The tutor-web: An educational system for classroom presentation, evaluation and self-study

Gunnar Stefansson

Science Institute, University of Iceland, Dunhaga 5, 107 Reykjavik, Iceland

Received 15 May 2003; accepted 18 September 2003

Abstract

A general Web-based system for use in education, the tutor-web, has been developed for storage and presentation of electronic slides for classroom use, along with reference material, examples and quizzes. The primary novelty of the system is the structured linkage between the various pieces of information, to maintain coherence and focus on the educational content in spite of being Web-based.

The system has been developed with an emphasis on teaching mathematics and statistics, but the design is generic. Students have full access to all material outside the classroom. Material within the tutor-web is organized into lectures and tutorials. The material can include text, equations and figures. The system can include multiple-choice questions and will automatically collect the returned answers as well as return rates and grades.

The system can be used for generating and presenting electronic slides for in-class use but it can also provide a textbook which links together the slides and associated reference material. A formal definition of the format of content is used, but it is also easy to extend the definitions. Given the formal formats it requires only minimal effort to modify presentation forms and add entirely new ones. The tutor-web is a developing project in an open environment, for general access and open to anyone interested in developing it further or using it.

The system itself is presented as are analyses based on students in one class, linking the performance using the new system to a written exam.

© 2003 Published by Elsevier Ltd.

1. Introduction

During the past few decades the use of electronic transparencies, or slides, has become steadily more popular. After about 1990 such slides came into general use at conferences but in teaching
circles only nearer the end of that century. It is clear that electronic slides can provide considerable work savings in the longer term since much of the material can be re-used.

On the other hand it is equally clear that such slide techniques are already outdated since some aspect of traditional teaching which need to be computerized cannot be fit within ordinary electronic slides. Items such as short additional material can be included in some slide generating tools (sometimes as so-called “speakers notes”) and can be a useful addition, allowing the lecturer to include at least some information in addition to the bullet points typically used on slides. However, extended optional or detailed material, quiz questions, Web-pointers and so on are usually not included within any system, and certainly not in a generally applicable or generic manner. In spite of this most systems for handling electronic slides are quite expensive. Thus, although features included in advanced presentation systems include movie clips, audio and so forth, these features are much less important than the aforementioned, as far as instruction is concerned.

In addition to these concerns additional issues crop up concerning programs which are not open or do not follow open standards. The greatest of these issues is that there is no guarantee that teaching material can be used in 10 years time if it has been developed using such programs. This is clear when it is considered that there is very little electronic material available today which was developed 10 years ago. The reason for this is not the age of the content but only the age of the presentation and storage methods. Thus, a considerable portion of material from a mathematics textbook from 1950 would be useful today (though particularly examples and figures) if the material was publicly available in a generic format. The same applies to text which may be several centuries old, as best seen from the interest in projects which strive to put old texts into new formats, made publicly available on Web pages.¹

When considering the various aspect of teaching, priorities can be set in many ways and these priorities will govern the design of a computer system to be used for teaching. The first priority here will be the possibility to put all teaching material into the system and to build the system around slides to be used in the classroom. Thus, photocopies from textbooks or examples should ideally not be needed except as set up within the system. Slides used in the classroom form a focal point for teaching material as the students see these slides when the teacher presents the material. It is therefore natural to use these same slides as a center when students look through the material outside the classroom. Similarly, it is a natural requirement that a student should be able to request a quiz on material which is being considered, i.e., the quiz should be directly linked to the slides or lectures.

In addition to these priorities, a system which is to be of general use for teaching must be generally accessible. For instructors who teach or lecture in more than one institute it is also a requirement that all material should be accessible to all such institutions. The only way to ensure this is to base the system exclusively on standard formats and open source software.

The following discussion will mainly deal with the design of a system which can, in addition to text and figures, handle mathematical symbols. It is imperative that equations can be handled in a convenient manner, but this is a serious failure of most systems for handling electronic transparencies. Notably, equations must not be handled as graphics except possibly for the purpose of

The storage format of equations must, for example, be one which can be easily edited.

The layout of material must be in such a manner that different categories of material can be linked. Thus one must be able to point to calculus when teaching statistics, yet the material must be separated into small units which can be studied individually. Given the form of presentation, the natural ordering is to group slides into lectures and lectures into lecture groups on a well-defined topic. These topics or collections of lectures will be termed tutorials.

This paper describes a system which handles all of these issues. The system goes by the name “tutor-web” and is completely accessible through the URL \(^2\) http://www.tutor-web.net. As the system is Web-based, it is primarily HTML \(^3\)-driven.

As seen from the above, the design features in the proposed system are not only considerably different from simple slide-projecting systems, but is also different from common campus-wide system for in-class use and remote learning, where the emphasis, at least in practical situations, is in uploading material to make it available to students, usually in a password-protected manner. There is no enforcement of links between slides, quizzes and other material and no natural linkage between courses or methods for students to seek alternative material within the system. As a rule there are no standards in formats used within or across system. As a result these schemes do not obviously contribute to joint use of material across institutions nor provide any guarantee that material will not be obsolete after a few releases of an operating system or commercial package.

The proposed system is also different from common open-source systems based on Zope \(^4\) or WikiWeb \(^5\) concepts or general Web-based approaches where, in many cases, the system is designed to be very open, encouraging general use of Web-based resources. The danger with these alternatives lies in the openness itself, firstly in that the material accessed by the students has no built-in quality assurance but more importantly the Web itself is completely unstructured and this does not encourage the focus required to solve hard problems, understand complex issues or learn difficult methods. On the contrary, too much flexibility can easily lead users of any system to spend the day pushing different buttons in search of something appropriate happening, rather than sitting down and tackling the issue at hand using the brain as a tool. Thus if it is correct that a plateau is being reached in the use of Web-based resources in higher education (Crooks, Yang, & Duemer, 2003), then the above arguments can be put forward to suggest that perhaps the use of Web-based resources can sometimes be too much of a good thing.

Although the system presented can be used by an independent-minded student as a tutor-system, it is not designed to be an intelligent tutoring system (see, e.g., Moundridou & Virvou, 2003). Rather, this is seen as a method for organizing into a structure all material normally used by a teacher in a classroom. Similarly, although the system can be used for remote-learning, this is not a primary design goal but a consequence of other goals.

Several systems exist for in-class assistance, administration, remote learning or content management or different combinations of these. Some of these systems are available under open li-
censes, such as the Whiteboard Courseware System,\(^6\) whereas others tend to be both expensive and closed source so these cannot be used freely or at finance-limited educational organizations. Some of these systems mainly provide content management, a mail interface and a bulletin board. These are very useful features, but they do not automatically provide uniformity in content definition, links between slides and quizzes, etc. Similarly, learning (content) management systems tend not to provide this linkage, which is an essential feature of the present system.

The tutor-web is not intended to be or replace a “distributed learning community” (e.g., Chang, 2003), many aspects of which are available to and popular with even the youngest generations of school-children through IRC,\(^7\) as there is at present no such mechanism built into the tutor-web.

2. The system

2.1. System components

The primary attribute of the tutor-web is the structure of material or educational content. A variety of methods for presenting such material have been developed, explored and made available, but these should be seen mainly as examples as it is easy to develop alternative presentational methods once structure is in place.

Material on the tutor-web is organized into a tree (much like a Linux file system), which should be viewed upside-down with the individual slides and questions at the bottom, as the leaves, but the system itself is the trunk (Fig. 1). A specific collection of slides is grouped together into a lecture and a few lectures (normally 2–10) make up a tutorial. The tutorials themselves are grouped into departments. A course in a physical school usually corresponds to several tutorials, which might very well cut across several departments.

Some care is needed when organizing the tutorials since the design must take into account several issues. A tutorial should be a collection of lectures which are built up around a single theme. A course which takes a full quarter or semester should therefore usually be split up into several tutorials. In a one-semester calculus course differentiation would be expected to take several weeks. A tutorial might be the basis for differentiation, i.e., the slope-of-a-curve concept, the formula for a generic (differentiable) function and some elementary rules of differentiation. Such material might provide enough content for 1–3 weeks of lectures and would be an adequate basis for a single tutorial since it would be clearly delimited. A second tutorial could then address more complex issues such as the differentiation of composite functions and/or trigonometric functions.

The reasons for the tutorial concept and the preferred size of tutorials (2–10 lectures) are several, most of which are simple and pragmatic. The most important reason is that if it is to be feasible to set up on-line courses, each course must consist of units which are small enough to be implemented within a reasonable time frame. The task of setting up from scratch a complete one-

\(^{6}\) http://whiteboard.sourceforge.net/.

\(^{7}\) Internet Relay Chat.
semester course with attention to all required detail is simply prohibitive. In comparison, setting up a handful of lectures is fairly trivial. Upon completing a draft version of the slides for these lectures, they can be test-run on students and subsequently developed further as needed. The total amount of work required for setting up content in an electronic teaching system includes slides, detailed material, examples and quiz questions which in total is much more than is required for traditional teaching preparation. For this reason, it is essential that a course can be built up piece-by-piece and this is more than enough to require the concept of tutorials as building blocks for the courses.

There are also pedagogical reasons for making the building-blocks smaller. In particular, this means that a student can complete a portion of the material and perform a self-evaluation based on smaller blocks than a complete course. This is an essential part of any system and is of course reflected in the preponderance of quizzes on selected topics within courses given in the physical world.
When starting a new tutorial it is possible to set up only slides and later add all other material. It is also possible to take an existing textbook and use this to generate automatically a collection of complete Web-based tutorials. In either case this results in a good basis for further development, as will be detailed later.

2.2. Defining a slide

The electronic transparency, or slide, is a basic unit of the tutor-web. A primary concept within the system is that a slide is a presentation of a fundamental idea, a specific piece of information, which could be termed an “infobit”.

It must be possible to transform slides, as well as all other content, to several different formats. Obviously a slide must be a piece of information which can be projected onto a computed screen, but options must also be available for printing and several display methods will be needed, including Web-based display, inclusion as a part of a printed textbook and so on.

For these reasons the components of the slide must be well-defined. Once the components have been defined they can be used repeatedly, in ways not imagined earlier. This is to some extent the same idea as in traditional slide presentations, where a graphic can be copied from one slide onto another or into a document, but the crux here is that these conversions can be made automatic since the format is predefined.

It is important to note that although a format has been defined, it is evolving and will be extended as needed in the future. The importance of having a standard format for the entire tutor-web cannot be over-emphasized, however.

In the first versions of the system a slide consists of a title, main text content, a main figure (graphic), explanatory material and an explanatory figure. Each of these can be omitted (except for the title) and there are several options on how the components are organized (see Fig. 2).

Fig. 2. A typical slide as presented in HTML format, with 2 types of text and 2 graphic types. Also shown are a variety of buttons.
In what follows, frequent references will be made to “text”. For most “ordinary” text slides, this “text” is just a collection of characters from the English alphabet with punctuation marks. In order to allow very simple additional formatting, such as bullet lists, the “text” can contain a few special symbols such as an asterisk (*) at the beginning of a line, for a bullet-item (automatically translated into a bullet in each presentation method). For equations the mathematical system \LaTeX\ 8 (Goossens, Mittelbach, & Samarin, 1993; Knuth, 1984) is used. Thus, “text” is either the former, largely unformatted ASCII text, or in \LaTeX\ format.

As seen below, a system which includes options of main and/or explanatory texts as well as main and/or explanatory figures already provides a number of options in laying out a slide. The selection of any or all of these is up to the presenter.

2.2.1. The title

All slides must have a descriptive title which in very few words explains the slide. This is considered an essential part of any slide.

The title of a slide can subsequently be used as the name of a subsection when printing the contents of a tutorial, and/or as a figure caption if no other caption is given.

In the present implementation of the tutor-web, the title is limited to letters of the alphabet. Due to processing methods used, the characters &, $, ! and # are reserved.

2.2.2. The slide body text

Most slides have some main content in textual form. This “text” can be a bullet-list, a paragraph, a collection of equations or any of the above. There are two main issues to be considered when a slide is organized.

First, from a pedagogical viewpoint it is important that the main text of a slide is well structured.

Secondly, for a slide to be legible during an in-class presentation (particularly in large lecture halls), it will by default be presented in large typeface. The slide text must therefore be concise.

These two main concerns go hand-in-hand to make legible and understandable slides. On the other hand, the content must of course also cover the concepts which the slide deals with and this can become quite difficult to reconcile with legibility and concise presentation.

Any additional material is better presented either separately altogether, i.e., not on the slide, or as explanatory material in smaller print in the form of a footnote.

These are the same general issues as apply to physical transparencies, i.e., of clarity both in terms of textual content as well as format. It is well-known that the best content can be destroyed if poorly presented whether through layout, choice of words or font sizes.

2.2.3. The main graphic

Many slides have one main “graphic 9” which illustrates the material described by the slide title. When both a main text and a graphic are used, both should fit on one slide, side by side.

---

8 A command set based on TeX.
9 In this document, a “graphic” is a figure, drawing, picture or photograph which is stored in an image format.
The “graphic” can in principle be any figure derived from any source. The format itself is described elsewhere.

2.2.4. Explanatory material

In addition to large bullet-point text, or the main body of a slide, some explanatory material often needs to be included on a slide. This may, for example, be much of the text which the presenter puts in a verbal presentation. This is not as important as the main text and can therefore be down-played in the presentation.

This material can in principle be as long or extended as needed. A great amount of explanatory material will, however, require some form of paging within a slide and to some extent this defeats the purpose of a slide being a concise presentation of a single piece of information. Slides requiring long-winded explanations would normally better be slit onto two slides or, alternatively, some of the material should be tucked away as reference material (possibly leaving key words on the slide).

In addition to the slide content presented in-class, the tutor-web has other methods for storing detailed material. Normally all the content of the explanatory section of a slide is also available in such form and therefore the explanatory part of the slide is usually not a part of a printed document.

The explanatory part of a slide can also be the only text of a slide, to go with a main graphic. This allows the graphic to take a more prominent role on the slide.

2.2.5. Explanatory graphic

An additional or explanatory figure is commonly added to a slide. The purpose of such a figure can be to enhance a slide which otherwise consists only of a main text, to go with an explanatory text in footnote-style or simply to complete a slide which already consists of another figure and texts.

Of course the explanatory graphic can also be the only graphic on the slide. For example, if some bullet points are used as the main text, the explanatory graphic can accompany this text as a down-played but important graphical explanatory figure.

2.2.6. Other parts of the slide

In addition to the above main pieces of information a slide can contain or link to several other items. These are in each case either optional or automatic. Most of these are discussed in more detail further on.

Quick pointers. The Web presentation contains pointers in a column on the left side of a slide. These are at present automatically put into all slides in a lecture, thus making jumps to specific lecture location easier.

The pointer label text can be defined in advance by the presenter. Alternatively it is generated automatically. The text must be very short, of the order of 8 characters.

Figure captions or information. Text can be stored for each figure. When viewed on the Web, most browsers will show this text if the mouse pointer is over the figure.

When available this text is used as a figure caption when a figure is printed.

Navigation: The button system. When slides are viewed over the Internet a series of buttons appears at the top of the slide. These appear on green background if an associated function exists but otherwise the background is red.
The buttons are always located in the exact same location on the screen, regardless of the size of the slide. Thus, when a presenter or student moves forward through the slides, the mouse does not need to be moved. This is crucial to the use of the system in a classroom and contrasts most Web-based systems where buttons tend to float up or down on pages according to the amount of content. This issue is the reason why the buttons are all at the top of the page, separate from the rest of the content. This is not a trivial concern, as first of all it is the reason why some presentation systems allow a single mouse click anywhere in a window to move a slide forward. Lack of standardization is also one reason why many presenters are seen to have problems when backing up is needed as this is not standardized even within some systems (see Fig. 3).

The main use of these buttons is to select the next (N) or previous (P) slides in a lecture, but in addition an up (U) button is used to get from an individual slide to an overview of the corresponding lecture. These three buttons are very important parts of the whole system and some care has been taken in developing their location and function. The previous and next buttons simply flip backwards or forwards through the slides in a lecture. When going “off the end” of a lecture, the user pops up into the lecture overview, where the titles of all slides in the lecture are shown.

A detail (D) button is used to access detailed information on the topic of the slide and an example (E) button is used to access examples. Each of these open new windows so that the user can maintain the same location in the slide view while temporarily considering more detail surrounding the slide.

Separate buttons are used to obtain a quiz question (Q) or a grade (G). These also open new windows to handle questions and grades. In either case a login procedure is initiated the first time one of these buttons is pressed, since a userid is needed to pair questions, answers and results.

Several other buttons exist or are under development. Some of these include an information (i) button, a button for alternate (A) educational material on the same topic (this might be another tutorial or even material in another Web-based system), a button for references (R), a button for a version of the slides in Portable Document Format 10 (pdf), and a button for author or copyright (C) information.

**Background.** The default background can be modified for a given lecture. This can be quite useful when a lecture or tutorial needs to stand out somewhat.

In principle this means that the background can also contain information, in addition to the slide text and graphics. This does, however, mean that information becomes quite condensed on a single slide and is quite difficult to do well. For some lectures, however, there are only few slides and not much on each one. In this case, it can be useful to maintain the background as a reference which the lecturer can point at to indicate issues which come up repeatedly.

It would be a simple addition to allow a separate background for each individual slide, rather than solely for entire lectures.

---

2.2.7. Further development

It would not be a complex addition to allow an optional location of graphics on the slides, e.g., by defining optional coordinates.

2.3. Details

Although electronic slides are useful for presentation purposes it is clear that this is in no way enough material when a course is taught or taken. In particular general details on the material is usually needed. Such detailed material can be linked to any slide on the tutor-web, available through a button (D). This material can be in either LaTeX format or as a simple ASCII text file.

The detailed material is available to a student browsing through the system and also becomes a natural part of a printed version of the course material. When a general figure is included on the slide to illustrate a point, this should be explained in in the detail part, in order to obtain a context in the printed version.

The concept of detailed material being immediately accessible from a slide is an important feature of the tutor-web. This links detailed explanations directly to the slide used in-class, the idea being that a student may not have completely understood the in-class presentation but observing the additional detail outside the classroom, when linked to the slide, should aid in the understanding. Verbal responses by some students indicate that this hypothesis may be correct, but it has not been evaluated quantitatively.

This design expressly avoids the link to the outside world, attempting to keep the student focussed on the material on the slide or in the lecture. This is considered an important part of the design since a general Web search will consume much more in terms of the student’s resources than a direct lookup into the detail slide.

When the detail slide is not enough some alternative is needed and it is planned to use a specific button for alternative material, which should again be a direct link, but not an open Web-search.

2.4. Examples

In many courses a special type of extended material is examples (or small case studies). This material is often in abundance and is in the majority of cases therefore only presented quickly if at all in the classroom but is essential for a student studying the material in depth. Similarly when a printed version of the tutorial is obtained, such material is essential.

On the tutor-web any slide can have associated examples. Such examples are written in either LaTeX or stored as simple ASCII text files.

On a pedagogical note, if a picture containing data is used on the slide, then the corresponding example(s) should at least contain a reference to this material.

In addition to examples being reference material, some need to be used as a part of regular classroom teaching. The easiest way to handle this is to use the example material both as a part of the explanatory part of the slide and as the slide example. This can, e.g., be done by using the exact same information in both places. Although not the neatest of all solutions, this is simple and it does not cause duplications in any screen shot or printed version since the printouts do not contain the explanatory information and the associated example is only shown in response to the example-button and not on the slide proper.
2.5. *Homework assignments*

Many courses include homework assignments as essential elements. These homework assignments are sometimes simple, i.e., in the form of a list of questions or problems to be solved, in which case the on-line quiz-questions form an alternative. In other cases the homework assignments may take the form of more extensive problems to be solved in detail and a part of the assignment is not just the solution but also the solution method. This is harder to enforce in a Web-based quiz system (but can in principle be done, piecemeal).

Paper-based homework assignment is currently implemented within the tutor-web as an HTML page available through a button (H) on a slide. The assignment is also printed as a part of the printed version of the tutorial. Naturally, the HTML page is derived from either ASCII text or a LaTeX file, as elsewhere.

This presentation needs to be improved, at least to the extent of making access easier.

3. Slide presentation methods

Slides on the tutor-web can be viewed using different formats, depending on the presentation medium and user preference. Several presentation formats have been developed and it is a simple matter to add other options. Each presentation form is processed automatically from the basic definitions.

3.1. Slides as HTML

Access to the tutor-web is through the Internet and therefore HTML is the primary presentation format.

A standard, verified HTML format is used on all pages of the tutor-web, whether they are individual static pages, pages generated by programs using batch processing (most of the material) or interactively generated pages (questions or grades). In addition, the pages are tested with a large number of different browsers.

3.2. Slides as PDF

During lectures it is possible to use high-quality presentations in PDF format. The method used in the first version of the tutor-web uses the “Prosper” package within LaTeX. The Prosper package enforces a high-quality style of presentation. This includes the use of large fonts and a limited amount of content on each slide.

This has the drawback that a slide with a lot of content in HTML format will not fit into the PDF format. In order to use the PDF presentation form anyway a lecturer can insert special comments which enforce page-splitting only in PDF-mode.

\footnote{http://prosper.sourceforge.net/}
A better future mode of processing would be to set up a special set of \LaTeX\ commands, replacing Prosper, to be used solely for generating the PDF versions of tutor-web slides automatically. This would give the high-quality of PDF while enforcing the same page layout as in the HTML version.

3.3. Slides as figures in printed documents

The final option currently available for presenting a slide is as a part of a printed document. In this case, the main content of the slide becomes a (small) figure in the document as will the slide’s main graphic.

A useful addition would be to allow more than one printing mechanism, for example a set of many slides-only per page and/or an “overview” format which would summarize the entire tutorial in one or two pages.

4. Using the system

4.1. Student use

For a student wishing to access the contents of a tutorial the simplest method is to access it using traditional Web-pages in HTML format through a browser. The student will first open the home-page of the tutor-web, i.e., “http://www.tutor-web.net” and select a “department”, e.g., “Mathematics Department”. Within the Mathematics Dept. several tutorials can be accessed, e.g., “Elementary linear algebra”, which is the tutorial “Math 121.0”.

At this stage, within the tutorial, the individual lectures of the tutorial are seen and accessible. A specified lecture can now be selected, e.g., “Matrices and systems of linear equations”, at which stage an overview of all the slides within this lecture are shown. When one of these slides is selected it appears along with all buttons to various links.

The student can now flip through the slides or look at underlying detail and examples, if the corresponding buttons are green. The slide also has the option of obtaining questions from the lecture (see Fig. 4).

4.2. Use in lectures

When slides are presented during lectures they can be used in one of two formats. It is possible to use HTML but this is more suited for general Web access and does not at present have the quality usually required for on-line presentations. Although the button locations are fixed at the top of the page rather than floating at the bottom, other issues include the difficulty of fixing font sizes applicable to both on-screen and projector-based viewing. Similarly, the presentation of mathematical content is still in its infancy on Web pages. Later work will include the use of DHTML to take into account at least the variable behavior of different browsers.

For in-class presentations a commonly preferred approach is to use PDF which is high-quality and machine-independent, at least until MathML becomes generally available (see Appendix A).
In the first version of the system the PDF layout of slides is only in one fixed format. The layout follows common presentation practices of large fonts and a minimal amount of text on each page. It would be a simple addition to allow more presentation formats to accommodate different needs for different courses.

5. Evaluation

Quiz questions are used for evaluating students. Since the questions are on-line, several issues are raised, which differ from those considered when giving in-class quizzes. A detailed investigation of the effects of using computerized quiz questions (such as given by e.g., Jodoin (2003)) is outside the scope of this paper, but an analysis of the relationship between quiz results and a final examination is given in a subsequent section.

5.1. Quiz design principles

The implementation of quizzes and individual questions is always somewhat dependent on design criteria which need to be laid out beforehand. The tutor-web is no exception to this. Some of the quiz and grading design issues are reflected in the design of the system itself, but of course other aspects are instructor-definable through the design of individual questions.

The system assumes that students are allowed to answer quiz questions at their own leisure, merely by pressing a button (Q). Thus, there is no built-in penalty for late answers, though of course an instructor may decide to specify a date at which a grade is output from the system.

Within the system questions are grouped together within a lecture. The question given to a student is selected randomly, but not with uniform probability. Within a lecture all questions have the same probability but some (lower) probability is also given to obtaining a question from previous lectures within a tutorial. This probability is reduced for older lectures but does not reach zero.
Answers to questions are of course randomly permuted. In addition, questions can include random numbers but it has been found that the use of random numbers is highly error–prone and therefore not recommended. Rather, fixed numbers should normally be used in questions, but there should be many variants. An exception to this is the generation of data sets for statistical analysis (see below).

Questions can include formulas, which is essential to mathematical instruction. At present the answers cannot include formulas. Therefore a mathematical question needs to be phrased so that answers are simple selections.

In live applications the students have been encouraged to request answers repeatedly until a decent grade is obtained. Thus, students who do not know the material can test their knowledge and, upon finding it wanting, go back to the on-line text or textbooks to come back to the quiz at a later date, up to the date of an examination in the physical world. This is considered a fundamental design principle.

In applications to date attempts have been made to make each question reflect some level of understanding rather than brute force memorization. Naturally the traditional approach of learning by rote no longer applies when a student is free to take an exam at home or in an Internet cafe. The questions implemented so far can be classified into a few categories, but they are all multiple-choice and in fact so far only three possible replies are designed for each question.

A very useful class of questions can be termed the “difficult task grouping style”. This method relies on a framework where the student is asked to undertake a single complex task, e.g., to do a statistical analysis such as a multiple regression analysis. Several questions within the lecture then state this as the framework and start out using the same wording. The final sentence in each question is different, however, querying about slightly different aspects of the problem at hand. This building-block approach is useful for the instructor since it allows very easy generation of a large number of (slightly) different questions within a lecture. This approach can of course be used to ask for detail which would otherwise be much harder if the data was only used once, but also this method implies that it is possible to give a much larger problem than would otherwise be justified within a single question. The student only has to work through the whole exercise once to obtain answers to a whole sequence of questions – assuming that the understanding is in place. This approach is particularly useful in statistics or any quantitative science but certainly also applies to many other fields such as introductory linear algebra or numerical mathematics (see Fig. 6).

The simplest group of questions is, however, the one-question, one-correct-answer type. The danger with such questions is that students may simply learn the answers with no understanding. A slight twist on these is to develop a collection of (similar) statements, some correct and some false. The multiple-choice query can then refer to one correct and two false statements. A setup like this one is easy to extend into a collection of several slightly different questions, making it harder to either guess or memorize answers.

The current system allows the use of a statistical package, R, 12 when a question is generated. This idea, which is borrowed from EduML 13 (see Appendix A), is extremely useful as it allows

---

12 http://www.r-project.org/.
the generation of structured, similar but not identical data sets and graphs for students to analyze or interpret. Thus, this method has particular potential for teaching statistics where a student can be asked to answer fairly simple questions, requiring understanding, based on analyzing randomly generated data sets.

5.2. Technical aspects

When a student accesses the questions for the first time a userid in the form of an e-mail address is requested along with a password. Subsequent uses of the system require the same userid and password.

The access is quite simple in that the student merely presses the Q button to request a question, selects an answer (currently a, b or c) and submits this answer. This results in a response in an other window, giving first an indication of whether this answer is correct and if not, what the correct answer would have been. Further, the answer window gives the number of questions requested and correctly answered from the current lecture.

A press of the G button is used to get an overview of all grades from all lectures, arranged by tutorial. If the instructor has formally registered the student for a collection of tutorials, then that set is included in the grade overview, but otherwise only tutorials for which the student has requested questions is used in the computation of a grade point average (GPA).

5.3. Return rates

An option of the tutor-web is to report a return rate. The original reason for implementing return rates rather than just grades is that there is a mandatory return rate on homework at the university where the tutor-web was first tested and a specified return rate on homework is traditionally a requirement for taking a final exam. However, it has been found that all students observe the mandatory return rates and they are therefore only an encouragement to work rather than an impediment to taking a final exam. Since students can request quiz questions at any time there really is no reason to abandon the use of return rate requirements and they do appear to have considerable benefits. A mandatory return rate is not enforced as such within the tutor-web, but merely reported so that the instructor can define such rates if desired.

In practice the return rates have been computed based on the number of correct answers to questions within each lecture. If a lecturer decides that a return of 8 correct answers per lecture constitutes a completed assignment, then a minimal return rate can be set to, e.g., 75% or 6 correct answers per lecture. With four lectures per week this implies that a student should answer 32 questions correctly per week. Depending on the structure and design of individual quiz questions, this can be made to range from a very light to an arbitrarily heavy work load for the students, but of course the requirement of a high return rate alone cannot enforce a work load exceeding that required to obtain 32 correct answers per week by pure guesswork in this example, which is on average obtained by guessing 96 answers per week, when there are three possible answers per question. This leads on to the need for formal grading schemes in addition to return rates.
5.4. Grading

As with other requirements, the implementation of a grading system is highly dependent on design principles made at the outset. The whole point of the grading system within the tutor-web is to allow the student to learn and thus improve the grade. It is not considered an issue whether this is done at the beginning or near the end of a semester. Given this basis, a student should not be penalized for initially testing the system with no knowledge. Rather, the grade should be based only on recent answers.

Grading has so far always been 0 for an incorrect answer and 1 for a correct one. When evaluating grades this can in principle be extended to other weightings.

It is very easy to store the 0/1-grades for each of the most recent questions per lecture and use these as a basis for a grade. This approach has been tested in one live course on the Web. In that first implementation the 4 most recent answers were used within each lecture. As seen in the analysis below, this does not seem to be enough, and in fact some students clearly keep on guessing until full marks are obtained from 4 questions in a row. In more recent versions the grade is therefore based on the 8 most recent answers.

Within the tutor-web a student can request a list of grades with the press of a button (G), thus allowing each individual to monitor personal progress. In addition, an instructor can monitor progress and rudimentary scripts are available to send automatically individualized results using electronic mail. This is considered important to encourage less-motivated individuals to using the system.

5.5. Cheating

There is no real solution to the issue of one student assisting another in answering questions. However, the present system makes this much, much more difficult than copying regular homework, since the questions are generated on an individual basis.

6. Entering material

Material can be entered into the tutor-web from several sources, but efforts have not been made to make these particularly user-friendly. The entire system can be (and has been) ported to different computers, but the following assumes that material is to be set up on the existing (semi-official) Web site. At present any content manipulation involves a Linux-literate individual at some stage.

6.1. New material from textbook in electronic format

The best way to enter a large amount of material into the tutor-web is from a document which already exists in ASCII text or LaTeX. In this case, it is very simple to make prototype slides as well as the additional detail and examples. It is also quite simple to convert from common word processing systems through LaTeX.
Tools have been developed to split up LaTeX files into explanatory and detail files based solely on prescribing where the slides split. These tools automatically generate all required subdirectories and files.

6.2. New material from scratch

Any new material should first go through some minimal design phase, if only to define what constitutes a tutorial, lectures and slides. Having set this up the task becomes one of first entering lecture and slide titles for an initial set of slides. This outline serves as a first basis around which everything else revolves. The precise layout is not important as it can fairly easily be changed, slides can be moved about, lectures can be split up and so on.

For new material it is usually simplest to start with filling in a set of prototype slides. Each slide can be a simple set of sentences, bullet text using an asterisk at the beginning of each line or LaTeX code, depending on what is needed. Graphics can subsequently be added as can general complexity, but the prototype slides form a second basis, which can in fact be used directly as a simple set of memos for use within and outside the classroom. For a lecturer who has given a course before, such a collection of place holders require considerably less effort to generate than the preparation for lecturing material for the first time.

New slides are given names within each lecture, usually of the form sl10, sl20, sl30 and so on, where the numbering is used to order the slides. Leaving spaces in the numbering scheme helps when adding new slides. Since the tutor-web currently exists as files in a Linux file system, these slides are implemented as directories which can be renamed like any other. Each directory contains the entire set of material associated with a given slide. The title slides are simple a text line in a file within each slide directory. These are the first things that need to be put in order.

Having obtained a set of slides the obvious next step is to copy the same material into the detail sections, again as place holders. In fact, if the initial slide text was put into the explanatory part of each slide only, then there will be no duplication when the content is printed. On the other hand it is likely that the desired end result will be to use the main text part of the slide and extensions in the detail sections so this will be only better for a short time and it will be a better long-term solution to have the material duplicated also in the printouts, until slides have been cleaned into final form and details have become true details containing real text rather than the simple bullets.

6.3. Adding individual slides

A new slide is added by giving it a name in the sequence, i.e., creating the directory. When the lecture is processed the slide is created with a default title and name but otherwise blank.

At this stage all text can be entered using a text editor on the appropriate files, possibly adding files which are not initialized by default.

6.4. Adding new lectures

Lectures are often created from earlier content. Since the lecture is basically a directory containing slides, this is easiest to do using simple recursive copy commands from an old lecture. Even
when the content is completely new it is still easiest to take an old (short) lecture as a basis, just to have the complete structure to start with, but to replace the content. Having set up a structure blank slides are added, their titles are defined. The whole lecture can now be processed and a printout will indicate whether the structure is correct, before moving on.

In terms of future work, very useful addition would be a collection of specialized LaTeX commands which could be used to generate entire lectures. This would provide a useful method of debugging the whole lecture before installing them on the tutor-web, potentially enabling an instructor with no Linux experience but only LaTeX knowledge full access to place content on the current tutor-web.

6.5. Processing

A special script is run to process an entire lecture so that all slides are set up for presentation, links are created and so forth. This could trivially be automated if the system were to be put into common use.

7. Special aspects

7.1. Security

Few security issues have been addressed within the system, apart from password protection of grades.

Coding of passwords needs to be improved but the only personal information stored in the system is grades for quizzes and these tend to form only a small part of the final grade in a real school.

Secure connections have not been implemented but will need to be considered if the system is to be put to more common use.

Cookies are used to store and transfer information about the user but these can be bypassed via command-line options, implemented for debugging purposes (and very useful). The security of the latter in particular needs to be investigated, but they are easy to remove if needed.

Students have not been observed to access sensitive information but this will be subjected to more stringent tests as CIS students will use the tutor-web in a live class in fall semester, 2003.

7.2. International aspects

The platform or system components of the tutor-web are entirely in English. There is no plan to internationalize the platform further. Thus a question is labelled “Question”, i.e., in English, the front pages are in English and so on. Further, though not of concern to users, comments and variable names in programs are in English, so as to allow joint efforts in development.

The educational material, i.e., the content itself, is independent of the platform language. Thus tutorials are available both in English and Icelandic at the time of writing. This is implemented simply by completely separating content and system components, thus enabling any language to be used for content, as illustrated in Fig. 5. Any language with a basis in the ISO 8859-1 8-bit
character sets can certainly be used, but provisions have not been made for 32-bit character sets. The only known problem with this approach is that, e.g., Japanese and Chinese are difficult to use with LaTeX equations.

The tutor-web is a registered member of Schoolforge, an international consortium on open methods and cooperation within the educational sector. One Schoolforge project has dealt with the coordination of “cyberschools” such as the tutor-web and it is hoped that at some stage such schools can cooperate so that, e.g., a student user of the tutor-web will see references and have access to related material in other systems in a simple manner.

Fig. 5. A bullet-point slide with equations in PDF presentation format, illustrating a non-English slide.

Fig. 6. A quiz question illustrating a design where a simple setup can be used to generate a large number of similar but not identical questions.

character sets can certainly be used, but provisions have not been made for 32-bit character sets. The only known problem with this approach is that, e.g., Japanese and Chinese are difficult to use with LaTeX equations.

The tutor-web is a registered member of Schoolforge, an international consortium on open methods and cooperation within the educational sector. One Schoolforge project has dealt with the coordination of “cyberschools” such as the tutor-web and it is hoped that at some stage such schools can cooperate so that, e.g., a student user of the tutor-web will see references and have access to related material in other systems in a simple manner.

14 http://www.schoolforge.net.
7.3. Copyright

The issue of copyright has been low-priority as the number of authors/contributors of content has been low in the first version of the tutor-web. The view so far has been that the material should be as accessible as possible. This concept does have to be formalized when the number of contributors increases, and some notion of authorship tends to be desirable in most systems.

The whole point of the tutor-web is that it is open. The logic for this has been discussed by various authors, probably more on the Web than anywhere else. Thus it is accessible to all, at no cost to the user, in the same manner as free or open software. Users will be allowed to download material to be used freely. On the other hand there will be a condition on always citing the author of a given text and that derived content remain open.

This methodology is well known with GNU software (through the GNU Public License, GPL) there are variations on open-oriented licenses for software (e.g., Artistic License, and a choice needs to be made between such approaches for text, e.g., GNU Free Documentation License and Open Content.

When adding material to the tutor-web it must therefore be kept in mind that this web is and will be open and freely accessible. Thus, the new material must also be open. In particular, material from text books can usually not be used without prior consent and authors must be aware that their material is made generally available.

8. Results from live testing

The system has been tested live in courses in the fields of mathematics, statistics and quantitative biology. It is of interest to evaluate various aspects of the system in terms of user (student) satisfaction as well as its contribution to student learning.

---

Table 1

Summary of answers to each individual questions on the tutor-web in in-class survey

<table>
<thead>
<tr>
<th>Question</th>
<th>Agree</th>
<th>Disagree</th>
<th>Undecided</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  I answered Web-questions today or yesterday</td>
<td>4</td>
<td>25</td>
<td>–</td>
</tr>
<tr>
<td>2  I answered questions this week or last week</td>
<td>22</td>
<td>7</td>
<td>–</td>
</tr>
<tr>
<td>3  I feel that I learn by using the tutor-web</td>
<td>14</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>4  It is easier to answer the last question than the first from each lecture</td>
<td>14</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>5  I prefer the tutor-web to regular homework</td>
<td>12</td>
<td>12</td>
<td>5</td>
</tr>
</tbody>
</table>

---


17 http://www.opensource.org/licenses/gpl-license.php.


19 http://www.gnu.org/licenses/fdl.txt.

A questionnaire was distributed and collected in a class on linear algebra and statistics. This course used all aspects of the tutor-web, i.e., in-class transparencies, printouts and on-line quizzes with mandatory returns and grades counting towards the course grade. It should be noted that this particular course is a mandatory course for students of food science, pharmaceutics and biochemistry, so this is a mathematics-oriented course for non-mathematics majors.

The following information was collected from the 29 students who participated in the survey. Clearly the students did not answer questions daily but most student answer questions at least every other week (Table 1). This is of course natural given the mandatory 75% of full returns which constituted 8 correct answers for each lecture or 32 correct answers per week.

Students’ views on the system differ widely. Half of the group preferred the tutor-web to regular homework. This should be viewed in the light of the fact that when regular (paper-based) homework has been mandatory in this course, a considerable fraction of the student have blindly copied homework including typographical errors to the level of nonsense. Similarly about half the group feels that they gain from using the tutor-web and this same conclusion is obtained both from the direct question (3) and the indirect one (4). One of these results (4) can be verified by looking at actual answers to questions, i.e., comparing the responses to the first and last question from each lecture, as will be done below.

The uneven distribution of answers to the first two questions make it difficult to perform cross-comparisons with other factors. It would for example be of interest to investigate whether those who answer questions regularly feel that their responses improve, but this is not really feasible given the sample sizes in each of the smaller response groups in questions 1 and 2.

On the other hand it is possible to cross-compare the answers to the other questions. In particular, comparing questions 3 and 5 (Table 2) demonstrates a strong relationship so that those who feel they learn from the tutor-web tend to prefer it to homework and vice-versa (notably this should not be taken to reflect cause and effect as there could be an underlying technophobic factor.

### Table 1
Analysis of responses to questionnaire: Comparison of individual preferences to opinion on whether the tutor-web has a positive influence on learning

<table>
<thead>
<tr>
<th></th>
<th>Prefer regular homework</th>
<th>Prefer tutor-web</th>
<th>Undecided</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learn from the tutor-web</td>
<td>2</td>
<td>9</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>Do not learn from the tutor-web</td>
<td>9</td>
<td>2</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>Undecided</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>12</td>
<td>12</td>
<td>5</td>
<td>29</td>
</tr>
</tbody>
</table>

### Table 2
Analysis of responses to questionnaire: Comparison of individual preferences to opinion on whether the tutor-web has a positive influence on learning

<table>
<thead>
<tr>
<th></th>
<th>Better last</th>
<th>No improvement</th>
<th>Undecided</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learn from the tutor-web</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>Do not learn</td>
<td>8</td>
<td>4</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>Undecided</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>14</td>
<td>11</td>
<td>4</td>
<td>29</td>
</tr>
</tbody>
</table>

### Table 3
Analysis of responses to questionnaire: Comparing students’ responses to whether they generally learn and whether their answers improve

<table>
<thead>
<tr>
<th></th>
<th>Better last</th>
<th>No improvement</th>
<th>Undecided</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learn from the tutor-web</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>Do not learn</td>
<td>8</td>
<td>4</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>Undecided</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>14</td>
<td>11</td>
<td>4</td>
<td>29</td>
</tr>
</tbody>
</table>
involved). This difference in opinions is statistically significant when viewing those who provide definitive responses.

This is not surprising, though of course the result does not indicate a cause and effect relationship. It is known (from personal interviews) that some students have serious problems when interacting with computers in any manners. These same students will have problems when encountering a computerized system and will prefer regular paper-based homework. They will also struggle every step of the way, typically clicking or double-clicking on wrong buttons and so forth and therefore commonly feel that they are only struggling and not learning.

Questions 3 and 4 are of course basically about the same thing and at first sight the responses appear very similar, but one of them refers to an opinion and the other requests an estimate, albeit informal. When cross-comparing the results (Table 3), however, it is seen that there appears to be no relationship between these two types of replies.

It is not at all clear how this type of result should be interpreted. On the other hand it is possible to evaluate this particular issue formally, based on submitted answers to Web-questions. A snapshot was taken of 1539 available replies by pulling out the first and last answer from each student to each lecture. Each individual answer can be 0 or 1 and on average one would obtain a grade of 1/3 by pure guesswork. From the 1539 answers students obtained on average 0.53 (s.e. 0.013) for the first question and 0.77 (s.e. 0.018) for the last question. Thus there is clear evidence that students enhance their ability to answer correctly as they move through the sequence of questions. This analysis used all possible replies. When taking only the lectures and students where at least 8 correct responses were obtained the average grade for the first was 0.55 but for the last one the average was 0.84 (n = 590, s.e. = 0.02 and 0.03, respectively).

On this topic of improving performance it must be recalled that when an answer is submitted (whether correct or incorrect), the correct answer appears as a part of the reply from the system. This is important since it affects the way the results can be interpreted. Notably, it is quite possible for a student to simple note what a response should have been and therefore “get it right” if, due to the random selection, the question reappears. Since there is a finite number of questions, and sometimes rather few, this is quite possible. On the other hand it is also possible that a student will simply realize what was wrong and therefore answer correctly later. The only way to distinguish between these two scenarios is to have a large database of questions, of which many are similar but not identical. Although this exists for some lectures, such a detailed analysis has not been conducted to date.

The development of a grade can also be viewed as a function of the sequence number of questions within each lecture, averaged across lectures and students. Taking only those cases when a student has completed 8 correct questions results in Table 4.

The answers on the tutor-web can of course also be compared to results on a final exam. A detailed examination would be outside the scope of this paper, but a simple analysis can be undertaken. The course described above had a total of three homework assignments in addition to the tutor-web. The assignments were intended to teach some basic statistical analyses in addition to getting the students accustomed to the use of a statistical package. Of these three homework assignments, the first is quite trivial and basically consists of printing out an assignment undertaken during computer class. The second assignment, although requiring considerable work, resulted in grades which were too tight to discriminate. The interest is therefore in using the tutor-web results and the third assignment to predict the exam grade. In addition to using the tutor-web
grade alone, one should consider whether a tendency to guess the answer on the Web can be observed. This can be measured through the total number of attempts made.

Given this, multiple linear regression was used to predict the course grade from the third assignment, the tutor-web grade and the number of answers submitted on the tutor-web. The results are given in Table 5.

Eliminating an outlier in the data (a single student with a very low tutor-web grade) does not affect the overall results as seen in Table 6.

Table 5
Summary of regression analysis predicting course grade from tutor-web grade (twgrade), number of trials on tutor-web (twnumtries) and final homework assignment (H3)

| Coefficients | Estimate | Std. Error | t value | Pr(>|t|) |
|--------------|----------|------------|---------|----------|
| (Intercept)  | -112.4552| 47.9280    | -2.346  | 0.023323*|
| twgrade      | 18.0986  | 4.8022     | 3.769   | 0.000466***|
| twnumtries   | -2.2812  | 0.9632     | -2.368  | 0.022123*|
| H3           | 3.8857   | 1.4755     | 2.633   | 0.011475*|

Significant codes: 0 *** , 0.001 ***, 0.01 *, 0.05 ,, 0.1 ++ , 1 .
Residual standard error, 20.11 on 46 degrees of freedom; multiple $R^2$, 0.4042; adjusted $R^2$, 0.3654; $F$-statistic, 10.4 on 3 and 46 DF and $p$-value, 2.416e-05.

Table 6
Summary of regression analysis predicting course grade from tutor-web grade (twgrade), number of trials on tutor-web (twnumtries) and final homework assignment (H3), after deleting outlier

| Coefficients | Estimate std. | Error | t value | Pr(>|t|) |
|--------------|---------------|-------|---------|----------|
| (Intercept)  | -182.9958     | 64.6281| -2.832  | 0.006904**|
| twgrade      | 24.7972       | 6.3194 | 3.924   | 0.000295***|
| twnumtries   | -1.9932       | 0.9644 | -2.067  | 0.044543*|
| H3           | 4.0457        | 1.4548 | 2.781   | 0.007888**|

Significant codes: 0 *** , 0.001 ***, 0.01 **, 0.05 *, 0.1 ++ , 1 .
Residual standard error, 19.78 on 45 degrees of freedom; multiple $R^2$, 0.4294; adjusted $R^2$, 0.3914; $F$-statistic, 11.29 on 3 and 45 DF and $p$-value, 1.205e-05.
The number of tries appears indeed to be an indicator of a student guessing at the answer, as this variable has a negative coefficient. Similarly, the coefficient of the two other grades is positive as expected. All variables are significant in explaining the variability in the exam grade.

As before, this cannot be taken as evidence of a causal relationship, since the conscientious students may simply be the ones who tend not to guess, but put work into each question on the Web, requiring fewer attempts and also perform better on the exam.

9. Future work

The concept of a “course” is not currently defined in the tutor-web. There is a need to define this concept as a collection of tutorials, already available. The reason for this is that a grade needs to be given for entire courses and not just for individual lectures, tutorials and the tutor-web as a whole.

As more tutorials become available in English it is hoped that the system as a whole will be of more general use, thus attracting more material and international cooperation in the exchange of teaching material. At the moment this can be done either through an individual instructor setting up the entire system locally, or through a partnership where material is submitted through e-mail and student access to a central machine.

To enhance this, easier and automatic methods are needed to submit material. This may be done using Zope or other systems which allow submission of material, but it will require some programming to ensure the consistency of the system. If LaTeX environments for the entire slide contents are developed then that would provide a comprehensive and useful framework for testing entire lectures locally using only LaTeX, before submitting to a server.

Consideration needs to be given to the possibility of making the tutor-web financially self-sufficient through advertisements, links to other Web-sites and similar methods.

Copyright issues need to be resolved before the system is set up for a large number of authors. Alternative presentation methods should also be considered. For example, it would be possible to present to the student working at home a more comprehensive view giving directly most links on the slide, with a smaller typeface. Thus, when the student flips through such slides the simultaneous view is into the original classroom slide, the associated examples and detailed expansions.

As Web-standards develop, new presentation formats for mathematics will become commonplace (see Appendix A) and will be adopted on the tutor-web.

The automatic collection of answers to quiz collections allows several future developments. Notably, questions can be labelled according to their level of difficulty when they are presented to the student and such summary information is useful for the instructor when deciding whether to add easier or more difficult new questions. More importantly, poor results from a sequence of a few questions could fairly easily be used dynamically to indicate to a student that the current level of knowledge is inadequate and a more elementary tutorial should be completed (or repeated) before this one is addressed again. This could go a long way towards adding an adaptive dimension to Web-based learning, as discussed, e.g., by Papanikolaou, Grigoriadou, Magoulas, & Kornilakis (2002).
Open Web-based educational systems which are in one way or another related should probably be interlinked in order to maximize their utility, whether by students or instructors. This includes open encyclopedias (both the reviewed GNUPedia\(^\text{21}\) and the more liberal Wikipedia\(^\text{22}\)). Such links could be included through a general “alternate information” button, but should open a separate, small, window so as to reduce the amount of distraction from the basic content of the tutorial.

10. Conclusions

The main aspects of the tutor-web system are the connections between electronic slides for in-class use, detailed material, examples and dynamic multiple-choice testing along with printing content in textbook style. Qualitatively, this approach has been found useful in live classroom work.

Quantitatively, the methodology has been evaluated in the classroom, both with survey-based methods and also through regression analysis using the tutor-web results as predictors of subsequent exam grades.

It is found that performance on the tutor-web is indeed linked to performance on a final exam, but this is of course not a declaration of cause and effect. A clear positive trend in performance on quizzes is seen over time, however. In years of formal paper-based homework, a large fraction of the student body has been found to copy others’ homework and this appears to be largely abolished. The survey further indicates that about half of the test-class prefers the on-line quiz form to paper-based.

Although there are clearly several improvements that can be made, this system has been found to be effective both in terms of making lectures more direct and in terms of encouraging individual students to work on their own homework.

This system could be used a basis for cooperation between schools or lecturers who would like to exchange teaching materials and methods in an open manner. Currently available software could be used, at least initially, but the software could also be set up in other locations. The next major step forward for this concept would be to include a number of lecturers, each possibly providing one or more lectures or tutorials. Such a bazaar-style approach\(^\text{23}\) approach would quickly provide each lecturer access to much more material than any individual would input.

Appendix A. Technical details: Standards and environments

A.1. \LaTeX{} and text

A slide’s textual content can be written in \LaTeX{} or as plain ASCII text. Simple transparencies, detailed additions and examples are commonly set up as plain text for quick prototyping or even for final form. Elementary additions are provided within the ASCII style, for bullet points and tables.

\(^{23}\) http://www.firstmonday.dk/issues/issue3_3/firstmonday/.
More complex slides will contain equations and these must be set up using LaTeX, as this is the closest thing to a standard in mathematical typesetting. Given that LaTeX is an option within the system, it is of course possible to make the slides as complicated as desired. Normally, however, each slide should be limited to a single concept and although graphics are a possibility within LaTeX, for reasons given elsewhere, graphics are normally included separately, outside the textual content.

Other textual content can be handled in the same manner, i.e., be either in LaTeX or ASCII format with simple additions.

When MathML will become generally available in browsers this will considerably improve the presentation possibilities of mathematics within regular Web pages, as opposed to the use of images as is needed at present. LaTeX, being the longest-lasting format for mathematics, will continue to be the basic storage format, but MathML will become the target code generated automatically by filters. MathML is not designed for end-users to write, but the general availability of MathML in browsers and filters will greatly enhance the presentation of mathematical equations on Web pages.

A.2. Conversions from other formats

Most common word processing formats can be converted manually for the system. Notably, documents with mathematical content can be converted from such systems, sometimes by using commercial add-ons which can generate LaTeX output. Similarly, many word processing programs can perform various save-as functions, such as “text” or “rtf”. In some cases this will automatically generate separate graphics files containing the figures in the document. These can then be converted to the PNG format (Roelofs, 1999) as described below.

Unfortunately the most common slide presentation programs have no options for saving the contents in any structured manner and hence it is very difficult indeed to convert from them to any other format. In some cases it has proven possible to use OpenOffice to read proprietary slide formats for transferring into the tutor-web. In these cases each slide needs to be handled individually and each item on each slide is cut and pasted into appropriate tutor-web files, normally as text. This cannot be done automatically and therefore it is unlikely that such programs will ever be used as a basis for maintaining a system such as the tutor-web. After conversion for the tutor-web it is therefore natural to maintain the content in place.

A.3. Image manipulation

When hand-drawing a figure on a computer, the program of choice is often xfig (for vector drawing) but for general image manipulation gimp is much more flexible. For scatter plots or line diagrams either R or gnuplot can be used. This can in principle be automated so that when

---

a user orders a picture it gets drawn in real time and in the format required at that instance. This has not been the selected methodology in this first version of the tutor-web. Rather, formats have been fixed and processing between formats thus simpler.

Conversions between most formats is not difficult, with the sole exception of WMF, 29 which is largely unknown except within commercial programs from a single supplier.

For automatic conversions the PPM family is mostly used.

A.4. Images: PNG for presentation

Slides can of course contain pictures or images. These are stored in PNG 30 format, which common browsers can handle, and (unlike many others) is not licensing-dependent. Further, this is a loss-less compression format (unlike other common formats, e.g., for photographs, which for some types of figures can result in considerable loss of information). The PNG format is therefore chosen as the basic image format throughout the tutor-web.

Naturally, some figures are generated as hand-drawings or line diagrams and are therefore basically vector-diagrams. Such figures would be better stored as either postscript 31 or as a set of commands for generating the pictures (e.g., as xfig files or R commands). This may be implemented in a later version of the tutor-web, by storing such basis-files and subsequently generating either PNG or postscript files as the case may be. Each format can then be generated using an optimal quality for each presentation.

A.5. Images: Postscript for printing

In order to print pictures stores in PNG format, these are converted to postscript. The problem with this approach is of course that some figures are initially generated in other than PNG format and sometimes even initially in postscript format. Thus a considerable loss of quality may result in the conversion process.

A.6. DHTML

Web pages are mostly HTML-based using verified HTML. 32 Some functions are implemented using JavaScript (Flanagan, 2003).

In addition to formal verification, the pages are further tested using several common browsers (Netscape, 33 Mozilla, 34 Konqueror, 35 Opera 36 and others). Browser incompatibilities are a considerable problem in developing a system as complex as the tutor-web.

---

29 Windows metafile format.
30 Portable network graphics.
31 Adobe postscript, see, e.g., http://www.adobe.com.
32 http://validator.w3.org.
33 http://home.netscape.com/.
A.7. Data bases for answers and grades

In the first version of the tutor-web a simple CSV-based format is mainly used to store data but data base commands in perl\(^\text{37}\) are used for extraction to ensure that at a later stage a true data base system such as PostgreSQL\(^\text{38}\) can be used.

A high-speed hash-based method is used, however, to look up and store final grades for each student in each lecture.

A.8. Linux

The environment for development is Linux. The reason for this choice is that the system should

- work on as many types of computers as possible
- not depend on any commercial components and producers
- be a powerful web server

As a result of this choice of environment, the server can be set up in any location, on any hardware which runs Linux. This includes cheap PC hardware as well as powerful servers. There is no commercial software anywhere in the system and most of it comes with the Linux operating system.

A.9. Programming: Perl and shell scripts

Most programs in the system are either perl programs or shell scripts. These are first used to set up the system, i.e., generate web pages, slides and printed documents as batch-jobs.

A perl program is used as a CGI script to deliver questions and process answers, such as registering grades in a database. Some perl additions are used, including those relevant to manipulating data bases.

A.10. Web server considerations

The Apache web server on a dual-processor pentium-based Linux system is used to serve http://www.tutor-web.net. Apache and Linux are chosen both for reasons of security and speed.

It would be possible to obtain further speedups through the use of, e.g., mod-perl, but speed is not a limitation at present.

A.11. Storage formats

The entire contents of the slide, examples, detail and quiz questions are stored as text files in a Linux file system. The formal definitions of the formats of all files would make it possible to insert the whole system into a well-defined database. This has been a part of the plan for the system from the start, but installation of the database has been postponed as it has not been essential to date.

\(^{38}\) http://www.postgresql.org.
Each component of each slide is therefore stored in a separate text file (title in one file, main text in one, explanatory text is separate etc) and each figure is stored in a file of its own.

The system is “fired up” by running (in batch) a sequence of commands which process all slides in a lecture to generate automatically all Web-files of interest, e.g., HTML or PDF files. When a user subsequently requests the pages they are all ready and ship from the server without any additional processing. With this setup the system is very light. On the other hand it is difficult to provide a student with pertinent per-user information if all the Web-pages are generic. It would for example be useful to have the ability to indicate to a student that a quiz remains to be taken. This is currently only done during question- or grade-requests.

A.12. Other possibilities

Some standards are available for teaching material. These include EduML, which is an XML schema and has been used for purposes similar to those presented here. Although EduML has several virtues, the principles are somewhat different. In particular EduML is very loose concerning organization of content but this is very strict on the tutor-web. In addition the tutor-web is set up to solve more issues than appear to be picked up in EduML in its present state, but on the other hand EduML gives much greater flexibility in the design of questions.

References


39 http://eduml.sourceforge.net/