An Open-Source LMS Evolves as Learning/Teaching/Testing Environment

Philip Grew, Ivan Longhi, Elena Pagani, Fiorella De Cindio, and Laura Anna Ripamonti

Computer Science Dept., Università degli Studi di Milano
{grew, pagani, decindio, ripamonti}@dico.unimi.it;
ivan.longhi@rcm.inet.it

Abstract. At the Dipartimento di Informatica e Comunicazione of the Università Statale di Milano an opportunity for exam automation was seen in an earlier initiative that had led to the development of an open-source learning management system. This LMS, known as "Just Learn It!" (or JLI!) was then applied to English-as-a-foreign-language placement tests for large numbers of incoming first-year students, thus replacing automated, paper-based testing systems such as QuizIt [1]. Initial studies have now been undertaken in using this on-screen system to automate written exams and mixed oral-written exams in other subjects, as well. The change in purpose has led to a number of considerations that have influenced the development of JLI! These, in turn, promise to ease the adoption of distance-learning technology in transitional and blended situations.

Keywords: Learning Management Systems; teaching environment; testing environment; teachers and students interactions.

1 Introduction

A Learning Management System ("Just Learn It!" – JLI!) has been deployed at the University of Milan, initially aimed at supporting e-learning in small enterprises. It has then proved useful in automating the test-administration process in several courses. Its adoption in courses concerning heterogeneoues topics drove its development, sometimes in unexpected ways. JLI! is currently used not only for student assessment, but also for e-learning. In this paper, we present a number of considerations under the point of view of both teachers and students about how JLI! changed their work. The paper is structured as follows: in sec.2, an overview of the platform is provided. In sec.3, we discuss the impact JLI! has on both teachers' work and teaching quality. In sec.4, we analyze how JLI! changes the students' behavior, as observed in the courses that adopt it. In sec.5, some concluding remarks are proposed.

2 Overview of JLI!

JLI! [3] was initially designed as a Learning Management System (LMS) for small-to-medium sized enterprises. It has recently been adopted as a testing platform for several Computer Science courses at the University of Milan and evolved into an integrated platform for teaching and student assessment. Fig.1 shows the main characteristics of JLI! and the basic functions it supports. These characteristics are discussed in greater depth in the following sections, after our description of the functionalities provided. Interested readers may refer to [3] for further details on JLI! architecture.

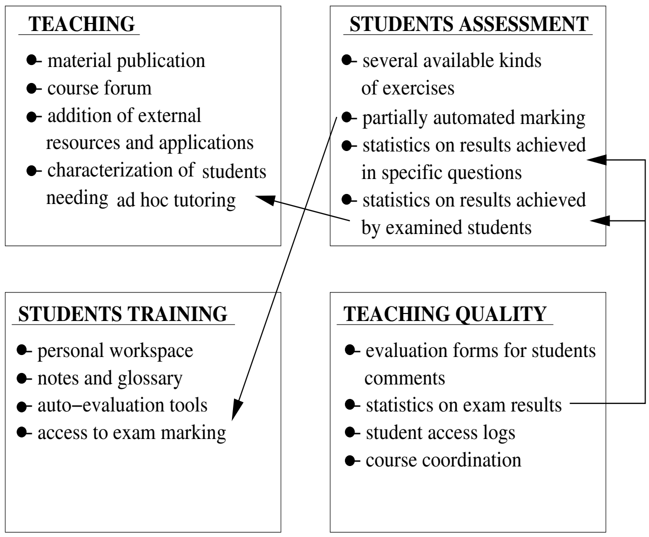

Fig. 1. Main JLI! functionalities and characteristics.

JLI! supports teaching by providing instructors with web-like publishing facilities. But it also allows community building by involving the teacher of a course with her or his students, thus boosting interactivity. JLI! includes tools to monitor students assignments, thus bringing to light the needs of certain students for special support in dealing with the course material. JLI! can be easily integrated with external resources and applications.

Student assessment is supported by allowing teachers to create and manage several kinds of learning objects, from essay questions to automatically graded quizzes of various types. Several statistics tools are implemented in JLI! In addition to easing the processing and publication of test results, these provide information either about a students performance on a specific object or about achievement across multiple attempts at passing the exam.

Teaching quality can be increased interactively through course evaluations provided by the students or on a pull basis by carrying out analysis of the statistics and the logs of the students use of the LMS. Indeed, each student owns a personal workspace in the frame of a JLI!-supported course, where all the work performed for the course is collected.

An important characteristic of JLI! is that it enables student authentication, which has a twofold benefit:

- course-related resources can be made accessible to students subscribed to a specific course only, thus obviating the need for ad hoc publishing by the teacher. Similarly, forum contributions are maintained confidential among the course participants;
- it eases the teacher's task of checking the identities of the students who take an exam, normally a subset of the students enrolled in the course. Absent students are immediately singled out. By contrast, when paper-based exams are used, students may unintentionally fail to turn in some or all of the exam sheets or maliciously claim that they took the exam when in fact they did not, thus creating unpleasant situations by accusing teachers of having lost their exam.

Two techniques may be used for student authentication. A temporary expedient, used for example to identify students during entrance exams on an ad hoc basis, consists in exploiting a password-generation algorithm implemented by the JLI! developer and then distributing the passwords to students when they arrive to take the test.

During the second stage of development of JLI! special attention was paid to integrating the LMS with the other existing and possible future tools and resources available to faculty and students in the information sciences programs. In order to achieve single sign-on (SSO), students are authenticated through lightweight directory access protocol (LDAP) on the SILab domain server, so as to assign students a single user account to be kept for the duration of their university career. A beneficial side effect of this policy is that, immediately following registration at the university, when students take the English placement test, they are forced to activate their SILab account and thus become acquainted with what is to become one of the most important resources at their disposal.

A further degree of integration also embraces the WebCen system. By developing an application based on Web Services, we have made a set of information about a student's "learning profile" available to teachers who use the JLI! system. This information consists of the fields that the registrar's office makes available to teaching faculty. When a teacher opens a student's user form, she or he can see which high school issued the diploma, year of enrollment, degree program, number of exams passed, and number of university credits acquired. The student services office views a set of dynamic data that would be hard for a centralized structure to keep up to date. When a student edits the email address at which she or he wishes to receive notices, for example, the Web Services interface transmits the information to the student services office and hence to the other faculty members. At its current stage of development, JLI! is paying particular attention to the integration of a teaching application with an important teaching-support facility. This both allows faculty members to provide public replies to student questions and assists them in building a learning community within their subject matter. Such communities that hinge on a given subject are nevertheless interfaced with one another, thus creating online space for the establishment of the larger community of students and faculty of information sciences.

3 Teachers' support

JLI! supports teaching both in lecture scheduling and during the whole student testing procedure. As far as the testing environment is concerned, JLI! offers several facilities that ease test preparation and evaluation compared to traditional paper-based systems. In JLI!, an exam corresponds to a `course`, composed of several learning objects, named `module`s. A `module` includes any number of exercises and questions, all of a certain type. `Module`s can be of 10 types, six of which are useful for testing.[1] In particular, a test `module` may contain:

- essay questions, whose marking is manually performed by a teacher once a student terminates the exam session;
- exercises with file upload: this kind of `module` is equivalent to essay questions, but it allows a student to write a response in a file including figures, drawings, plots or complex formulas, rather than having at her or his disposal only a form for ASCII text insertion;
- True-false questions, multiple-choice questions, Likert quizzes or cloze (fill-in-the-blanks) quizzes. These `module`s are automatically corrected by the system, which compares the response provided by a student to a correct answer input *a priori* by the teacher. Both the questions and the distracters (possible responses) of multiple-choice modules can be scrambled to prevent students from cheating.

All these `module`s may exploit multimedia content (audio, video) for questions and quizzes, which has proved useful, e.g. for English language exams. This also makes it simple to reproduce a classroom slideshow for asynchronous reference.

This `course` = { heterogeneous `modules` } structure makes it easier to use different types of exercises for a given exam than with traditional paper-based systems, which require the teacher to prepare an appropriate number of copies of each module, distribute a copy of each module to students, and then sort out the different parts, for example holding essay questions for manual correction while delivering multiple-choice quiz forms for scanning with QuizIt. The way QuizIt is used at the University of Milan makes it error prone – mistakes can be unintentionally inserted by the clerks in charge of preparing the forms – and time consuming – the exam text must be given to the clerks well in advance of the test date and results require several days to be delivered to teachers. By contrast, JLI! does not incur in these problems. JLI! gives teachers more autonomy in adapting exam texts also soon before the exam. Automated marking is provided for several learning objects besides multiple choice questions. Furthermore, with JLI! merging the results achieved by a student in different modules is

[1] The remaining types of `module`s are discussed in sec.4.

automated. Teachers can freely adapt results by variously weighing the different modules. They can also intervene in the automated marking process, changing the responses to be considered correct *a posteriori*.

JLI! has also proved to ease teachers' work in the case of courses shared by two degree programs. Both the Computer Science degree and the Information Technology for Telecommunications degree of the University of Milan involve a Computer Networking course. Part of this course is common to both degrees, but certain topics are discussed in only one of the two degrees. JLI! allows a teacher to prepare `module`s concerning the common topics, which are copied in both `course`s (such as the "layers 1-6" `module` in fig.2), and then add to each `course` appropriate `module`s according to the respective syllabus. When a student logs in, she or he automatically accesses only the `module`s for the `course` she or he is enrolled in; which is considerably simpler than managing to deliver each student the appropriate test sheets depending on the student's major.

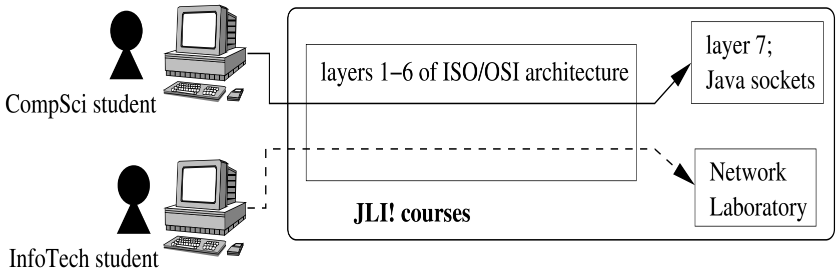

Fig. 2. Example of modules reusage.

A further facility, which is currently under development, allows test authors to build an **exercise repository**. Questions and quizzes are randomly extracted from the database when a student logs in, thus automatically building different exams for different students, which makes cheating extremely hard. Several issues are involved in the deployment of this feature. Questions should be classified according to difficulty, so that the exams served to different students are comparable from this point of view. Random selection of questions is already implemented within a given module but involves no classification of the questions themselves.

A number of advantages have been reported elsewhere when using JLI! for our English-language courses [2]. Using it for other bricks-and-mortar courses highlighted different features. Some exams include both a written and an oral part. With JLI! a threshold for access to the oral part can be set, defined on the basis of the results a student achieves in automatically corrected modules. If those modules' results are shown as soon as the student terminates them, students know immediately whether they may proceed to the oral part. Moreover, teachers do not need to grade the manually-corrected modules for low-scoring students, thus expediting the exam procedure.

The JLI! interface has proved useful during oral exams as well. Indeed, it allows quick retrieval of the examined student's written test and comparison of that student's results to the class average, thus highlighting weaknesses that can drive the discussion. It was found that incorrect answers may be less serious than they appear: misunderstandings arise from some students' meager aptitude in expressing their thoughts in writing. JLI! integrates tools to aid publishing exam results (fig.3). Teachers can mark open-answer questions directly on screen. The grades attributed to each module are shown in JLI!, and can be exported to files, either as HTML or ASCII text with tab-separated fields. This latter format can be easily imported in spreadsheets for further result editing or processing.

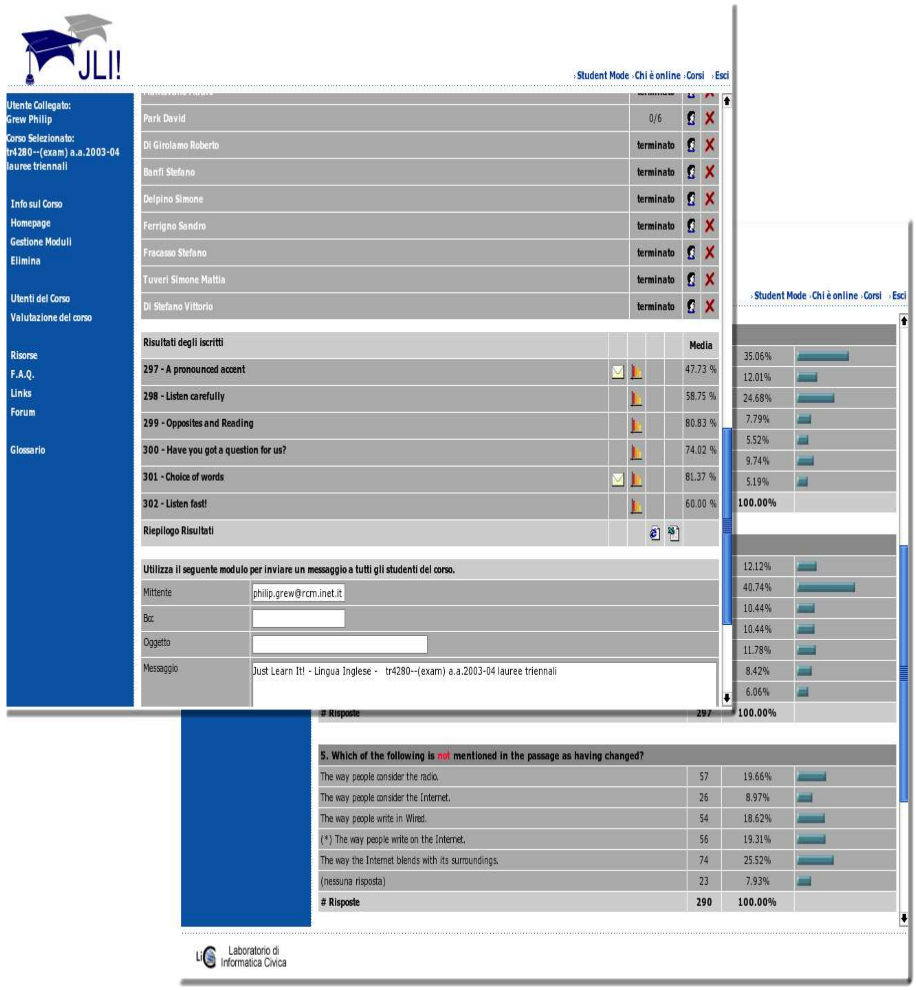

Fig. 3. JLI! screenshot showing the teacher's interface to exam marking.

As support for teaching, JLI! builds a database of past exams, which can be useful to monitor the results attained by students in specific topics of the course syllabus. For instance, questions about certain topics that regularly produce low scores may reveal that students find that topic hard to master, thus requiring the teacher to spend a larger amount of time discussing it in future editions of the course. Similarly, average results below expectations may show that the overall syllabus is far too heavy for students, or that course prerequisites are not met in previous courses, thus requiring that the syllabus be adapted both in terms of the topics involved and in its comprehensiveness. A history of each student's attempts to pass the exam may reveal students needing particular tutoring to deal with course topics. Usually, identifying such students is very difficult, given that required courses average some 150 students per section.

Online course registration enables teachers to identify the students registered for the course beforehand, along with their e-mail addresses. JLI! allows broadcast email to all registered students, for instance to publish urgent notices about the course, which might otherwise go disregarded if posted on a traditional bulletin board or web page. At any time teachers can check how many registered students still need to take the exam.

4 Learners' support

JLI! also supports students during the learning process. As mentioned, JLI! includes four types of `module`s that are not for testing use. Three of them allow students to access contents related to a course – such as the class schedule agenda, slides used by the teacher, related papers in the literature, a course forum, and so on. Moreover, a *self-access* `module` is available with the hidden-answer questions `module`. Using this `module`, students can prepare for exams with previous tests: the exam exercises and questions are served for a student to answer. After that, the student can see the correct answers supplied by the teacher and estimate whether she or he is ready to take the exam and what topics require further studying. Similarly, this self-evaluation tool can be used by students *before* taking an advanced course, in order to gauge whether they fit the course prerequisites or not and, perhaps, to "fill the gaps" in their preparation.

It is often noted that "students learn more during exams than during lessons." Certainly, one is more likely to remember information impressed on a mind under stress than words that flow by while one is relaxed – and possibly inattentive – in the classroom. JLI! makes it possible to show students the correct answers on automatically-corrected `module`s. In case of mixed written/oral exams, this feature may help students prepare the oral part by studying in greater depth the topics they failed on the written part. In any case, students discover and remember the correct answers and can check their mistakes once at home. Similarly, a teacher may make available the markings for manually-corrected modules once they have been completed. By monitoring students' accesses after the exam [2] it has been observed that the number of students checking the corrections of their exam in JLI! is far greater than the number of students who show up at office hours for this purpose, with obvious resulting benefits in learning.

Subscribed students may submit their own course evaluation. This facility, together with the statistics that can be collected about exam results, may help teachers adapt the course syllabus to student behavior, and possibly cooperate for a better coordination of their respective courses.

5 Conclusions

An LMS originally designed for distance learning was shown to be a useful tool in automating the test-administration process, even when the users included several language teachers who were not necessarily enthusiastic about computerization. The availability of the tool has drawn these courses and others, all aimed at large groups of students where tracking individual performance is a daunting task, closer to a situation that in fact consists of blended learning. This achievement, though no mean feat in itself, is the rather unsurprising outcome of having an excellent learning-management tool available.

Less predictable, however, has been the ability of feedback from teaching faculty to influence programmers' design choices. Sound modular design principles adopted at the outset have meant that a system originally conceived as a vehicle for distance learning has evolved into a more versatile component of several student-management and learning-content systems. Together, these systems make up the information-automation support context for a community where no clear boundary need exist between flesh-and-blood contact and computer-mediated communication. Thus the distance-learning paradigm can be fruitfully applied even over very short distances such as those encountered in a very large lecture hall.

Acknowledgments

This work was partially funded by the Italian Ministry of Education, University and Research in the framework of the FIRB "Web-Minds" project.

References

1. Chemware, "QuizIt Online Testing", `http://chemware.co.nz/quizit.htm`
2. Grew, P., I. Longhi, F. De Cindio , L. A. Ripamonti, "Applying an LMS to Large Language Classes", *Proc. IASTED International Conference on Web-Based Education (WBE'04)*, Feb. 2004.
3. De Cindio, F., I. Longhi, L. A. Ripamonti, "Design Issues in Developing a Learning Environment for Small-sized Enterprises Communities", *Proc. Intl. Conf. on Technology-Enhanced Learning (TEL'03)*, pp. 133-140, Nov. 20-21 2003.