usage of this adapter is explained next, with two representative screenshots taken from the design and enactment of a learning situation that will be later discussed.

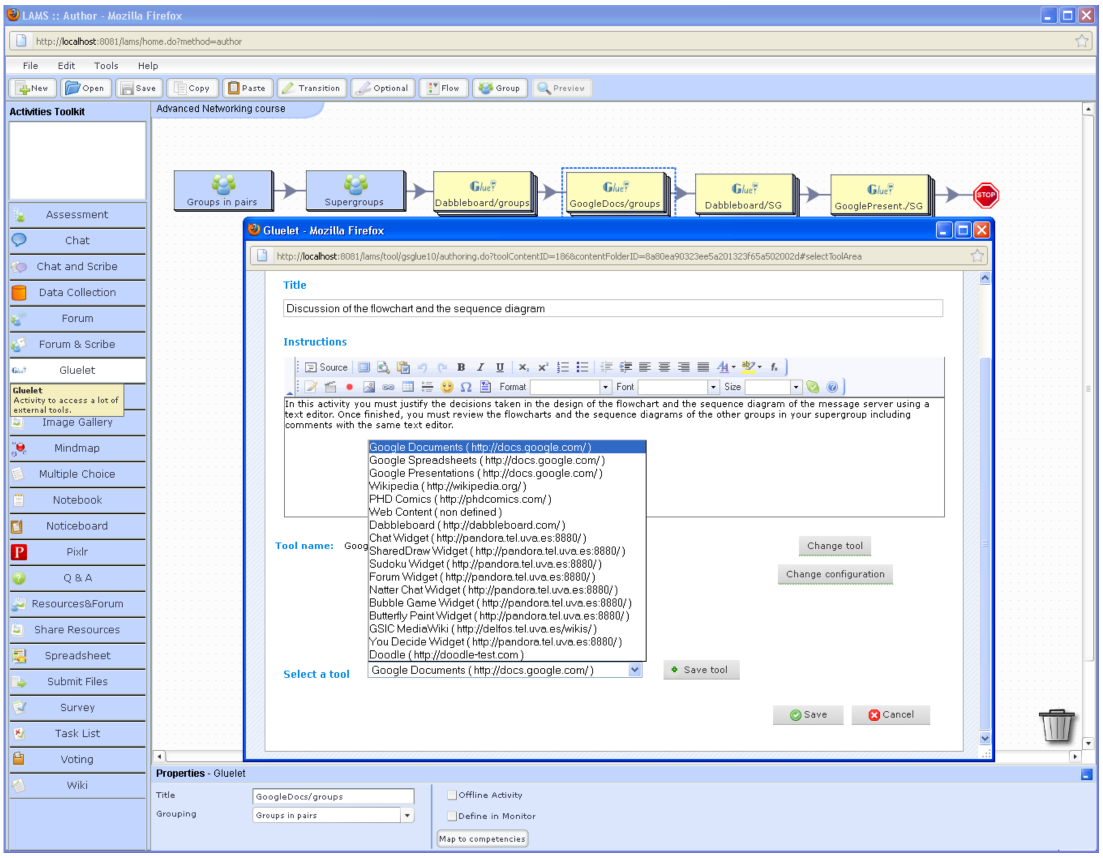
Figure 2 is a screenshot representing the design of a lesson in the LAMS authoring environment. A new "gluelet" icon that enables the use of external tools is available on the left side, as another LAMS activity. The learning sequence includes four activities, three external tools (Dabbleboard, Google Documents and Google Presentations) and two different groupings. The basic settings of a "gluelet" activity are also depicted in the figure. These settings are the title and the description of the activity, as usual, and a drop-down menu for the selection of an external tool. Once selected, and if the tool allows an initial configuration, a web form is rendered below in the same screen for the educators to configure the tool.

It is noteworthy that once the LAMS adapter was developed, around twenty external tools were automatically available under the "gluelet" activity, since many tool adapters had been previously developed. This is a major advantage of the GLUE! architecture. Furthermore, from now on, every new integrated tool will be simultaneously available in Moodle, LAMS and MediaWiki. The few requirements imposed by the GLUE! contract, and the use of well-documented and widespread specifications, reduces the development effort, thus facilitating the implementation of new adapters and encouraging third-party developers to progressively increase the number of available tools in GLUE!.

The lesson containing the learning design is deployed as usual in the LAMS monitoring environment. After a manual group population, external tool instances are automatically created and configured, being ready for the students. Figure 3 shows a screenshot with the enactment of the lesson at one particular moment. This figure represents a Google Document embedded in LAMS, whose settings were detailed in Figure 2. A couple of students are synchronously and collaboratively working on the document. The headings and the structure of the document were provided by the educator by uploading an initial file in the configuration of the tool, while the drawing that briefly appears in the screenshot was generated with Dabbleboard in the previous activity, being manually inserted in the document by the students later.

Enactment of collaborative learning situations in LAMS: A case study

Educators delivering Advanced Networking, a third-year course in the degree of Telecommunication Engineering in the University of Valladolid, Spain, are planning to realize a collaborative learning situation during November 2011, with the 48 students enrolled in this course. In this situation, students must design, discuss and present the flowchart and time sequence diagram of a message server, as part of a practical lesson about network sockets. Students have one week, including two face-to-face sessions in the laboratory of two hours each, to successfully accomplish these tasks.



Title

Discussion of the flowchart and the sequence diagram

Instructions

In this activity you must justify the decisions taken in the design of the flowchart and the sequence diagram of the message server using a text editor. Once finished, you must review the flowcharts and the sequence diagrams of the other groups in your supergroup including comments with the same text editor.

Tool name: Goog

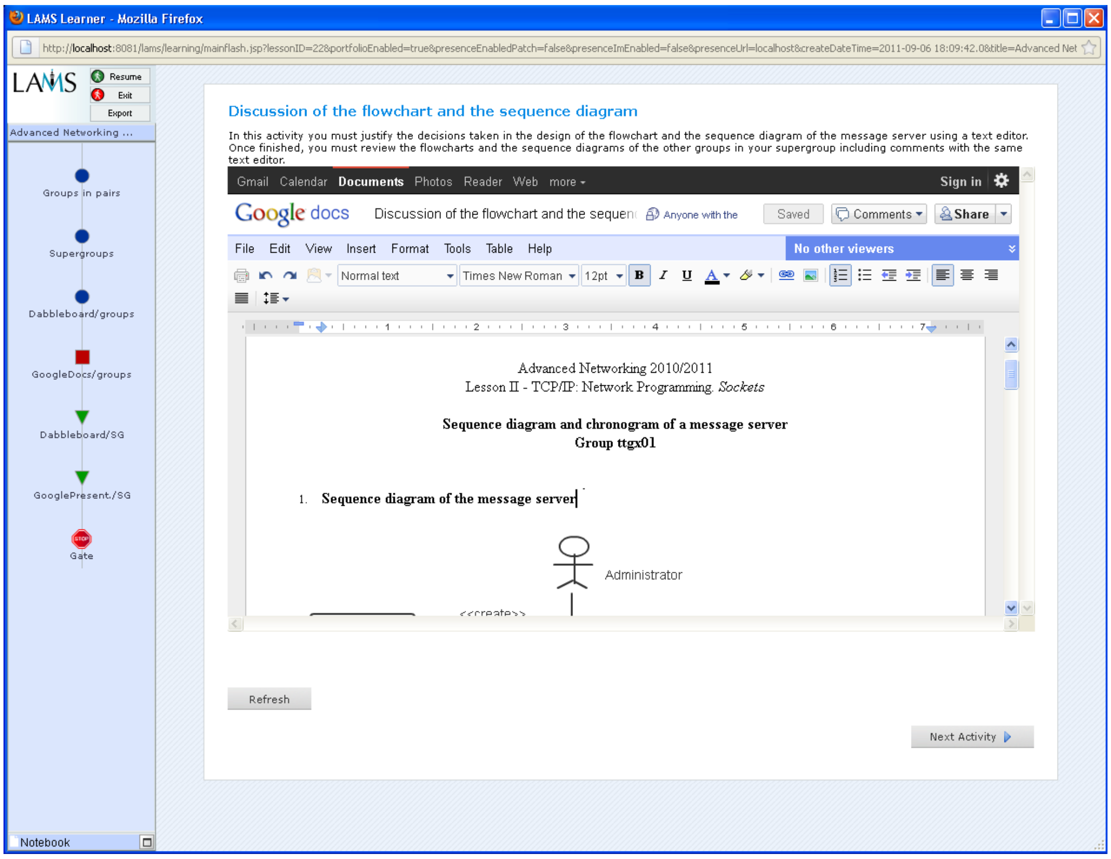
Google Documents (http://docs.google.com/)
Google Spreadsheets (http://docs.google.com/)
Google Presentations (http://docs.google.com/)
Wikipedia (http://wikipedia.org/)
PHD Comics (http://phdcomics.com/)
Web Content (non defined)
Dabbleboard (http://dabbleboard.com/)
Chat Widget (http://pandora.tel.uva.es:8880/)
SharedDraw Widget (http://pandora.tel.uva.es:8880/)
Sudoku Widget (http://pandora.tel.uva.es:8880/)
Forum Widget (http://pandora.tel.uva.es:8880/)
Natter Chat Widget (http://pandora.tel.uva.es:8880/)
Bubble Game Widget (http://pandora.tel.uva.es:8880/)
Butterfly Paint Widget (http://pandora.tel.uva.es:8880/)
GSIC MediaWiki (http://delfos.tel.uva.es/wikis/)
You Decide Widget (http://pandora.tel.uva.es:8880/)
Doodle (http://doodle-test.com)

Select a tool: Google Documents (http://docs.google.com/)

Title: GoogleDocs/groups

Grouping: Groups in pairs

Figure 2: Design of a lesson in the LAMS authoring environment using GLUE!


LAMS Learner - Mozilla Firefox
http://localhost:8081/lams/learning/mainflash.jsp?lessonID=22&portfolioEnabled=true&presenceEnabledPatch=false&presenceImEnabled=false&presenceUrl=localhost&createDateTime=2011-09-06 18:09:42.0&title=Advanced Net
LAMS
Resume
Exit
Export
Advanced Networking ...
Groups in pairs
Supergroups
Dabbleboard/groups
GoogleDocs/groups
Dabbleboard/SG
GooglePresent./SG
Gate
Gmail Calendar Documents Photos Reader Web more
Sign in
Google docs
Anyone with the
Saved
Comments
Share
File Edit View Insert Format Tools Table Help
No other viewers
Normal text
Times New Roman
12pt
Administrator
<<create>>
Refresh
Next Activity
Notebook

Discussion of the flowchart and the sequence diagram

In this activity you must justify the decisions taken in the design of the flowchart and the sequence diagram of the message server using a text editor. Once finished, you must review the flowcharts and the sequence diagrams of the other groups in your supergroup including comments with the same text editor.

Discussion of the flowchart and the sequen

Advanced Networking 2010/2011
Lesson II - TCP/IP: Network Programming. *Sockets*

Sequence diagram and chronogram of a message server
Group ttgx01

1. **Sequence diagram of the message server**

Figure 3: Enactment of a lesson in the LAMS learning environment using GLUE!

The educators have designed this situation, so that students work initially in pairs to propose a first solution. Later on, groups of three or four pairs (called supergroups) are formed to give feedback to their group partners and to agree on the final flowchart and time sequence diagram. More specifically, the timeline of this situation is as follows: the work in pairs is due during the first face-to-face session; the review of the initial diagrams is due at home during the week; and the agreed final diagrams and the presentation are due during the last face-to-face session. This learning design promotes groupwork and enforces reaching a gradual consensus by means of a well-known collaborative learning flow pattern called pyramid (Hernández-Leo et al. 2005).

This situation would benefit from its design and enactment in LAMS for several reasons. First, the design is clearly defined, so that a lesson containing the activities can be generated and deployed in advance, supporting a later formation of the groups at run-time, according to students' preferences. Besides, the use of a collaborative learning flow pattern that promotes the realization of tasks in order fits with the LAMS representation based on sequenced activities. In addition, LAMS can easily handle the creation and management of the numerous groups, and provide features to help educators tracking the work and discussions of each group during the realization of the activities. Nevertheless, this situation cannot be designed and enacted in LAMS because the required tools are not provided: LAMS does not include neither shared whiteboards, nor presentation tools, while built-in collaborative text editors (e.g. notebook or wiki) offer a very limited functionality.

In this context, GLUE! could provide the tools required for the realization of this collaborative learning situation. For example, among the available external tools, Dabbleboard is a shared whiteboard that can be useful to draw the time sequence diagram and the flowchart; Google Documents is a collaborative text editor that can support the discussion in groups; and Google Presentations is a collaborative presentation tool that can be employed to explain the findings and the final drawings to the rest of the class. Therefore, after implementing the LAMS adapter, this collaborative learning situation was successfully designed and deployed in LAMS by the authors of this paper (Figures 2 and 3), as a preliminary step aimed at detecting problems and testing. Nevertheless, real Advanced Networking students are intended to realize this situation within this semester. It is noteworthy that, since the LAMS adapter was designed and implemented taking into account the main distinctive LAMS features, the Advanced Networking educators can add transitions and branches in their design, monitor students' work, and share this lesson with other peers, as usual.

Significantly, an important functional limitation was detected during the enactment of this collaborative learning situation: LAMS does not support the reuse of tool instances in different activities of the same lesson, neither does the GLUE! adapter for LAMS. Therefore, students could not review their supergroup partners' diagrams in a new LAMS activity, once the work in pairs is accomplished. A simple solution would be to review the documents in the same LAMS activity, modifying the groupings from "pairs" to "supergroups", but LAMS does not allow changes in activities that have started. However, LAMS supports adding new members to a group at any time, and GLUE! supports the update of users sharing external tool instances, so educators could sequentially grant access to the documents each supergroup must review, by manually modifying the group members. This additional burden is cumbersome and error prone and so, the implementation of the functionality to reuse instances is prioritized in the improvement of the current LAMS adapter. This functionality would enable peer review of previously generated artifacts, even though it is not natively supported by LAMS. The reuse of instances could be implemented as part of the advanced settings of the "gluelet" activity in the authoring environment. The LAMS adapter would thus offer educators the opportunity to create and configure new instances or reuse some of the existing ones in that lesson. Since instances are not created until the deployment of the lesson, the LAMS adapter could only show educators an abstract definition of the available instances in the authoring environment, identifying them, for instance, with the name of the activity and the group.

Finally, it is noteworthy that the same conceptual learning design with the same external tools was enacted and realized in Moodle by real students during November 2010, as part of the previous course on Advanced Networking, and no problems were reported. This shows that the tool life cycle supported by GLUE! enables the design and enactment of the same conceptual design in different VLEs. Moreover, considering the distinctive native VLE features in the design and implementation of the corresponding VLE adapter, leads to a seamless integration of external tools that exploits the potential of each particular VLE.

Conclusions

This paper has presented the requirements and design decisions that led to the implementation of an adapter that enables the integration of multiple existing external tools in LAMS through the GLUE! architecture. The implementation of this adapter follows both the GLUE! and LAMS integration contracts, thus supporting the life cycle of external tools, as well as the main distinctive LAMS features regarding monitoring, learning sequences, design reuse and groupwork. The effort required for the development of the LAMS adapter is similar to that of *ad hoc* integration modules, but with the important advantage that every new integrated tool requires a much lower effort and is automatically integrated in multiple VLEs at the same time.

A representative collaborative learning situation that includes several external tools, a sequence of activities, and different group settings has been designed and enacted in LAMS as a case study. This case study illustrates the integration of external tools in LAMS with GLUE!, and takes also advantage of the distinctive LAMS features. The only limitation found affects the reuse of instances of the same tool in different activities, which is not a problem of GLUE!, but a consequence of the current implementation of the LAMS adapter. This limitation is expected to be overcome before real students realize this situation.

References

Alario-Hoyos, C. and Wilson, S. (2010). Comparison of the main Alternatives to the Integration of External Tools in different Platforms. *Proceedings of the International Conference of Education, Research and Innovation* (ICERI 2010), Madrid, Spain, 3466-3476.

Alario-Hoyos, C., Velasco-Villanueva, D., Bote-Lorenzo, M.L., Gómez-Sánchez, E., Asensio-Pérez, J.I., Vega-Gorgojo, G. and Ruiz-Calleja A. (2011). GLUE! Contract Specification 1.0, *Technical Report*, GSIC-EMIC, University of Valladolid, Spain. URL: http://www.gsic.uva.es/glue/, last visited, September 2011.

Bower, M. and Wittmann, M. (2011). A Comparison of LAMS and Moodle as learning design technologies - teacher education students' perspective. *Teaching English with Technology – Special Issue on LAMS and Learning Design,* 11(1), 62-80.

Catherall, P. (2005). *Delivering E-Learning for Information Services in Higher Education*, Chandos Publishing, Oxford, UK.

Del Blanco, Á., Torrente, J., Marchiori, E.J., Martínez-Ortiz, I., Moreno-Ger, P. and Fernández-Manjón, B. (2010). Easing Assessment of Game-based Learning with <e-Adventure> and LAMS. *ACM International Workshop on Multimedia Technologies for Distance Learning (MTDL 2010)*, Florence, Italy, 25-30.

Dillenbourg, P., Baker, M., Blaye, A. and O'Malley, C. (1995). The evolution of research on collaborative learning. In E. Spada & P. Reiman (Eds) *Learning in Humans and Machine: Towards an interdisciplinary learning science*, Elsevier, Oxford, UK, 189-211.

Fuente-Valentin, L., Pardo, A. and Delgado-Kloos, C. (2011). Generic Service Integration in Adaptive Learning Experiences using IMS Learning Design. *Computers & Education*, 57(1), 1160-1170.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J.M. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional, Reading, MA, USA.

Ghiglione, E. and Dalziel, J. (2006). Design principles for LAMS version 2 and he LAMS Tools Contract, *Proceedings of the TenCompentence Conference Workshop,* Barcelona, Spain.

Hernández-Leo, D., Asensio-Pérez, J. I. and Dimitriadis, Y. (2005). Computational Representation of Collaborative Learning Flow Patterns using IMS Learning Design. *Journal of Educational Technology & Society,* 8(3), 75-89.

Weller, M. (2007). *Virtual Learning Environments: Using, Choosing and Developing your VLE.* Routledge, New York, NY, USA.

Wilson, S., Shaples, P. and Griffiths, D. (2008). Distributing education services to personal and institutional systems using Widgets, *Proceedings of the First International Wokshop on Mashup Personal Learning Environments*, Maastricht, The Netherlands, 25–32.

This work has been partially funded by the Spanish Ministry of Science and Innovation (TIN2008-03023, IPT-430000-2010-054) and the Autonomous Government of Castilla y Leon, Spain (VA293A11-2).

Carlos Alario-Hoyos is currently a PhD student and a teaching assistant at the University of Valladolid, Spain. He received his MSc degree in telecommunications engineering from this University in 2007. He is currently a PhD student and a teaching assistant at this University. His research interests include software applications and platforms for the support of learning. Address: School of Telecommunication Engineering, University of Valladolid, Paseo de Belén 15, 47011, Valladolid, Spain. Email: calahoy@gsic.tel.uva.es.

Please cite as: Alario-Hoyos, C., Bote-Lorenzo, M., Gómez-Sánchez, E., Velasco-Villanueva, D., Asensio-Pérez, J., Vega-Gorgojo, V. and Ruiz-Calleja, A. (2011). Integration of external tools with GLUE! in LAMS: requirements, implementation and a case. In Cameron, L. & Dalziel, J. (Eds). *Proceedings of the 6th International LAMS & Learning Design Conference 2011: Learning design for a changing world* (pp 40-48). 8-9 December 2011, Sydney: LAMS Foundation. http://lamsfoundation.org/lams2011sydney/papers.htm