Design Principles for LAMS Version 2 and the LAMS "Tools Contract"

Ernie Ghiglione and James Dalziel
Macquarie E-Learning Centre Of Excellence (MELCOE)
Macquarie University, Sydney, Australia
ernieg@melcoe.mq.edu.au & james@melcoe.mq.edu.au

Background

LAMS (the Learning Activity Management System) is a system for educators to author learning designs using a drag and drop interface, as well as run these designs with students, and monitor student progress. Initial LAMS development began in 2002, and a live prototype was first demonstrated at the Valkenburg group meeting in Vancouver in February 2003 (Dalziel, 2003a). It was subsequently trialled in a range of university and school contexts in 2003 and 2004, with the official Version 1 release in December 2004 (LAMS Foundation, 2004), followed by the release of LAMS as open source software in early 2005 (LAMS Foundation, 2005a).

While LAMS was "inspired" by Educational Modelling Language (EML) and IMS Learning Design (IMS LD), its main initial focus was on activity tools for learning design, not as a reference implementation of EML or IMS LD. Although the initial development of LAMS attempted to implement IMS LD, a series of challenges meant that this was not achieved in the first version (Dalziel, 2003a; 2003b).

A range of formal evaluations of LAMS have been conducted, such as the JISC (Joint Information Systems Committee) trial of LAMS in UK higher and further education (Masterman & Lee, 2005) and the BECTA (British Educational Communications and Technology Agency) evaluation of the UK Specialist Schools Trust trial of LAMS (Russell, Varga-Atkins & Roberts, 2005), as well as other research and evaluations (for example, see the list at LAMS Foundation, 2005b).

Complementing the LAMS software is the "LAMS Community" (LAMS Foundation, 2005c), an online community of practice for LAMS users to discuss the use of LAMS and share LAMS learning designs through a repository (Dalziel, 2006). In addition to the feedback from formal evaluations such as those described above, LAMS users have provided extensive feedback through the LAMS Community forums and through personal contact with LAMS developers.

While many educators and students have used the first version of LAMS to create impressive learning designs and experience rich online learning environments, their use has encouraged them to request further features and improvements.

Additionally, since the LAMS application was originally created as a proof-of-concept, the system lacked a clear specification or blue-print. Moreover, due to the high demand for new features in LAMS, the early development team started to increase the code base without a sufficiently robust architecture that could scale to handle hundreds of simultaneous users.

Most importantly, the experience of using LAMS led educators to request fundamental new features, such as: the ability to edit a lesson (or unit of learning) while it was running with students; an activity tool “plug-in” architecture that supported easy development and installation of new tools; branching within learning designs; use of media types other than just text (videos, audio, etc); the ability of students (and teachers) to export out of LAMS a comprehensive record of learning (“portfolio export”), internationalisation to support languages other than English, etc (for further details, see LAMS Foundation 2006). The combination of a need for a scalable architecture together with support for a wide range of fundamental new features made it clear that a new version of LAMS was needed.

Requirements for a new LAMS

While the educational concept of LAMS was sound, key areas for improvements were identified based on numerous consultations and feedback from educators and users. These areas can be summarised as:

- Pedagogical support and reusability
- Collaboration
- Usability and interface
- Technical improvements

Each of these areas is described in detail below.

Pedagogical support and reusability

As part of improving LAMS’ pedagogical support and reusability, educators required a variety of new features mainly related to greater flexibility in modifying running learning designs. Educators wanted LAMS to behave in a similar fashion to a real face-to-face course where a teacher is able to modify the activities “on-the-fly” while giving a lesson. Their rationale is that in real classroom environments, it is not uncommon for teachers, while delivering a lesson, to change the activities they had planned to convey to the students due to unforseen circumstances. An example is when a teacher realizes that the students’ knowledge of the subject she is teaching is greater than she expected. In face-to-face environments, teachers can opt to change the activities as they have planned to better suit their students, and this ability to adapt a lesson “on-the-fly” was a key feature that educators wanted replicated in LAMS.

Another important requirement was branching within learning designs. Educators wanted to group students within a class based on criteria of their choice and let each group do separate sets of activities according to their criteria. In previous version of LAMS, educators who wanted to use different pedagogical methodologies were limited to the “Optional” tool which allows students to choose from a set of activity tools, but this was limited to a selection from a single set (not selection of several sequences of activities, ie, different branches). Educators wanted a richer set of branching approaches, including student-selected branches, teacher-allocated branches, and automated system allocation to branches (based on prior activity tool data such as students that have attain a certain score in a quiz activity or posted a specific number of postings in a forum).

A different requirement was the ability of students (and also educators) to export a comprehensive record of activities conducted with LAMS to be stored outside the system (such as in an e-portfolio or Personal Learning Environment). This export should be a static set of files that does not require any connection back to the original LAMS server (so that the record can be kept indefinitely, regardless of when a LAMS server may be shut down).

In LAMS Version 1, grouping activities allowed teachers to set a number of groups into which students would be allocated by the system in random order. Although this was helpful in certain educational scenarios, educators requested the additional features of allow students to select their groups or the ability for the teacher to allocate students into groups at runtime. Building on the earlier feature, educators also wanted to be able to set a number of students per group (not just a number of groups).

Given improvements in bandwidth and storage, the need to use richer multimedia objects in online lessons is greater than at the time of initial development. Educators now expect that all learning design activity tools to support video, audio, images and other types of rich media.

Many educators who had used the LAMS Version 1 authoring had incorporated "Noticeboard" (ie, plain text) activities into which they described an "offline" activity that students would conduct at that point in a sequence. This approach allowed a single learning design to capture a flow of both online and offline tasks. Educators asked that this potential use of LAMS (as a record of offline tasks) be formalised as an optional setting for any activity tool. To extend this feature, an "Instructions" area was provided to allow for further instructions (either to educators or students) about how to run either online or offline activities. A file upload facility was added for both the online and offline instructions area to allow storage of related materials such as printed student worksheets, advice sheets for educators, background articles, etc. This set of features adds a new dimension to LAMS in that it is no longer just an e-learning system, but rather it becomes a generalised lesson planning environment in which "e-delivery" of any given task is an option.

Educational research supports the concept that student learning benefits for self-reflection as part of the learning process (eg, Laurillard, 2001). In LAMS Version 1, certain tools included writing a reflective comment in a notebook as part of the overall activity. Following feedback on the potential benefits of including this step in all tools, it was added as an optional setting for all LAMS V2 activity tools.

In support of system interoperability, LAMS V2 included an export option for LAMS sequences using the bindings proposed in Level A of the IMS Learning Design specification.

Collaboration

In the area of collaboration, educators wanted more flexible ways to create learning designs collaboratively with other educators. This request included the ability to provide multiple shared areas for designs for different groups of educators, including a shared area for each course (rather than only one shared area for educators as in LAMS V1).

Also, greater collaboration among educators also meant sharing the workload of monitoring learners in a lesson, so tutors or other teaching assistants can collaboratively assist and monitor learners along side each other and a course convenor. A course convenor can have monitor access to all classes, whereas tutors can be assigned only to monitor their own classes.

Usability and Interface

One of the most important requirements was the internationalisation of LAMS so that it could support languages other than English. As LAMS V1 only support the English character set, LAMS V2 was built to support a wide range of character sets including double-byte characters (such as Chinese and Japanese) as well as Arabic (including right to left text placement).

Another important aspect of the LAMS interface is the look-and-feel. Since LAMS is used integrated with other learning environments, it was important that the look-and-feel could be tailored to match that of the integrated learning environment, therefore making the user experience seamless across both systems. To support easy changes to look and feel, LAMS V2 uses a well specified set of Cascading Style Sheets (CSS) throughout the system.

Educators also wanted to use rich-text editors to create rich web content for their activities. This included the ability to change font sizes and colours, and include images, flash movies, create tables, etc.

In addition to making the LAMS interface more intuitive and user friendly, accessibility requirements (W3C Level A) were implemented to assist learners with specific disabilities.

Technical improvements

Robustness, scalability and performance were the key aspects that technical teams planning to implement LAMS at educational institutions with 35,000+ learners were most concerned about. LAMS V2 needed to support a large number of total and concurrent users, and incorporate technical features to assist this, such as database and application server clustering.

LAMS version 2 Architecture

Once the requirements outlined above were assessed and analysed, the planning phase started to create the specifications for LAMS Version 2. Given the requirements, the LAMS 2 architecture was designed to accommodate these requirements and provide a foundation for future features and improvements.

To support the easy development and installation of new activity tools, LAMS 2 implements a modular architecture where activity tools can be added on-the-fly to a LAMS 2 server. In order to provide such modularity, LAMS 2 implements a Tools Contract. The Tools contract is a set of expected behaviours and APIs that each tool needs to implement to communicate with LAMS Core. The LAMS Core has modules for Authoring, Monitor, Administration and Learner.

The new architecture proposes a clear and defined separation between tools and core service responsibilities/functionalities.

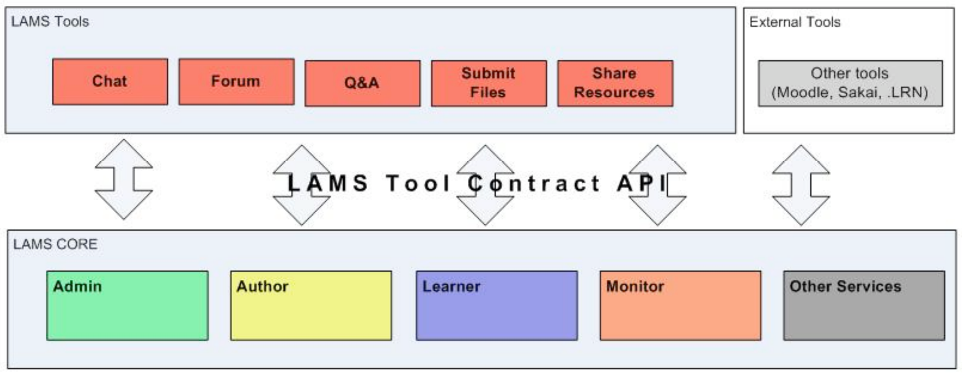


Figure 1: LAMS 2 Architecture

The LAMS 2 architecture abstracts the interface for communication between the core and the activity tools.

Activity tools in LAMS V2 are almost completely independent web applications that interact with LAMS core modules and services through the Tools Contract. This contract establishes the expected behaviours and APIs that each tool needs to implement to be instantiated as a LAMS activity tool.

By following this Tool Contract, not only can a LAMS activity tool be used within the LAMS system, but also external tools that implement it can become LAMS Tools.

In this way we intent to foster a community of tool developers that might already have activity tools that they would like to be used in the context of LAMS and are not required to learn all technical aspects of LAMS. By implementing the Tool Contract, they can use their tools within sequences in LAMS.

LAMS 2 Tool Contract

In formal terms, the LAMS Tool Contract is a set of expected behaviours, registered URLs and API calls that a LAMS Tool has to implement to communicate with the LAMS Core.

An activity tool interacts with the LAMS Core via URL calls and direct Java calls. It implements 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 be able to communicate with the tool. However, external tools that might not be written in the same languages as LAMS 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 it.

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.
- Administration
  Uses the tool's admin screen, which may be used to configure the tool
- Learner
  Calls the tool to copy tool content and set up tool sessions. Uses the tool's learner and export portfolio screens.

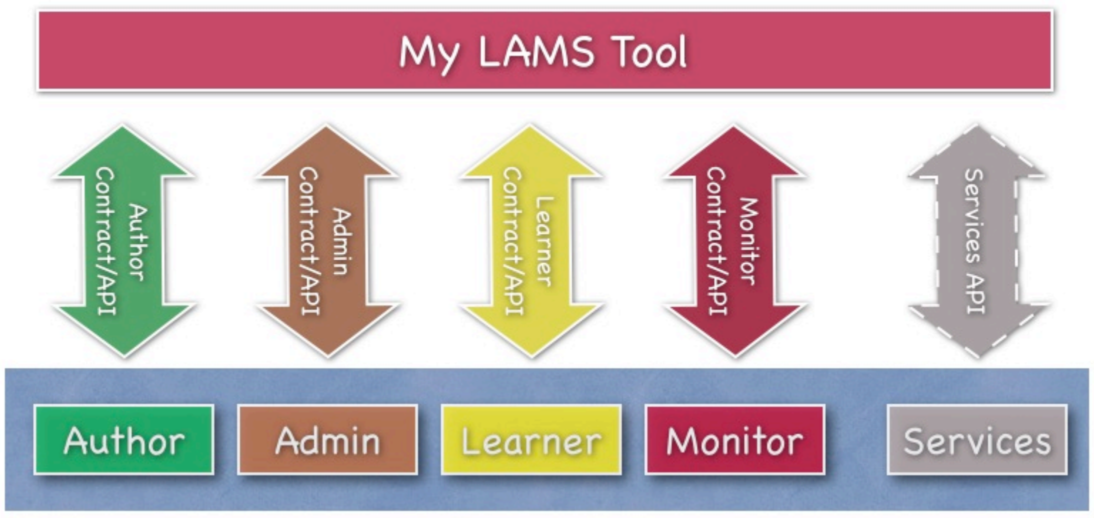


Figure 2: LAMS 2 Tools Contract

In this following section we describe in detail the LAMS 2 Tools Contract and its requirements.

Author Contract

In this section we describe the expected APIs and behaviours that an activity tool must fulfil to communicate with the LAMS Authoring module.

Authoring URL

The tool must supply an authoring module, which is called to create new content or edit existing content. This authoring module is accessed by a URL that the tool registers with the LAMS Core when it's deployed, so when an Author double clicks on the activity tool this URL will be called to author content for this activity.

Tool Icon

The tool needs to supply a tool icon so it can be displayed in the Author and Monitor interfaces.

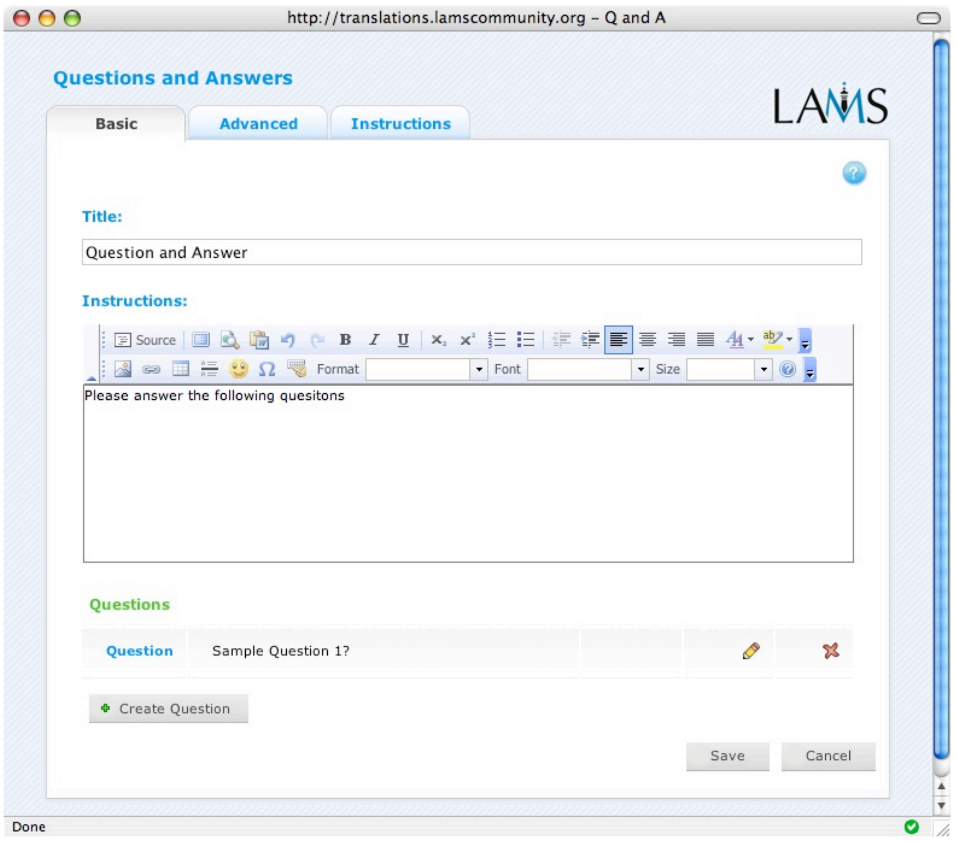
Default content

The default content is the initial data that will be displayed when the tools is authored for the first time.

Authoring UI requirements

Authoring UI (User Interface) data consists of general Activity data fields that depend on the activity purpose and the Tool specific data fields.

The LAMS authoring interface for tools have three tabs: Basic, Advanced and Instructions. Each of these tabs contains certain mandatory fields as well as tool's specific fields according to the purpose it serves. The Basic tab displays the basic set of fields that are needed for the tool depending the purpose it serves. Additionally, LAMS requires two mandatory fields (mainly for consistency purposes): Title and Instructions. The Advanced tab displays the extra fields that would be used by experienced LAMS user to set the behaviour of the tool at runtime. Finally the Instructions tab displays the "instructions" fields for teachers, where the author can specify the accompanying information for teachers on how to perform or deliver this particular activity. These instructions can be of two types: Online instructions and/or Offline instructions.


http://translations.lamscommunity.org – Q and A

Questions and Answers

Basic | Advanced | Instructions

Title:

Question and Answer

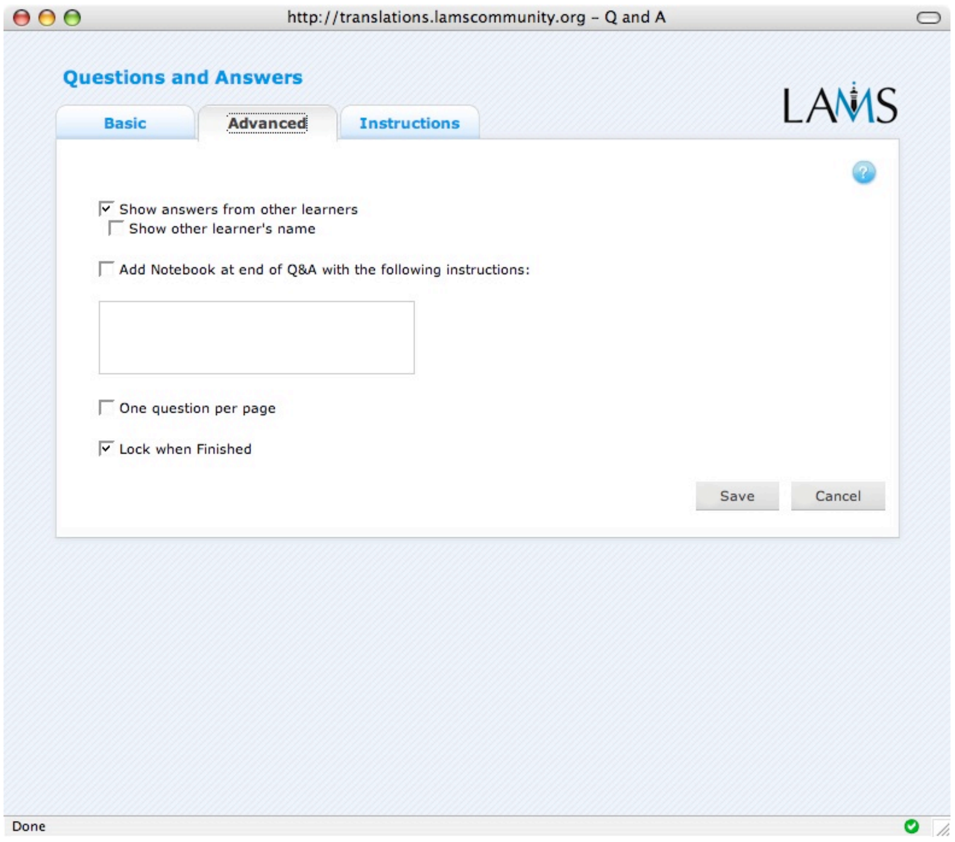
Instructions:

Please answer the following quesitons

Questions

Question	Sample Question 1?

Figure 3: Tool Authoring - Basic tab

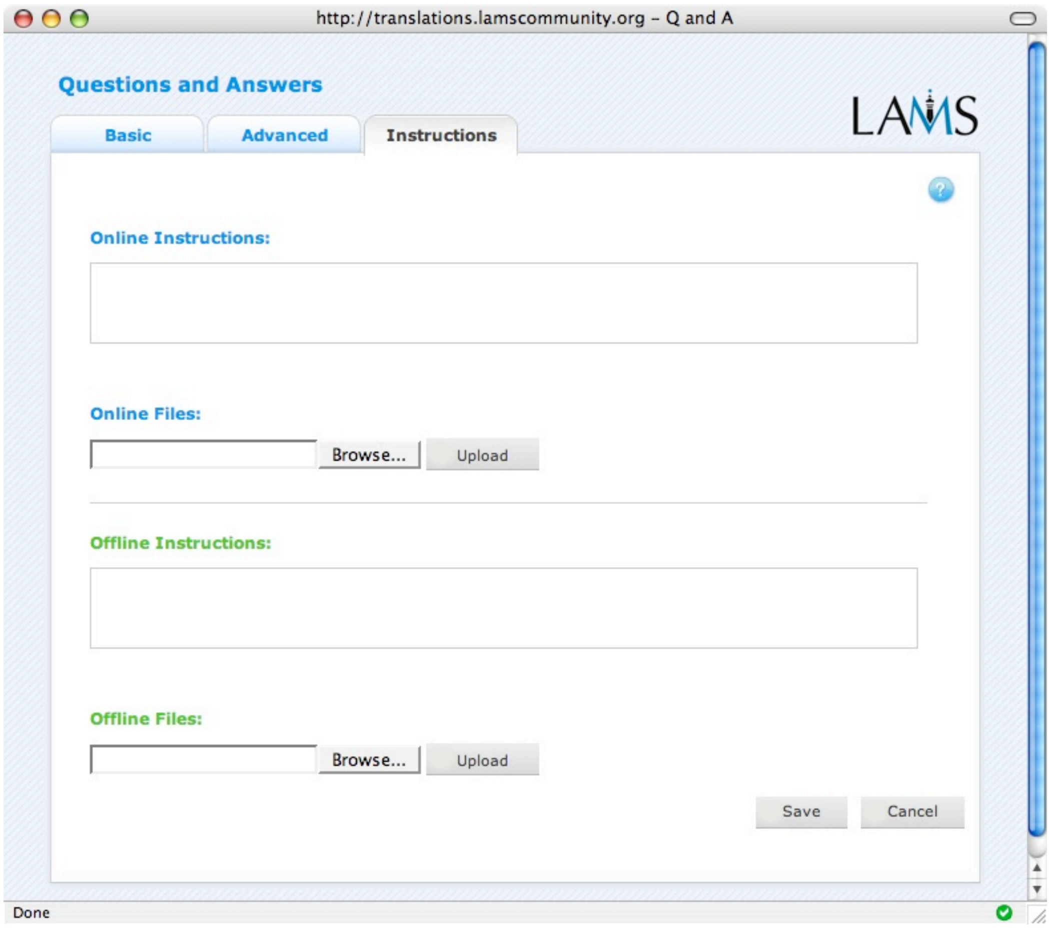

http://translations.lamscommunity.org – Q and A

Questions and Answers

Basic | Advanced | Instructions

- [✓] Show answers from other learners
  - [ ] Show other learner's name
- [ ] Add Notebook at end of Q&A with the following instructions:
- [ ] One question per page
- [✓] Lock when Finished

Figure 4: Tool Authoring - Advanced tab


http://translations.lamscommunity.org - Q and A

Questions and Answers

Basic | Advanced | Instructions

Online Instructions:

Online Files:

Browse... Upload

Offline Instructions:

Offline Files:

Browse... Upload

Save Cancel

Done

Figure 5: Tool Authoring - Instructions tab

Preview

The tool must be able to show the specified content as if it was running in a lesson (Preview mode). During Preview, the author steps through the design as if she were a student. In Preview, normally tools act exactly the same as they do for normal learner interface. The exception to this is tools that require interaction between users or would take a long time (e.g. a week) to complete. For example, if a tool needed two people to do something before either of them could finish, then that would have to work differently in Preview as there is only one person in preview.

Export tool content

Additionally, the tool must be able to export its tool content as part of the overall learning design export. The format of the serialization for export is XML.

Contextual Help

The tool should supply help information page(s) which enables the user to find useful information about authoring or working with this tool.

Data Exchange

Tools must register the type of inputs and outputs they can handle with the LAMS Core. The idea is that a tool can pass data to another tool that requires it. These are simple data types as string, text, integer, float, etc. For example, a quiz tool may pass each learner's quiz score to a later branching tool, and then the branching tool decides which branch each student goes to according to their overall quiz score.

Monitor Contract

Learner Progress URL

The learner progress URL allows the teacher to monitor the contribution of a particular student to a particular activity.

Monitor URL

The screen given by the Monitor URL is the main monitoring screen for the tool. As with the authoring screen, there are a series of tabs which allow the user to select from various functions.

The Monitoring screen has the following tabs:

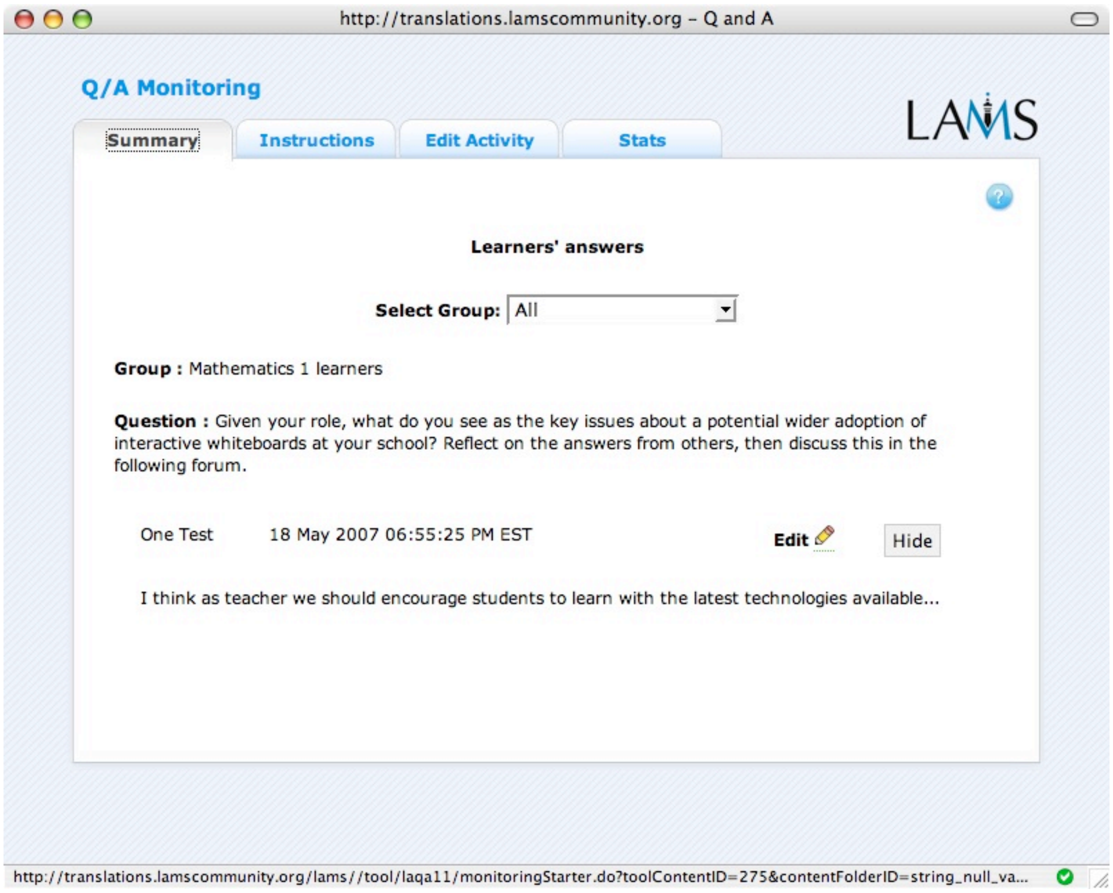
- Summary: displays a summary of all the learners' responses and allows learner entries to be modified or hidden.

- Instructions: displays the online and offline instructions, entered during authoring to provide instructions for tutors or teaching assistants on how to run or use this activity with learners.

- Edit Activity: allows a teacher to set or modify the tool content.

- Statistics: displays usage statistics for a tool e.g. the number of users completed, the percentage of correct answers, etc.

Note that a tool may include other tabs if it wishes to display something that does not fit on one of the standard four tabs.


http://translations.lamscommunity.org – Q and A

Learners' answers

Select Group: All

Group : Mathematics 1 learners

Question : Given your role, what do you see as the key issues about a potential wider adoption of interactive whiteboards at your school? Reflect on the answers from others, then discuss this in the following forum.

One Test 18 May 2007 06:55:25 PM EST

I think as teacher we should encourage students to learn with the latest technologies available...

http://translations.lamscommunity.org/lams//tool/laqa11/monitoringStarter.do?toolContentID=275&contentFolderID=string_null_va...

Figure 6: Tool Monitor - Summary tab

Export Portfolio URL

This screen allows the user to export in plain HTML the contribution of one or all participating students. The concept is to record what the learner has seen on their screens. The portfolio may be for an individual learner, or may be for the whole class (this is accessible only by the teacher).

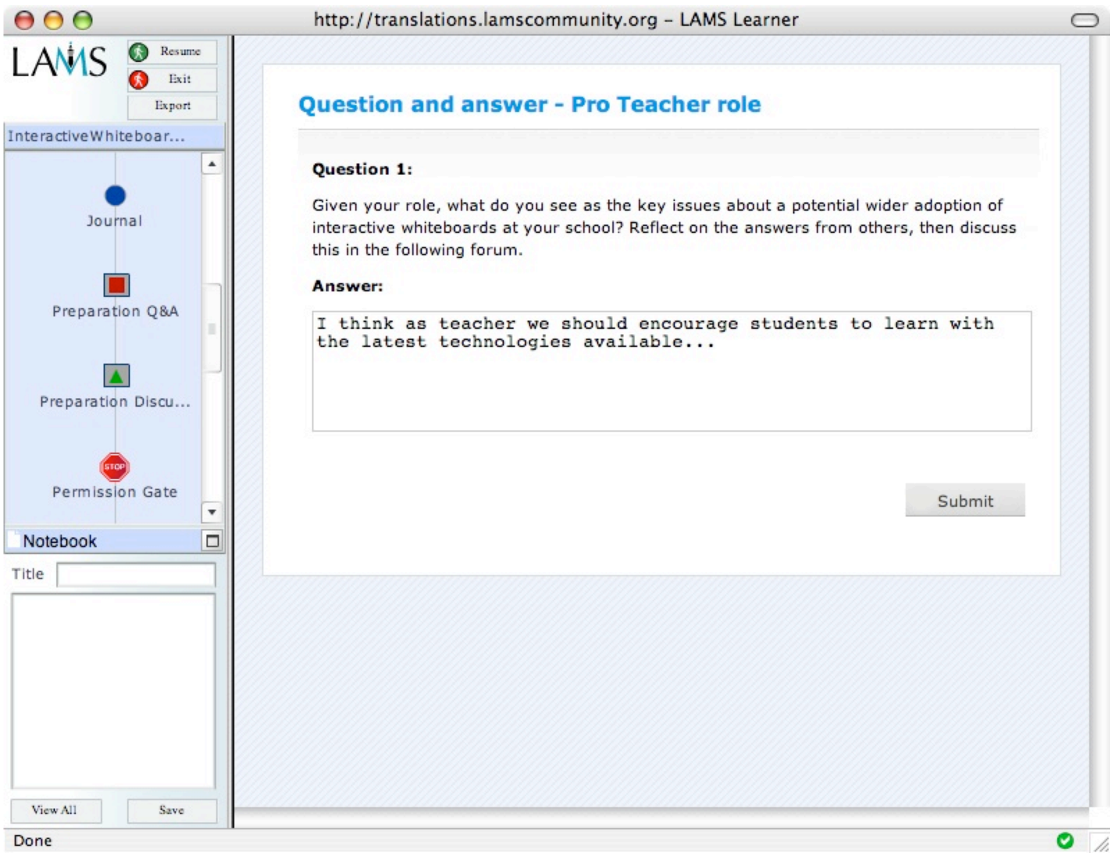
Modifying Tool Content

At runtime, teachers might need to change the content of an activity. Therefore the tools must provide an interface to change its contents while the activity is running.

Learner Contract

Learner URL

This is the URL that "plays" the content to learners and is responsible for tracking and recording the progress of the user. When the user has completed the activity tool, then the tool notifies the progress engine.


http://translations.lamscommunity.org – LAMS Learner

Question and answer - Pro Teacher role

Question 1:

Given your role, what do you see as the key issues about a potential wider adoption of interactive whiteboards at your school? Reflect on the answers from others, then discuss this in the following forum.

Answer:

```
I think as teacher we should encourage students to learn with
the latest technologies available...
```

Figure 7: Tool Leaner's Page

Export Portfolio URL

This screen allows the learner to export in plain HTML the contributions of the learners in this learner's groups – that is, a record of each screen seen by this learner during the sequence of activities.

Admin Contract

Administration URL

All tools define some technical metadata on installation. Tools must also supply a URL which may be used for updating this information, along with any tool specific configuration details.

The administration URL should also allow access to any administrative tasks e.g. reviewing a tool log, runtime stop/start of the chat server, or a monitor method which shows which Chat rooms are active.

External Tool Wrappers

LAMS version 1 has been integrated with a number of open source and proprietary learning management systems. After releasing these integrations, educators have pointed out that they would like to use the LAMS tools together with the tools from their own learning management system within a LAMS learning design.

Thus in LAMS 2.0, we have conceptualised external tool wrappers for different learning management systems that could allow their tools to be "LAMS enabled".

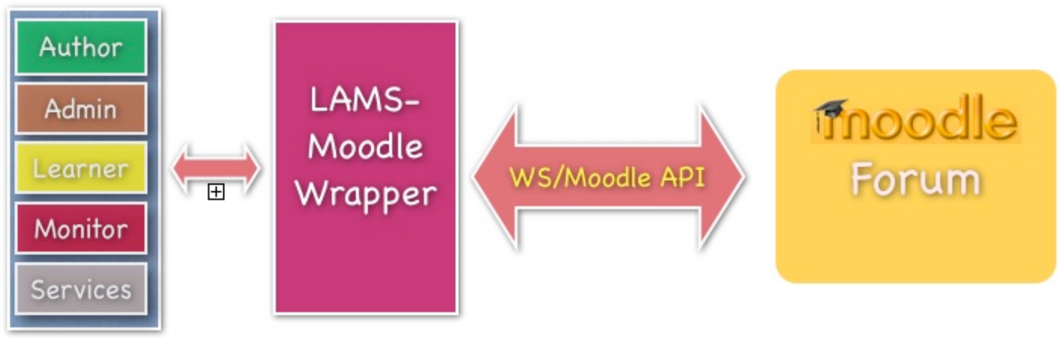


Figure 8: LAMS External Tool Wrapper

These tool wrappers allow the external tools to comply with the LAMS Tools Contract as well as handling authentication and authorization. By doing so, these external tools can be included in LAMS sequences. Figure 8 and Figure 9 show an example on how a tool wrapper for a Forum tool from Moodle would be able to work within LAMS.

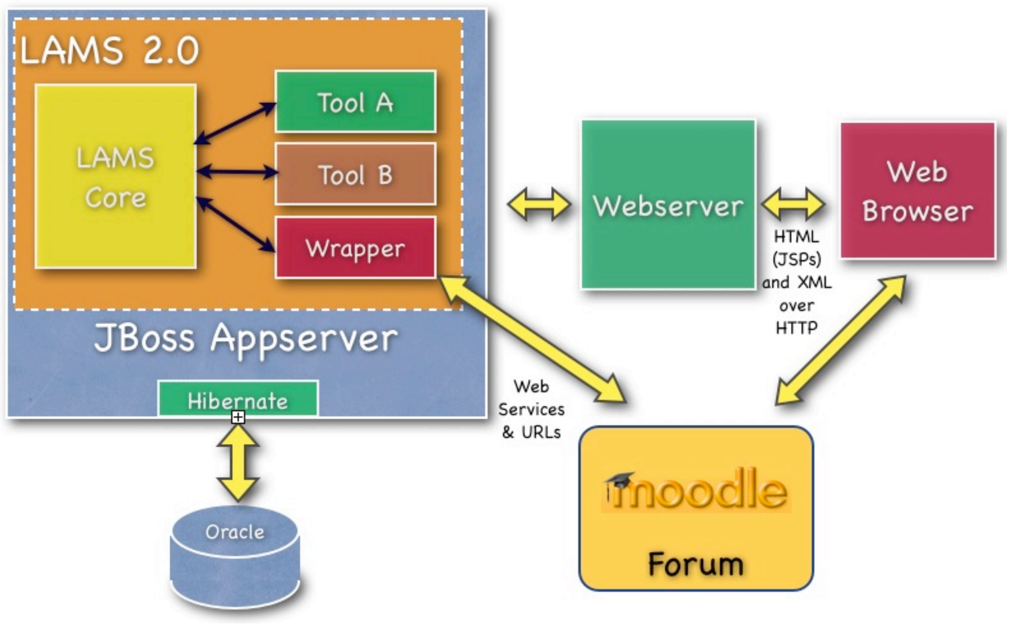


Figure 9: Moodle External Tool Wrapper

Conclusion

The design of LAMS version 2 was based on extensive user feedback as well as the outcomes of formal evaluations of LAMS version 1 implementations. New and improved features were incorporated into LAMS V2 to: improve pedagogical support and re-use; enhance collaboration among educators (as both authors and monitors); improve the user interface and overall user experience; and to provide greater robustness, scalability and reliability. Key features supported by LAMS V2 include: edit-on-the-fly, branching, rich multimedia; “offline” activity information; optional reflections for all tools; a wider range of grouping options; “portfolio export” and support for languages other than English.

One of the key architectural improvements of LAMS V2 was the introduction of the “LAMS Tools Contract” – a set of expected behaviours about how an activity tool interacts with the LAMS Core “workflow engine” in terms of four areas: Author, Monitor, Learner and Administration. The Tools Contract fosters easier creation and integration of new activity tools into the LAMS framework, as well as integration of external tools via a “tools wrapper”. The LAMS Tools Contract also provides a set of innovative functionalities that could inform the next generation of interfaces between e-learning platforms and activity tools.

References

Dalziel, J. R. (2003a). Introducing LAMS: An EML/LD “inspired” system. Presentation for the Valkenburg group meeting, Vancourver, Canada, February 23rd, 2003.

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

Dalziel, J. R. (2006). The design and development of the LAMS Community. [Online] http://lamscommunity.org/dotlrn/clubs/educationalcommunity/lamsresearchdevelopment/forums/attach/go-to-attachment?object_id=311748&attachment_id=311750

LAMS Foundation (2004). Trans-Tasman e-learning first: New Zealand hosts global LAMS launch. [Online] http://www.lamsfoundation.org/news/index.html#3

LAMS Foundation (2005a). Innovative e-learning system to be launched as open source software this week. [Online] http://www.lamsfoundation.org/news/index.html#4

LAMS Foundation (2005b). Articles/Reports/Research on LAMS. [Online] http://www.lamsinternational.com/CD/html/resources.html#ArticlesResearchLAMS

LAMS Foundation (2005c). LAMS launches the LAMS Community: the birth of "open source teaching." [Online] http://www.lamsfoundation.org/news/index.html#6

LAMS Foundation (2006). LAMS 2.0 Feature List. [Online] http://wiki.lamsfoundation.org/display/lams/LAMS+2.0+Feature+List

Laurillard, D. (2001). Rethinking University Teaching: A Conversational Framework for the Effective Use of Learning Technologies. London: Routledge.

Masterman, L. & Lee, S. D. (2005). Evaluation of the Practitioner Trial of LAMS: Final Report. [Online] http://www.jisc.ac.uk/uploaded_documents/LAMS%20Final%20Report.pdf

Russell. T., Varga-Atkins, T. & Roberts, D. (2005). Learning Activity Management System Specialist
Schools Trust pilot . [Online] http://partners.becta.org.uk/page_documents/research/lams.pdf

Spring Framework (2006). The Spring Framework. [Online] http://www.springframework.org/