Experiences Developing Low-cost Courseware for an On-site Taught M.Sc. Course

Gavin Russell & Ian Pitt
University College Cork, Ireland

g.russell@cs.ucc.ie
i.pitt@cs.ucc.ie

Gary Couse
Cork Institute of Technology, Cork Ireland

gcouse@cit.ie

Abstract

This paper describes experiences designing and developing a low-cost courseware system to support the taught M.Sc. course in Multimedia Technology in the Computer Science Department of University College, Cork, Ireland using a combination of both original and modified open-source components. The requirements analysis, design, development and testing phases of the system are all described. Finally, observations on its first year of operation are presenting including suggestions for why the completed system fell out of use before the years end.

Keywords: E-Learning, Courseware, Open Source, Multimedia, Web Design

Introduction

In the Summer of 2000 it was decided to investigate the feasibility of implementing a courseware management system for the taught M.Sc. in Multimedia Technology course in University College Cork, Ireland. The course had been running since the previous October and it was hoped that the experiences of that first year could be built on in order to develop a system to cater for the day-to-day needs of both the staff and students of the degree. It was also hoped that this would reduce the administrative load on staff thereby freeing up time so that they could concentrate on academic rather than administrative tasks (such as collecting assignments, booking resources, and facilitating group discussion between students).

Restrictions

Before deciding on the actual features of the finished system certain restrictions had to be kept in mind. This system had to be developed at little or no cost to the Computer Science Department for whom it was being developed, and there was no external funding available (This was partly due to the system’s origins as a student project in the M.Sc. program). This automatically precluded software learning systems such as *Blackboard* (http://www.blackboard.com) or *WebCT* (http://www.webct.com).

As the site was going to be used by lecturers *and* incoming students, it was necessary to plan for an initial launch date of 1st September (giving lecturers a month to familiarise themselves with the system and to prepare content before classes started in October). As a result, there were only ten weeks for research and development. Testing could be carried out in the remaining four to five weeks (before the start of the academic year).

The only way to feasibly meet the financial and time constraints was to make use of freely available software. Open source software being particularly attractive in this regard as it overcame any possible licensing or copyright problems.

Paper Accepted as a Regular Paper

Since the open source community is more active on Unix Systems (and their variants) it was decided to run the main courseware server on a UNIX or LINUX platform. Other factors which suggested these platforms included reliability, stability, remote administration, and cost (the variant chosen – FreeBSD – was, as its name suggests, free). Apache was chosen as the web server.

Features

The short-term goal of the project was to provide the M.Sc. in Multimedia Technology with an Extranet to support course activities. The long-term goal was that this Extranet would be a scalable and flexible solution that would meet future requirements. However, the initial requirements and features of the system had still to be decided upon. After brainstorming sessions, analysis of existing (and more costly) solutions (Blackboard, WebCT and Top Class) and interviews with both staff and students, the following list of possible features suggested itself:

- A structured gateway to course material
- A Support section (FAQ & Tutorials)
- A Project management/Progress area that will indicate to the student their progress in the degree
- A Student Project Gallery
- Bulletin Boards
- A Student CV/Resume Database
- A links area where students and staff can submit URLs of interest to the others
- A Homework Submission Tool (with time stamping)
- A Calendar of Events
- A Web-based Email System
- A tool for converting notes to the PDF format
- Personal homepages for each student maintained by the Courseware system

The intention was to use this list to provoke debate and discussion amongst staff and the current students undertaking the degree in order generate a final list which would be approved by both the lecturers (the *producers*) and students (the *consumers*) of the system.

A survey was used to gather opinions on the importance of each of the proposed features. The survey was carried out using two online form-based questionnaires: one for the lecturers and one for the current students, leaving plenty of space for students and lecturers to make their own suggestions.

When all the results were collected 50% of the students had replied and all but one of the lecturers. Given the relatively small sample of people polled (25 students and 4 lecturers) the results could not be considered authoritative but of particular interest were the following:

- Several students were reticent about CVs being put on the site. They felt that all students in the class should be considered equal, and given the nature of a taught masters course, some students would have much more impressive CVs than others, possibly intimidating them.
- In a question asking which format they would prefer teaching materials to be made available in, students overwhelmingly preferred HTML (72%), while no lecturers chose this option. The options preferable to lecturers (PDF, Powerpoint Documents, etc.) had a low conversion overhead in common. The significance of this would become apparent once the project was completed.

Once the survey was finished the final list of features was drawn up based on its findings and several follow-up interviews:

- An Assignment Submission and Reporting Tool
- A Calendar
- A Link Management System
- A structured gateway into the course material (i.e. lecture notes, etc.)
- Bulletin Boards
- A Resource Booking System (for Cameras, etc.)
- A Student Notice Board

This list represented the features that both staff and students felt would be useful if they had to use the system. We would also have to ensure that each component had a similar look-and-feel so that users would not be presented with a selection of different navigation systems and layouts, depending on which component they were using.

Implementation

As there was no one open source product that provided *all* of these services at the time each component part of the system had to be approached separately. For each individual component a list of criteria were drawn up which it was decided were essential if the components were to meet the requirements of the final courseware system. The feature sets of each of the candidate open source products were then examined. Any that met all the criteria would be modified to fit into the final system so that it looked like one seamless product.

For example, candidate solutions for *each* component were tested to see if they met the following general criteria:

Robustness - All components should function as independent processes that do not affect other processes or the system as a whole. The system should also be fault tolerant (i.e. it should be able to recover from incorrect user input or restart automatically in the event of a system crash or a power failure).

Cost - Due to the financial restrictions on this project, all software utilised in this project should be either freeware, open source or should already be licensed within the department.

Easily Maintained - All components should be easy to maintain.

Scalability - All components should be scalable, i.e. components should remain functional and efficient regardless of the number of users, or the amount of data stored, etc.

Well-commented, structured and documented code - All code should be well documented and commented to facilitate structural modifications when and where necessary.

Security - Directory structures should be designed in such a way that permissions applied to them will correspond to logical security entities (e.g. if all the calendar files were put in one directory, that directory could be password-protected, thus protecting the calendar)

Customisation – It should be possible to customise a component with regard to graphics, their look & feel, etc. so as to fit in with the web site identity. This would eliminate many solutions for which the source code is not available.

Administration - All relevant components must have administrative facilities. For those components that need regular administration (e.g. the Assignment Submission and Reporting Tool and the Link Management System) the interface should ideally be web-based.

Product Support - Where external scripts are concerned, there should be adequate online support in the form of documentation, FAQ's, discussion forums or availability of the author.

Components

Along with those listed above the individual components of the system had to meet extra criteria unique to their particular function (Russell & Van Leeuwen, 2000) The combination of the general and specific criteria was used to evaluate the available products and helped determine which were most suitable. The final choice of product for each component is listed below (several of the alternative products that were also examined are listed at the end of the paper).

Calendar

Matt Kruse's *Calendar Script* (Kruse, 2000) was chosen here since the script met all the criteria. The installation of this script also proved to be the most straightforward of the three calendar scripts evaluated (Burgdorf, 2000, Forbes, 2000 & Kruse, 2000)

Link Management

A component was also needed that would allow students to submit the URLs to various online resources that they thought might be of use to their classmates. In the end it was decided to use *Links* (Gossamer Threads, Inc, 2000). This system was free for non-profit use and was a perfect match for the needs of the system (The others products examined included DBMan by Gossamer Threads and the *Linkmatic Web Director*y (DePasquale, 2000))

Course Material Database

During the process of adapting *Links* to meet the requirements of the Links Management component it became clear that it would be simple enough to customise it to be used as an interface to the course materials that lecturers would be providing. During the initial meetings with the lecturers of the course, they made it clear that the presentation of lecture notes, assignments and practicals, should only require a minimum amount of effort on their part.

By using *Links* as an interface, it would be possible for them to keep their course material on their own servers (or any other preferred server) and simply add links to those files via *Link*'s administration page. They could then use a HTML template to mark-up their notes. Moreover, the links database allows for any kind of URL to be added, so it would also be possible to link to PDF files, graphics, QuickTime movies, etc.

Bulletin Board

Once the criteria for the bulletin board system had been arrived at it was decided to first evaluate the bulletin board system already in operation in the Department of Computer Science in UCC (*Ultimate Bulletin Board*, Infopop Corporation,2000). This met all the criteria for this component, and since this system was already installed and running it was decided that the forums required for the multimedia group web site should simply be added to the existing system and linked to from the multimedia web site.

Resource Booking System

Both staff and students wanted to be able to reserve workstations and equipment for certain time slots and, if possible, allow for block booking of the same time over a number of weeks. The idea of this Resource Booking System was introduced very late in the development of the project and although the chosen solution (Snellman, 2000) could have been adapted to the needs of the project, it eventually became clear that the modifications could not be completed in time and the component was not included in the final system.

In the time since the project was finished, several free resource-booking systems have become available on the Internet such as *e-allocator* (Dragon Technology Ltd., 2000)

Student Notice Board

The student notice board is the page that all students are presented with when they enter the student area. This was implemented by the project team with a Perl script (Russell & Van Leeuwen, 2000).

Assignment Submission

After reviewing the alternatives (Meltze, Michalski, 2000 & Frederic, 2000), it was decided that there was no product or single piece of code that met the specific criteria laid out for the Assignment Submission and Reporting Tool and therefore a system needed to be written by the project team (Russell & Van Leeuwen, 2000). This was written in PERL and provided the following functions:

> **Assignment Creation** – When a lecturer set a new assignment for his/her module, they were also able to use the Assignment Submission and Reporting Tool to name the assignment and give it a due date so that students could submit their homework to it.
>
> **User Management** – The system had a user management facility. Staff could add, edit and delete new student users as well as new staff users.
>
> **Secure Upload Facility** –This system allowed individual users to be assigned specific directories to upload their files to.
>
> **Reporting Facility** – The system included a method for showing which students had handed in their work and when.

Integrating the Components

Once the individual components were ready, it was just left to tie all these disparate elements together into a seamless system. This mostly involved adapting each product so that they presented the same look and feel, and presenting them to the user within a site architecture that was easily understood and easy to navigate. Designing a usable architecture and layout was considered an essential part of the final product (Nielsen, 2000 & Rosenfeld, Morville, 1998).

Having each component conform to the same look and feel involved modifying the code of any products not written by the project team. This required a greater understanding of the product than would normally be expected for their general use. Their code had to be scrutinised, graphical interface sections identified and modified to conform to the graphical layout of the site (see Figures 1, 2, and 3 for examples of the layout).

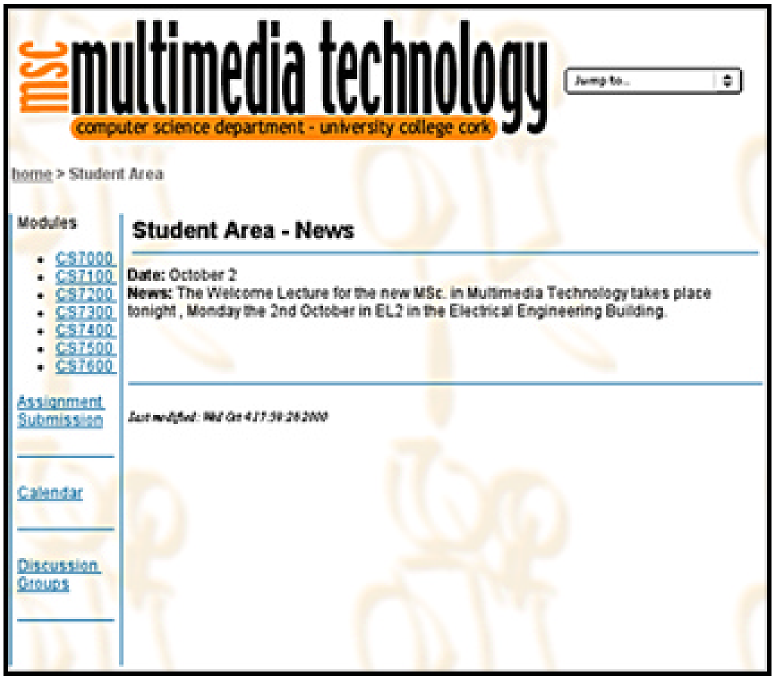
HTML templates were also provided for individual lecturers to use if developing pages of their own.

Testing

The main bulk of the M.Sc. in Multimedia Technology site targets two distinct user groups, namely the lecturers and the students. One of the lecturers began testing the staff area from approximately four weeks prior to the launch date, while the previous year's students and the incoming students also had the opportunity to test the student section of the site. The few errors reported (mostly typographical) were easily corrected.

The system was finished and ready to use on September 30th 2000, one day ahead of schedule. Both of the major constraints had been met: the 10-week development period and the (non-existent) budget. Its use during the following academic year was then monitored.

Experiences in the First Year and Beyond



home > Student Area

Modules

- CS7000
- CS7100
- CS7200
- CS7300
- CS7400
- CS7500
- CS7600

Assignment Submission

Calendar

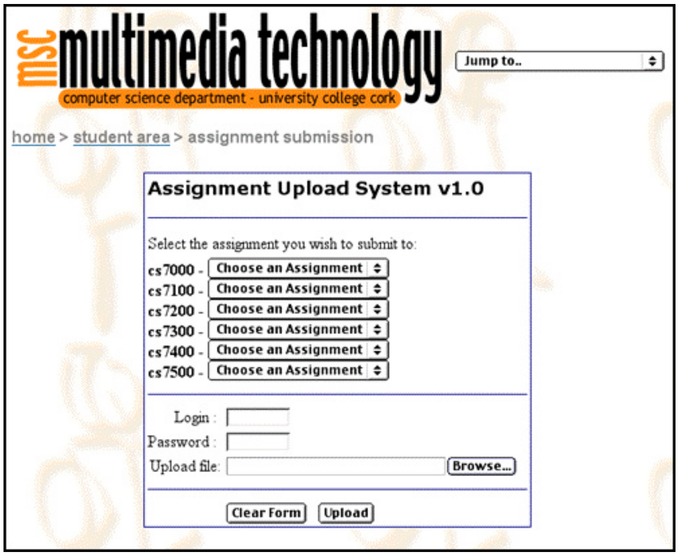
Discussion Groups

Student Area - News

Date: October 2
News: The Welcome Lecture for the new MSc. in Multimedia Technology takes place tonight , Monday the 2nd October in EL2 in the Electrical Engineering Building.

Last modified: Wed Oct 4 17:59:26 2000

Figure 1: A Students main page after logging in, showing the notice board.



home > student area > assignment submission

Assignment Upload System v1.0

Select the assignment you wish to submit to:

cs7000 - Choose an Assignment
cs7100 - Choose an Assignment
cs7200 - Choose an Assignment
cs7300 - Choose an Assignment
cs7400 - Choose an Assignment
cs7500 - Choose an Assignment

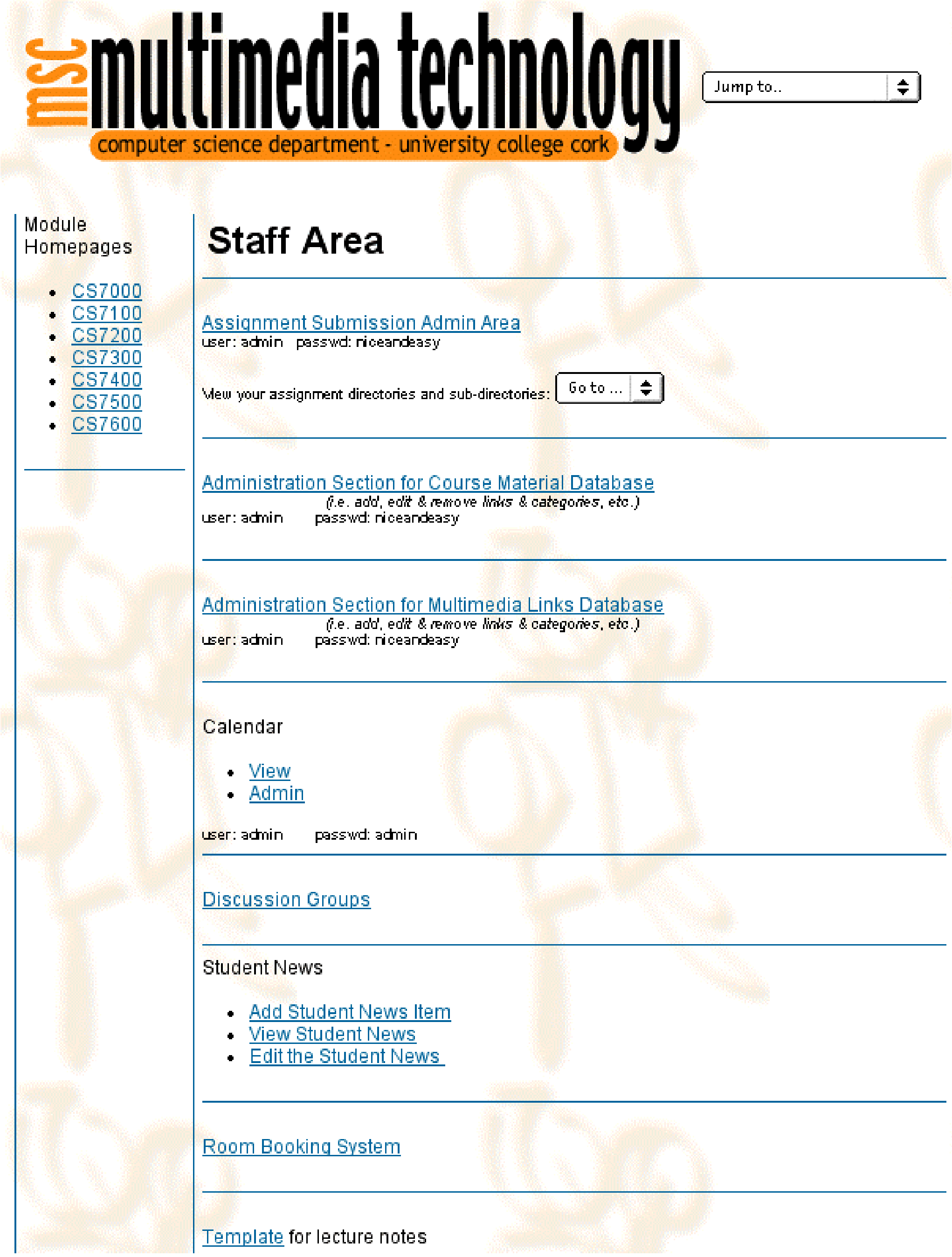
Login :
Password :
Upload file:

Figure 2: Assignment Submission Page

The most obvious thing that was noticed during the site's first year was that the look and feel of the site began to fragment over the year as individual lecturers (most of whom had considerable web design experience due to the nature of the course) began to modify the templates they had been provided with in order to suit their own requirements. They reported that some of the elements of the template (in particular a menu sidebar, as seen in Figure 1) took up too much screen space, which they felt they could better utilise. They had originally approved the layout of the templates but it was through their daily use that this problem became apparent. The fragmented look that this introduced led to different navigation systems being used in different parts of the site (something it was hoped could be avoided). As a result the cohesive nature of the site began to deteriorate, and combined with the areas of the site that were rarely updated (again, these were areas that had been approved during the design phase of the project but were found to be impractical to upkeep during the day-to-day experience of running the course) meant that large areas of the site fell into disuse and as a consequence students began to lose interest in using the site.

The following academic year proved no better. All that remained in use was some of the original look-and-feel of the site (via the original page layout templates, graphics, logos etc.).

The project was a success from one of point of view: It proved that an integrated courseware system could be created in a short period of time for little or no money. Despite their alteration, the continued use of the templates also proved that a uniform look-and-feel was important. However, the more important goal - that a system could be developed that would be of long-term use to both students and lectur-



msc multimedia technology

computer science department - university college cork

Module Homepages

- CS7000
- CS7100
- CS7200
- CS7300
- CS7400
- CS7500
- CS7600

Staff Area

Assignment Submission Admin Area
user: admin passwd: niceandeasy

View your assignment directories and sub-directories:

Administration Section for Course Material Database
(i.e. add, edit & remove links & categories, etc.)
user: admin passwd: niceandeasy

Administration Section for Multimedia Links Database
(i.e. add, edit & remove links & categories, etc.)
user: admin passwd: niceandeasy

Calendar

- View
- Admin

user: admin passwd: admin

Discussion Groups

Student News

- Add Student News Item
- View Student News
- Edit the Student News

Room Booking System

Template for lecture notes

Fig 3: Sample Lecturers Web Page after logging in

ers - was not met. Although the system was used initially, both student and lecturer interest in it waned throughout the year.

One factor that doomed the project from the start was an adherence to conventional web design practices. A mistake many people make when designing commercial web pages is that they organise and label the site from the perspective of the producer (the providers of the information or service represented by the site). Structuring a site based on concepts of importance only to producers is counter productive. Consumers expect to find the information laid out in a way that is of use to them. All good commercial websites take the consumer's needs and expectations as the primary focus for their design (Morkes & Nielsen, 1997) .The mistake in this case was to apply this principle to a courseware site. This project was focused around a list of the features mostly requested by students. Each of these features was implemented and it was assumed that the site would be driven by the students' requirements. However, more attention should have been given to the source of the information they would be accessing. Rather than enquire of the lecturers what they would like to see on the site it would have been better to have gauged the level of work they, as producers of the content, were capable of providing on an ongoing basis (in addition to their normal workload).

The nature of an *on-site* learning support system is different to that of a distance learning or *off-site* system. In the case of distance learning the maintenance of the website is an essential part of the teaching process. It is the primary communication tool possessed by the lecturer. Student-to-student interaction can most efficiently be facilitated through electronic means supported by the courseware (especially for one-to-many communication). Student-to-lecturer interaction can also be best facilitated in this way (since asynchronous electronic communication overcomes time-difference issues which are typical of the distance learning model). However, in on-site learning schemes (i.e. where students are attending lectures in a central location) the courseware system is no longer the primary communication tool, the students can easily communicate with each other and student-lecturer communication primarily occurs face-to-face. As a consequence of this, maintaining courseware systems can entail *extra* work, where as in distance learning schemes, maintenance of the courseware systems is the main pedagogical task of the lecturer (other than course design). Therefore, for an on-campus course of this kind, the emphasis of the site should have been on facilitating additional forms of asynchronous communication between students and lecturers which would lighten the lecturers workload *and* accommodate the students, and, as a result, encourage both to use of the system (e.g. E-mail allows students and lecturers communicate without having to go to the trouble of finding a timeslot they are both available to meet; distributing notes online eliminates the need for photocopying on the part of the lecturer and allows students view their notes wherever they have an internet connection, and so on). If lecturers are not capable of maintaining this level of interaction for whatever reason then a courseware system is not required.

An early prototype of the system might have helped here as the lecturers would have had a better idea of the work involved. The testing phase of this project focussed primarily on finding bugs whereas it might have been better spent on usability (from the point of view of the lecturer as well as the student). Any component which the lecturer felt they could not maintain throughout the year could have been removed. This would have left students with a smaller courseware system, but one that was fully supported by the staff. In essence, the *producer and consumer* model should have been changed to one where the producer (the lecturer) is *also* the consumer (i.e. the lecturer interacts with the site as much as the student does, which is usually not the case in the traditional model where the producer just provides the content and it is the consumers that interacts with it.)

Summary & Conclusion

This project was designed from the start to facilitate the teaching of an on-campus taught Masters. A courseware system was required that could serve the class at little or no cost to the Department running the course. This goal was achieved by using various products freely-available on the internet, modifying their code so that they could be integrated seamlessly into the main system, and complimenting these with components written specially by the development team to provide functionality not otherwise available.

Even though it is not still in use the project proved that it is possible to implement a complete courseware system within tight time and budget constraints. But it is only advisable if you are able to maintain the site technically. Open source products frequently provide some sort of technical support, but it tends to be of an ad hoc nature. If your organisation lacks the technical support necessary to create and maintain such a site it cannot be recommended. Using and maintaining open-source products and/or writing the courseware yourself can require a large skill base (including CGI scripting, PERL, PHP, Databases, SQL, HTML, HTTP, JavaScript, Configuring Web servers, security, and so on).

Watching the site in its first year of operation revealed a possible mistake in the way the site design was approached. By basing the features of the final system on other more costly systems, and on "wish lists" on the part of staff and students, the practicalities of the situation were overlooked. A courseware system is, at its most basic, a communications medium to aid student/lecturer interaction. Regardless of whether you are using a commercial or custom-built site, participation on the part of the staff must be taken into account beforehand. You must establish the minimum work the producers are willing or capable of doing in order to maintain the site. These results will shape the foundation of the site. On this foundation you can add other features over time as staff and students familiarise themselves with the system and adapt to its potential.

Another problem that arose with this system is the inconsistency with which the content was approached. A dedicated site administrator or editor would have greatly benefited the system and would be suggested to anyone considering a similar approach. This person would be committed to ensuring that the site is up to date and consistent, and would be able to help lecturers use the system if necessary. Their job would be to monitor the content of the site and as such would not necessarily need any technical skills. A person like this could have prevented many of the problems encountered with this project by encouraging staff to keep their content up-to-date, enforcing interface guidelines for the site, monitoring student use of the site and so on. While lecturers would have different approaches to their own classes, certain elements of the a courseware site have to remain consistent to facilitate their use by students (much in the same way Operating Systems have human interface guidelines).

Finally, the experience with the Multimedia Technology website has also pointed to interesting research directions which we hope to explore, the foremost of which is the ideological difference between e-commerce sites (which are consumer led) and e-learning sites (which, from experience, tend to be producer-led systems).

References

Blackboard, Blackboard Inc. Retrieved September 2000 from http://www.blackboard.com

Burgdorf, D. C. (2000). *WebCal*. Retrieved September 2000 from http://awsd.com/scripts/webcal/

DBMan, Gossamer Threads Inc. Retrieved September 2000 from http://www.gossamer-threads.com/scripts/dbman/

DePasquale, J. (2000). *Linkmatic Web Directory*. Retrieved September 2000 from http://www.getcruising.com/crypt/ linkmat.html

Dragon Technology Ltd..2000, *e-allocator*. Retrieved September 2000 from http://www.e-allocator.com/

Frederic, T. *eUpload*. Retrieved September 2000 from http://www.ftls.org/en/examples/cgi/eUpload.shtml

Forbes, C. (2000). *Calendar of Events*. Collin Forbes Retrieved September 2000 from http://kuoi.asui.uidaho.edu/~collinf/index.html/calendar/about/ AAdp

Global Exchange Technologies, Inc. *Database Doctor*. Retrieved September 2000 from http://www.web-bazaar.com/cgi/

Infopop Corporation, *Ultimate Bulletin Board*. Retrieved September 2000 from http://www.infopop.com/

Kruse,M. (2000). *CalendarScript*. Retrieved September 2000 from http://www.mattkruse.com/scripts/calendar/

Links. Gossamer Threads, Inc., Retrieved September 2000 from http://www.gossamer-threads.com/scripts/links/

Meltze, K., Michalski, B. *Writing CGI Applications with Perl*. Addison Wesley, 2001 (The script originally evaluated for the project is now included in chapter 7 of this book)

Morkes, J., Nielsen, J. (1997). Concise, SCANNABLE, and Objective: How to Write for the Web. *Sun Microsystems*. Retrieved September 2000 from http://www.useit.com/papers/webwriting/writing.html

Nielsen, J. (2000). *Designing Web Usability*. New Riders.

Rosenfeld, L., Morville, P. (1998). *Information Architecture: for the World Wide Web*. O'Reilly & Associates.

Russell, G., & Van Leeuwen, J. (2000). Extranet and Courseware for the MSc in Multimedia Technology: Architecture and Implementation. Project presented for M.Sc. in Multimedia Technology, University College Cork

Snellman, J. (2000). *CALBOOK*. Department of Mathematics, Stockholm University. Retrieved September 2000 from http://www.math.su.se/~jans/ (product no longer available at time of writing)

Top Class, WBT systems, Retrieved September 2000 from http://www.wbtsystems.com

WebCT, WebCT, Inc, Retrieved September 2000 from http://www.webct.com

Biographies

Gavin Russell has been a lecturer in Computer Science, University College, Cork, Ireland since 1997. Before joining the university, he worked in the Web Publishing and Internet service provision businesses. His main research interests include: Web Based Learning, Interactive Multimedia on the World Wide Web and Secure Internet Payment Systems. He received an M.Sc. in Multimedia Technology in 2000 and is currently undertaking a Ph.D. in Computer Science at University College, Cork.

Ian Pitt has been a Lecturer in the Department of Computer Science at the National University of Ireland, Cork, since 1997. He studied Music and Drama before going on to take a masters degree in Music Technology. He then took up a research post at the University of York, UK, working on the design of auditory human-machine interfaces. After completing his D.Phil he spent a year as a post-doctoral research fellow at Otto-von-Guericke University, Magdeburg, Germany, before moving to Cork. His main research interests are speech, sound, and the design of human-machine interfaces, particularly multimedia interfaces and interfaces for use by blind people.

Gary Couse has been a lecturer in Computing at the Cork Institute of Technology since 1998. He received his M.Sc. in Computer Science in 1994 from University College Cork where he has also lectured since 1996. His research interests include object-oriented programming and computer aided learning.