A Web-Based Course Management Tool

Chittaranjan Mandal  
Chittaranjan.Mandal@cse.iitkgp.edu.eu.org  
School of IT, IIT Kharagpur, India

Vijay Luxmi Sinha  
vluxmi@rediffmail.com  
School of IT, IIT Kharagpur, India

Christopher M P Reade  
Chris.Reade@kingston.ac.uk  
Kingston Business School, Kingston University, 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. Plans for developing this tool using a web-services component model are also described along with the relevance of this for recent open standards for 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
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 Previous Versions of WBCM

We have already implemented and used two versions of WBCM. These two 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 were also almost removed.

The latest version, in particular, supports sections within a large class and distributed storage repositories. The earlier versions relied on all the submissions and databases being

![Figure 1: Architecture of WBCM package.](image_url)
located in a central location. 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 latest version enables 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
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
• Client-Server architecture for providing the human interface (the web based aspect of the utility)

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). 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 files in the file 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 file 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 us 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.

The new distributed architecture was developed to alleviate load on a single host. 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 file repository is decided by the administrator. A database and a file repository may reside on same host-as shown in Figure 1, where Host-1 has both a Postgres database and a file repository, Host-2 has only a file 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 file 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).
4. Technical details

The important components here are 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. These are described below.

4.1 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 in two databases. The database which is common to all the courses (course database) has two tables. The first table “coursepool”

![ER Diagram]

Figure 2: Online submission relation.
has the following attributes: course code (varchar) and course name (varchar) having course code as primary key. The second table “current courses” has: course code(varchar), year(integer), semester(char), database host ip, repository host ip, repository host user, repository host directory root as attributes and (course code, year, semester) as primary key.

The tables for WBCM are divided into two categories. One category of tables keeps only the attributes necessary to identify an entity. The other category of tables keeps information that ensures non-repudiation. For example “staff” and “student” belong to the first category of tables (see figure 3). Other tables falling under the first category keep information about assignment categories, assignments, sections, groups, schedules, etc. The “submission log” and “assignment marks” belong to the second category of tables. As the name suggests, the table “submission log” keeps the information about the online submissions made. It has attributes: assignment category(integer), assignment number (integer), roll/group(varchar/integer), submission date(date), submission time(time), host, proxy submission(boolean) as attributes, where assignment category(integer), assignment number(integer), roll/group(varchar/integer) is the primary key. The value of each attribute in this table are automatically generated and extracted by the script without student intervention, at the time of submission.

```
+==================================================+
<p>| Table &quot;staff&quot;                                    |</p>
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>staffed</td>
<td>character varying(20)</td>
</tr>
<tr>
<td>staffteacher</td>
<td>character varying(1)</td>
</tr>
<tr>
<td>password</td>
<td>text</td>
</tr>
<tr>
<td>primary key: staffid</td>
<td></td>
</tr>
</tbody>
</table>
+==================================================+

+==================================================+
<p>| Table &quot;student&quot;                                  |</p>
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roll</td>
<td>character varying(15)</td>
</tr>
<tr>
<td>Passtouch</td>
<td>Boolean</td>
</tr>
<tr>
<td>Username</td>
<td>character varying(30)</td>
</tr>
<tr>
<td>Section</td>
<td>integer</td>
</tr>
<tr>
<td>Password</td>
<td>text</td>
</tr>
<tr>
<td>primary key: roll</td>
<td></td>
</tr>
</tbody>
</table>
+==================================================+
```

Figure 3: Staff and Student Tables
Likewise the table “assignment marks” has: assignment category(integer), assignment number(integer), roll/group(varchar, integer), staffid (varchar), run(integer), marks(integer), comment(text), host-attributes and primary key assignment category, assignment number, roll/group, run and this table also falls under the second category of tables. All the information is generated and extracted by the script, except marks and comment, which are given by the evaluator. Other critical information which requires non-repudiation is about proxy submission, re-submission, notices, attendance, etc.

4.2 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 files 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.

4.3 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 identification 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-specific password file is maintained which contains a user name and encrypted password for each of the group members. In our scheme the password file 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.
4.4 File repository directory structure

The submission directory structure is shown in Figure 4. In addition to this we have a similar assignment description directory structure. As shown in Figure 4, 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 identifier 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.

![Diagram of repository directory structure]

Figure 4: Repository directory structure.
5. Features
The purpose of our web-based utility is to provide a simple yet powerful interface for managing courses, along with the flexibility of online submission for the students. The top-level course management web page is shown in Figure 5. The interfaces are either public or protected and they are linked as shown in Figure 6. Note that the users of WBCM are: course administrator, staff (teaching and/or supporting) and students. WBCM has got 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

<table>
<thead>
<tr>
<th>Course code</th>
<th>Course name</th>
<th>Year</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBS406001</td>
<td>PGDIT Programming and Data Structure</td>
<td>2003(Autumn)</td>
<td>L S T</td>
</tr>
<tr>
<td>CS13002D</td>
<td>Programming and Data Structure - Sec 4</td>
<td>2002(Spring)</td>
<td>L S T</td>
</tr>
<tr>
<td>CS23005</td>
<td>Algorithms</td>
<td>2002(Autumn)</td>
<td>L S T</td>
</tr>
<tr>
<td>CS43001</td>
<td>Compiler Construction</td>
<td>2003(Autumn)</td>
<td>L S T</td>
</tr>
<tr>
<td>CS43051</td>
<td>Computer Organization and Operating Systems</td>
<td>2003(Autumn)</td>
<td>L S T</td>
</tr>
<tr>
<td>KGP406001</td>
<td>PGDIT Programming and Data Structure</td>
<td>2003(Autumn)</td>
<td>L S T</td>
</tr>
<tr>
<td>SL406001</td>
<td>PGDIT Programming and Data Structure</td>
<td>2003(Autumn)</td>
<td>L S T</td>
</tr>
</tbody>
</table>

Figure 5: Course Management Web page
Figure 6: Interface navigation diagram.
6. Towards web-services for LMS components

There have been many recent steps towards opening up learning management systems through development of open standards. For example, standards designed to de-couple learning content from proprietary packages such as SCORM and more recently the IEEE Learning Object Model (see, for example, Diffuse 2002). 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 (Downes 2002) discusses these points and John L. Hall also 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).

The current WBCM system has great potential to be developed into highly reusable components which could be made to fit in with other LMS components and work alongside them. In particular, the fact that it is web-based will make it relatively easy to turn it into a collection of web-services. 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. This involves the use of XML for describing data to be transferred, SOAP to wrap messages for delivery via HTTP, and standards for describing services (in a machine-processible form) and for the discovery of services. There are other projects starting to make use of web-services with learning management systems, such as (Ternier et al 2003) which describes possible use of web services with the ARIADNE learning object repository system (Duval et al 2001).

A web-services version of WBCM will require the development of appropriate (implementation neutral) XML schemas for the various forms of information required and supplied by WBCM at its interfaces. This, in turn will mean finding good generalisations of some of the current forms of information. As an example, consider the nature of "marking and feedback" for which XML schemas could be developed. These could allow highly flexible (extendable) forms of marking schema (e.g. with criteria, weighting) and different forms of marks. This would then allow a course designer to design a marking scheme (using an XML schema to describe it) and have this automatically 'plugged in' to attach to an assignment (with associated automatic processing).

There are several benefits to be had from a web-services design for our WBCM. Web services would help to isolate more of the implementation and design decisions for the current WBCM 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 with other course management or learning object management components should be greatly enhanced.
Our new design is likely to be based around the following key services

**A Submission service:**
This would authenticate users and receive submissions as XML documents containing all the relevant information for storage (submitter, course run, etc.). The submission would return (at least) an acknowledgement, but could be adapted to provide a 'tracking' service. This could be in the form of a URL which would be queried for feedback and marks at a later date, or a delivery by email.

**A Marker service:**
This would authenticate users and retrieve items (XML documents) for marking, giving access to appropriate marking schemes (to be used to add marks and comments). This service would allow updating the server information to record new marks and feedback. The service would also provide for generation of summaries for individuals, groups, classes and would keep track of unmarked parts.

**A Course Administration Service:**
This would 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.

**An Assignment creation service:**
This would authenticate users and allow creation based on a marking scheme (criteria, 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.

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. Services have following characteristics:

1. An interface (e.g. web-page) is provided from where a user makes request for the service.
2. An action script (known to the interface) is executed using the server/client machine resources to complete the request.
3. Actions may be dependent on context in the sense that each time a user asks for the action the script is invoked with a different form object/value depending on which, action is taken. This is the case for WBCM services, e.g. submission information is kept according to submitter, course, assignment, etc.

The functionality of the web interface and the action script is inherently different, but they co-operate to get context specific work done with the involvement of the user. So we may have interface or/and action script providers. (Standards become particularly important when modules are required to be linked.) Each action script needs to have an XML specification listing all the attributes. E.g. An action script which takes action for providing a submission service may come with an XML file having the following form:

```xml
<service>
   <action>
      <link>URL</link>
      <attribute>
         <name=course_code datatype=varchar description=course code>
         <name=year datatype=number(4) description=year>
         <name=semester datatype=varchar description=semester>
         <name=assignment datatype=number description=Assignment>
         <name=roll datatype=varchar description="student identity">
      </attribute>
      <technology dependent information>
         <software>
            <hardware>
         </technology dependent information>
      </action>
   </service>
```

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

7. Conclusions

We have described the current architecture of a web-based course management tool (WBCM) which 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 (assignment tutorials, references, articles, internet resources, question paper etc.) management, although it currently only keeps information about online tests and assignments.

Plans for the evolution of this tool into a web-services form have also been described. This planning is aimed at improving the tool's potential for integration with other course management tools. It will also open up some of the modularity for other developers who might want to use some services but not all of them.

References


