Computer-Assisted Learning Using the Web

Mike Joy and Michael Luck
Department of Computer Science
University of Warwick
Coventry CV4 7AL
United Kingdom
{M.S.Joy,mikeluck}@dcs.warwick.ac.uk

Abstract

As student numbers increase, and as more flexible methods of teaching are required both for traditional higher and further education courses and for more novel distance learning, the challenge facing teachers has not, arguably, been greater. Fortunately, the recent advent of Internet computing and the World-Wide Web has provided a ready-made infrastructure for supporting attempts to deal with these issues. In this paper, we describe a *Web*-based system for instruction and course delivery to provide quality education that caters for these new and differing instructional needs.

1 Introduction

The recent rapid advance of Internet computing in its many guises has changed the way many of us go about our daily work. The speed with which changes have occurred has been dramatic, and we are already at a stage where we can exploit this infrastructure for serious and complex purposes which would otherwise be difficult or even impossible. The shift of emphasis in use of the World-Wide Web, for example, from individual information retrieval tasks to wide-scale applications is one demonstration of this.

Such applications are particularly prevalent in the education sector, in which there has been a period of great change and expansion. As a result of increased numbers of students in further and higher education, problems with existing methods of teaching have been highlighted and amplified, and new methods and techniques have become necessary.

In this paper, we describe a *Web*-based system for instruction and course delivery to provide quality education that caters for differing instructional needs.

2 The World-Wide Web

In a fundamental shift of paradigm, the key to unlocking the power of information technology is no longer the computer but the interconnection of computers, The World Wide Web is a loose catch-all concept that refers to globally distributed hypertext information [4]. Hypertext technology relies on browsers to make connections with remote machines for access to the information they contain. There are three important points to note about this. First, browsers are available for all types of hardware and software platform, and many are free. Second, the hypertext languages in which the

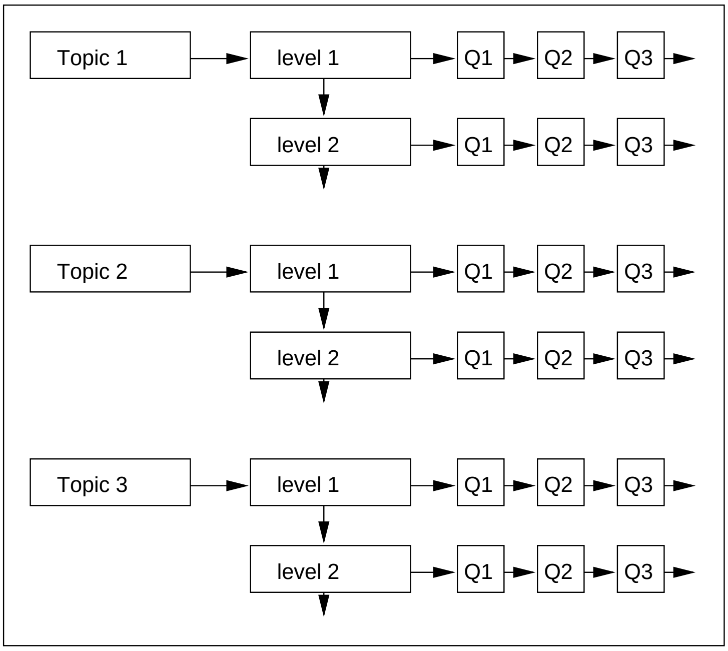

Figure 1: Levels of Questions Within Topics

information is represented, generically referred to as HTML (Hypertext Markup Language), are subject to standards and are reasonably uniform and consistent. In practice, this means that there are specifications for core HTML [1, 2, 3] enabling documents to be written which can be viewed correctly on all conformant browsers. Third, both of these are continually being developed and upgraded with the addition of new capabilities and functionality, but within existing constraints so that upward compatibility is maintained. This means that newer and more advanced tools are able to access information created using older tools without any changes necessary to the underlying materials. Furthermore, the information contained in hypertext form on the Web can also be downloaded and stored locally on a PC for standalone use without connection to a network, if so desired.

The World-Wide Web appears to be eminently suitable as an environment in which to base the development of computer-assisted learning delivery mechanisms and materials. It is *de facto* standardised but rapidly being improved and updated. It supports the complex organisation of information of many kinds, but facilitates easy modification and adaptation as well as allowing hooks to external services. By adding extra functionality for particular instructional purposes, we can develop a new and effective way of teaching that uses existing equipment and infrastructure, requires minimal extra development, and empowers the student.

Materials in our computer-assisted learning system (CAL) are organised by topic within a course. A course consists of a set of collections of related information and questions. For each topic, there are questions at several levels of difficulty, with ordered sequences of questions within a level. A student will typically progress through the levels of a topic by answering all questions on a level and then moving to the next, but it is possible to drop through to the next level at any point. This organisation, which is illustrated in Figure 1, structures courses, and it is in this context that the developed software functions.

3 The CAL System

The CAL system at Warwick consists of three major components, as illustrated in Figure 2:

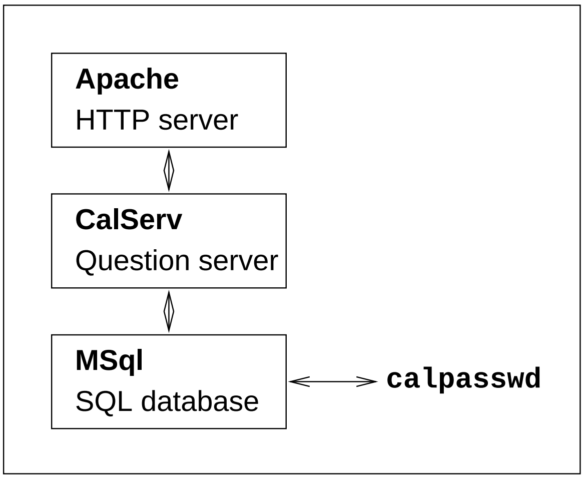

Figure 2: The CAL System Components

- an HTTP server
- an SQL database
- the CAL question server

Using a Web browser, a user initially connects to the CAL server. This delivers them hypertext documents, containing both tutorial material and links to exercise questions. When the user selects a question, a connection is made with the CAL question server (known as CAL*serv*), which returns a document containing a question. The user then chooses a response to that question, and this is returned to the CAL server which replies with an appropriate message.

The CAL server communicates with a database to store information about users' successes and failures when answering questions, so that feedback can be provided for them (and for the course tutors). Each time a question is answered, the result is stored together with an identification of the user. At any point, a user may request CAL*serv* to return a document outlining their recent performance. Tutors may similarly request statistics from CAL*serv*, either for a specific user, or global information for a course.

CAL*serv* stores questions as a collection of files containing markup of two types — HTML markup (enclosed by `<...>`), and extra markup (enclosed by `[...]`) which we refer to as *CALserv markup*. Each file contains a single question, its correct answer, and the desired responses for both correct and incorrect user choices. The question files are identified by the suffix `.ques` so that the HTTP server can redirect requests to CAL*serv*. When a question is requested, CAL*serv* preprocesses the file to extract just the question, and returns an HTML document. When the user indicates an answer to CAL*serv*, it is compared against the correct answer contained in the file, which is again preprocessed to return an HTML document containing the desired response.

4 Summary and Conclusions

We have developed a Web-based computer-assisted learning system that has been deployed and tested on different undergraduate courses attended by several hundred students in total. There are many possible modes of operation of the system, including its use in assessements by which a student's performance contributes to their overall course grade. So far, however, we have been using the system to augment existing methods of instruction rather than replace them. In this mode of operation, we maintain information about students but do not use it for assessment. Instead, this information is used to provide direct feedback to the students themeselves, and can also be used by instructors to identify any students having difficulties. The response has been generally favourable, and we are confident that our paradigm works, is easy to maintain, and is extensible in the future. Indeed, the architecture lends itself well to easy modification so that, for example, other forms of dialogue with the user can be supported, with only minimal changes.

The system has been deployed and tested on different undergraduate courses attended by several hundred students in total. The response has been generally favourable, and we are confident that our paradigm works, is easy to maintain, and is extensible in the future. There is no reason why more general dialogue with the user should not be supported, which can be pursued with minimal changes to the existing software.

We view our work as a *snapshot* of development over time. Though successfully deployed to significant effect, it is still a prototype proof-of-concept system. In that respect, we recognise that functionality and performance can be enhanced further. For example, the limited number of question types that are supported by the CAL*serv* markup can be further developed and extended. Despite these potential improvements, our basic structure and organisation are sound.

Future work will include such extensions, redevelopment using the Java language, enhanced security features through encryption technology, and so on. However, this will all be done within the context of the existing structure. Moreover, the integration of the WWW-based CAL software with other course management tools [5, 6] will go a long way to providing for our students a sophisticated and useful learning utility worthy of the late 1990s.

References

[1] T. Berners-Lee and D. Connolly. Hypertext markup language - 2.0. strange, November 1995. RFC 1866, `ftp://ds.internic.net/rfc/rfc1866.txt`.

[2] T. Berners-Lee, L. Masinter, and M. McCahill. Uniform resource locators (url), December 1994. RFC 1738, `ftp://ds.internic.net/rfc/rfc1738.txt`.

[3] P. Flynn. *The WorldWideWeb Handbook.* International Thomson Computer Press, 1995.

[4] P. Gilster. *Finding it on the Internet, Essential Guide to Archie, Veronica, Gopher, WAIS, WWW and Other Search Tools*. Wiley, 1994.

[5] M. Joy and M. Luck. On-line submission and testing of programming assignments. In J. Hart, editor, *Innovations in Computing Teaching*. SEDA, 1995.

[6] M. Luck and M. Joy. Automatic submission in an evolutionary approach to computer science teaching. *Computers and Education*, 25(3):105–111, 1995.