

A Step from CAA toward MAA: A Test-question Repository

Philip Grew, Elena Pagani
Dept. Information Science and Communication
Università degli Studi di Milano, Italy
{grew,pagani}@dico.unimi.it

Ivan Longhi, and Andrea Cardinale
Fondazione Rete Civica di Milano
ivan.longhi@rcm.inet.it
Accenture S.p.A.,
andrea.cardinale@accenture.com

ABSTRACT

The authors have previously been active in developing a learning management system, applying it to computer-assisted assessment (CAA) tasks, improvising in-class activities using students' cellphones as text-messaging terminals, and devising architectures and prototypes to apply ad hoc networking to distributed computer-mediated communication in the classroom. The need to easily exchange test questions among various server platforms has now led us to experiment with the creation of an online repository for test-question trading.

This paper describes the development of a repository originally designed to import and export questions from one learning-management platform to another. In keeping with the open-architecture principles that our project has adhered to from the outset, the repository is an external application. Not only does this enable other conventional platforms for wired, client-server e-learning, such as Moodle, to access questions in the repository, it reduces test questions to a bare-bones format optimal for importing into our content server for mobile devices, thus representing a first step toward mobile-assisted assessment (MAA).

KEY WORDS

testing and assessment issues of web-based education; mobile e-learning; innovative teaching and learning technologies for web-based education; authoring systems for web-based education.

1 Background and Aims

Over the last several years, the authors and their colleagues have designed a learning management system (LMS), originally for use by small businesses[1], and applied it extensively to computer-assisted assessment[2]. This LMS differs from many tools that support web-based teaching and learning in that it is neither focused on one specific purpose – like assessing students or publishing course content or maintaining a learning community – nor does it integrate all these tasks into a single centralized system. While the “JLI-Just Learn It!” LMS[3] can certainly be used as an all-in-one platform, it was designed from the get-go to integrate with other, existing systems[4].

For example, this openness to integration made it possible to introduce JLI! while preserving existing class fo-

runs hosted on the FirstClass[5] community server[6] run out of our department[7]. The forum functionality is one of many that can either be incorporated into the JLI! server or integrated as an outside service (see Figure 1). It should be noted that FirstClass is a proprietary system not designed to allow such integration, so the JLI! structure had to bear the brunt of the compromise. Open architecture also made it possible to outsource authentication to an existing system for assigning students computer-lab access through LDAP[8] and to import student-status data from the registrar's restricted-access database. JLI! has thus been used extensively in our testing program for English as a foreign language.

The learning-content core of the JLI! platform was designed to maintain SCORM compatibility[9], so that content may be packaged into a transferable .zip file, for example. JLI! accomplishes this at the level of the *module*, which – when applied to assessment content – means a series of test questions of one given type (e.g. either multiple-choice or fill-in-the-blanks but not mixed). This is a practical solution that packages exercises together with any associated media, since the .zip file may contain audio, video or graphics. However, as Meyen and Miller aptly point out, while such a solution indeed isolates an “object-based instructional unit,” this does not correspond to “the smallest useful standalone content component that can be used as a part of a lesson”[10]. In other words, situations may arise in which it is desirable to transfer content with greater granularity.

In practice, there may often be little reason not to simply copy the instructional content from one course or module to another, especially because the reuse of course materials typically takes place during the process of editing or preparing materials for a new course or a new test. In the case of our own actual assessment practice, using JLI! as the platform for serving questions, a testing module might contain from, say, eight to 50 multiple-choice¹ questions. One or two new questions could easily be manually pasted into a module in edit mode (see Figure 2, below) or an entire new set of questions could be prepared in .odt format[11] for importing into JLI! Such practices show their limitations in two main situations. The first

¹For the purposes of this work, the terms “multiple choice” and “single choice” both refer to questions with one correct distracter (as opposed to “multiple-answer” questions).

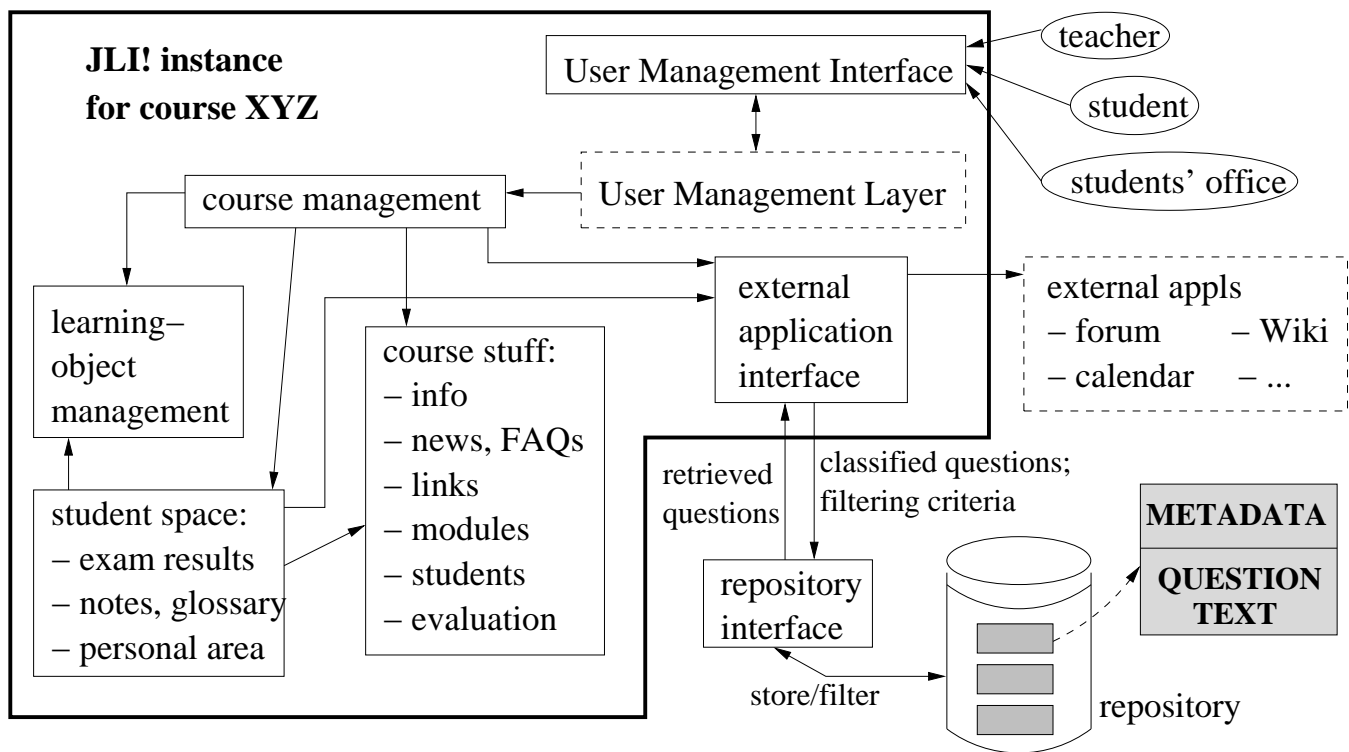


Figure 1. JLI! architecture.

arises when more than one instructor is involved in preparing materials, which obviously calls for more sophisticated collaborative tools than simple copy-and-paste or attaching word-processor files to an e-mail message. The second arises when test questions need to be recycled onto platforms other than the one on which they were originally created. This second case also highlights the issue of granularity in content reusability, i.e. we will want to consider test questions on a one-by-one basis, perhaps with some metadata to help us sort them out.

The need to cross over from platform to platform has in practice been especially felt in the case of mobile devices, in part because of the authentication possibilities such devices offer. Indeed the authentication potential is the very reason two-dimensional bar-code technology has now transferred from education [12] to air travel[13]. These approaches to authentication share a core principle with the technique the authors currently apply to secure on-screen assessment events: pre-existing electronic credentials are cross-referenced to exam candidates' physical presence at the test center. However, rather than locking down the terminals used to access test content as done with wired PCs[14], test content would be locked 'up' to mobile devices residing in verified hands.

The authors have for some time been involved with experiments using mobile devices for mediated classroom interaction, including pop quizzes. While much can be accomplished in this arena even with such primitive media as texting on cellphones (which have the advantage of be-

ing universal among our students)[15], we have also been experimenting with more sophisticated systems. The distributed architecture we envision[16] makes a repository that is networked in the traditional fashion an interesting point for synchronization.

Furthermore, because the repository is already external to the LMS (see Figure 1), it can obviate the need to interface the server that optimizes content for a variety of mobile devices with an LMS designed to offer a wealth of features to clients who are logging on through traditional web browsers. Like our LMS, the repository supports a variety of question types (cloze, true/false, etc.) but in simplified format ready for export or editing.

2 Editing and Exchanging Test Questions

Let us take as an example the creation of a multiple-choice question for a required first-year English as a foreign language (EFL) course. Once the question has been typed or imported into JLI!, its only significant parameter is the indication of one correct distracter. Other information, such as how much time is available to answer the question, the reading or listening passage on which the question is based or an image to which it refers, is not embedded at this level but at the level of the module.

As can be seen in Figure 2, one of the possible edits is the addition of a distracter to the multiple-choice question. The JLI! interface always displays one distracter field more than the current number of distracters, thus allowing

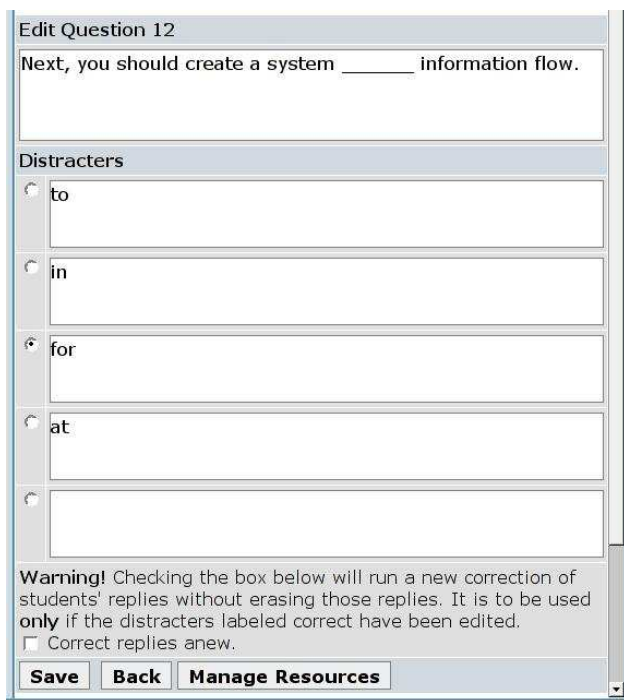


Figure 2. A multiple-choice question in author mode

unlimited distractors to be added. In practice, few teachers use more than six distractors, however.

Figure 3 shows the same question in preview mode, the way it would look after a student has correctly answered the question but not yet turned in the module containing it. The export function is not visible in either Figure 2 or Figure 3 because, when the repository is enabled, export from JLI! takes place at the level of the entire module. The export button brings up a list of all the questions in the module for the teacher to choose from by checking them off. Each question is then separately equipped with the metadata that will accompany it in the repository.

The export dialog requires the instructor to assign the selected question a category, either from those existing in the repository or by creating a new one. The instructor is also asked to rate the question's level of difficulty. These functions were added to JLI! so the existing (familiar) interface could be used as a client to the repository. It should be noted, however, that authentication is kept separate. Because this is independent at the repository level, no problems arise when the same instructor has different LMS installations that she or he uses to access the repository.

3 The Repository

The repository we are using was originally developed as a student BA project[17] using XML-RPC specifications[18] with the specific objective of creating an online environment for sharing and organizing questions for use of different LMS platforms. Once exported from JLI!, our example

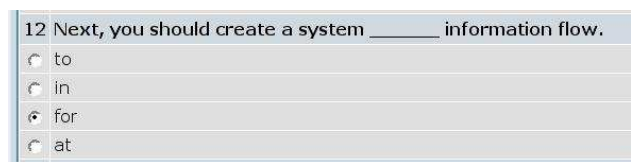


Figure 3. The same question as presented in JLI! to the student

question thus appears listed in the repository as shown in Figure 4.

This window is the result of a repository search for questions in English at level 2 in the grammar category. Every question in the repository has the question type attribute, supported types being: multiple choice, multiple answer, true-false, cloze, and Likert-scale. While category is a mandatory attribute of every question in the repository, assigning a level of difficulty at the time of export is optional (and in practice often taken care of later for several questions in a series, possibly after statistics on their use show how difficult students found them). The language attribute may also be null (i.e. left blank by the teacher).

Each question also has an owner and, unless it is shared (an attribute corresponding to the icon in the first column of the list in Figure 4), will be visible only to that user. Whether to display questions uploaded and shared by other users is thus an option set for each search. A list of questions in the repository can be generated by a search of attributes, as shown in the example, and/or by searching for a text string.

Figure 5 shows our same question open in the repository for detail view and possible editing or deletion. The repository makes it clear that this is a single-choice (i.e. multiple-choice) question but it should be noted that the question used as an example reflects how instructors authoring a question into a system may be limited to the multiple-choice format (by platform compatibility, among other factors) and thus tend to apply this format for a closed-end cloze question. The question-list view shown in Figure 4 gives no indication of distractors, these being displayed at the level of single-question view shown in Figure 5. In addition to keeping the question list readable even in the case of rather long distractors, this choice enables the list to show questions of all types (whose view at the single-question level is, of course, not comparable). Cloze questions imported from JLI! may also have long answers because, while they are designed for automated correction via text-string comparison, numerous variant responses may be considered correct. Other LMSs taken into consideration during repository design include: Chef[19], Moodle[20], Multicommunity[21], ATutor[22], and Spaghetti Learning[23]. The use of SCORM specifications and open-architecture principles should assure that developing repository clients for these other LMSs will not prove excessively onerous. Our sample question in XML-RPC format for import-export looks like this:



Figure 4. Our example question among others listed in the repository following a simple search

```

<!-- question token containing ID,
      category, and language -->
<question id="12" category="Grammar"
      language="en">
  <!-- question text -->
  <text>Next, you should create a
    system _____ information
    flow.</text>
  <!-- question difficulty -->
  <difficulty>0</difficulty>
  <!-- question type carried over from
        question ID -->
  <singlechoice>
  <!-- list of distracters with attribute
        set for correct option (one
        required, one allowed) -->
    <choice correct="0">to</choice>
    <choice correct="0">in</choice>
    <choice correct="1">for</choice>
    <choice correct="0">at</choice>
  </singlechoice>
</question>

```

4 A Wireless LMS

Because the use of mobile devices (PDAs, 3G cellphones, tablet PCs, etc.) and of wireless networking technologies (especially Wi-Fi and Bluetooth) is now widespread, these technologies may be used for mobile learning, as many experiments in the literature[24, 25, 26, 27] show. Usually, these experiments deploy a centralized architecture, with the mobile devices used either (i) for personal work,

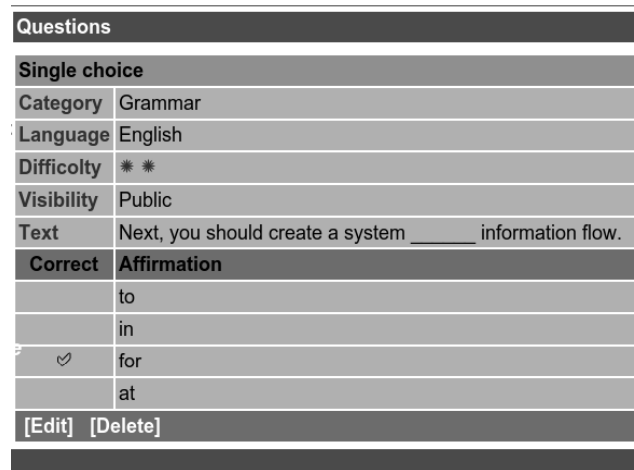


Figure 5. The question displayed in the repository

without interacting with other devices; or (ii) for interactions channeled through a central server. But centralized architectures do not scale well to systems with hundreds of users, such as a university campus. Mobile ad hoc networking has the potential to allow direct interaction among devices, thus guaranteeing scalability and enabling cooperation even in the absence of other infrastructure.

Our vision[28] involves an ad hoc wireless-network infrastructure that will allow users to interact and share content from anywhere, including roaming. In particular, a *virtual classroom* can be built using only the teacher's and students' devices, allowing them to share lesson-related material and to store it on personal devices for subsequent

work. In this context, the repository described above may be used as a sort of “meeting point.” Instructors can download content related to a planned lesson onto their laptops. Such content is then shared among students in class through an ad hoc network using their own devices. Content produced during the lesson is stored on the instructor’s device and then synchronized with the repository, which thus represents the *memory* of the work done offline. This also may be set up to allow a learning object to be reused by another instructor or by students who were absent.

We experimented[29] with modifying a centralized architecture – namely ActiveClass – to make it distributed. The virtual class consists of two components: a repository that synchronizes relevant information about students and courses, and the application for user interaction. Through the former, instructors can populate their laptops before a lesson, e.g. with students’ profiles and authentication data. The latter performs access control, supports polls, supplies a question-and-answer system, and appropriately shows contents through a browser, by using the information previously downloaded onto the teacher’s device. Poll results, questions and answers, contents produced during the lesson, and updated student profiles are then uploaded to the repository when the teacher again has access through the wired/wireless campus backbone.

We implemented a prototype of such infrastructure, with portable PCs and a PDA as clients over both Bluetooth and Wi-Fi as wireless technology. Experimental results have held out great promise, confirming the soundness of a distributed approach to support offline work, while using repositories to maintain content alignment.

5 Conclusions

Integrating the repository into our existing content platform was possible thanks to open-architecture principles. In practice this meant, first of all, that system integration was achieved without forcing changes in tools and methods on teaching faculty who use various systems for managing their specific course content, including course-specific Wiki servers and forum systems.

Now, the openness is proving useful as new systems are developed to support mobile technology. The sad fact is that some existing content would not be worth the effort to preserve without the prospect of reusing it for different purposes. This is especially true when such content now exists on platforms that do not incorporate automated export functionalities. It is often easier simply to retype test questions.

However, the design of a repository as an open system for organizing and editing test questions offers attractive potential for making “economical” use of retired test questions as material around which to construct a pop quiz. If such a pop quiz can be easily managed over an ad hoc network of the devices students already carry with them to class, the ‘old’ questions for the pop quiz might be easily selected from among those that had stumped the greatest

number of students during a previous exam held on browser screens in a locked-down laboratory.

What remains to be tested is the extensive application of the repository to content-synchronization as mobile devices actually become a feature of everyday classroom life. New content developed specifically with mobile devices in mind will also need to be accessible for later reference in the same browser interface used for other course materials. Enabling such a process to take place seamlessly appears likely to create an additional role for the repository.

Acknowledgments

This work has been partially supported by the FIRST Project “Wireless platform for supporting mobile ubiquitous e-learning”.

References

- [1] F. De Cindio, I. Longhi, and L. A. Ripamonti, “Design Issues in Developing a Learning Environment for Small-sized Enterprises Communities”, *Proc. Intl. Conf. on Technology-Enhanced Learning (TEL’03)*, Milan, Italy, 2003, 133-140.
- [2] P. Grew, I. Longhi, F. De Cindio, and L. A. Ripamonti, “Applying an LMS to Large Language Classes”, *Proc. IASTED International Conference on Web-Based Education (WBE’04)*, Innsbruck, Austria, 2004, 480-484.
- [3] Ivan Longhi, Philip Grew, *JLI!-Just Learn It!*, <http://sourceforge.net/projects/jli>.
- [4] P. Grew, I. Longhi, and E. Pagani, “Functional Architecture of a Web-based Distributed System for University Curricula Support,” *Proc. IASTED International Conference on Web-Based Education (WBE’05)*, Grindelwald, Switzerland, 2005, 332-337.
- [5] Open Text Corporation, “FirstClass Communications Platform,” <http://www.firstclass.com/AboutFC/>.
- [6] Rete Civica di Milano, <http://www.retecivica.milano.it/>.
- [7] Community Informatics Lab, “Rete Civica di Milano,” <http://www.lic.dico.unimi.it/>.
- [8] J. Hodges and R. Morgan, “Lightweight Directory Access Protocol (v3): Technical Specification”, *RFC 3377*. Work in Progress. Sept. 2002.
- [9] Advanced Distributed Learning, “Sharable Content Object Reference Model,” <http://www.adlnet.gov/scorm/>
- [10] E.L. Meyen and J.R. Miller, “A System for Creating and Managing Reusable Learning Objects,” *Proc.*

- IASTED International Conference on Web-Based Education (WBE'07)*, Chamonix, France, 2007, 353-358.
- [11] OpenOffice, "Writer," <http://www.openoffice.org/product/writer.html>
- [12] K. Nagaoka, "A Response Analyzer System Utilizing Mobile Phones" *Proc. 4th IASTED Intl. Conf. on Web-Based Education (WBE'05)*, Grindelwald, Switzerland, 2005, 579-584.
- [13] B. Hem, "Cell phone may take boarding pass's place" *The Houston Chronicle* (October 12, 2007).
- [14] G. Biacchi, P. Grew, and E. Pagani, "A Trusted Assessment Environment in a Network-booted OS," *Proc. IASTED International Conference on Web-Based Education (WBE'07)*, Chamonix, France, 2006, 415-420.
- [15] P. Grew and E. Pagani, "Channeling the Bricks-and-mortar Lesson onto Students' Devices," *Proc. IASTED International Conference on Web-Based Education (WBE'06)*, Puerto Vallarta, Mexico, 2006, 160-165.
- [16] Philip Grew, Elena Pagani, "Towards a Wireless Architecture for Mobile Ubiquitous E-Learning," *Proc. Intl. Workshop on Learning Communities in the Era of Ubiquitous Computing*, Milan, Italy, 2005, 20-29.
- [17] A. Cardinale, *Un approccio user-driven al consolidamento di un LMS open-source* (BA Thesis, Università degli Studi di Milano, 2004).
- [18] "XML-RPC" <http://www.xmlrpc.com/>.
- [19] "Chef" (now part of the Sakai project) <http://www.chefproject.org/>.
- [20] "Moodle" <http://moodle.org/>.
- [21] "Multicommunity" <http://multicommunity.di.uniroma1.it/>.
- [22] "ATutor" <http://www.atutor.ca/>.
- [23] "Spaghetti Learning" <http://www.docebo.org/>.
- [24] ActiveCampus Project, "Active Campus – Explorations in Community-oriented Ubiquitous Computing," University of California at San Diego, <http://activecampus.ucsd.edu/>.
- [25] Lancaster University, "The Guide Project - Overview" <http://www.guide.lancs.ac.uk/>.
- [26] K. Yatani, "Musex Project" <http://www.dgp.toronto.edu/~koji/project/musex.html>.
- [27] M. Ancona, G. Dodero, V. Gianuzzi, O. Boccini, A. Vezzoso, A. Traverso, and E. Antonacci, "Past Project" <http://www.beta80group.it/past/>.
- [28] Philip Grew, Elena Pagani, and Francesco Giudici, "From E-Learning to M-Learning: Architectures to Support University Teaching" in *Architecture Solutions for E-Learning Systems*, C. Pahl, ed. (Idea Group Inc., 2007).
- [29] G. Caracciolo, *Deployment of a Distributed Platform for Wireless E-Learning* (BA Thesis, Università degli Studi di Milano, 2005).