FUNCTIONAL ARCHITECTURE OF A WEB-BASED DISTRIBUTED SYSTEM FOR UNIVERSITY CURRICULA SUPPORT

Philip Grew, Ivan Longhi, and Elena Pagani
Department of Information Science and Communication
Università degli Studi di Milano, Italy
email: {grew,pagani}@dico.unimi.it; ivan.longhi@rcm.inet.it

ABSTRACT

Several tools have been proposed in the literature to use the web to support learning. Usually, these tools are designed for a very specific purpose, namely, to provide information about a course, to support student assessment or as a platform for maintaining the learning community. However, in a university curriculum all those independent functions contribute to training students and should be merged harmoniously, as well as include student-services offices.

In this paper, we describe the architecture of a distributed system that provides modules to perform these functions, in such a way that the modules are strictly integrated into the overall system and interactions among the parties involved – teachers, students, and university offices – are greatly facilitated. The architecture was designed so as to accompany and tutor students throughout their whole program, meanwhile allowing faculty and staff to monitor student performance with respect to the chosen curriculum. The architecture is almost completely deployed and is already in use at the Department of Information Science and Communication of the University of Milan, Italy.

KEY WORDS

web-based education; innovative web-based teaching and learning technologies; educational portal; distributed architecture, testing and assessment.

1 Introduction

Many tools have been proposed in the literature to provide web-based support for teaching and learning. Usually, such tools have a specific purpose – e.g., assessing students in a specific course or publishing course information or maintaining a community for teachers and students – but are not integrated with one another. Sometimes their use is not flexible with respect to the needs and skills of students and teachers. All-in-one platforms are often not designed to integrate with other systems already deployed and to remain in use.

An integrated environment providing such functionalities and oriented toward supporting interactivity among those involved has several advantages. A workgroup can be created that involves students in the course material. Exams results can be monitored and used to adapt teaching either within a specific course or across the whole curriculum. Better coordination among courses and teachers can be achieved, thus providing better quality teaching. University staff can monitor the progress of each student, easing the burdens of administration.

In this paper, we propose a distributed architecture that integrates three main functionalities: support for teachers and students in the learning process, with particular attention on promoting interactivity; facilitating and partly automating student-performance assessment; feedback from teaching and testing, so that statistics on test results can be used to improve teaching quality.

The choice of a system based on the open-source development model was a foregone conclusion, given the situation. Aside from the obvious ethical obligation of an Italian public university to implement, insofar as possible, public-domain, non-proprietary, and European-made solutions, budgetary constraints would have made any licensed-software option prohibitive. Many of the considerations that had led to the original decision to develop Just Learn It! (known as “JLI!”) still held true (see [2] p. 135). JLI! started as an open-source LMS to foster distance-learning know-how among small and medium enterprises in the Lombardy Region using an adaptation of the Adept project [6].

Other factors specific to our teaching context contributed to our choice to pursue integration through further development of this system. Previous experiences with platform adoption had made our department leery of any commitment to server-side systems that we could not modify ourselves. Conversely, the open nature of JLI! and its modular design assured that functions supplied by other platforms already adopted here could either be delegated to those systems or subsumed by JLI! on a case-by-case basis.

Many diverse systems continue to be used on campus, making proposals for yet another centralized solution untenable. Our objective was not to import contents and data but to make the diverse systems run collaboratively, without duplicating functionalities already provided by dedicated platforms but rather focusing development efforts on features strictly related to serving our course content and processing our test results.

Because the automated test-correction facilities in JLI! had been developed specifically to meet our testing needs, it offered a guarantee of suitability that no other

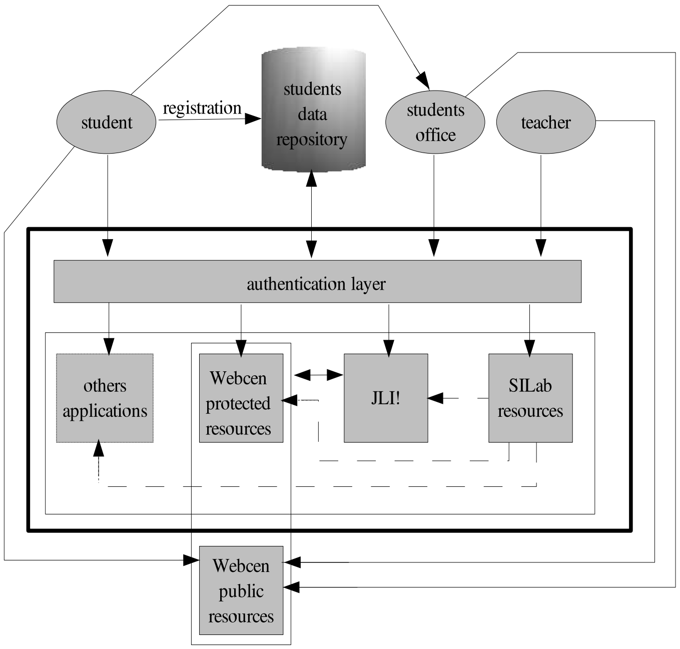

Figure 1. Overall system architecture.

platform could match. By using standard protocols to integrate this platform with the other systems we use, we were able to leave many things as they were while offering the added testing functionalities. Various courses have forums and online areas based on technology related to the subject of the course, in keeping with the principle enunciated by Linser [8] that "how you teach is what you teach." These include WikiWiki websites [9], a custom assembled quiz application called "Mirror" [10], and various forum applications. Subsequent development of Mirror, for example, had been hampered by its closed-source nature.

One of the most important forum applications with which JLI! integrates is our FirstClass [11] community server. Although this is still widely used, various attempts to integrate other services with it over the last ten years had taught us that such integration was not part of the original design plan. Naturally this affected our consideration of the other platforms grouped by Wenger [7] in the same type class as FirstClass, such as Blackboard [12], which requires an Oracle database, and all-encompassing solutions like WebCT [13].

The system is currently deployed and under further development at the Department of Information Science and Communication of the University of Milan [14].

The paper is structured as follows: Sec.2 explains the aims of the system. In Sec.3, the overall system architecture is explained, along with an example of its use, and some details of the main system components' internal structure are given. In Sec.4, the advantages offered by our system are discussed, and its current implementation status is described. Sec.5 contains the conclusions.

2 System purposes

Using the web to support didactics as is traditionally done, simply allows teachers to provide information about their courses in electronic form, substituting the paper-based noticeboard and a copy center with equivalent virtual documents. The communication model among teachers and students did not change dramatically.

At the Department of Information Science and Communication of the University of Milan, most courses have from 100 to 200 students. With such large numbers, supporting student-instructor interaction and managing exams is a hard job. The use of course forums is still sporadic,[1] mainly because its installation, management and customization are left to teaching faculty, though not all teachers have computer skills. However, the infrastructure proposed in this paper aims at alleviating these problems.

In addition to providing web-based information publishing facilities, our infrastructure aims to boost interactivity by supplying tools that create communities of people who cooperate online within the framework of a given course. All the people involved in such a community know their respective identities, whereas teachers often know only a few students, especially in large classes, the ones who talk in class. Linser [4] discusses how supporting interactivity and creating a sense of belonging to a community helps boost the exam performance of less interactive students. The architecture we outline supplies tools to prepare and mark exams and publish the results, through a learning management system (LMS) that is integrated with

[1] At least in our Dept.

other tools for community maintenance. Administrative tasks are supported by providing the student-services office with easy access to student data and course notices posted by teachers.

As a result, easier management of exams allows teachers to collect information and statistics about students' evaluation of the course and students' exam results, which can be used to adapt teaching to the real needs of the students, as they emerge from the system. This task may involve either an individual teacher or the teacher community in dynamically adapting the whole curriculum to students' needs, by better integrating and interfacing different courses.

These goals are achieved via a distributed system that "tutors" each student throughout his/her university career. The distributed system includes all the major resources available to students and teachers. It can also be easily extended by adding new components.

3 System architecture

Fig.1 shows the overall system architecture. Already implemented modules and interfaces are drawn with solid lines while dashed lines are used for components that have yet to be fully developed. We describe how the system operates through an example of its use, which is already ongoing at the Information Science and Communication and the Computer Science Departments of the University of Milan. Details on the modules' internal workings are provided below.

Before courses start, first-year students must take a preliminary English language test, whose purpose is to place them into an English course of the appropriate level [1]. Students' data is recorded in the student data repository and a login is created for each of them, which allows student authentication for access to all the protected resources in the system. It remains valid until the student graduates. The main resources students have at their disposal are the department laboratory (SILab) – used for practical activities to train them in the subjects of different courses and for accessing the internet – and various systems on which they log in to access course material.

Information about courses can be made available to students either via WebCen (the server for web-based teaching [3]) or via JLI! Both platforms have been deployed at the Computer Science and Information Science departments at the University of Milan. They differ in their primary purpose and essential features. Basic information can be provided on WebCen using pre-formatted HTML pages provided by the platform and maintained by instructors. Teachers may decide to add additional HTML pages, whose implementation is left to them. The material supplied by WebCen may be either public or protected. In the latter case, student authentication is achieved via the student login and password mentioned above.

Either through the internet or WebCen students may access the information associated with a certain course and maintained on the JLI! platform, using their student login. The data shown by JLI! and the authorization to modify it depend on user identity, according to the authentication procedure. JLI! can be used both for didactic support and for exam management. JLI! is similar to WebCen as far as its ability to provide course information and additional material. However, JLI! provides a richer set of tools: it enables access control, it logs students' participation, it lays the groundwork for a community of teachers and students, and it eases management of all the tasks related to exams, from preparing tests to marking them (which is partly automated), from publishing exam results to collecting statistics on those results. In particular, JLI! is currently used to conduct the English placement test, in addition to exams for several other courses. The value of applying placement tests in core subjects such as mathematics, physics and logic is currently under evaluation. These tests could both help last-year college students choose the appropriate university curriculum, giving them an estimate of the skills they need to be successful in a given program, and aid the university in organizing preliminary courses for students who lack the needed prerequisites. The results of the tests will be available to the students' office through direct access to JLI!

When students take the English test, as a preliminary operation they are asked to enter their data, i.e., their chosen major and a valid e-mail address, which can be useful to both the students' office and to teachers in case there is a need to contact the student. This information is maintained by the students themselves, thus preventing the university from having outdated, inaccurate or contradictory data. University faculty and staff can use the authentication system to access JLI! and WebCen in order to edit platform content or display information about students (exams results, number of passed exams, and so on).

A system to distribute certificates to students who log on to the SILab machines is currently under development, so that those certificates allow them to access either WebCen or JLI! without having to enter their password again. The resulting architecture has been designed so as to easily allow other modules whose access is controlled by the authentication layer to be added.

The internal architecture of major system components is described in greater detail below.

JLI! JLI! was initially developed as a component of a system to implement a learning environment for small businesses [2]. In particular, JLI! implements LMS functions. It was subsequently oriented mainly to providing tools for testing students. JLI! is currently used for testing and learning support in several courses offered by the University of Milan Department of Information Science and Communication.

The JLI! architecture (fig.2) includes a user-management interface that allows access to JLI! tools, subject to user authentication via login and password. The JLI! authentication system uses the account repos-

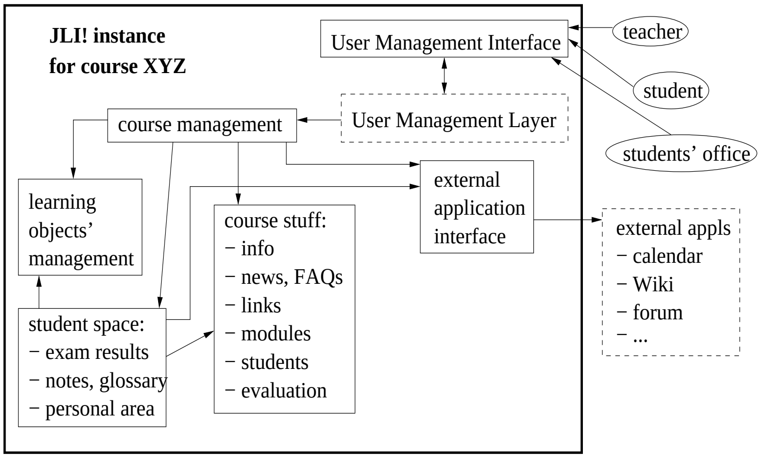

Figure 2. JLI! architecture.

itory that provides students access to SILab resources, thus greatly simplifying both student authentication and security-management tasks. Moreover, JLI! logs user access to the system. Re-use of SILab user IDs and passwords is accomplished through LDAP [5].

Only students enrolled in a certain course can access the JLI! data for that course. Students are registered for required courses they are supposed to take during the current semester – by the students' office – and may subscribe to the elective courses they choose. In the latter case, registration is accomplished the same way as exam registration is currently handled: students send an e-mail to the teacher with the relevant data (their name and student number). This information is used by the teacher to build a CSV file for upload into JLI!

Each course is composed of several `module`s – i.e. learning objects – each of a given type. Basically, three kinds of modules are available: *testing* modules, which make it possible to build different kinds of exercises, questions, and quizzes for the course's final exam, *self-evaluation* modules used by students to gauge their preparation, and *theoretical* modules, which can be used by teachers to make additional resources on course topics available. Teachers may create and edit courses and `module`s, as well as sign up new students to whom to assign courses. They may provide students with information about the course, news, a course glossary, and links to additional HTML pages, external applications or a forum. After an exam session, teachers may use JLI! to mark the test modules, if they have assigned essay questions. The system automatically corrects multiple-choice and fill-in-the-blank questions, and alike. It also tabulates exam results and gathers statistics. These statistics can also be accessed by the office of student services to monitor students' career progress.

Registered students have a personal space at disposal, which allows them to create personal notes and a glossary for the course, access the `module`s they have been assigned, monitor their exam results, and practice with the exam topics via self-evaluation `module`s. Evaluation forms are available for students to provide teachers with their comments about the courses.

For further details on the internal structure of JLI! and considerations about experience in using it, interested readers may refer to [1, 2].

WebCen. WebCen was devised as a portal to provide information about courses supplied by the Information Science and Communication and the Computer Science departments. It was designed with the main purposes of both offering users a uniform interface for different courses, and offering teachers a platform to easily publish materials related to the courses they are responsible for, regardless of their skills. Indeed, teachers of courses not related to computer science may be uncomfortable with the task of creating HTML pages themselves.

The WebCen structure (fig.3) allows access to a list of the courses available during the current academic year. For each course, information about the teacher, the schedule, and the course program is available, along with notices and forms for students' evaluation of the course. A teacher may decide to add additional materials, such as his/her own slides or notes about the course topics, links to related sites, and the course bibliography. Related links may involve a forum, independently managed by each teacher and perhaps relying of different technology. All the resources provided are public, i.e. no access control and logging is performed by WebCen. Teachers may limit access by implementing filters themselves on the linked material. WebCen pages for a given course are edited through forms accessible to the instructor in charge. These enable instructors to:

- fill in pre-established fields (e.g. office hours);
- upload files (e.g. slides);

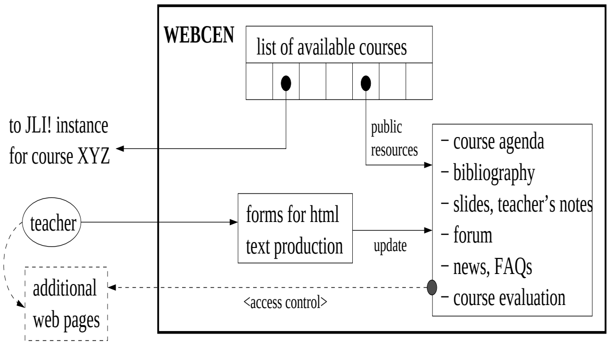

Figure 3. WEBCEN architecture.

- add links (to forum or – possibly protected – web pages).

Until recently, no further tools were available via WebCen. WebCen was used as a traditional web site, where students could find information about their courses, published by the teachers. No interaction among teachers and students was supported, unless explicitly provided by teachers as an add-on *external* to WebCen.

The interfacing between WebCen and JLI! is nearly complete. In the future, we plan to adopt WebCen as the unique portal for course information, providing general data through unprotected web pages. The interaction among students taking a course and between the teacher and the students will be supported through the tools provided by JLI!, which make it possible to build the virtual community that is proving to greatly enhance both teachers' and students' work. These topics will be better described in Sec. 4.

SILab. The SILab (*Scienze dell'Informazione* Laboratory) supplies students with resources and tools to practice the topics taught in our courses (fig.4). Each student owns a home directory, which (s)he can use for his/her data, programs, and exercises, and can freely navigate in Internet. Among the available resources students have at their disposal, are compilers for the various programming languages taught in the courses, packages for computer graphics, network simulation, and so on.

Access is controlled via the login/password pair assigned to a student when (s)he enrolls. This remains valid until (s)he graduates.

4 Discussion

Several advantages have been reported by both teachers and students who have used infrastructure described.

First of all, duplication of information has been eliminated by maintaining student data in a unique repository, thus avoiding data inconsistencies. Giving responsibility for the maintenance of information to the people it regards guarantees that the data is correct and up to date, thus decreasing the workload previously required of university staff to maintain information beyond their control.

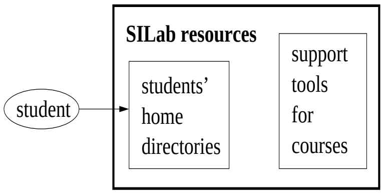

Figure 4. SILab structure.

Teachers can find out which students are registered for a course beforehand, including their email addresses. In addition, the JLI! student-authentication system may be used to limit access to course contents to registered students only. Moreover, by knowing the number of students registered for a class, a teacher can determine at any time how many students still have to take the exam. JLI! allows email broadcasting to all registered students, for example to send out urgent notices about the course that would otherwise be disregarded if published on a traditional noticeboard or webpage.

The use of JLI! for exams allows teachers to monitor students' results so as to collaborate in harmonizing course syllabi. For instance, collecting statistics about results achieved in Computer Networking exams may reveal that students need greater insight into interrupt-treatment mechanisms, which are taught within the framework of the Operating Systems course. The visual interface and the statistics tools provided by JLI! help teachers detect such needs, thus enabling course plans to be adapted to student behavior. Potentially, workloads can be re-balanced among courses if JLI! shows that some courses are too burdensome and students achieve poor grades in them. This task is supported by facilities of JLI! that are already deployed, which allow registered students to submit their own course evaluation .

Finally, the database produced by the use of JLI! will allow the department to easily derive information about the starting point of the students (thanks to entry tests), the time they spend before graduating, the relationships between their initial skill, the grades they achieve in different exams, and their final graduation score, and so on. Some of this information is currently collected manually through the hard work of university staff, given that it is fundamental in evaluating the quality of the education provided.

5 Conclusions

This paper describes an integrated infrastructure that supports students, teachers, and student-services offices throughout the whole course of study. The primary infrastructure components are already implemented and are currently in use by both teachers and students. The process of integrating them into additional existing infrastructure is nearly complete.

The use of an open architecture means that system integration can be achieved without forcing changes in tools and methods on teaching faculty who now use a variety of systems for managing their specific course content. Thus the course-specific WikiWiki servers and forum systems already in use for certain classes can be preserved and integrated into the SILab system used for WebCen/JLI!

Further development has also been planned. A big problem still to be coped with is maintaining information consistency among multiple sites. Currently, information concerning both teachers, such as their office hours, telephone number, and room, and courses, such as lesson times, classrooms, and schedule, are maintained not only on WebCen but also on the official departmental website and potentially in JLI! The management of these sites may be left to different people. When an update is performed, all those concerned must be informed so as to maintain consistency among information copies. Inconsistencies create student confusion and result in inquiries to both teachers and student-services offices requesting clarification.

The design of a comprehensive portal has been undertaken for the two departments that supply curricula related to computer science. This portal should integrate both WebCen and JLI!, with all modules referring to a unique repository for basic information available in several places, so that consistency will be automatically provided by dynamically building webpages that use repository data. In particular, WebCen, linked with JLI!, will be the sole site as far as teaching materials are concerned. By contrast, the departmental site will maintain information only about departmental research activity and about administration. As a subsequent step, the portal will be interfaced with the university portal.

Acknowledgments

We wish to acknowledge Dr.Raffaella Folgieri and Dr.Marco Reggio for their help with the WebCen structure.

This work has been partially supported by the Italian Ministry of Education, University and Research in the framework of the FIRB “Web-Minds” project.

References

[1] 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)*, Feb. 2004.

[2] 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)*, pp. 133-140, Nov. 20-21 2003.

[3] G. Valle, “Continuous Learning Relaxing Time and Location Constraints in a Regular Curricula Degree Offered by University of Milan”, *Proc. Intl. Conf. “Challenges and Opportunities of Vocational Training and E-learning in EU Accession Countries”*, Nov. 12-13 2003.

[4] A. Ip and R. Linser, “Simulated Worlds: Rapid Generation of Web-Based Role-Play”, *Proc. Seventh Australian World Wide Web Conference (AusWeb’01)*, Apr. 21-25 2001.

[5] J. Hodges and R. Morgan, “Lightweight Directory Access Protocol (v3): Technical Specification”, *RFC 3377*. Work in Progress. Sep.2002.

[6] Adept Project `http://adept.sourceforge.net`

[7] E. Wenger, “Supporting Communities of Practice: a Survey of Community-Oriented Technologies”, *Report to the Council of CIOs of the US Federal Government*, `http://www.ewenger.com/pub/pubpapers.htm`, Mar. 2001

[8] R. Linser, “Role-play Simulation in teaching and learning”, *Seminari at the Department of Information Science and Communication of the University of Milan*, Jun. 21st 2004. `http://www.fablusi.com/`

[9] W. Cunningham and B. Leuf, “The Wiki Way. Quick Collaboration on the Web”, Addison-Wesley, 2001.

[10] Centro di servizio per le tecnologie e la didattica universitaria multimediale e a distanza (CTU), “Mirror Project”, `http://www.ctu.unimi.it/mirror.asp`

[11] Open Text Corporation `http://www.firstclass.com/`

[12] Blackboard Inc. `http://www.blackboard.com/`

[13] WebCT `http://www.webct.com/`

[14] Laboratorio di Informatica Civica `http://jli.dico.unimi.it/`