Web-Based Course Management and Web Services

Chittaranjan Mandal and Vijay Luxmi Sinha
Indian Institute of Technology Kharagpur, India
Chittaranjan.Mandal@cse.iitkgp.edu.eu.org
vluxmi@mla.iitkgp.ernet.in

Christopher M P Reade
Kingston University, UK
Chris.Reade@kingston.ac.uk

Abstract: The architecture of a web-based course management tool that has been developed at IIT, Kharagpur and which manages the submission of assignments is discussed. Both the distributed architecture used for data storage and the client-server architecture supporting the web interface are described. Further developments of the tool making use of web-services are also described along with a discussion of the relevance of this for recent open standards and future learning management systems.

Keywords: course management, distance education, web-services

1. Introduction

Managing online assignment submissions made via email becomes a real challenge (especially when class sizes are large) because it requires huge amounts of storage space and file management skills to process submissions efficiently. The problem is aggravated when more assignments are given as the course progresses. Distributing the responsibility of evaluation amongst staff may alleviate the problem of submission management to some extent, but this requires managing the collation of assignment marks. Centralized management of marks is also desirable to assess student progress in a course.

Our Web-based course management (WBCM) system provides easily navigable structure to all online submissions and a centralized web-based interface for submission evaluation. A customized online submission interface is generated in accordance with requirements for each assignment as specified by the staff concerned. Student progress tracking, group and individual assignment organization, assignment evaluation and marking, grade maintenance and distribution, online submission, online attendance are the important features of WBCM. In essence WBCM automates and integrates several diverse aspects of course management.

After discussing the motivation behind WBCM, subsequent sections describe the system architecture and important technical details and features of the implemented WBCM. The original intention of the developers was to evolve the tool into a complete learning management system for release as open-source. However, with the recent trend towards open standards for learning management systems, a path has opened up for developing this tool into a standards-compliant component for use alongside other LMS components.

The paper concludes with a discussion of planned further development of the tool using a web-services model and notes the relevance of recent open standards for learning management systems.

2. Motivation for a Web-based system

There were several motivations behind developing this system. The work started as a system to ease the handling of laboratory based courses. The requirements, however, turned out to be fairly general and minor additions were required to make the system handle almost any kind of course. We also had the need to handle courses running at a distance at the various extension centres. After developing two versions of this system we decided that another version with comprehensive features was required. These are described in more detail below.

2.1 Laboratory courses

Although not unique to laboratory-based courses, several factors that require significant administration are often concentrated in such courses (especially when there are multiple assignments given throughout the course, each having its own deadline and guidelines for completion). Completed assignments need to be collected, evaluated and marked for assessing students' progress. Assignments may be given to individuals or groups. To further assess students' progress examinations and online tests may be conducted. Attendance records may need to be kept and some marks may be awarded for attendance. The marks for each student need to be tabulated and at the end of the course grades need to be distributed. Physical dissemination of material is difficult and involves generating and handling a lot of paper. Keeping a record of the attendance is costly in terms of time. It is also extremely difficult to keep track of individual progress manually. Such a situation is a clear case for the need of automated support through an electronic course management system.

2.2 Distance education

There are additional benefits that may be had from managing a course over the internet using a web-based course management (WBCM) system. Such a system can be highly effective in bridging geographical distance, which is an important concern in India.

In the Indian Institute of Technology, Kharagpur University we run a distance education programme. We have several centres where this programme runs and it has been extremely difficult to manage the programme running over these centres and monitor the progress of students at there centres centrally from our main campus.

Our web-based course management system has provided an excellent solution to this problem. Teachers, too, tend to get geographically separated from the students for other reasons, such as having to travel to conferences. As a result, they either have to curtail their travel or make complex arrangements to deal with the absence. Here again a web based course management scheme has provided useful advantages.

2.3 Development of WBCM

We have already implemented and used three versions of WBCM. Two earlier versions were a success with staff and students, further motivating us to add more features. The packages were used for large classes. Courses run with these packages helped in reducing logistic problems for assignment from near impossible to trivial. Logistics for evaluation were simplified as submissions were available online all the time and transparent evaluation meant more student satisfaction. Dependence on printers was almost completely removed.

The third version, in particular, supports sections within a large class and distributed storage repositories. The earlier versions relied on all the submissions and databases being located centrally. This scheme has its advantages and disadvantages. Centralized data storage takes off all responsibility of storage management from the end users (in the capacity of instructors). However, it does increase the burden of storage on the central server. Also, end users sometimes feel out of control of the submissions of their courses.

Support for sections in the third version enabled common or distinct assignments to be given for separate sections. Also, students may work individually or in groups. This affects the assessment mechanism, but that is supported. Support is also available for moving students between sections.

3. Architecture of WBCM and technical details

The distributed databases (Postgres) and file (Linux) repositories, the web server (Apache) and the corresponding cgi-scripts and the http clients (tested with Netscape, Konqueror, IE) are the main components of WBCM architecture. To describe the WBCM architecture, shown in Figure 1, we divide it into two parts: Distributed architecture for data storage, and Client-Server architecture for providing the human interface (the web based aspect of the utility). We also give details of the important components: the database design, the cgi-scripts (which form the backbone of the system), the authentication mechanism, the repository directory structure and the distributed storage mechanism.

3.1 Distributed architecture for data storage

WBCM manages course data for many courses and multiple runs of each course.

A schema and directory structure (for storing scripts and assignments) is created for a course run and replicated to manage multiple course runs.

In addition to having an independent database for each course run, a database which we call the "course database" is used for keeping data about all the course runs (list of courses, list of course runs).

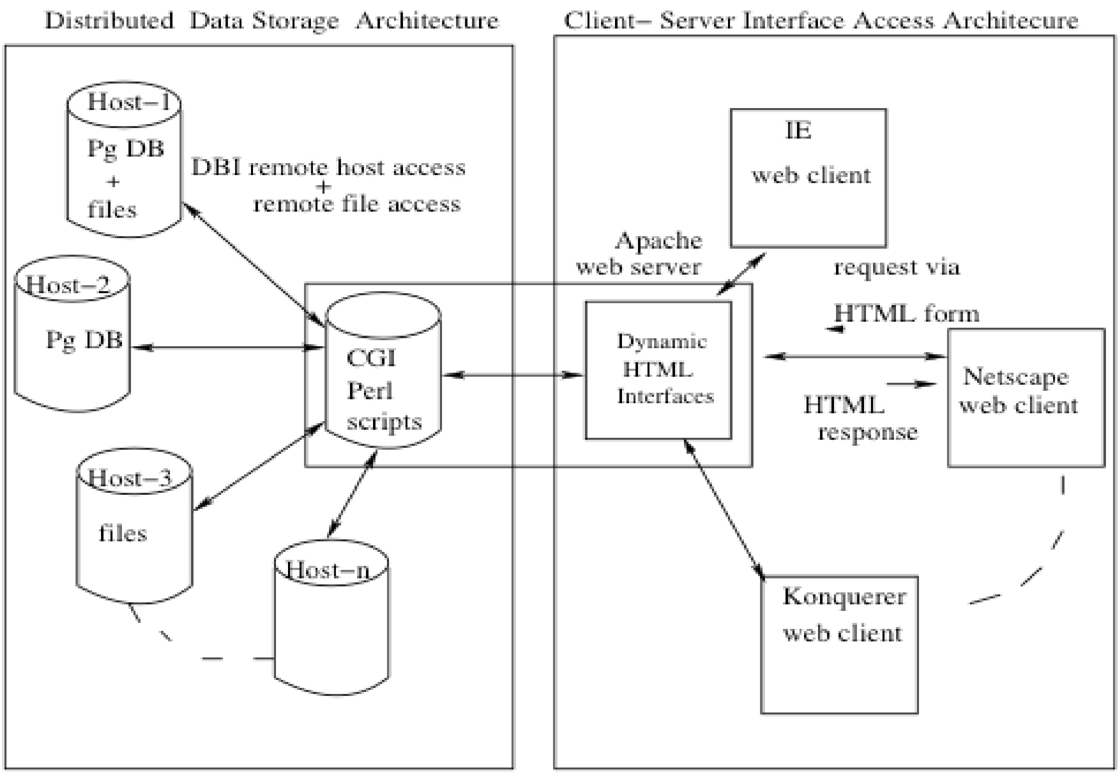


Figure 1: Architecture of WBCM package

Thus we have a cluster of database, instead of single database which allows

- the databases to reside on different hosts;
- the use of simple common schema for each course except for the "course database".

For each course run, assignments and the submissions are stored directly as f les in the f le system, under suitable directory structure, which allows reconstruction of a path to them independent of their content. Other information, relating to courses, students, staff, etc. is also stored in databases. This may be considered as "control" information. Whenever there is a choice between storing information in a f le or a database, the later is preferred, as operations on a database are performed at a high level.

Our experience with previous versions of WBCM, which had a centralized architecture for data storage has guided as to adopt a distributed architecture for the same. The following observations were of critical importance during the run of previous versions of WBCM, convincing us of the need to change to a distributed data storage scheme:

- The disk and database storage requirements grows considerably with increase in the number of courses managed by WBCM.
- Classes may have very large numbers of students -e.g. for interdisciplinary courses, translating into large storage requirements.

A new distributed architecture was developed to alleviate load on a single host in the third version. Thus avoiding the impact of an increase of courses managed by WBCM creating a bottleneck. WBCM data are stored in multiple hosts as shown in Figure 1. The host used for storing either a database, or a f le repository, is decided by the administrator. A database and a f le repository may reside on the same host, as shown in Figure 1 where Host-1 has both a Postgres database and a f le repository, Host-2 has only a f le repository, Host-3 has only a Postgres database. A host may be used for storing data about more than one course run.

3.2 Client-Server architecture for interface access

There are two kinds of CGI resources for WBCM, namely authenticated and unauthenticated. We use an authentication mechanism provided by the Apache web server.

As shown in Figure 1 a web client makes requests via an HTML interface to WBCM generated by Perl scripts on the web server. All the HTML pages of the system are dynamically generated (i.e. when they are requested, the corresponding CGI script is run on the server and result is displayed). This is desirable as up-to-date information must be made available. For authenticated resources, when the web client makes a request, the server sends a response requesting authentication information. The user is prompted to enter their username and password and this information is submitted to the server for verification. If the information is correct, the server sends a response to the original request made by the web client.

To obtain the response for the web client request, the web server executes an appropriate WBCM script, which in turn queries the database and accesses the f le repository to generate a dynamic HTML page.

Dynamically generated pages require server resources. The scripts can be executed from different web servers, distributing the server burden. Automatic distribution of the server load is an interesting topic for improving performance of the system. Currently, we are expecting users to move to a different server whenever performance degrades considerably or they may be instructed to use different servers when all (or most of) the possible users are at one place (e.g. during a class test)

3.3 Database design

WBCM manages a huge amount of information. The following entity-relationship diagrams and description of the tables gives an overall picture of the database design for WBCM. The ER diagram in Figure 2 shows relationships that affect online assignment submission. Each assignment comes under an assignment category, which may be an online test, an exam, a lab assignment, etc. Each section should have assignment categories associated with it. Student groups can be made for assignment submission that are under the same assignment category. Student cohorts may be divided into sections and an assignment may require individual and/or group submission.

All the information about each course is distributed between two databases.

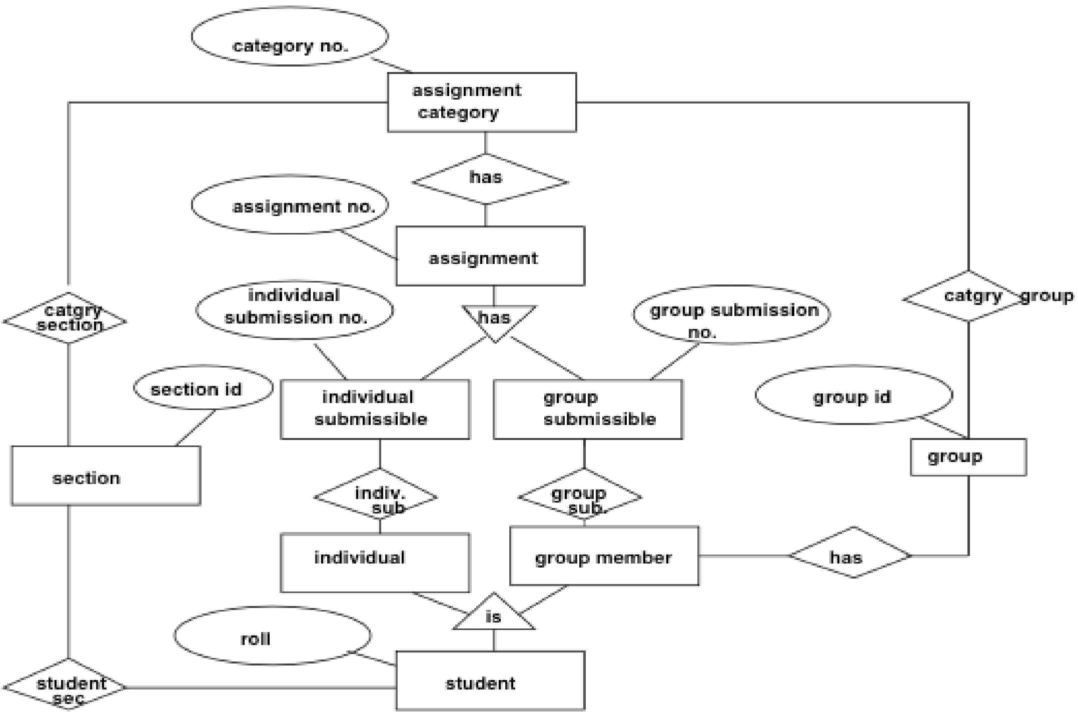


Figure 2: Online submission relation.

3.4 CGI scripts

CGI resources are distributed among directories to facilitate www authentication. Directories are for admin, faculty, faculty and supporting staff, student and public access. For each course run, a soft link is created to each of above-mentioned directories except the public access directory. The course run directory also stores all the necessary f les for www authentication.

The web pages are generated dynamically by querying the database for up-to-date information. A Perl DBI module is used to make connections (remote/local) to Postgres databases. Perl eval statements along with Postgres support for commit and rollback are used for transaction control

3.5 Authentication

User authentication is an integral aspect of such a system. We extensively use the authentication mechanism provided by Apache (based on .htaccess). We do not describe the mechanism used by Apache here. Instead the interested reader is referred to the Apache documentation available at http://httpd.apache.org/docs/.

Use of http authentication requires some pre-planning by the programmer. Planning starts with the identif cation of different groups of expected user who should authenticate themselves. For WBCM these different groups are “administrator”, “faculty”, “evaluator” and “student”. All the cgi resources used by these groups are kept in their respective directory (i.e. each group has a different directory).

Each course has its own group of “faculty”, “evaluator” and “student”. When a course is started by an administrator, a local course directory is created under the above-mentioned directory as a link. Under these directories a course-specif c password f le is maintained which contains a user name and encrypted password for each of the group members. In our scheme the password f le is maintained by a script, as group members may change. We make sure that for students who already have an account on the system hosting WBCM, the username and password for WBCM authentication is the same. Otherwise the students are given a roll/user name and some initial password automatically. For the current scheme staff should have an account with the system hosting WBCM.

3.6 File repository directory structure

The submission directory structure is shown in Figure 3. In addition to this we have a similar assignment description directory structure. As shown in Figure 3, the submissible directory structure may keep submissions for different runs of courses. That is why the immediate sub directory is named ”course run (1..n)”. This should be the identif er of the course run - in our case, a combination of the course code, year and semester. Classes are divided into sections, requiring a sub directory for each section (sec-1, sec-1..sec-m). For each section we have an assignment category having assignments, giving: category-1..category-o followed by asgn-1..asgn-2 levels of sub directory. Assignments are organized individually or in groups, so submissions are kept under roll and group subdirectories for individual and group submissions respectively. It is possible to distribute the repository over several machines.

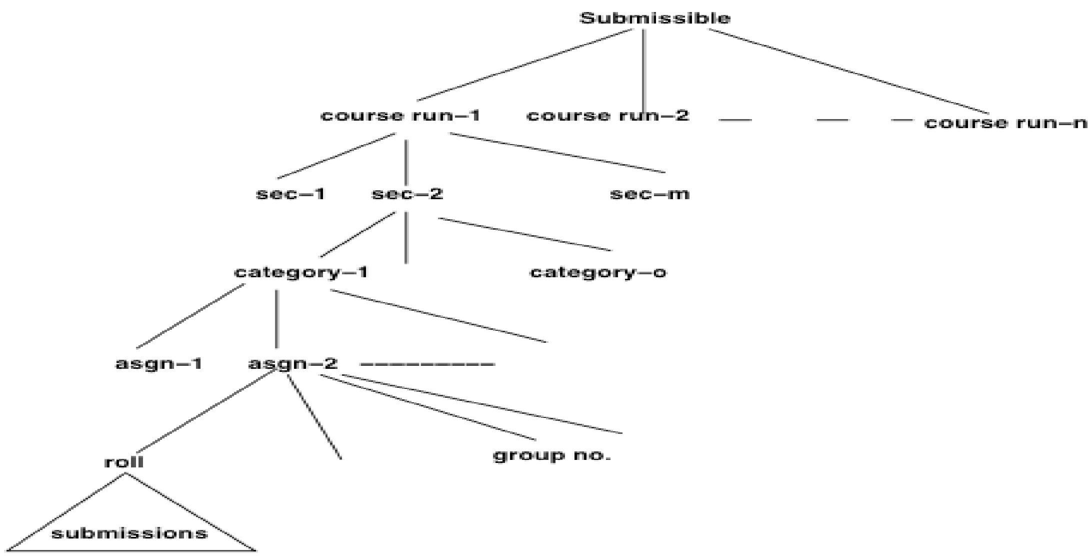


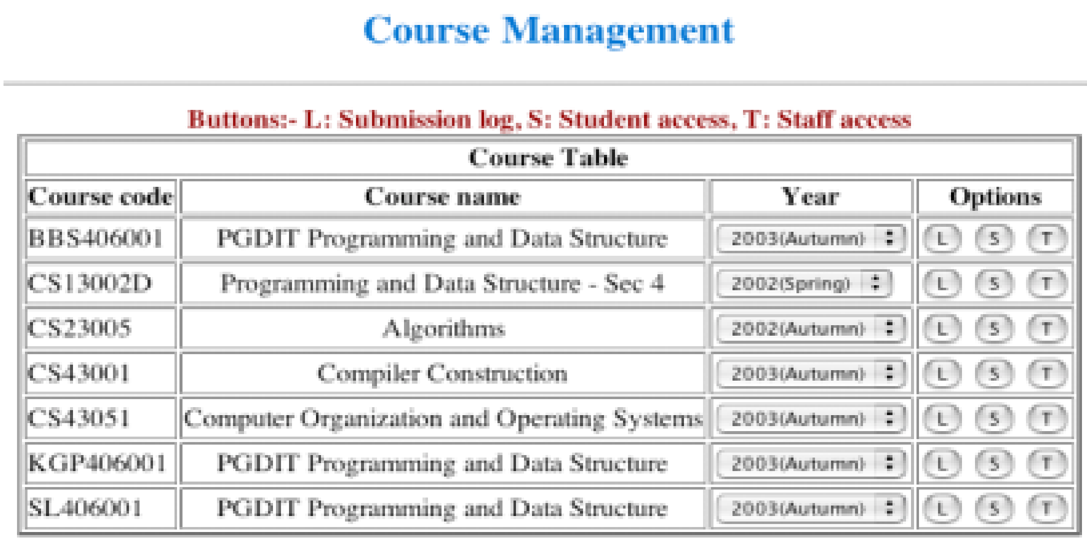
Figure 3: Repository directory structure.

4. Key features of WBCM

The purpose of our web-based utility is to provide a simple yet powerful interface for managing courses, along with the f exibility of online submission for the students. The top-level course management access page is shown in Figure 4. The interfaces are either public or protected and they are linked as shown in Figure 5. Note that the users of

WBCM are: course administrator, staff (teaching and/or supporting) and students. WBCM has few publicly viewable pages, which are mainly logs of important information pertaining to a course.

The current WBCM has the following features: admin, course table, assignment management, student/staff management, assignment/submission evaluation and submission log. The admin feature allows courses to be added or deleted and initial staff to be assigned to a newly added course. The course table lists all the courses currently availing of the course management facility and also has navigation buttons for staff, students or the submission log of each course. Assignment management allows for the addition, modification and deletion of assignments. Necessary consistency guards are enforced. Student/staff management allows for the addition or removal of staff/student from a course. Students can submit assignments via a submission link in an assignment table which lists all the assignments currently in the course. Assignment evaluation allows staff to evaluate assignments and provide comments or justifications for the evaluation. Submissions may be re-evaluated, keeping a record of older evaluations. A particularly useful feature is the submission log which shows the status of all submissions and evaluations at a glance.


Course Management

Buttons:- L: Submission log, S: Student access, T: Staff access

Course Table			
Course code	Course name	Year	Options
BBS406001	PGDIT Programming and Data Structure	2003(Autumn)	L S T
CS13002D	Programming and Data Structure - Sec 4	2002(Spring)	L S T
CS23005	Algorithms	2002(Autumn)	L S T
CS43001	Compiler Construction	2003(Autumn)	L S T
CS43051	Computer Organization and Operating Systems	2003(Autumn)	L S T
KGP406001	PGDIT Programming and Data Structure	2003(Autumn)	L S T
SL406001	PGDIT Programming and Data Structure	2003(Autumn)	L S T

Figure 4: Course Management Web page

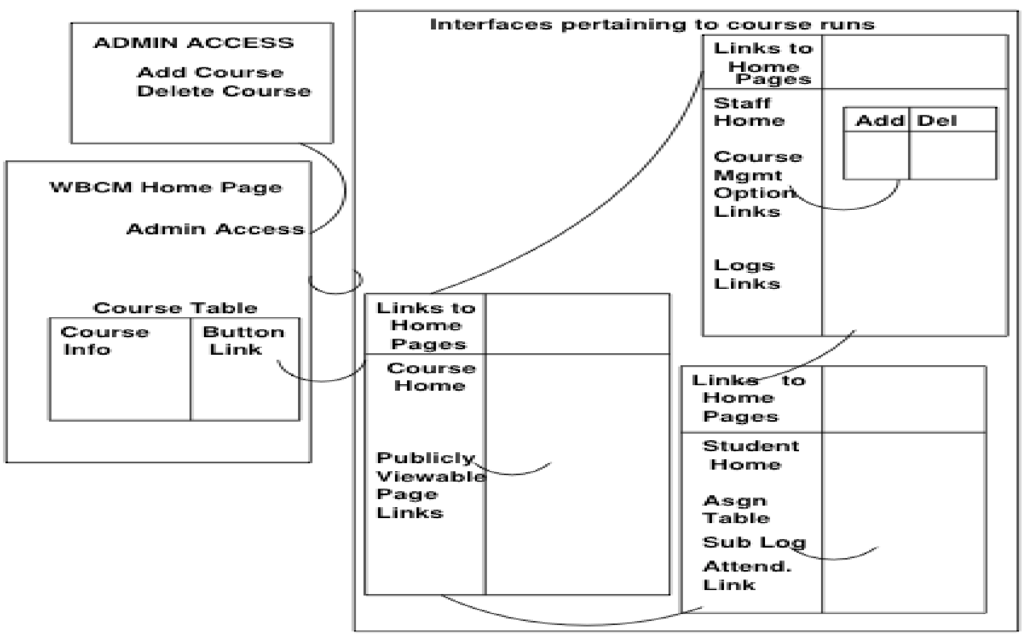


Figure 5: Interface navigation diagram.

5. Learning management systems and Web Services

In this section we discuss recent standards work for learning management systems, the relevance of web services, and the current development of WBCM as a collection of web services.

5.1 Standards for interoperability and LMS components

There is now significant attention being paid to the development of open standards for learning management systems. See, for example, the centre for educational technology interoperability standards (CETIS http://www.cetis.ac.uk/) and also Diffuse (2002) for details. Many of the current standards developments are being directed at the management of learning content and interoperability issues. This is motivated by the need to de-couple content from proprietary packages and to ensure that content is not locked-in to specific platforms. For example SCORM (ADL 2003) is a standard reference model for shareable content and the IEEE LOM is a standard being developed for learning object metadata. The potential use of metadata and related standards has also opened up many new possibilities for a more federated approach to constructing cooperating learning management systems and components. Stephen Downes (2002) discusses these points in describing designs for a distributed learning object repository network (DLORN).

However, there is also a drive for learning management systems to inter-operate with other systems effectively. Other open standards are needed to support this as well as to encourage a more flexible, component-based view of learning management systems. The IMS Global Learning consortium developed standards in 1999 and 2002 for inter-operability of Enterprise systems (IMS 2002). These address issues of transferring data between Learning Management systems and Enterprise systems such as student record systems.

In the UK, the more recent JISC e-Learning technical framework [http://www.jisc.ac.uk/index.cfm?name=elearning_framework] is very relevant to our work. In the architecture described by Scott Wilson (2003), a service-oriented view of components in learning management systems is discussed with an application layer and common service layer as well as a user agents layer. WBCM provides several application layer services (course management, group management, assessment, grading) making use of common services (e.g. for authentication, authorisation, statusInfo).

5.2 The potential of Web-services

The benefits of a service-oriented architecture for distributed systems are mainly to do with loose coupling of components so that systems become more flexible and components are easier to add and change. The recent advent of web-services is likely to have a big impact on the future design of learning management systems. John L. Hall points out that: "From an operational point of view, the LMS and its key components—content management, user administration and system administration—should be 100 percent Web-deployable, requiring no additional client applications." (Hall 2003). Web-services (see, for example Cerami 2002) are based on recent standards maintained by the W3C (http://www.w3.org/). Their purpose is to enable components of distributed applications and other services to be provided over the web with a language-neutral, platform-neutral, and vendor-neutral interface. There is significant interest from the IT industries (including the involvement of all the large IT companies) in driving this new approach to distributed systems forward. The key factor is the development of global standards to enable this. The standards are based on the use of XML for describing data to be transferred and SOAP http://www.w3.org/TR/soap12-part1/ to wrap messages for delivery via HTTP. There are also standards for a language for describing services in a machine-processable form (Web-Service Description Language -WSDL http://www.w3.org/TR/wsdl) and for directories for the discovery of services (Univeral Description, Discovery and Integration – UDDI http://www.uddi.org/). Web-services are designed to be available over the web both for human-readable access (via a browser), and also for access by other software (and other services). The fact that web-services can even be discovered, and bound to, dynamically at run time is an important new feature for distributed computing and will eventually enable complex services to be created by dynamically combining simpler services.

For learning management systems, web-services will support the service-oriented architecture proposed by JISC (discussed earlier), and also ensure simpler integration of services through the web-services standards, giving end-user/developers much more control over the design of an LMS.

Some proprietary LMS products are starting to advertise web-services features. A related academic project is Ternier et al (2003) which describes possible use of web services with the ARIADNE learning object repository system (Duval et al 2001).

5.3 WBCM and Web-services

The WBCM system is now being developed as a collection of web-services so that it can provide highly reusable components which could be made to fit in with other LMS components and work alongside them. The fact that it is web-based already, makes this transition relatively easy. The web-services version of WBCM involves the development of appropriate (implementation neutral) XML schemas for the various forms of information required and supplied by WBCM at its interfaces. Some open standards such as those proposed by the IMS Enterprise standard are already available for this.

One of the aims of the web-services version of WBCM is to isolate more of the implementation and design decisions from its main functionality and interface. For example, hiding details of the distributed nature of the repositories, and abstracting from some of the current specific data details. With the development of standards for parts of the XML schemas used in interfaces, the potential for easy integration and inter-operability with other course management or learning object management components will be greatly enhanced.

As an example, consider a service which reports assessment marks or grades for a student on a course. In earlier versions this would be a facility in a web page which would (via http and CGI) retrieve information from a database and present the information in a web page. In the web-services version this is decomposed into a back-end web-service which, when invoked, would deliver the information in XML according to a documented schema. The front-end presentation as a web page from the XML is simple to achieve with XML technologies. However, it is also simple for other services (such as a student record service) to directly process the XML and retrieve marks from the web-service. Clearly, such a direct link is always possible with appropriate programming. What is new is the ease with which the inter-operability link can be made through the generation of an XML schema for the data and standard technologies to (i) generate a service description and (ii) to process such a description and automatically use the service.

There are also evolving standards for security aspects of web-services (including authentication, message integrity, non-repudiation, etc.) so that more generalised solutions to security can be explored.

Our new design is based around the following application level service provisions

5.3.1 Mark reporting service:

This retrieves and reports on marks for individual students or groups of students for an assignment, returning data as XML (as described previously).

5.3.2 Local marking services:

The current WBCM interface provided a web-based marking facility. However, in some cases the necessity of a web access could be a disadvantage. For example, if an instructor wishes to mark assignments while on a train, web access as a necessity is problematic. A better scheme might be for him to have a copy all assignment submissions available to him locally. This would require the instructor to first download all (or some of) the submissions to his local m/c. Then he would have to mark them and record his comments and finally upload the marks. Two web services are clearly involved, one to download the submissions of a set of students and another to upload the marks with comments. A local application would be desirable to properly organize the marks and the comments. The local application could very well be third party, though it would be desirable to have simple prototype applications available as part of the tool. This would make the tool more usable.

5.3.3 Course administration service:

This will authenticate users and allow updates to course information, including generation of a new run (with duplication of information where appropriate). The service allows students to be added or removed from a course. Similarly instructors may be added or removed from a course. Differentiation of capabilities of instructors is possible. Only primary instructors may possibly perform a proxy submission for a student or allow resubmission of a student assignment after marking is over. Also, only primary instructors may add/modify/delete assignments. The administration service could assign or retract such capabilities from course instructors.

5.3.4 Assignment creation service:

This will authenticate users and allow creation based on a marking scheme (criterea, weightings etc.) to be generated by interaction between the user (designer) and the service. The service would return a URL for identification of the assignment. The URL might also be a link to an XML document for the assignment description and marking scheme.

Services are also required by the above services for: accessing information on courses, instructors, time slots, enrolled students, etc.

The functional definition of "Learning Objects" as given in (Downes 2002) is broad enough to cover WBCM components. It would be desirable to develop components that may be pulled and incorporated in any LMS (for example). These components provide service, which in turn can be tailored to users' needs. So we have a "sharable service provider learning object" inherently different in nature from "sharable content object" [SCORM – see, for example http://www.oasis-open.org/cover/scorm.html] except in the sense that both of them are intended to be shared in a learning economy (Downes 2002).

The key services as described earlier may evolve to become sharable objects. Each service provider component has its user interface and script for completion of the service.

Furthermore, a WBCM component may be developed as a sharable component to enhance the functionality of an existing system (Open or Proprietary Service Layer Component) (Downes 2002). The components access interface should provide options for customization of the services.

6. Conclusions

We have described the architecture of a web-based course management tool (WBCM) that has been developed and is in current use in India. The main benefits of using the tool have been: its support for the co-ordination of many courses, even with very large class sizes (100-300); its simple, descriptive and consistent layout of interfaces, making it quite easy to learn and use; the extensibility of the architecture deriving from the modularity of WBCM.

The current WBCM could easily be extended to add features for complete courseware management (assignment tutorials, references, articles, internet resources, question paper etc.), although it currently only keeps information about online tests, assignments and course notices.

We have discussed the current evolution of this tool into components in a web-services architecture and how this fits in with recent views of LMS designs based on loosely coupled components as proposed in the JISC architectural framework (Wilson 2003). We believe that web-services provide for both a service-oriented architecture and full deployment over the web.

The importance of open standards for such developments has been discussed and, in particular, the potential impact of web-services for learning management systems. The new development of WBCM is aimed at improving the tool's potential for integration with other course management tools and LMS components. It will also open up some of the modularity for other developers who might want to use some services but not all of them.

Although the current system has security aspects implemented using features of the Apache web-server, it is clear that further developments of proposals for XML-based security standards for web-services are likely to have an impact on the design of WBCM in the next two years. For example the Security Assertion Markup Language v2.0 is under development (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security) (SAML 1.1 was ratified by OASIS in September 2003), and OASIS work on Web Services Security is ongoing (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss).

References

ADL. (2003). "Sharable Content Object Reference Model Version 1.2" [online] accessible from http://www.adlnet.org/

Cerami, E. (2002) "Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL" O'Reilly, Feb 2002, ISBN: 0-596-00224-6

Diffuse (2002) "Electronic Learning Standards", [online] Diffuse Consortium Report, Information Society Technologies Web. http://www.diffuse.org/eLearning.html

Downes, S. (2002) “Design and Reusability of Learning Objects in an Academic Context: A New Economy of Education?” Submitted to eLearning: una sfida per l’universita, Milan, November 12, 2002 [online] http://www.usdla.org/html/journal/JAN03_Issue/article01.html

Duval, E., Forte, E. et al. (2001) “The ARIADNE Knowledge Pool System”. C.ACM, May 2001, 44(5), pp73-78.

Hall, J.L. (2003) “Assessing Learning Management Systems”, [online], Feature Article for Chief Learning Officer. January http://www.clomedia.com/content/templates/clo_feature.asp?articleid=91

IMS (2002) “IMS Enterprise Specification, v1.1” [online] http://www.imsglobal.org/enterprise/entv1p1/imsent_infov1p1.html

Ternier S., Neven F., Duval E., Leuven K.U., Macowicz M., Ebel N. (2003) “Web services for Learning Object Repositories: a Case Study - the ARIADNE Knowledge Pool System” Poster at The Twelfth International World Wide Web Conference, May 2003 , Budapest, Hungary [online] http://www2003.org/cdrom/papers/poster/p203/WWW2003 Poster.htm

Wilson, S. (2003), “The case for a technical framework to support MLEs and e-Learning” Oct 03, CETIS report [online] http://www.jisc.ac.uk/uploaded_documents/Case-for-ELF.doc (See JISC E-Learning Frameworks and Tools Programme http://www.jisc.ac.uk/index.cfm?name=elearning_framework)

