

Design and Implementation of Interactive, Web Based Courses

Thomas Ebner, Christian Magele

Institut für Grundlagen und Theorie der Elektrotechnik (IGTE), Graz University of Technology,
Kopernikusgasse 24, A-8010 Graz, Austria, E-mail: ebner@igte.tu-graz.ac.at, magele@igte.tu-graz.ac.at

Thomas Dietinger

Institut für Informationsverarbeitung und Computergestützte neue Medien, Graz University of Technology,
Schießstattgasse 4a, A-8010 Graz, Austria, tdieting@iicm.edu

Abstract: Setting up virtual, Web based courses has become very popular among teachers recently. Still, most of them are not familiar with the proper implementation of training material for the Web. This was, at least in parts, also true for the authors of this paper. In this paper a comprehensive course construction concept using novel Web technologies is proposed, featuring simplicity and practicability especially for non technically experienced authors. All vital features like a learning environment, treatment of written basics and interactivity are briefly discussed and introduced with a pilot course.

1. Introduction

Today most universities and an increasing number of big companies all over the world feel the necessity to offer virtual Web based courses to educate their students or employees. Therefore, many teachers and authors of Web based lectures are in search of a complete concept to create virtual and interactive courses from text based scripts and books used for traditional lectures. Beside an excellent mastership of the subject area, there are a lot of additional requirements like programming experience, knowledge of Web technology and so on.

The Institute for Fundamentals and Theory in Electrical Engineering at the Technical University of Graz mainly deals with the numerical calculation of electromagnetic fields and the optimization of electromagnetic devices. Up to now, the use of Web technology was more or less reduced to the application of a browser. But the change of teaching and learning paradigms, the need for spatial and temporal independence for local students as well as “far distance” availability for non local and worldwide spread students, make it necessary to offer virtual courses. There have been some initial experiments with online electromagnetic field calculation examples using the CGI and TCL scripts to remotely control a Finite Element Software package and its Pre- and Postprocessors [Preis 1997]. Nevertheless, there has been neither a real concept how a virtual course should be designed, nor the technical know how to implement it, which, however, seems to be a common problem. This paper proposes a way to overcome these difficulties at the example of a highly interactive Web based course of the lecture “Optimization in Electrical Engineering”. Although the implemented course comes from the field of engineering, many of the suggested solutions, which had to be totally programmed by the authors, can be adapted for other fields. This is even more applicable as upcoming tools will virtually simplify the process of treating dynamic contents. Also users, who are no experts in Web technology, will be able to use these tools.

Basically, the task should be divided into two major parts: The Web based learning environment, which builds up the backbone of any good course and the handling of the contents. The content itself is again divided in two elements:

- Theory and basics
- Examples and Interaction

Nearly of same importance is the evaluation of the students advancement and their self assessment which accompanies and completes a course.

2. The Learning System

The quality and comprehensibility of the course content is definitely one of the most important parts when teaching a certain topic with the support of the Web. Unfortunately the course pages can not exist alone, at least some sort of navigation like leafing forwards and backwards, a table of contents etc. has to be provided. Also it would be nice if the learning and teaching process is supported by communication and collaboration facilities between distant students and teachers and among students themselves to utilize the possibilities of the Internet. Of course it is desirable that authors do not have to invest much time and work to accomplish this, in contrast routine tasks like interlinking pages should be performed automatically and it should really be easy to configure an environment that optimally embeds courses, so that authors can fully concentrate on the actual content creation process. To provide all this is exactly the purpose of a learning and training/teaching system! After evaluating a few available systems [WBT Systems 1999] [WebCT 1999] we decided to use GENTLE (GEneral Networked Training and Learning System [Dietinger 1998] [Dietinger 1997]) which is a research project currently under development at IICM (Institute for Information Processing and Computer supported new Media at the Technical University of Graz).

GENTLE is based on a Hyperwave Information Server [Maurer 1996] and only requires a JavaScript capable browser like Netscape 4.x or Internet Explorer 4.x (or greater) on the client side because the whole user interface is implemented using HTML, DHTML and JavaScript. The working prototype is also called Hyperwave Training Space. The WBT system currently distinguishes four different types of users and provides different features and functionality for them. These types are participating students, teaching staff like trainer and tutors, courseware authors and system administrators. Users can also belong to several types at the same time (e.g. an administrator could be a student at one course and a teacher and author at a different one). Within the environment various so called *rooms* separate different scenarios and tools provide additional functionality to the users.

Typically users enter the environment through the *registration desk* where they can register for a personal account, read the introduction and an overview to the system or look at the list of available courses. After the users have registered they will get a unique login name and a private space on the courseware server, which is called *study room* and can be entered upon successful identification. The study room is the main working place and also provides access to all other rooms. Here students will get a list of suggested courses (depending on the users' profiles), may enter enrolled or review already finished courses. Teachers and authors may also create new courses by using a tool called *course wizard* which upon activation asks a few questions concerning the lecture (like aim of the course, prerequisites required to master it, time and place of live lectures if appropriate, tutors etc.) and automatically creates a course skeleton and some introductory HTML pages. The course content itself can easily be added either by using the Web browser or via drag and drop by using the *Hyperwave Neighborhood*, a Windows 9x/NT Explorer extension. Access to the *administration office* is only granted to teachers, authors and administrators and is used to maintain user accounts, teams, courses and modify system settings. Of course not all functions are available to all three user types in the same manner, e.g. teachers can suggest or assign their own courses to different students but may not remove other courses or delete user accounts. Although students do not have access to the administration office they can directly change their own profile by altering the settings of the electronic business card. Most parts of the business card like photo of the student, the name and other personal data may be made public and can be viewed by other members of the system whenever they click on a visible user name. This has the positive and important effect that a personalized virtual community can be created because users can imagine how other users look like or what their hobbies are, which simplifies teamworking and collaborating. The study room also contains a personal Web space where the students can develop their own projects or work together in shared areas.

If students select one of the listed courses in their private study rooms they will be moved to the *course room* of the specific course. The course room provides access to the course content, which is dynamically interlinked (including e. g. the creation of a table of contents) and adaptively presented, which means that it can vary depending on the students preferences. Furthermore this room offers different tools for communication like a messaging system which can be synchronized with Internet mail, a discussion forum with sophisticated functionality that surpasses e.g. Usenet News, and personalized and typed annotations. Annotations can be

directly attached to a certain area within the HTML content page (or to the whole page if it is non HTML) just by selecting the text area and adding the note. Afterwards an icon indicating the annotation and heading the selected area will be visible only for the author of the note, or for a team or the public depending on the assigned access rights. Annotations can also contain attachments and a certain type like remark, question, answer and supporting argument. If a student poses a question note, the tutors of the course will be notified automatically by the messaging system so that they can answer the question as soon as possible. After doing that, the student that asked the question will also get a notification that it has now been answered. To minimize question answer dialogs and thus supervising effort the system supports various sophisticated searching techniques like fulltext query within every part of the system and a course specific section for storing background material called the *background library*. This should improve the chance that students can find an answer to their questions on their own.

This just describes the current state of the system at the time of writing, however as we know huge enhancements like more tools that automate routine tasks, more support for collaboration and personalization and also a hybrid version that combines the advantages of online and offline learning and teaching are planned.

3. The Theoretical Part of the Lecture

The theoretical part of a technically oriented lecture in general consists of written text, figures, equations, tables and references. Standard Multimedia contents like mpeg videos or spoken audio files can easily be added without special plug-ins – which was a major requirement for our course. Nevertheless, the preparation of high quality multimedia content often exceeds many authors capabilities.

But even generating a single, multipage document caused great problems at the time this project started. So far, no proper HTML editor is known to cope with automated page numbering, figure numbering, equation and table numbering as well as chapter and section numbering and header and footer insertion which are standard features of nearly any text editing systems like WinWord or Latex . HTML editors like MS FrontPage98 or Netscape Composer were and still are supporting single pages only and deal with multiple pages in a unsatisfying way. To overcome this problem a couple of small utilities were developed using Perl programming language following the model of the Latex text editing system [Wall 1996] [Lamport 1994]. Each figure, table and equation is assigned a reference name in the plain HTML text which is processed and the respective numbering is inserted in the correct way. Headers and footers and any desired HTML parts can be inserted at marked positions in a similar way.

To ensure reusability of individual course chapters within other courses, a modular concept was adopted. Each single chapter is treated as an independent module with few hyperlinks to other chapters and no links to the Internet. Modules can be detached from one course and easily reused within another one by deactivating or reorganizing the hyperlinks. Instead of a flood of links within course pages, advanced searching and browsing facilities and additional reference links are provided via the background library.

But even inside a single chapter one can easily get lost somewhere in the hyperspace, between the current page and some auxiliary pages which are referred to and displayed [Dietinger 1997]. The authors have overcome this problem by keeping the current page always visible. Figures, tables, equations and even references which are not part of the actual viewed hypertext page are overlaid in extra windows on demand and closed automatically when the user browses to the next main page. This supports the generation of short and efficient pages.

Another feature, well known from offline education software but not easily available for Web based systems until Dynamic HTML, is a series of figures with a corresponding explanation successively displayed at the same location of the main page. Series of pictures with correlated text and little animations were implemented in Java Script using Cascading Style Sheet (CSS) features. Applying DHTML, figures and text can be separated and text remains text in contrast to “slideshow” figure/explanation series. Animations can be easily inserted. It has turned out that this is a good way that even complex subjects can be understood more easily.

Another special element is used to visualize program flowcharts or more complex (sequential or parallel) processes. An image map is controlling accordant images and illustrations with DHTML which may again be combined with animations. Applying such elements has the positive effect, that normally inactive text is enhanced with some interactivity. The students are called to click the image maps and serials, watch the

animations and changing figures according to their learning style and speed, which is much more challenging than plain text. Java Script program modules can easily be reused in other pages or courses with only few adaptations required.

4. The Interactive Part of the Lecture

There are two philosophies to allow interactivity for online courses:

- the client solution
- the server solution.

Examples executed at the client computer, mostly implemented as Java Applets, do not need an additional server and could be also used in offline mode (e.g. on a CD-ROM edition). Many authors strenuously realize sophisticated examples as Applet programs, but really complex examples like Finite Element calculations are absolutely impracticable with a client solution. Server side run programs are much more powerful, but the interface is hard to implement, and an additional online, administrated server is needed to provide full performance for both the learning environment and the examples. Therefore, we use a combination and benefit from both methods.

Simple numerical and theoretical examples are realized as client applications. For example, the student is asked to derive analytical formulas and the result appears directly within the page without a Hyperlink to click. This is the same method often used in text books, where the solution of examples could be looked up in the back of the book, but much more comfortable and faster. Simple numerical examples may either use the click and show system which is also applicable for ordinary text questions or the students are asked to enter their solutions which are validated by the system. The students receive either a confirmation that their results are correct or an invitation to redo the calculation with a text depending on the number of trials already performed. The design and implementation of these objects is fast, efficient and technically not too difficult, in particular because there already exists a growing number of commercial applications supporting such DHTML objects.

To deal with more complex examples, we have been using a server side solution. Like in modern class room lectures where computer applications are often used to support and improve the traditional lessons, Web courses can make use of these applications if they can be remotely controlled. Application programs in our implementation are MATLAB [Mathworks 1999], Fortran optimization routines and a Finite Element Package solving electromagnetic field problems. Using these software the design of very complex, real life simulations and examples is mainly reduced to conceive the example itself while its implementation is strongly supported by the program packages. The functional elements for user input are standard HTML forms and input data are transmitted using the "Post" method. Input data for training examples are mostly parameters – in the optimization case strategy parameters and problem parameters, formulas, sometimes instructions – and the output data are pictures, text, numerical data and animations packed into HTML pages. Online calculated animations are provided as image frames and displayed using an appropriate Java Applet.

The amount of data transfer over the Internet can be kept fairly low this way, all calculation and generation of HTML output is done on the server. The performance bottleneck of the whole chain is the application server. If there are too many students to be served at a time a splitting to multiple application servers can be done [Kovács 1998]. We have experience with about 150 students doing extensive Finite Element calculations within a month which was done with one server in a satisfying way. Since the calculation period of our examples lasts from some seconds up to some minutes a pure CGI compliant interface using Perl scripting language is sufficient. A simple, but relatively easy to implement form of remote controlling is possible if the application supports a scripting language and the transfer data is passed via files. Therefore our data transfer between applications and CGI is mostly handled by files and DDE (Dynamic Data Exchange), but it could also be worked out by another direct link between server programs and Web server (NSAPI, ISAPI) or with FastCGI, depending on the remotely controlled application [Göschka 1998]. A semaphor concept using file locking guarantees error-free concurrent multi-user service of the application server. The whole system structure can be seen in [Fig. 1].

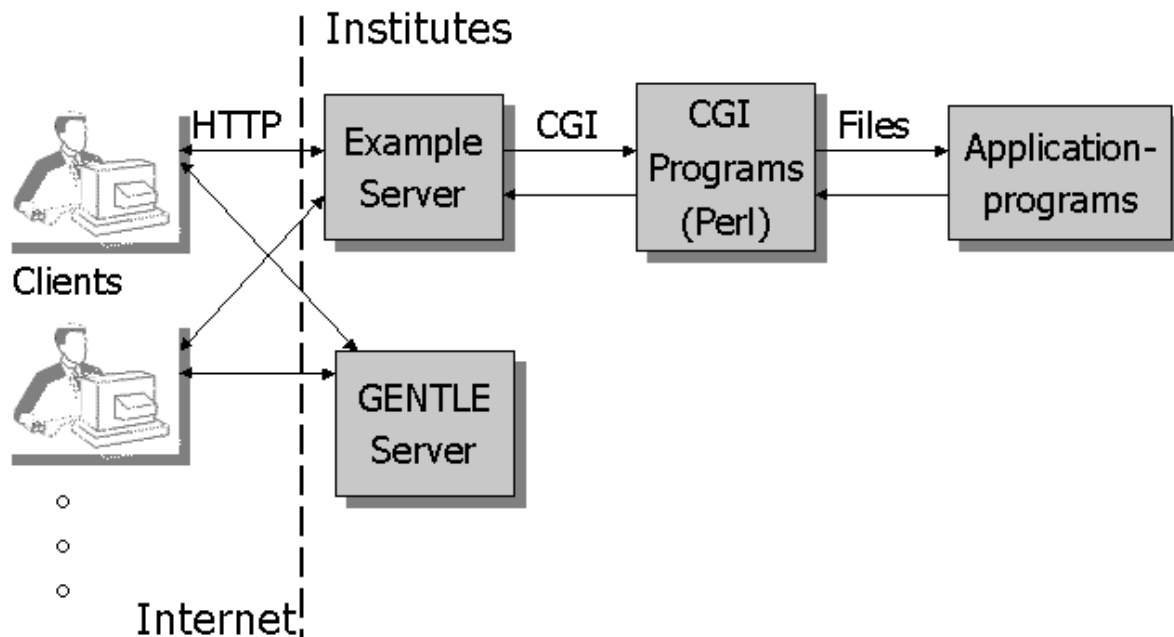


Figure 1: The learning system

5. Evaluation of the Student's Progress

Beside the most obvious form for evaluation in Web based courses, the multiple choice quiz, written statements, reports or exams with open questions are other possible options. Either an automatic grading (easy to gain with multiple choice quizzes) or manual assessment may be used by the teacher. At the time of writing of this paper no evaluation concepts were implemented in the GENTLE learning environment. Despite a rather high level of personal engagement, we decided to implement question forms with open questions and questions to be answered by a statement which are both graded manually. Although designed for self assessment, the objects described in the interaction section are very well suited for test and evaluation purposes. A number of questions can be based on a server based example module which has to be used by the student to answer the question. For example, the students are asked to run an optimization problem with different optimization strategies and interpret the results. Answers to the questions are automatically formatted and prepared for database access by CGI scripts. To ensure objectivity, examinations are held at fixed times in computer rooms with a person in charge.

6. Conclusions and Further Development

All necessary parts to successfully create a Web based course have been briefly discussed in this paper. The GENTLE system fulfills all requirements for an excellent courseware and the further development of the software benefits from persistent feedback of the course authors. Nevertheless also other courseware systems like WebCT or Topclass are suitable choices [WBT Systems 1999] [WebCT 1999].

The "first time" development of the quoted Java Scripts, HTML modules, Perl programs, CGI programs etc. is expendable and time consuming. Once available the modules can be adopted for other courses very quickly with few adaptations. Up to now the time consuming implementation of proposals has been reserved to technically experienced people with programming skills, but forthcoming application software like Dreamweaver2 will ease the task significantly [Macromedia 1999]. The most difficult job remaining will be the adaptation of server based software especially because no practicable solutions for that can be expected in the near future.

Currently, there is much work in progress to apply solutions for interactive tutorials for Finite Element Software applications also in offline mode. In the future we plan to transfer many more lectures from the traditional way of teaching to the virtual one with a high grade of interactivity. Concurrently, further progress takes place in the development of the GENTLE virtual learning system and new features like standardized evaluation methods will be added.

7. References

[Preis 1997] Preis K., Bárdi I., Biró O., Hoschek R. and M. Mayr, A Virtual Electromagnetic Laboratory for the Classroom and the WWW, *IEEE Transactions on Magnetics*, Vol. 33, No. 2, pp. 1990 – 1993, March 1997

[WBT Systems 1999] TopClass from WBT Systems, <http://www.wbt systems.com/>

[WebCT 1999] WebCT Homepage, <http://homebrew.cs.ubc.ca/webct/webct.html>

[Dietinger 1998] Dietinger T., Maurer H.: GENTLE – General Network Training an Learning Environment, Proc. of *ED-MEDIA98/ED-TELECOM98*, ISBN: 1-880094-30-4, pp. 274 - 280, Freiburg, Germany, June 1998, <http://wbt.iicm.edu/gentle/papers/edmedia98.pdf>

[Dietinger 1997] Dietinger T., Maurer H.: How Modern WWW Systems Support Teaching and Learning. Proc. of *International Conference on Computers in Education (ICCE) 1997* (Ed. Z. Halim, T. Ottmann, Z. Razak), Kuching, Sarawak Malaysia, December 2-6, 1997, pp. 37 – 51., <http://wbt.iicm.edu/gentle/papers/icce97.pdf>

[Maurer 1996] Maurer, H. (1996). *HyperWave: The Next Generation Web Solution*, Addison Wesley Pub.Co., UK

[Wall 1996] Wall L., Christiansen T., Schwartz R. L., *Programming Perl, 2nd edition*, O'Reilly & Associates, Inc., 1996, <http://www.perl.com/pace/pub>

[Lamport 1994] Leslie Lamport, *Latex – A Document Preparation System*, Addison-Wesley Co. Inc., Reading, MA, 2. ed, 1994

[Mathworks 1999] The Mathworks, Inc., 24 Prime Park Way Natick, MA 01760-1500 USA, <http://www.mathworks.com>

[Kovács 1998] Kovács F., Szitnyai G., Monostori K., Charaf H., Vajk I., Teaching Control Theory Interactively, Proc. of the *15. IFIP World Computer Congress, 1998*, Aug. 31 – Sep. 4, (ed. Gordon Davies), Vienna/Austria and Budapest/Hungary, pp. 579 - 588

[Göschka 1998] Göschka K.M., Riedling E., Web Access to Interactive Database Training: New Approaches to Distance Laboratory Work at the Vienna University of Technology, Proc. of the *15. IFIP World Computer Congress, 1998*, Aug. 31 – Sep. 4, (ed. Gordon Davies), Vienna/Austria and Budapest/Hungary, pp. 349 - 360

[Macromedia 1999] Macromedia Dreamweaver, Homepage, <http://www.macromedia.com/>

Acknowledgements

This work was supported by the “Austrian Fonds zur Förderung der wissenschaftlichen Forschung” under the Grant No. F003.