

USING MOODLE ACTIVITIES WITHIN LAMS

by **Ernie Ghiglione, Marina Rodríguez Aliberas*** , **Lluis Vicent***, **James R. Dalziel**

Macquarie E-Learning Centre Of Excellence (MELCOE)

Macquarie University, Sydney, Australia

* Escuela de Ingeniería y Arquitectura de la Salle

Universidad Ramon Llull, Barcelona, Spain

{ernieg, james}@melcoe.mq.edu.au; {vicent, st16827}@salle.url.edu

Abstract

In this paper we discuss the implementation of the LAMS Tool Contract to include Moodle learning activities within the highly visual context of the Learning Activity Management System (LAMS). We aim to show how learning tools, from different Learning Management Systems, (LMS) can be included in a visual learning design environment such LAMS using a common interface (LAMS Tool Contract). In addition, we present an example of tool interoperability where learning activities can be used in complex and sophisticated learning designs using LAMS.

1. Introduction

LAMS is a system for educators to author learning designs using a drag and drop visual interface, as well as run these designs with learners and monitor learner progress [1][2]. In the second generation of LAMS (or LAMS v2), one of the critical design requirements was to include any learning activity or tool - not just LAMS activities - to LAMS so these activities could be orchestrated and used in sequences of learning activities or learning designs.

In order to achieve this requirement, an application-programming interfaces, registered URLs and a set of expected behaviors were agreed upon so any tool that implement them, would be able to interact with the LAMS Core.

The next section formally introduces the LAMS Tool Contract and then we continue demonstrating an implementation of Moodle learning activities within LAMS.

2. LAMS Tool Contract

The LAMS Tool Contract is a set of expected behaviors, registered URLs and API calls that a LAMS Tool has to implement to communicate with the LAMS core modules.

Activities interact with the LAMS Core via URL calls and direct Java or webservices calls. These tools implement interfaces defined in the LAMS core, and makes use of known LAMS services supplied by the Core. Native LAMS tools are written so they use the Spring Framework (Spring Framework, 2006) to allow the LAMS Core to communicate with the tool. However, external tools that might be written in other languages can also be used as Native LAMS tools with an external wrapper that permits the external tool to behave in the way that the LAMS Core expects.

Each tool interacts with the following LAMS Core modules:

- **Author:** calls the tool to create/update or delete tool content. Uses the tool's authoring screens.
- **Monitor:** uses the tool's monitoring screens to follow learners' progress.
- **Administration:** uses the tool's admin screen, which may be used to configure or set up the tool's definitions.
- **Learner:** calls the tool to copy tool content and set up tool sessions. Uses the tool's learner and export portfolio screens.

These four main core modules are essential for a learning design system in which teachers/instructors can configure and setup the behaviour and content of each learning activity (author), monitor the learner's progress through the learning process (monitor), execution of the lesson plan with the students (learner) and instantiation and configuration (administration).

Other tool interoperability approaches like IMS Tool Interoperability, for example, are limited in the interfaces they provide, which unfortunately defeat its interoperability requirements of learning design.

All native LAMS tools “comply” with this contract, but in the case of existing external learning activities from other Learning Management Systems (LMS), the need of an external tool adapters is required [3].

3. External Tool Adapters

Tool Adapters act as bridge between LAMS and the external tool. The external tool adapters allow the external tools to comply with the LAMS Tools Contract as well as handling authentication and authorization. These adapters are a software implementation that “wraps” the external tool and provides the interfaces required by LAMS Tool Contract so it can be used within LAMS. As these tools are added into LAMS, they automatically “inherit” all the characteristics and attributes of native LAMS tools such as “workflow awareness”, grouping, branching, conditions, outputs and tool interoperability (all the exported learning design containing the content of the external tool as well, making it reusable in other platforms where the external tool is present).

Usually these external activity tools might be part of an LMS. However, these tools can also be standalone services on the Internet with open APIs, for example: Flickr, Youtube, Google Docs, Calendar, Picasa, etc.

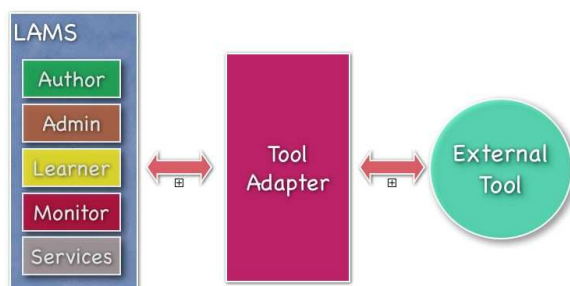


Figure 1. External Tool Adapter

For the purpose of this paper we will focus on the Moodle Tool Adapter.

4. Moodle Tool Adapter

The Moodle Tool Adapter is a tool adapter that permits Moodle tools to be used natively (as LAMS Tools) within a LAMS and Moodle integration.



Figure 2. Moodle Tool Adapter

The main challenge we faced with Moodle Tools is the same as we confront with other LMSs: Moodle tools instances are heavily tied to a Moodle course. Therefore Moodle tools cannot be created or instantiated without a course.

LAMS (and any learning design for that matter) has a design stage (authoring) and a runtime or enactment stage. In the design stage, the teacher sets up the activities, their content and, using transitions, sequences these activities creating a learning design. The runtime or enactment stage, the created learning design is instantiated to be run with the learners. Each learner goes through the activities as the teacher sequenced them.

Moodle and other LMSs tools don't really have this concept. If they would, a teacher would be able to create a Forum outside the course and then instantiate one or more instances of this Forum in one (or more) of her courses. Currently, LMS tools allow teachers to create activities/tools *in a course*.

Trying to do a work around this can be difficult as it requires slight changes to the LMS tool as it has to overcome this limitation. In this particular case with Moodle, the modifications required in the Moodle Forum code did not exceed more than 70 lines of code, and that includes additions and modifications.

4.1. Authoring Moodle Activities within LAMS

Once these Moodle changes are in place and the tool adapter is added to LAMS, then the Moodle tools appear in LAMS like any other learning activity that teachers can use in their sequences of learning activities.

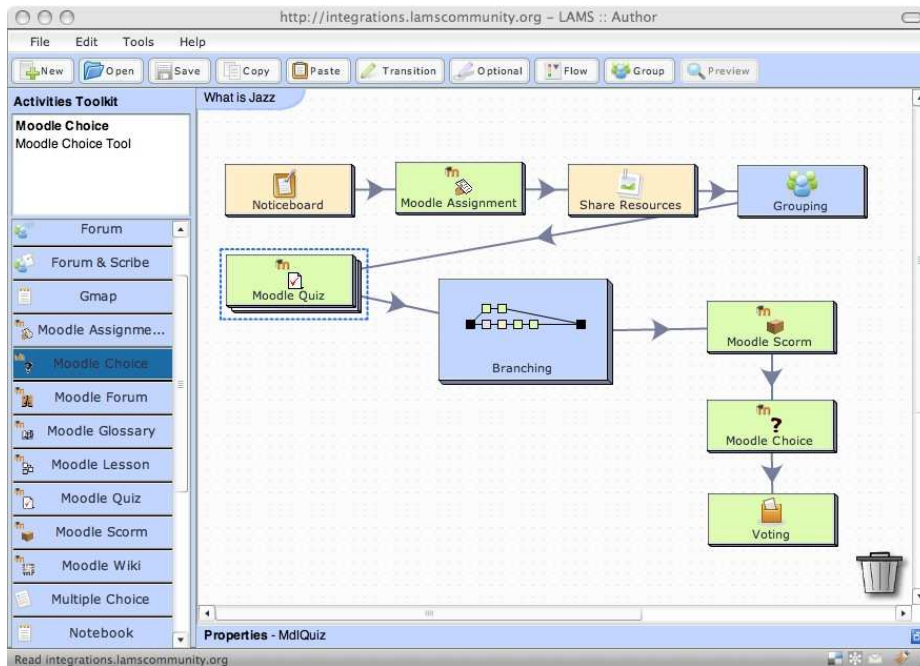


Figure 3. Authoring Moodle within LAMS

As seen in Figure 3, the Moodle activities can be drag-and-dropped and used as any other LAMS activity. Most importantly, when the Moodle activities are authored, they provide their own interface (see Figure 4) and not a proxy interface. In this way teachers who are familiar with these Moodle tools don't need to learn anything new.

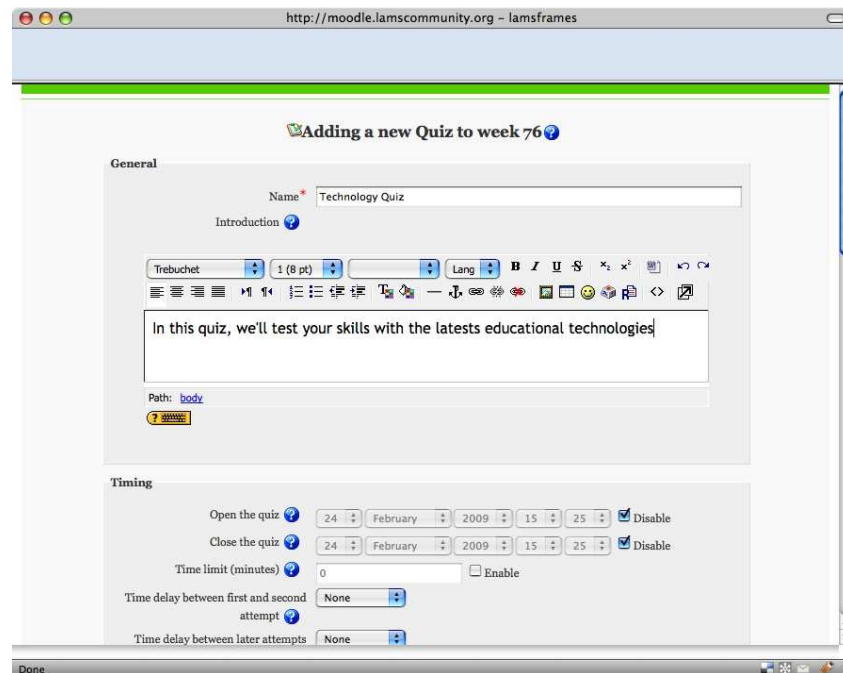


Figure 4. Authoring Moodle Quiz

Additionally by implementing a completion state, these Moodle activities are now workflow aware. They record when a learner completes one activity and proceeds to the next. As in with another LAMS activities, these activities inherit Groups and Outputs that can be used to create adaptive learning paths with Branching. Although the Moodle activity in itself might not even know that it is grouped, its tool adapter can instantiate it as many times as Groups are needed.

4.2. Running LAMS lessons with Moodle activities

A LAMS lesson is a running instance of a design. After a learning design is authored, a teacher can assign a particular learning design to a group of learners, creating a lesson.

Once a lesson is created, learners run the lesson following the sequence of activities designed by the teacher. When the learners get to the Moodle activities, they see the exact same interface for the Moodle activity. This interface is not a proxy but the actual learner interface for this Moodle activity (see Figure 5).

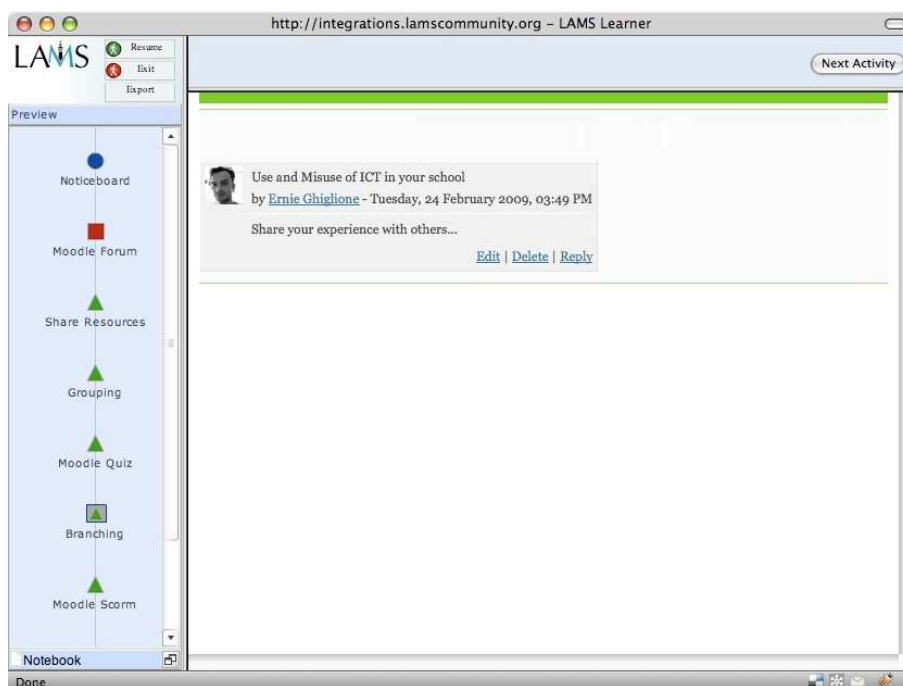


Figure 5. Moodle Forum within LAMS

Similarly to the authoring interface, the LAMS Monitor interface allows teachers to provide learners' progress in real-time. For Moodle activities within LAMS, these can display the teacher's view of a particular instance of a Moodle activity, achieving a similar result.

Output from Moodle activities can also be used to create branching paths. For instance, we can specify a LAMS branching activity that uses a Moodle Forum output like "number of postings in a thread". According to the number of posting per each learners, we can create a condition stating that, "if a learner posted more than five comments in the forum thread, then he/she should follow a particular learning path". Given the Moodle outputs (implemented by its tool adapter), then we can create complex adaptive learning designs using external tools.

A working implementation of this work is available for the public to experiment with using the Moodle LAMS Integration servers [4]. In there, there are already 13 different Moodle activities added in LAMS and more are expected to be included shortly.

5. Conclusions

We have shown how using the LAMS Tool Contract provides a flexible architecture to incorporate learning activities to create elaborate learning designs using LAMS' highly visual

environments. The usage of tool adapters makes it easier for developers and teachers to add external non-LAMS native tools within LAMS while taking advantage of all the benefits of LAMS tool features.

We believe this is an important development as it gives other LMS the ability to leverage on LAMS visual learning design environment as well as branching and sequencing capabilities, without the need to significantly modify their existing systems.

In this paper we have shown how this can be achieved with Moodle, but we have also implemented tool adapters for other LMSs (Sakai, .LRN [5]) as well as Internet tools such as Dimdim Internet conferencing activities, Pixlr drawing tools, etc.

6. References

- [1] Dalziel, J. R. (2003a). Introducing LAMS: An EML/LD “inspired” system. Presentation for the Valkenburg group meeting, Vancouver, Canada, February 23rd, 2003.
- [2] Dalziel, J. R. (2003b). Implementing learning design: The Learning Activity Management System (LAMS). Author, A. & Writer B. (2003). In G.Crisp, D.Thiele, I.Scholten, S.Barker and J.Baron (Eds), *Interact, Integrate, Impact: Proceedings of the 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education*. Adelaide, 7-10 December 2003. [Online] <http://www.ascilite.org.au/conferences/adelaide03/docs/pdf/593.pdf>.
- [3] Ghiglione, E., & Dalziel, J. (2006). Design principles for LAMS version 2 and the LAMS “Tools Contract”. *Proceedings of the TenCompetence Conference Workshop*. UPF - Barcelona, Spain. June 21st, 2007.
- [4] Moodle LAMS Integration Servers Online, <http://moodle.lamscommunity.org/moodle>.
- [5] .LRN LAMS Integration Servers Online, <http://dotlrn.lamscommunity.org/>.