

WebTeach: an Integrated Web-based Cooperative Environment for Distance Teaching

Franco Bagnoli

Dipartimento di Energetica, University of Florence
Via S. Marta, 3 I-50139 Firenze, Italy bagnoli@dma.unifi.it.

Fabio Franci

Dipartimento di Sistemi e Informatica, University of Florence
Via S. Marta, 3 I-50139 Firenze, Italy fabio@dma.unifi.it.

Andrea Sterbini

Dipartimento di Scienze dell'Informazione, University of Rome 1
Via Salaria, 113 I-00198 Roma, Italy sterbini@dsi.uniroma1.it.

Abstract

We present the WebTeach system, currently used at the Engineering Faculty, University of Florence, Italy. The system is composed by three parts: WebCheck, a web database interface for test management; WebTest, a multiple-choices quiz generation system and WebWrite, a collaborative web publishing tool. WebCheck is used for managing subscriptions to tests and publication of results. It allows teachers to organize exam sessions and students to consult the results of their tests by a web interface. As a benefit, it allows students to be notified by e-mail in case of sudden cancellation of tests. Some of tests are generated using WebTest, which is a suite of perl libraries and front-end applications for the generation and automatic evaluation of multiple-choices quizzes. It is particularly suited for the generation of mathematical tests with L^AT_EX expressions. WebWrite is an implementation of the TWiki system, which appears as a web site where all pages can be edited using a simple syntax, resembling e-mail text. Moreover, files can be appended to pages like e-mail attachments, and there is a versioning control system. WebWrite is used for publishing didactic material and will become the generic interface-managing tool of the WebTeach system.

1 The WebTeach project

The *WebTeach* project started three years ago as a small on-line system to manage students scheduling for examinations. At present it has grown to include three different tools – *WebCheck*, *WebWrite*, *WebTest* – and enhancements are continuously being added.

In the traditional organization of Italian Universities teachers are supposed to organize themselves for what concerns exams management. This approach is clearly very unsatisfactory, especially considering the recent reform of the University which has introduced much more

compressed courses.

Since all teachers and most of students in our Engineering Faculty own a computer connected to the Internet, it is evident that all services should be web-based. There are many systems designed to ease teachers and students interacting by the web. One of them is offered by the central administration of the University of Florence [1] and has been developed by the University staff, another is offered by the Engineer Faculty [2] and has been developed by a private company.

These services present two main drawbacks: central-

ized administration and reluctance against the introduction of new features.

The development of the *WebCheck* tool was encouraged by teachers, unsatisfied by the possibilities offered by the existing tools and unable to convince the central administration to modify the existing software.

We started from a simple web-based interface to a database for the management of examination scheduling and publishing of the corresponding results on the web. We planned a long-term development, stimulating users' (teachers and students) participation to the design. Users' collaboration was motivated by promptly implementing their suggestions, and distributing the ordinary administration among teachers and secretaries, in order to avoid the "system manager bottleneck". We exclusively used free software both because its license allows us to adapt it to our needs, and because we would like to release it to the open software community.

System enhancements have been implemented by adapting existing free software. The *WebCheck* tool is described in details in Section 2.

The main enhancements introduced are the *WebWrite* and the *WebTest* tools. The first is an implementation of the *WikiWikiWeb* concept [3], appearing as a web site in which all pages can be edited using a simple syntax, resembling e-mail texts. This tool allows teacher and students to share didactic material and communicate among them. Moreover, it can serve as a generic interface, as described in details in Section 3.

The *WebTest* tool originated by the need of generating multiple-choice mathematical quizzes from L^AT_EX sources. We were unable to find a free tool for it, so we developed the one described in Section 4.

Finally, a survey of actual use of the system and future perspectives are presented in the last sections.

2 WebCheck

The *WebCheck* tool is based on a set of perl [4] scripts interacting with an SQL-database [5] where all data are stored. This leads to good performance in accessing information and a certain level of robustness of the system. There are three classes of users in *WebCheck*: administrators, teachers and students.

Administrators have full powers and can register other administrators, teachers and (when needed) students. At present, the name of courses and other global

setting can be inserted only by administrators, in order to keep a uniform "style", which is still fluctuating. As soon as this stabilizes, we shall let teachers insert and manipulate also these data.

As already mentioned, the main feature of *WebCheck* is the management of the scheduling of examinations. After having asked to an administrator to be registered into the system, teachers have to activate a course for a given academic year, activate a test for a given course, and eventually publish the results for a given test. Teachers can also extract cumulative reports for each course.

From a student's point of view the first thing to be done is registration. In the University of Florence, each student is assigned a unique id number, which is used as the identifying id in our database. All Florentine students are already registered in a central database, which is accessible by themselves through a web interface for filling-in their study plan and performing other administrative tasks. We have simply written a web client that, during registration, tries to connect to this central server using the student's credentials. If successful, the student is granted access and, if it is the first time that he/she uses our service, a record is added in the *WebTeach* database.

However, in the past years we discovered that freshmen's data are entered in the central database with a certain delay, so that the new students cannot register for their first examinations. Moreover, we were asked to manage also pre-entrance tests, i.e. to deal with users external to the central database.

In order to distribute this administrative task, we chose to let teachers enter students' registration, either one at a time (for example collecting their id during lessons) or massively after a test using a simple text file. Moreover, we used the unique fiscal code (assigned to everybody by the Italian government) as an alternative identification id.

After being authenticated, a student can edit his/her personal data, register for a test, or read the corresponding results. Teachers are free to choose whether the list of students registered to an examination or the results are public (viewable by all students) or private. This allows teachers to comply with the Italian privacy law, which forbids the diffusion of personal data (such as a result of an examination) without permission.

In case of last minute change in the scheduled tests

or some other important information to be broadcasted, teachers can send email messages to students using the addresses inserted by students themselves. In case of no address, students are notified as soon as they enter the system. In the near future, it will be possible to send messages to students using SMS messages.

There is no supervised check on data stored in the database, since we use an approach based on “community reputation”, which is reasonably enforced by the identification requested when logging into the system.

Recently, we have adopted a cookie-based authentication method with time-tickets, at the level of the http server. This permits to use the *WebCheck* authentication mechanism for any web application. In particular, it is used by *WebWrite*, as illustrated in the next section.

Even if *WebCheck* was born for a specific task, it resulted a very flexible tool. For instance, there are no constraints on the meaning of tests or courses names. One teacher can publish survey results from a set of partial tests or manage the queue for an oral examination simply using the test mechanism, or control students’ access to some resource, such as a laboratory, a seminar or other activities by creating “tests” belonging to a special “course”, namely “seminars”. There is no need to pass through the administrators.

3 WebWrite

The implementation of *WebCheck* is rather classic: a web interface to a database. The drawback of this kind of approach is that any change in the database structure or in the front-end requires a certain amount of “qualified” work, that cannot be delegated to teachers (or students).

This drawback is especially important when the overall design is still fluctuating. When we were asked to include more flexible communication tools between teachers and students, for sharing didactic material or collecting FAQs, we started considering different approaches.

We first considered e-mail like threads, like Hypernews [6], but then switched to a more flexible approach, based on the WikiWikiWeb [3] concept.

WikiWebs appear as usual web sites in which all pages can be directly edited by users through the web browser itself. A simple syntax is used to add contents to the web without knowing HTML: the philosophical approach is to input simple text as one does when writ-

ing an e-mail. It is server’s task to present this text in a nice way.

In the Wiki jargon a web page is termed “topic”, and a homogeneous set of topic is termed a “web”. A topic name is usually distinguished because it is written with uppercase letter in the middle, as for instance **WebTeach**. The presence of a topic name is automatically recognized, and the systems adds either the hyperlink to the corresponding page, or signals the possibility of creating the missing page. This favors a top-down approach to writing: authors start from the index, and then populate the web by clicking on the orphan links.

There are several different implementations of WikiWikiWebs, and also other similar approaches to cooperative environments, either free or proprietary. We have chosen TWiki [7], due to its active community of developers.

Topic contents are stored as text files, allowing the use of normal UNIX tools like grep to perform searches, RCS for version control and so on. In the source file the formatting elements are kept at minimum: emphasized text is simply surrounded by asterisks or underscores (an e-mail convention), bullet lists are marked by whitespaces followed by an asterisk and a space, URLs are just plainly written, and so on. Authors are allowed (but discouraged) to use HTML formatting.

During the visualization phase, the text is elaborated in order to format it as HTML, inserting bold, italics, bullet lists, hyperlinks, etc. There is the possibility of inserting “dynamic” commands to include other topic, insert the user’s name or the date of the day, and so on. In particular, web indices are plain pages containing just a dynamic command. The text is then embedded into a template, which furnishes the appropriate “skin” including buttons for navigation, searches, editing, etc. The templates are just text files with several dynamic commands.

The actual template can be selected by site preferences, web preferences, user preferences or specifying a field in the URL. As most of TWiki configuration, these preferences are selected by editing particular pages. We exploited the template mechanism to translate the interfaces to Italian, without (almost) affecting the TWiki code.

When editing, a simple text area with the source is presented, so that the author is not distracted by formatting tags. This favors focusing on contents rather

than on appearance.

Each topic can be classified by means of several category indices, e.g., lesson, course, prerequisites, related topics. The student can find his/her way through the didactic material both by doing full-text searches or by following the category links. TWiki allows a very easy definition of new categories: it is sufficient to edit a particular page and add the new term. A web can contain multiple category schemes applicable to a topic. This allows both a rough form of work-flow and a finer classification of each topic.

All topics can have files attached, i.e. uploaded to the server. This makes simple the distribution of didactic material. The uploaded files can be linked in the topic text, thus allowing the inclusion of images/multimedia files in the page shown.

TWiki allows the definition of access rights at the level of site, web-wide and at single page level. User administration is easy, because the definition of groups of users is itself stored as a topic (editable only by the administrator group). A hierarchy of groups can be designed by group-inclusion. The TWiki authorization mechanism relies on the server “Basic” authentication scheme. We replaced this mechanism by a custom handler based on cookie and the *WebCheck* database, in order to integrate the two systems without even touching the TWiki sources.

TWiki provides an automatic notification of changes that can replace bulletin board systems and even mailing lists. Mailing list management is performed by users themselves, just by editing a particular page in a web.

As in most portals, TWiki implements a simple full-text search engine with optional regular expressions. Mistakes or misuse are avoided by storing all the versions of a TWiki topic. No page can be deleted by the normal user.

3.1 TWiki extensions

Most of *WebWrite* users are physicists, and the need of inserting mathematical formulas using the *normal* \LaTeX syntax was soon expressed.

We also liked the idea of attaching an existing \LaTeX document (or a postscript figure, or a Microsoft Word documents, etc.) to a page and include it into the text as usually done for GIF images. Moreover, we would like to insert plot of functions, without the burden of

creating a GIF image and attaching it to the page.

Finally, we considered that most of discussions in Wiki sites assume the form of threads, with questions, comments, answers developing in a hierarchical manner. This is clearly possible using the usual Wiki unstructured editing, but implies a strict discipline, not usual among students (or teachers). Moreover, we would like to have different permissions for editing a topic and for adding questions or comments to it.

We started modifying the TWiki code (thanks to its free-software approach), but we soon realized that this would imply losing the contact with the TWiki developers community, thus introducing a weakness in the support and preventing the possibility of easily upgrading the software. On the other hand, the rest of TWiki users were not happy of adding heavy and useless (from their point of view) features to the generic TWiki code.

So we decided, in accordance with the TWiki development pool, to design a generic plugin API (which is now part of TWiki) that permits to selectively include new features.

Plugins can be activated on a web-by-web basis. They extend the system by adding functionality in several steps of the page creation: at the beginning (in order to handle the current session), just before rendering the topic, for each line, at the end of the rendering, just before saving, at the end (to handle a possible redirection).

3.1.1 \LaTeX fragments in the topic text

\LaTeX is the *lingua franca* for writing mathematics. Support for formulas in HTML is at the moment rudimentary (if not absent). In a future we shall be able to use MathML but for the moment the best way to show math in web pages is either by using translators from \LaTeX to HTML (e.g. TTH [8], LaTeX2HTML [9], hevea [10]) or by a Java applet.

To keep browsers' requirements minimal, we have chosen to implement a plugin for automatically transform to HTML the \LaTeX fragments present in a topic. We chose LaTeX2HTML because of the superior quality of its output, but the conversion is quite slow (it may produce several small bitmaps). In order not to affect the view pipeline the plugin is activated at save time, using a cache mechanism. The page produced is, anyway, passed through the normal rendering mechanism so that automatic links (topic names, URLs) present in the \LaTeX part can be handled by the TWiki machinery.

3.1.2 2D and 3D plots

We have developed a plugin that allows the writer to embed 2D and 3D plots in the topic text. 2D and 3D plots are created with the free software `gnuplot` [11] and shown as embedded pictures in the page. In this case, also, the bitmap generation is done at save time to keep the visualization faster.

3.1.3 Threaded discussions

TWiki already allows a shared editing mode for cooperation with students on the course material. In a minimal setting the teacher can leave the course topics editable from all students except maybe for some particular pages she/he wants to keep read-only (e.g. program course, exercises, etc.).

To make the cooperation easier, better structured and to allow free and even anonymous comments we have designed a plugin to add threaded comments to each topic of a web. The comments are kept in a separate web with different authorizations if needed. Each comment is a topic and is related to a topic/comment in the main web or in the comment web. The titles of all comments to a topic can be shown under the topic as an indented tree (like in a newsgroup). The view of comment tree can be enabled/disabled through the skins mechanism.

3.1.4 Calendar of important events

The calendar plugin is accessible through a dynamic command and permits to include a small monthly calendar in each page. The events can be defined in the same or another page using simple text, or can be extracted from the *WebCheck* database using a CGI query.

4 WebTest

Traditional written tests are handled through a series of open questions. The marking task is quite heavy, and this prevents the usage of supervised tests for monitoring classroom uniformity of learning and homework.

We started developing a system for the generation and automatic correction of multiple-choices quizzes, named *WebTest*. Since all of former users of the *WebTeach* system were either physicists or mathematicians, one of the first requirements was that of using

\LaTeX sources. Moreover, many teachers had already developed quiz databases using another hand-made software, and we tried to maintain compatibility with it.

For bureaucratic reasons our teachers prefer to handle tests on paper, rather than on-line. Moreover, computer rooms are not common in our University.

The quiz database is a \LaTeX file with special tags that marks the beginning of groups of questions, of questions themselves, of right and wrong answers. A group is composed by homogeneous questions (same difficulty), generated by hand or dynamically from a template using the Template Toolkit [13] tool.

In the presently developing version, the database is specified by an XML file, which can be generated starting from a \LaTeX source. This allows a much easier checking of \LaTeX syntax than starting directly from an XML source.

The actual test is defined by a specification file, which is a text file specifying the total number of different tests, the number of questions to be sampled from each group, and a number of other variables to be interpolated into the test template. The syntax of this file is compatible with *WebWrite*.

By combining the database, the specification file and a template (using the Template Toolkit tool), the system generates a \LaTeX file, and from it the paper printout. It is possible, by using a different template, to generate on-line tests using PDF (and the *AcroTeX eDucation bundle*). The system checks for duplicated answers, missing right answers, etc.

Teachers can exploit the *WebCheck* tool to generate a number of tests that matches the registered students, and even generate test personalized for every student (for instance, having the name of the student pre-printed on the paper sheet).

The answers to the tests are specified using a string of characters, which has to be copied on a text file by teachers. We are investigating the possibility of digitalizing answers by optical correction systems or even reporting them by means of private cell phone (widely used by Italian students) SMS messages. Markers are computed and optionally inserted into the *WebCheck* database in a semi-automatic way.

It is already possible to use *WebWrite* for sharing the database of questions among teachers. We are working to use it also as a graphical front-end to the *WebTest* suite.

5 Survey of *WebTeach* usage

The system is currently used in the Engineering Faculty of the University of Florence, managed by the Department of Applied Mathematics [12].

WebCheck is used by more than 70 teachers for about 130 courses, 450 tests (each with an average participation of 25 students), and a total of about 4500 students. Its use is steadily growing.

A previous version of the *WebWrite* tool was used to produce documentation for three courses (computer science, advanced geometry and calculus) and as a “virtual lab” of dynamical systems and statistical mechanics. This material is being migrated to the present version (which uses the new plugins described above) together with new material in the Physics field.

The *WebTest* tool is currently used by more than 20 teachers, and served for the entry test to all freshmen in the Florentine Engineering faculty (about 1000 tests).

6 Near future development

As a more general setting we would like to distribute the content over different servers, not always on-line, and transparently manage the updates. This would allow us to design distance learning settings where several schools, not always on-line, share common discussion spaces.

The simplest approach is to keep multiple copies of webs (one for server), all copies are read-only except for a single server. Synchronization of servers content is done through `rsync`. A more general approach could use CVS or replicated databases for the topic storage layer.

The Wiki instrument is especially suited to assist teaching to jailed students. They share the same inconvenience of far-distance teaching, without the possibility of consulting teachers and friends at will.

Since the jail administration is highly concerned about security, one has to carefully control the traffic of information, especially the possibility of arbitrary external contact via e-mail, HTTP or other protocol. The solution we are experimenting is based on two distributed Wikis (one internal, the jail, properly secured, and one external, connected to the Internet) whose content are synchronized at scheduled times. This implies a rigid separation of writing areas in order to avoid arbitration of changes to files.

Both servers shares the same accounts (for students and teachers). The internal server is accessed by jailed students, and the external server is accessed by teachers and other support people. By removing all unnecessary services but the Wiki one, we hope to meet the security level required by jail administration, still providing a useful tool to students.

There are a certain number of improvements that we would like to introduce in the next version of *WebTeach*. In general, enhancements are going in the direction of total integration of *WebTeach*, and the use of *WebWrite* as a content-manager tool. This task presents some difficulties since TWiki’s template system is not very sophisticated. We would like to integrate it with the Template Toolkit one [13], and display the *WebCheck* queries inside a *WebWrite* page using dynamic commands.

We are going to formally start a public CVS based project to coordinate the *WebTeach* development.

References

- [1] <http://stud.unifi.it:8080/>
- [2] <http://www.ing.unifi.it>
- [3] Bo Leuf and Ward Cunningham. “The Wiki Way: quick collaboration on the web”. Addison Wesley. 2001.
- [4] <http://www.perl.org>
- [5] MySQL, <http://www.mysql.org>
- [6] <http://www.hypernews.org>
- [7] <http://www.TWiki.org>
- [8] <http://www.hutchinson.bellmont.ma.us/tth/tth.html>
- [9] <http://www.latex2html.org>
- [10] <http://pauillac.inria.fr/~{maranget}/hevea/>
- [11] <http://www.gnuplot.org>
- [12] <http://didattica.dma.unifi.it>
- [13] <http://www.tt2.org>