

# Interactive Web-based Teaching for Computing in an Engineering Degree\*

GEOFFREY G. ROY and PETER L. LEE

School of Engineering, Murdoch University, South Street, Murdoch WA 6150, Australia.

E-mail: geoff@eng.murdoch.edu.au

*The effective implementation of web-based teaching and delivery of education and training materials depend on a number of key design and implementation parameters. Central to these is the ability to capture and hold the student's attention. With this done it is then possible to develop a learning environment which is at least as effective, if not more so, than traditional chalk-and-talk approaches. This paper presents some of the developments we have initiated in our own Engineering degree programme. Our objectives are twofold: (1) to create a more cost effective way to deliver complex and rapidly changing curriculum content; (2) to enhance the learning experience for the student and to encourage the development of self motivated learning skills. Our efforts have thus been focused on the development of interactive activities which can be built into web pages. These are designed to enhance the learning experience and to maintain and encourage interest in the subject matter as the student progresses through the course. Web-based courses will also provide us with the ability to offer our courses off campus in external mode.*

## INTRODUCTION

THE RECENT developments in web-based delivery tools provide a range of challenges in engineering education. We are now forced to re-think the traditional chalk-and-talk delivery methods and consider a wider range of delivery options. We must, however, also consider very carefully the pedagogical issues surrounding the teaching/learning processes.

While we may feel comfortable with traditional approaches the new technologies provide us with the tools to challenge these positions, and open up the teaching/learning questions for some rethinking.

There is a wide range of motivations for developing new teaching processes, and perhaps the key ones are:

- To provide for a more student-centred learning environment where students can access teaching materials and progress through their courses at their own pace, and perhaps in their own space (i.e. at home, or work place).
- To improve the average quality of teaching and learning by using better quality resources to support staff who may not be able to provide a uniformly high quality in their teaching delivery.

There are a myriad of other issues which contribute to, or distract from, improving the quality of the teaching/learning experience. Some of the more important ones include:

- the effectiveness of the teaching/learning process;

- the levels of enjoyment and enthusiasm exhibited by students;
- the ability of students to manage their time in what might be a new learning environment;
- the costs of preparation and delivery.

Within the School of Engineering at Murdoch University a commitment has been made to work towards the development of web-based teaching across a wide range of units. We are fortunate in the fact that we are currently implementing a completely new engineering degree, so we have the opportunity to do it without having to carry the baggage of well established units. We naturally began with first year units, with other units coming on stream over subsequent years.

The on-line courses have been developed around a number of general principles which we believe make it a viable approach to teaching and, from our own observations, popular with the students. Some of the key elements in the design of our on-line units are:

- the provision of a navigation system that allows the user easy access to all parts of the curriculum from a top level menu;
- e-mail access to lectures/tutors;
- a discussion group to encourage sharing of problems and ideas amongst students;
- on-line access to all the necessary organisational issues surrounding the operation of the course;
- on-line assessment to aid students in measuring their level of understanding of content and concepts;
- tools for monitoring student progress, including support for students to plan their own work programme.

\* Accepted 14 May 1999.

These are not perhaps unique but taken together they form an essential part of providing students with the necessary information in a timely fashion and at their own convenience.

## EDUCATIONAL CONTEXT

The use of various forms of multimedia technology in teaching and learning is a rich area of research, and it has attracted a wide range of interest groups from educational theorists to domain specialists attempting to build environments to match their own specialist needs. In many cases the results of this work are less than conclusive or never fully implemented and tested.

We recognise the importance of navigation paradigms on the learning experience. Barker and King [1] have outlined an approach to performance evaluation based on four categories: quality of the end-user interface design, the level of engagement of the users during use, the level of interactivity for the user and the ability to tailor the environment to suit individual needs and the educational experience.

It is also clear that the instructional designer must have a clear set of goals for the learning process and be able to match these to the ability and needs of the student-user as well as to the educational objectives of the teaching domain [2–4]. This includes having a clearly defined (and perhaps described) model of the subject being taught and the underpinning concepts necessary for the user to reach the required level of understanding.

Communications between student and teacher is central to many teaching and learning experiences. In [3, 5, 6] a range of issues concerning the facilitation of student/teacher communication channels is raised. While we may have many good ideas on how to facilitate these communication channels there is still much work to be done in facilitating them, and evaluating their effectiveness.

In an area of application closer to the needs of teaching in engineering and technology a recent ACM SIGCSE conference [7] included a wide range of papers on the use of computer-based (including Internet) technologies in computer science education. Topics covered include on-line assessment of programming projects, software project management, teaching of computing concepts (e.g. data structures and algorithms) as well as the quality of the learning experience.

## TEACHING ORGANISATION

On-line teaching begs the question of the role of face-to-face teaching through lectures and small group tutorial sessions. It is our view that face-to-face contact will most probably remain an essential element in a quality teaching and learning

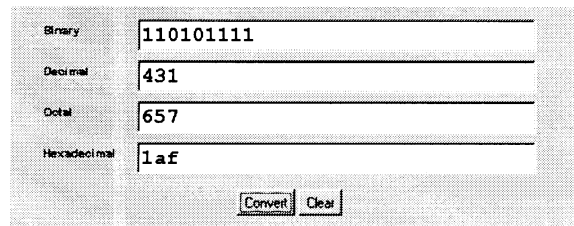


Fig. 1. Number conversion applet.

environment. Distance education makes this more difficult to achieve. In such environments it may be argued that the on-line teaching modes provide a real advantage without the practical presence of face-to-face contacts.

Traditionally, lectures are intended to provide curriculum content and tutorials/laboratories assist in reinforcing/practising this through more detailed examination of content material. On-line modes offer a different approach. The content delivery, the reinforcement and practice can be provided through the on-line/interactive activities. The face-to-face activities have a new focus—to provide inspiration, encouragement and reinforcement in the teaching and learning process. In our on-line teaching the approach we have adopted is:

- Lectures provide an opportunity to introduce the subject matter in context, with the aim of generating enthusiasm for the subject. In this approach there is less pressure to cover content and more time to develop the context for the curriculum content. The detailed content is available via the web pages.
- Tutorials may continue to provide opportunities to look at example problems, perhaps expanding on the details which are provided on the web pages. More importantly they provide an opportunity to monitor student progress and to provide guidance on study methods and time organization.

### A FSA to process real numbers...

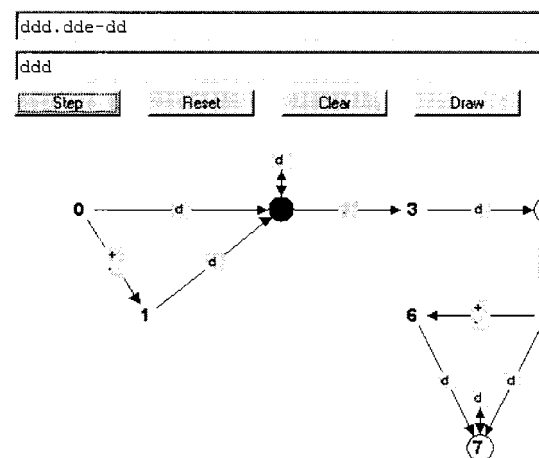


Fig. 2. Finite-state automata applet.

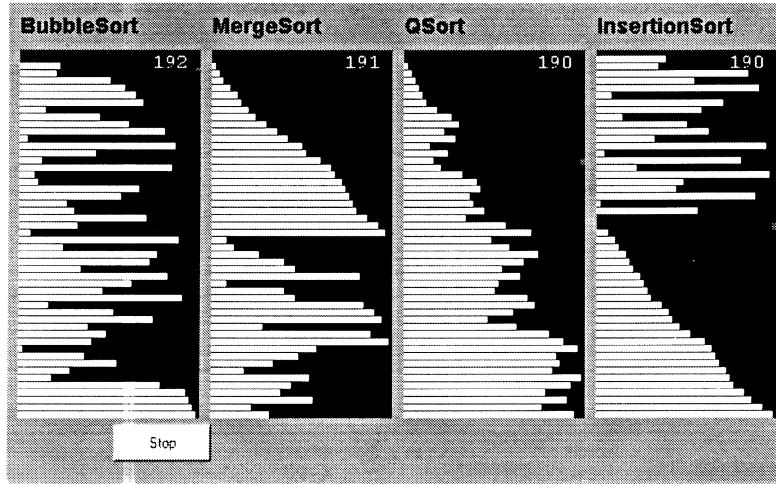


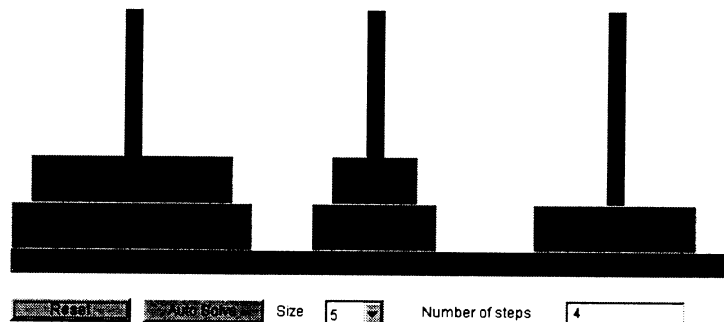
Fig. 3. Sorting algorithm applet.

**WEB PAGE CONTENT**

A well designed web-based course provides opportunities which are not easily achieved in a lecture room situation. While it is possible to bring a computer into the lecture theatre for demonstrations, a web-based approach allows each student to operate their own simulations or example problems on a one-to-one basis. This, we feel, offers considerable potential for holding the attention of the student as they progress through a particular learning experience. This premise has encouraged us to develop a range of interactive activities which we embed into web pages. These are intended to develop and enhance the learning outcomes.

These developments are central to our approach to on-line teaching methods. It is not sufficient to simply convert lecture notes, or written texts into web pages—there is a whole new paradigm to be cognisant of. User interaction, both physically and intellectually provides the essential basis for an effective teaching environment in a web-based delivery mode. Some examples of our developments follow. In each case we present a snapshot from a web page which is presented in context (of a problem to be solved, or concept to be appreciated).

1. **Number conversions:** An understanding of number systems and being able to convert from one radix to another is a common element in all computing courses. We have developed an applet (see Fig. 1) which can be embedded in a web page which facilitates this function. The aim is to encourage students to experiment with numbers in different systems so that some underlying appreciation is achieved. The student can type in a number value (in any radix) and see the converted values in the other fields.
2. **Finite state automata:** These are, of course, an essential element in the teaching and understanding of computing methods. We have implemented a simulator (see Fig. 2) which can be embedded into a web page. The student can enter an input stream of symbols for processing, and then step the machine through its operation, observing the transitions and state changes until the stream is successfully processed, or an invalid symbol is found. This particular FSA processes streams of symbols intended to represent IEEE floating point numbers.
3. **Sorting algorithms:** These are a usual part of all introductory computing courses. The simulator we have built (Fig. 3) shows graphically the sorting processes of four commonly used



(C) 1998 Geoffrey G. Roy, Murdoch University

Fig. 4. The Towers of Hanoi.

### Linear Programming Simulator

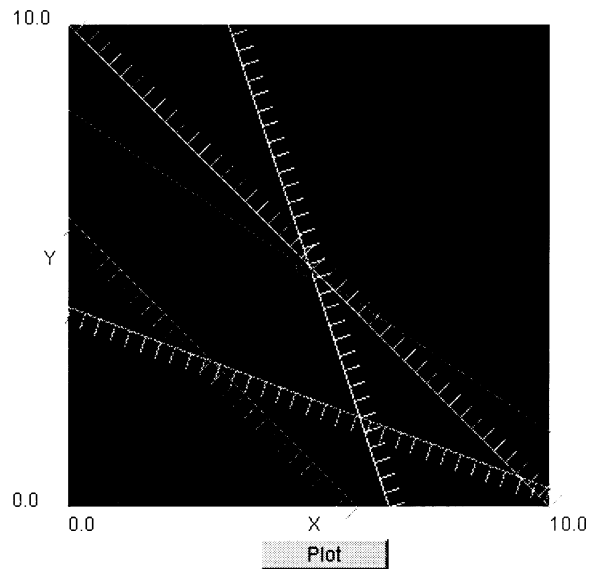
Click and drag target on objective function to move it

2.0	X +	3.0	Y +	1.5	Objective Function
1.2	X +	2.3	Y	=	13
0.9	X +	2.4	Y	>	10.0 <input checked="" type="checkbox"/>
1.0	X +	1.0	Y	>	6.0 <input checked="" type="checkbox"/>
2.0	X +	2.0	Y	<	20.0 <input checked="" type="checkbox"/>
3.0	X +	1.0	Y	<	20.0 <input checked="" type="checkbox"/>

min X 0.0    max X 10.0    min Y 0.0    max Y 10.0

X-value     Y-value

Objective value



Concept: Peter L. Lee, Applet: Geoffrey G. Roy, (C) 1998, Murdoch University

Fig. 5. An LP simulator.

algorithms. The students are not only able to see the algorithm in action, but see the relative performance of the algorithms in terms of the time it takes, as well the actual number of steps as computed by the complexity measure.

- Recursive algorithms:** Recursive algorithms are difficult to explain and perhaps even more difficult to display graphically. We have made one attempt using the Towers of Hanoi problem as shown in Fig. 4. In this example the user is invited to try to solve the problem manually (after having the algorithm explained). The applet will also solve the problem graphically, showing each step.
- Linear programming:** LP is one of those classic problem areas in which we expect some level of understanding, and apart from quite trivial examples it is always done using rather 'black box' computer packages. Simple examples are

done by hand—pencil, ruler and paper. In an attempt to make this introduction to LP more interesting we have built a simple LP simulator where we can represent simple (2-D) problems, with a small number of constraints, in a way that the user can manipulate the constraints and the objective function (Fig. 5). The applet does not solve the LP, it just provides a means of drawing it and allowing the user to slide the objective function around—to see the effect (on variable values) and to see where optimal and near-optimal solutions might lie.

- RSA encryption:** Teaching the concepts of number theory and searching for interesting applications some interesting examples has led us to RSA encryption as a useful application. While the concepts are important to understand the computations are difficult without computer support. We have developed an applet

E	<input type="text" value="3079"/>	<input type="text"/>
P	<input type="text" value="1531"/>	<input type="text"/>
Q	<input type="text" value="431"/>	<input type="text"/>
N	<input type="text" value="659861"/>	<input type="button" value="Prime Tester"/>
D	<input type="text" value="636319"/>	

Input Text

Encrypted Text

De-encrypted Text

Fig. 6. RSA encryption.

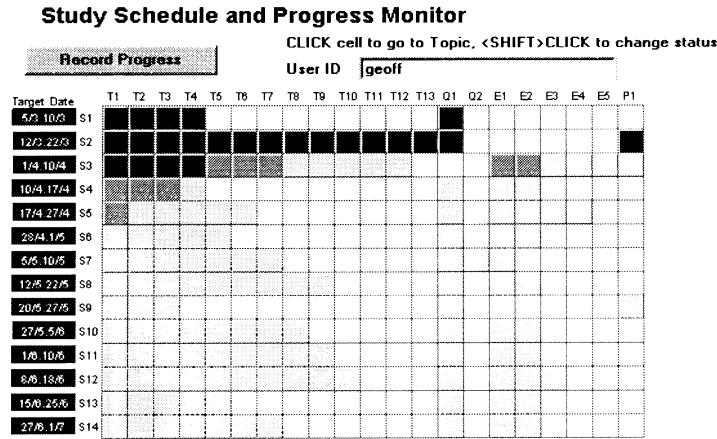


Fig. 7. The study schedule planner and progress monitor.

which provides the user with access to all the important parameters as shown in Fig. 6. The user must select suitable prime numbers for E (the encrypting exponent), P and Q (the prime factors of N). The inverse D, of  $(P-1) \times (Q-1)$ , is computed thus allowing a piece of text to be encrypted and decrypted.

**PROGRESS TRACKING**

Keeping track of student progress is not as easy as it first seems in an open browser environment. We have developed a prototype that uses a database residing on the server computer. Each student is required (requested) to record their progress in the database. We have constructed a browser interface for both the students and lecturing staff to monitor and analyze progression rates. An image of the student record screen is shown in Fig. 7. Here we see the current state of user 'geoff', showing what parts of the course are complete, in progress and yet to be attempted.

This type of progress monitor presumes that the content course is to be attempted in a sequential manner. While there are no real constraints on the students, the general idea is that they start with topic 1 in section 1 and proceed serially. This approach is typically most suitable for first-year courses where study/planning skills are still being developed.

Figure 8 shows an approach which is more suitable for later year students where we can presume the existence of reasonable study skills and well planned work habits. In this case we offer alternative pathways through a complete course, from the green (upper left) node to the red (lower right) node. Where sequence is important we describe this explicitly by the arrowed pathways. Users record their progress by adding notations to the nodes to indicate an 'in progress' or 'completed' status.

A system that automatically tracked a student's progress through the material, and even diagnosed that the student required extra tuition or could

progress faster would be very useful. Such a system would enter into the world of intelligent tutoring systems, an area of very active research. The underlying difficulties concern isolating study/learning activity from just 'browsing'. The latter is perfectly valid, but may not pass as 'deep' learning.

The problem of receiving student submission of assignments and returning marked versions to students is still not fully solved. Issues revolve around ensuring that submissions received originate from the student in question, and not submitted from elsewhere. While password and other security codes can ensure that a submission comes from the named account, it is less clear how we can guarantee that the appropriate student is operating the account.

We are currently experimenting with an on-line test facility as shown in Fig. 9. Here the instructor develops a set of multiple choice questions, with answers and explanations. After attempting the

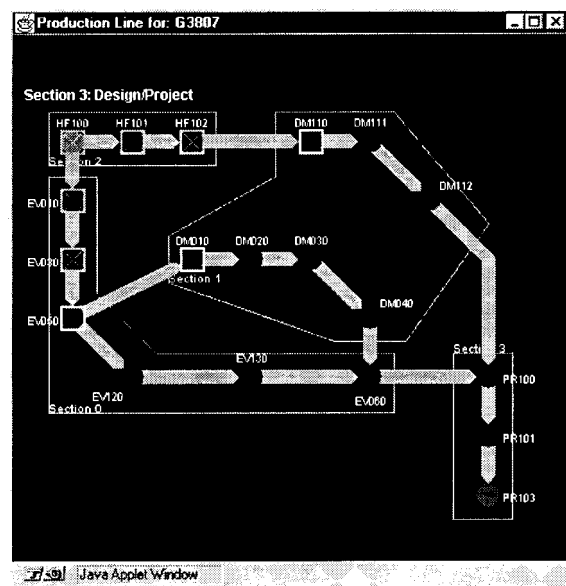


Fig. 8. Study planning and progress tracking.

**School of Engineering - Online Test**

	Question		Score (%)
<b>Q</b>	Which of the following complexities would result in the fastest solution to a problem?		15
<b>A1</b>	<input type="text" value="O(NlogN)"/>	<input type="checkbox"/>	?
<b>A2</b>	<input type="text" value="O(N*N)"/>	<input type="checkbox"/>	?
<b>A3</b>	<input type="text" value="O(logN)"/>	<input checked="" type="checkbox"/>	✓ ?
<b>A4</b>	<input type="text" value="O(N)"/>	<input checked="" type="checkbox"/>	✗ ?
<b>A5</b>	<input type="text"/>	<input type="checkbox"/>	?

Question No. 
No. of Questions

Applet started

Fig. 9. An on-line test.

questions in the set, the user requests them to be marked. The results are shown to the user and the score captured in the on-line database which can be accessed by the instructor. Students are required to provide a password to access the questions, thus providing some level of authentication.

### SOME IDEAS FOR THE FUTURE

There are a number of issues still to be addressed in our implementation of our on-line courses. We highlight some of these issues here so that others may also be inspired to solve these problems. We have some already in development, as prototypes, but others are still just ideas.

#### *Electronic submission and marking*

We need effective tools for annotating submissions and returning them electronically. There are tools in some commonly used products (e.g. Microsoft Office), but they most often do not allow the flexibility that the 'pen and paper' provide for annotating assignments and reports.

#### *Electronic whiteboard*

The use of electronic whiteboards, whereby a lecturer could use a writing tablet in one location and have the image displayed to a number of students in other locations and vice versa still does not seem to be straightforward—at least in an open browser environment. This display may be 'live' or 'delayed', i.e. accessible by a student some time after the whiteboard interaction took place.

The advantages of such interactivity to

display equation development and problem solving strategies are obvious.

#### *Remotely operated experiment*

One of the key challenges remaining will be the role of experimentation. 'Dry' laboratories or computer-based simulations do not provide the full rich learning experience of laboratory experiments. Operating some experiments over the web is possible but the communications bandwidth available and the safety issues involved currently limit the scope of such activities. The future may solve the former, but is unlikely to solve the latter.

### STUDENT RESPONSE

We have had one year's experience with web-based teaching and the feedback we have from students is very encouraging. We can say that we are doing at least as well as more traditional methods with some indications that it may be better. Some particular responses from students include:

- positive support for having course materials available any time, and especially from home;
- reduction in numbers of formal lectures;
- an appreciation that the quality of the web-based materials is supportive of a productive study programme.

Longer term learning outcomes are more difficult to judge. We certainly expect to achieve outcomes at least as good as traditional teaching methods, though this is still to be proven.

### REFERENCES

1. P. Barker and T. King, Evaluating interactive multimedia courseware—a methodology, *Computers in Education*, **21** (4), (1993), pp. 307–319.

2. J. D. Young, The effect of self-regulated learning strategies on performance in learner controlled computer-based instruction, *Educational Technology Research and Development*, **44** (2), (1996), pp. 17–27.
3. G. Blissett and M. Atkins, Are they thinking? Are they learning? A study of the use of interactive video, *Computers and Education*, **21** (1/2), (1993), pp. 31–39.
4. R. Yildiz and M. Atkins, Evaluating multimedia applications, *Computers in Education*, **21**(1/2), (1993), pp. 133–139.
5. D. Laurillard, *Rethinking University Teaching: A framework for the effective use of educational technology*, Routledge, London (1993).
6. B. Collis, *Tele-learning in a Digital World: the future of distance learning*, International Computer Thompson, London (1996).
7. Association for Computing Machinery, Special Interest Group, Computer Science Education, *ACM SIGSE Conference, Integrating Technology into Computer Science Education*, Uppsala, 1997.

**Geoffrey G. Roy** is the professor of software engineering in the School of Engineering at Murdoch University. He has BE and M.Eng.Sc. degrees in Civil Engineering, and Ph.D. from the University of Melbourne. In recent years he has held academic positions at The University of Western Australia as Head of the Computer Science Department, then leading the Computer Science Programme at Murdoch until the recent establishment of the new School of Engineering and its software engineering programme. He currently has particular interests in the development of on-line teaching support tools to assist in supporting the school's move towards a stronger on-line delivery mode.

**Peter L. Lee** is the professor of Instrumentation and Control Engineering in the School of Engineering at Murdoch University. He has a BE degree in Chemical Engineering from the Royal Melbourne Institute of Technology and a Ph.D. from Monash University. In recent years he has held academic positions as the head of the Department of Chemical Engineering at the University of Queensland. His current work is focused on the establishment of the new School of Engineering and the development of a totally new curriculum using on-line and other modern delivery modes.