DOCUMENT RESUME

ED 427 715 | IR 019 266

AUTHOR	Lester, Callum R.; Robinson, David A.; Hamilton, Neil M.
TITLE	Using Java and Dynamic HTML To Develop Collaborative, Computer Assisted Learning.
SPONS AGENCY	Scottish Higher Education Funding Council, Edinburgh.
PUB DATE	1998-11-00
NOTE	7p.; In: WebNet 98 World Conference of the WWW, Internet and Intranet Proceedings (3rd, Orlando, FL, November 7-12, 1998); see IR 019 231.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Child Health; *Computer Assisted Instruction; Computer Networks; Computer Simulation; Cooperative Learning; Cooperative Programs; Curriculum Development; Foreign Countries; Higher Education; Instructional Design; Interaction; *Material Development; *Medical Education; Models; *Multimedia Materials; Patients; Programming Languages; *Shared Resources and Services; *World Wide Web
IDENTIFIERS	HTML; Information Infrastructure; Java Programming Language; Scotland

ABSTRACT

The construction of a very fast network backbone between all Scottish universities, in conjunction with the formation of a common core curriculum for Scottish medical schools teaching child health, has provided the schools with an opportunity to provide collaborative, computer assisted learning (CAL) across the World Wide Web, delivered by a central site. CAL, embracing the latest technologies of Java, Dynamic HTML, Javascript and MPEG-1, was successfully developed and implemented in the form of interactive, multimedia Model Patients. This paper discusses: materials and methods used, including the World Wide Web, Java, and Dynamic HTML; student user experiences to date; and implementation issues, including hardware and software requirements, delivery speed, and design. It is concluded that use of the Web as a delivery medium for CAL is both feasible and practical with the provision of high bandwidth networking; collaboration between institutions allows for resources to be shared from one server, leading to efficient use of developers and content contributors and lowering delivery costs and administration needs. (Author/AEF)

Using Java and Dynamic HTML to Develop Collaborative, Computer Assisted Learning

Callum R. Lester
Medical Faculty CAL Unit, University of Aberdeen, Scotland, UK
E-mail: c.lester@abdn.ac.uk

David A. Robinson
Medical Faculty CAL Unit, University of Aberdeen, Scotland, UK
E-mail: d.a.robinson@abdn.ac.uk

Dr Neil M. Hamilton
Medical Faculty CAL Unit, University of Aberdeen, Scotland, UK
E-mail: mil001@abdn.ac.uk

Abstract: The construction of a very fast network backbone between all Scottish universities in conjunction with the formation of a common core curriculum, provided the medical schools with an opportunity to provide collaborative, computer assisted learning across the World Wide Web, delivered by a central site. Thus computer assisted learning, embracing the latest technologies of Java, Dynamic HTML, Javascript and MPEG-1 video, was successfully developed and implemented in the form of interactive, multimedia Model Patients. This paper will present our development, design and delivery issues, what methods we used to address them and report on student user experiences to date.

Introduction

The Child Health Medi-CAL project is the product of the convergence of new pedagogical policy with the emerging Internet technology.

Background

As a result of the General Medical Council (GMC) guidelines [GMC 1993], the medical schools teaching child health in Scotland founded a Common Core Curriculum to enable sharing of teaching resources and materials. The fact that much of the material is relevant to child health training for nurses provided an opportunity to collaborate further.

The GMC guidelines also encouraged the introduction of IT learning resources, which lead to the choice of the World Wide Web (WWW, W3, the Web) as an ideal delivery medium for Computer Assisted Learning (CAL) to supplement the curriculum [Robinson et al. 1998]. This would allow one center to serve all the medical schools in Scotland.

The Scottish Higher Education Funding Council (SHEFC) is committed to supporting the infrastructure for this type of collaboration. Delivery speed for this kind of material is crucial, especially for downloading large video files and to address this SHEFC have funded a very fast 155Mbit/s fiber optic network backbone between the main academic institutions of Scotland. To encourage use of this Metropolitan Area Network (MAN), SHEFC offered funding for projects which would be effective and innovative in their use of the available bandwidth.

The maturation of browsers into sophisticated multimedia presentation devices and the arrival of the WWW programming language Java, and Dynamic Hypertext Markup Language (DHTML) allowed the creation of fully interactive multimedia applications. There was also the advantage of having potentially nothing to install onto the computers to be used. The resource could then be completely centrally administered without requiring to give notice of upgrades, errata and additional applications.

The Aberdeen University CAL Unit has extensive experience in producing software in the form of "Model Patients" [Hamilton et al. 1998]. These applications supply information in a multimedia form with text, sound, images and video, which is also interspersed with questions and decisions allowing the students to interactively manage a patient's illness, a task which cannot be readily achieved in traditional forms of teaching.

The CAL Unit is therefore well placed to tackle the above project by producing similar Model Patients on the Internet using DHTML and Java. It is also able to give clinicians carte blanch to conceive new innovative forms of interactivity to simulate real life decisions and responses which clinicians make.

Summary

We are contracted to construct 30 multimedia Model Patient applications delivered by a WWW server using the latest Internet technology (Java, DHTML, MPEG-1, etc) over a period of eighteen months. These were aimed at undergraduate medical and nursing students.

The scripts and multimedia used to produce the learning material in the applications were provided by clinicians from all of the collaborating institutions.

Materials and Methods

World Wide Web

The need to deliver model patients to all the universities in Scotland, resulted in the WWW becoming the obvious choice as the delivery medium, reinforced by the development of the ultra wide bandwidth offered by the Scottish academic MAN.

WWW is a distributed information retrieval system on the Internet that places the information in multimedia hypertext pages which are downloaded and viewed in web browsers (such as Microsoft Internet Explorer and Netscape Navigator). It has the advantage of being independent of network topologies and client operating systems allowing one version of the applications to serve all universities, no matter what hardware/software their IT strategy implements.

WWW hypertext pages were originally static and passive, only *displaying* information, not allowing any user interaction other than choosing which hyperlink to follow. Although Model Patients could feasibly be developed using this technology alone, the result would be crude, cumbersome, slow to respond and very limited in its diversity of interaction.

The advent of forms and server-side Common Gateway Interface (CGI), scripting using languages such as Perl, offered improved functionality, allowing several kinds of interaction. Unfortunately the user still has to wait for the server to process the choice and respond to the user which, during network congestion, can be slow and frustrating, and can therefore discourage usage. Diversity of interaction is still ultimately limited to what can be implemented using forms.

Java

Around 1995, Sun Microsystems developed a new Object Oriented (OO), distributed, secure, architecture-neutral, interpreted computer programming language called Java. Originally aimed at imbedded programming in consumer devices, it became apparent to Sun that the language was ideally suited for applications on the WWW thereby inserting unlimited interactivity into HTML pages. Application Programming Interfaces (API's) were added to the core library for Graphical User Interfacing (GUI's) and TCP/IP networking. Design constraints had the priority of platform independence and robust built-in security.

Java is therefore a very powerful tool for producing CAL. As a programming language it offered full control of CAL design supplemented by inherent qualities of OO and multi-threading. Small applications called Applets can be imbedded into web pages to provide any form of interaction the developer desires. The GUI allows Human Computer Interfaces (HCI), facilitating user interaction, to be quickly and easily developed. The networking API allows server side communication for centralized auditing and logging.

The CAL Unit immediately embraced this technology and subsequently, the original interactions (multiple-choice questions, multiple-answer questions, true-false questions and interactive image maps) provided by the legacy stand-alone software, were quickly emulated and improved. A modular approach was adopted, creating one applet for each type of interaction. This negated the problem of extensibility, a limitation of the original large software engine, as new types of interaction could be easily added at any time by simply writing a new applet and plugging it in. This would not interfere with the existing software. This approach had the advantage of easy debugging, due to smaller size and complexity, and a new applet could be written without any knowledge of the existing applets, thereby accelerating development.

Applets were designed to be as generic as possible taking, parameters from the HTML pages for such information as the number of answers, the text for each answer and whether it is correct, response for the answer, image map target coordinates etc. By creating custom GUI components with built-in features such as automatic text wrapping, the user interfaces were designed to present varying numbers of answers and lengths of answer text. Object Orientation was extensively exploited and a series of sub-modules (classes) were written and re-used throughout the applets to save code replication and provide a similar look and feel throughout.

Due to Java's properties as a full blown programming language, clinicians writing scripts for the Model Patients have been encouraged to conceive innovative, new forms of student interactivity to further put students into realistic problem solving situations. One idea is allow "multiple outcomes" whereby the possibility for student to misdiagnose or prescribe the wrong treatment is a valid option and the scenario is followed through to its conclusion.

Java is a rapidly developing programming language and since the project started a new version (1.1) has been released by Sun. This version contains several improvements that were thought to be useful to the project. These include:

- A much-improved event model to detect user input etc making lightweight components far easier to write.
- Archiving and compression of class files – Java Archive (JAR) files which decreases the number of requests the client makes to the server and the amount of data to be downloaded, thereby decreasing download time. The perceived performance increase makes the applications much more user friendly and usable by students.
- Just In Time (JIT) compilers dramatically increased the speed of execution of the code. This allows the potential of very complex applets with intensive calculations which could be used for "on the fly" simulations etc.
- The use of encryption and certificates to upgrade Java's security model to allow applets to write to local clients' hard drives which was previously forbidden.
- Java Beans – reusable software components that developers program which can subsequently be manipulated visually using application builders. These could allow rapid application development by non-programmers to extend the applet set. This has yet to be fully investigated.

It was deemed necessary that there should be a dynamic record kept of all pages visited by the student. This ensures that pages, which should be more appropriately visited *after* visiting preceding pages in the Model Patent, could not be accessed without warning. This would be bypassed by a review option. It was feasible to use the new security model to implement this, by updating a log file on the local disk.

However, the additional administration involved in the installing of public encryption keys and certificates in the trust library of every computer requiring to access the applications, prohibited this route. Instead a server-side application was implemented, which communicates with applets imbedded in the web pages, keeping a log file listing the visited pages, on the server for each client. Thus the goal of central administration is preserved.

This additionally provides the user with a history utility which displays the pages previously visited by the student and allows them to directly navigate to each of those pages.

The implication from Sun that future versions of Java would not be compatible with the previous versions to 1.1 led to a decision to write the code to target the latest version. This has tied anyone using the resulting applets to using 4th generation browsers (such as IE 4 and Netscape 4) as previous generations did not have a Virtual Machine (VM) which was 1.1 compliant.

Dynamic HyperText Markup Language

A spin-off from the 4th generation browsers presented itself in the form of a new technology called Dynamic HTML (DHTML) DHTML is a term used to describe the interaction between HTML, Cascading

Style Sheets (CSS) and scripting languages such as Javascript. In essence DHTML builds on the static nature of HTML by allowing authors to specify the exact position and style of elements within a web page. This contrasts with the previous model whereby individual browsers were tasked with information presentation based on a set of guidelines. In addition the web page elements can be further manipulated using scripts. Parts of a page can for example be hidden from the user until either a mouse enters a certain area of the screen or a mouse button is clicked.

DHTML offers far more control over the look and feel of the Model Patients enabling them to seem like custom applications rather than HTML pages in a browser. It has therefore been used to address many design considerations. To standardize the look and allow faster and more intuitive development of the pages, CSS's were implemented to have separate divisions in which text and components were then dynamically formatted, coloured and positioned on the page.

We had to consider the screen resolution that students would be most likely to use. This was determined to be 800x600 pixels. We subsequently decided to target our design of the pages to this resolution and to eliminate any need for scrolling at this selected size. To reduce perceived download time and create emphasis on certain points, specified words are given links to text or image panels that are hidden on the page. When the mouse passes over these words, the panels are then made visible and positioned dynamically. This was found very effective for presenting images, summarizing key points and providing optional or expanded information - all within the desired page resolution.

To allow the pages to look and feel more like a custom application, a navigation/title bar, placed at the top of every page, was implemented. This contained the title of the Model Patient, the copyright and logo of the funding body (SHEFC) and three interactive buttons that offer functionality to go forwards and backwards through the pages as well as link to a help page.

Additionally, each Model Patient application is given a button in a parent page. The result of pressing the button is an invocation of a script that closes the original browser window and opens a new one at the Model Patient's title page. Since browser controls are superfluous and distracting in this case, we removed them from the window, emphasizing the appearance of a native application and offering more white space for the pages themselves. The navigation bar described above provides the necessary controls.

The browser window was also given a fixed, non-customizable size of 800x600 pixels to force the pages to be rendered in the way they were originally designed, completely eliminating the need for scrollbars with the exception of the use of a screen resolution lower than 800x600.

Results

At this time, the testing of the Model Patients has mostly been limited to Windows 95 and Windows NT operating systems using Internet Explorer 4. Due to the fact that the latest version of Netscape Navigator is not yet sufficiently Java 1.1 compliant, platform and browser neutrality will be addressed upon release.

To test the usefulness and usability of the applications, on-going student beta testing was performed. Groups of four students are split into pairs and asked to use the applications with minimum supervision and encouraged to discuss their experience aloud. Relevant comments were duly noted and appraised.

- The feedback was very positive with the students intuitively navigating the model patients with little or no instruction.
- They enjoyed their experience of the software and intimated that they would readily use the resource if made available to them. They were impressed with the quality of the video and the functionality of the Java Applets.
- Criticisms were aimed at the video instructions that did not explain clearly enough, that they could be played before the download completed. This has now been rectified.
- The difficulty level of some of the material was not considered high enough, this will have to be rectified by the contributing clinicians.
- Download time for the pages was of primary importance to the students in terms of usability. It was considered that if response times were slow this would act as a de-motivator to using the applications. However, during the testing, the universal opinion was that the Model Patients were sufficiently responsive.

Discussion

The basis for implementing the 30 applications is now complete and the desired functionality is fully implemented. Model Patient scripts can now be quickly and easily processed into applications.

Hardware and Software Requirements

The implications arising from the requirement of a 4th generation browser have to be confronted. Most of the universities at this time have implemented the Microsoft Windows 3.x operating system for which there is no adequate browser software available. The browsers will require an operating system such as Microsoft Windows 95 or Windows NT. Windows NT is the favored choice due to the benefit of its inherent security, although this does requires a very high level specification of computer. A minimum suggested specification would probably be an Intel Pentium 133MHz processor (or equivalent) and 32 megabytes of memory.

Fortunately all the Scottish institutions are committed to upgrading the computers, available to students, to run the Windows NT Workstation operating system. This will subsequently allow a robust DHTML and Java 1.1 compliant browser to be installed (such as Internet Explorer version 4.x). This is a difficult and expensive undertaking but will have the bonus of pushing forward the level of computing sophistication available to all students.

Delivery Speed

Delivery speed has been targeted as a priority, since unresponsiveness will discourage usage of the resource and thus has been given due consideration throughout the project.

We are collaborating with the IT departments of all the participating institutions to attempt to ensure that all the students have access to computers connected to a fast network, which itself should be connected to the MAN.

The most likely source of delay is the download of large video files. This is partly addressed by MPEG-1 type videos when viewed by Microsoft's media player. The partially downloaded file can be played while the rest if the file is still downloading. This is only useful when the download speed of the date is at least as fast as the data is played. As simultaneously downloading, uncompressing and playing of the video places a significant demand on the CPU, a high specification machine is required.

The video files were placed on a fast video server connected directly to the 155Mbit/s network backbone via a fast Ethernet interface. As some students may only have access to slower network connections, videos may also be sourced from CD-ROM. To reduce the *perceived* download time we are considering placing a relevant task for the student on the pages where the videos are accessed. This would keep the student busy during the transfer of the video data.

The introduction of JAR files, to incorporate the individual Java class files and compress the data to be transmitted, will reduce the number of requests to the server by the browser and improve download time for the applets, thereby increasing the startup speed for the interactive components.

The loading of images into hidden panels imbedded in largely text based pages, appearing when specified words are moused over, increases the perceived responsiveness since the students should be reading the text during the downloading of the pictures.

Design

HCI considerations for the look and feel of the applications have been carefully thought through and the resultant design has been universally popular. The text was considered to be clear and easy to read with no problems for colour-blind people. Use and navigation were considered intuitive with little instruction necessary.

Conclusions

The use of WWW as a delivery medium for CAL has proved to be both feasible and practical with the provision of high bandwidth networking. Collaboration between institutions allows shared resources to be served from one server leading to efficient use of developers and content contributors and lowers delivery costs and administration.

Tools such as Java and DHTML are very powerful and attractive in the production of CAL when the relevant technical expertise is available to the content providers.

DHTML married with Javascript produces browser hosted web pages with look and feel like intuitive, user friendly custom applications. Design and layout are major considerations for the applications to be popular in use.

Java as a full blown programming language with many programming interfaces bolted on, allows an infinite diversity of user interaction which is only limited by the imagination and programming skill of the developer.

We have successfully used these tools to produce CAL applications, which have been well received by students and clinicians alike. CAL on the WWW is a reality now and will continue to improve with the development of the relevant technology.

References

[GMC 1993]. Tomorrow's Doctors. Recommendations on Undergraduate Medical Education, *General Medical Council* (GMC), December (1993).

[Hamilton et al. 1998] Hamilton, N.M., Furnace, J., Duguid, K.P., Helms, P.J. & Simpson, J.G. (In press). Development and integration of CAL: a case study in medicine. *Medical Education*, August (1998).

[Robinson et al. 1998] Robinson, D. A., Lester, C. R., & Hamilton, N. M. (1998). Delivering Computer Assisted Learning across the WWW. *Computer Networks and ISDN Systems*, 30 (1998) 301-307.

Acknowledgements

This work was funded by a grant from The Scottish Higher Education Funding Council (SHEFC) Universities Metropolitan-Area-Network Initiative (UMI) Phase 2. The authors wish to thank the following people, whose help has been invaluable: Iciar Frade, Joanne Hammersley-Riach, Shirley Moore, Wilma Morrison, Tracey Sinclair, Keith Duguid, Prof. Peter Helms, Sandra Martin, John Sangster and Gordon Stables. The authors also wish to thank all of the contributing clinicians, without whom there would be no model patients.

NOTICE

REPRODUCTION BASIS

☒ This document is covered by a signed "Reproduction Release (Blanket) form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").