

Accessible Technology for Networked Learning

Organised by: Chris Jesshope

A Content Management System for the TILE Managed Learning Environment

Chris Jesshope and Zhenzi Zhang

Hull University and Massey University

C.R.Jesshope@dcs.hull.ac.uk & j.zhang@massey.ac.nz

Abstract

This paper will describe the TILE MLE, which comprises a learning content management system and an educational delivery system; it will focus on the former as the delivery framework has been described elsewhere. From a pedagogical point of view, content creation and delivery are the two key factors in web-based learning systems. When supported by the Internet, content delivery are relatively cheap. But creating high quality online learning content is very expensive. Currently different organizations are creating similar content for their own need, with little reuse of content. This is a waste of valuable resources and is the reason why there is an imperative need for standards-compliant content management systems to facilitate content development and deployment. E-learning content management systems face the challenge of collecting, organizing, managing, maintaining, re-using, delivering and targeting the content. We must differentiate between a content-management system and authoring tools for content creation. The latter are used to create content that is organised into a course. This may be animations, graphics, text, audio, video or other multimedia segments. These learning objects are organised and catalogued by the learning content management system and this creates units of study or on-line courses that can be navigated and perhaps monitored. That unit of study usually has structure, which may include hierarchy and precedence. An LCMS should support the management of any content level, which includes multimedia segments, learning objects or units of study. The design of such a system is described here, together with an analysis of the issues considered.

Introduction

There is currently a rapid uptake in is e-learning. E-Learning enables the learner to learn without the restriction of time and distance. It can increase the accessibility of traditional education, increase the learning efficiency and facilitate collaboration. There are already many existing systems in this market, such as Blackboard and WebCT, but there is still a large body of research being undertaken in developing next generation systems. Much of this research is in the arena of defining common standards[refs], aimed primarily at cataloguing and the reuse of educational material. There are also developments, such as the TILE managed learning environment (MLE), Gehne et. al. (2001), being undertaken, which have enhanced the traditional, thin-client web architecture, to provide flexibility and scalability in the delivery of educational material, no matter where the student is or what internet connection they have (or not!). This paper looks at the design requirements for an authoring tool for TILE.

We will investigate both the technological constraints and the various users' needs in designing such a tool. We have considered a variety of standards for the description of both on-line course packages as well as student models. We also consider issues such as interfaces to other standard authoring tools, e.g. web editors, multimedia packages etc. The output of these third-party tools must also be integrated into the course structure as learning objects and so we must develop open standards for that integration. This is not

so easy, as TILE is designed to support student learning models and hence must be able to track the student's progress. The TILE delivery architecture includes a server on the student's computer to achieve this even when the student is browsing bandwidth unfriendly material off-line, from CD say. The problem we face in authoring, is that TILE may not be aware of student actions if the media produced by third-party tools contains choice in the form of hyperlinks. We propose a solution to this situation in this paper. We also describe a version control system that enables multiple authors to collaborate on course development. Also using a hierarchy of capabilities, we enable material, once placed in a repository and if licenses and permissions allow, to be reused and re-annotated within other courses. This paper describes the schema and techniques to maintain a coherent, structured corpus of educational material.

E-Learning Systems Requirements

Generally, an e-learning system comprises three key components, they are infrastructure, services and content Lennox (2001). *Infrastructure* is the software that allows learning to be created, managed, delivered and measured. It can be divided into a Learning Management System (LMS) and a Learning Content Management System (LCMS). *Services* involve the planning, customization, integration and management of the e-learning application. *Content* can be categorized according to subject, preferred format, student's progress and language requirements. The content's origin might be off-the shelf, customized or custom designed by a lecturer. E-learning system's infrastructure, services and content are complementary to each other. Content is the core of LMS and LCMS. And most services are delivered through the LMS and authored in the LCMS.

First we look at the general requirements for an e-learning system and identify how the TILE system meets these needs, a good starting point is as proposed by Singh (2000):

Accessibility. Knowledge can be access at anytime and anywhere.

Flexibility. The e-learning environment can be customized to an organization's needs.

Extensibility. The system must allow for additional components to be integrated easily.

Reusability. Content can be reused by creators or consumers

Interoperability. The system should allow content and other data to be exchanged and shared by separate tools and.

Scalability. The system should permit access to potentially hundreds of thousands of users and large content repositories.

Security. The security of data, information, or knowledge should be promise in the system.

Standards compliance.

TILE meets the first of these requirements, though the use of web-based technology, and the second and third through the use of a data-base application and servlets that communicate with it. Points 4 and 5 are achieved by dividing the course into structural components that are held in the data-base and learning objects that are created in an open environment and linked using the authoring tool described here. Of course the latter also requires the us satisfy 8, by using meta-data standards, (Rover, 2002) for describing the course and possibly other standards for packaging the material so that it can be migrated between different systems, (IMS 2001). Finally, if a student learning model is used, then there must be standards to describe this model and how the course material relates to it (EML 2001). TILE provides a unique solution to point number 6 by distributing a part of the server's functionality onto the student's computer. Thus the more students that use the system, the more computing power is brought into play. Although this sounds simple, it actually introduces a non-trivial, distributed database synchronisation problem and many security issues, as described in Zhang (2001).

In addition to the above features we believe there is also a requirement for strict version control, to enable many users to concurrently author a single course unit or parts thereof. This should allow for multiple versions, change audits, rollback to earlier versions and purging of old versions. This is similar to good software engineering practices in source code control systems..

The division of labour in e-learning

An LCMS focuses on content creation, reuse and management, and can says Lennox (2001), compress the lifecycle of capturing, delivering, managing and measuring knowledge and learning reuse content in many different ways. Content can be selected and used at very granular level. The general goal of course is to provide a lower content maintenance cost. An LMS, on the other hand focuses on delivery, learning activities and student competency management. They have three common areas: content, user and management (Rengarajan, 2001). Integration of LMS and LCMS together has the advantage of sharing a common content repository and unified

schema, and this has always been the aim of the TILE system.

The TILE LCMS

TILE is an integrated system for the management, authoring, delivery and monitoring of education at a distance. TILE is different from current web-based e-learning systems, as TILE aims to provide more accessibility, flexibility and scalability to the various users (institution administrators, lecturers and students). It has an unique architecture which allows users to access the LMS no matter they have internet connection or not (Gehne 2001).

Before we introduce the TILE LCMS, several concepts need to be defined. First we define a *Learning Object* to be a self-contained piece of educational material that contains content and/or assessment based on specific learning objective. We also define the *Learning content* as the material that is used to convey subject matter; it may include raw media elements such as text, graphics, audio, some form of interaction etc. We define *Metadata* is the data which is used to describe a learning objects, an example is the LOM model (IEEE 2002). An LCMS uses metadata to organize, search, reuse or to protect learning objects and contents. A *Course unit* is defined as a component of a course, possibly the whole course that comprises a set of structured learning objects and where both the learning objects or the nodes in the structure may have metadata associated with them. This structure may provide hierarchy, for example as an index to the material, or it may provide precedence, such as a set of prerequisites. TILE supports both of these.

The TILE LCMS does or will implement the following features:

Provide one or more repositories to store learning content.

Maintain the structure, metadata and learning objects as separate entities. The former two are held as relations in a database and the latter are treated as standard html objects and referenced by URLs.

Provide three level of re-usability, e.g. course unit, learning object and raw media material can and should all be re-usable.

The system should allow the author to create learning content and raw media elements in an open environment and not constrain the choice of tools used.

A version control system and protocols to allow collaboration of multiple authors on a single course unit. The system will keep track of all versions and who was responsible for any changes until old versions are purged.

Authorization is imposed by course unit, at any level. Only authorized users can update the learning objects or metadata.

Intellectual property protection features will be provided to allow the author to decide who can re-use the content and there is support for communication between the authors and the re-users.

In addition to metadata, personalized labels can be applied to course units and learning objects to facilitate searching.

The repository of LMS is a subset of the LCMS repository. Only one version of an LCMS course unit and learning objects will be visible to the learners, the published version. The TILE delivery system already supports versioning and automatic synchronisation with newly published versions of course units and learning objects.

We will support a range of standards for importing and exporting compliant learning content, e.g. SCORM (Rover 2002), IMS (IMS 2001) and possibly EML (EML 2001)

The problem with standards is that there is not just one. There are many organizations developing learning standards, these include Institute of Electrical and Electronics Engineering (IEEE 2002), the Advanced Distributed Learning Network (ADLNet), the Aviation Industry CBT Committee(AICC) and the Instructional Management System Global Learning Consortium (IMS). The Sharable Content Object Reference Model (SCORM) is a set of interrelated technical specifications built upon the work of the AICC, IMS and IEEE to create one unified content model (Rover 2001). The US federal government has already announced that any e-learning product it uses must be SCORM-compliant. That is just one factor that is driving the acceptance of SCORM as a de facto standard for e-learning (Brennan 2001). This new standard however, does not fully support student learning models, course unit

personalisation and how the learning models interact with the content. Figure 1. Shows a conceptual model, taken from a presentation on EML by Jocelyn Manderveld of the Open University of the Netherlands. EML is one of the developing standards for modeling this kind of interaction in courseware.

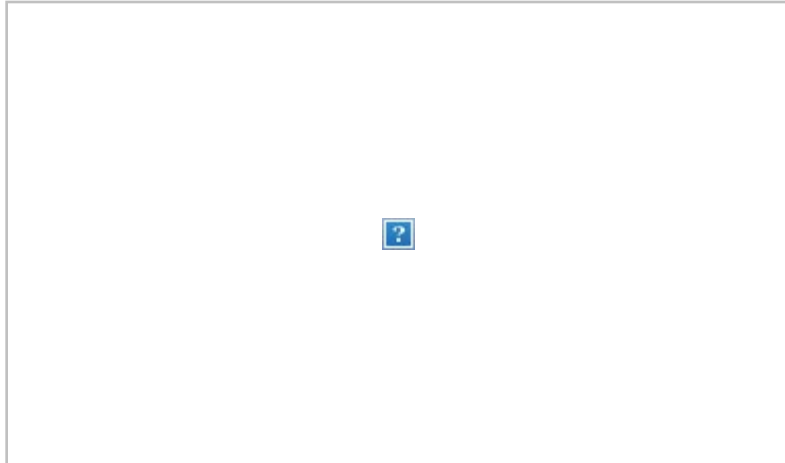


Figure 1. The EML conceptual learning model.

EML is not the only development in this area but it is probably the most complete. Its XML binding, a technical manual and further details can all be found at the EML web site (EML 2001). Others standards in this area include the Tutorial Markup Language (TML), which was developed by the University of Bristol, UK, and is described in

<http://www.ilrt.bris.ac.uk/mru/netquest/tml/>.

TML is an interchange format designed to separate the semantic content of a question from its screen layout or formatting. The language is designed to support several different types of question within the same content

model and is essentially a super-set of HTML. Another is the Learning Material Markup Language (LMML). LMML is an implementation of the XML binding of the teachware-specific meta-model described in (Süss 2000). LMML is extensible and it therefore represents a family of various languages. There is a short introduction in the LMML tutorial at <http://daisy.fmi.uni-passau.de/pakmas/lmml/11/doc/en/html/>. These are just a few of the ongoing developments. It should be noted that because SCORM is a set of interrelated standards, developments such as EML may yet be incorporated into it.

TILE LCMS Architecture

TILE LCMS architecture is illustrated in Figure 2. It shows that on the education providers computer(s) the TILE LCMS comprises a LCMS server and an LMS server and labeled versions of courses are published from one to the other. Each has its own repository for storing course units and learning objects. The TILE LCMS client is the authoring tool described in this paper and it is a Java application on the lecturer or teacher's computer. It is used to edit and bind raw media segments, learning contents, learning objects and course units locally, even while off-line. If the LCMS user logs into the server, they can browse all the course information held in the repository and retrieve information to be held locally, either for updating, or for inclusion into other course units, either as is or in modified form. The user can browse any version of course unit structure, check out course units or learning objects (or both), check in new course unit, check in edited course unit that the user previously checked out, import or export a course unit that is compatible to SCORM standard.

Security is clearly important and the TILE LCMS server is in charge of the end user's authentication and authorization, managing and delivering learning contents. For example, all material in the development database has capabilities. Controllorship can be allocated by administrators to course units and can then be delegated to others, so that at any point in the tree there will be various permissions on who can access and modify that material.

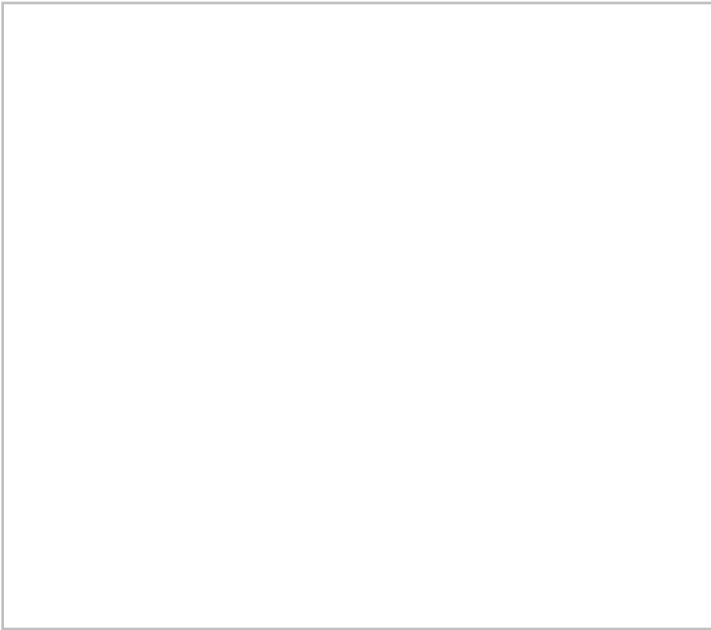


Figure 2 TILE LCMS architecture

TILE has two databases on the education providers computer(s), one is the LCMS or development repository, the other is LMS or publication repository. A history of all versions of course units, learning object and learning content are kept in the LCMS database. Only one version of any course unit will be published to the LMS database.

Design of TILE LCMS

User roles

There are three user roles in the TILE LCMS system. They are *course administrator*, *course controller* and *reviewer*. A course administrator is able to set up new courses, and specify course controller. A course controller is the person who has the responsibility for the course unit, e.g. author or maintainer of the specific course unit. A course reviewer is the user who can access the LCMS but has no responsibility with the specific course unit. A reviewer for example will be able to reuse material but will not be able to modify it, although he will be able to modify some part of the corpus of material so that he can copy structure there. Course units of TILE LCMS are organized in a tree like structure. Learning objects are located in the leaves of the tree. Sharing is achieved by copying the structure of a course unit but by sharing the learning objects. The data requirements of the structure are small compared to the learning objects, which may be multimedia. Also reuse may require re-annotating nodes with different meta data for different purposes. All copying of course units is deep, i.e. a node will always be copied with all of its sub-structure right down to, but not including, the learning objects.

Authority control applies to every node of the structure tree. Access authority propagates down the tree, i.e. if the user is the controller of one node, he is the controller of all the descendent nodes. A controller of one node can also add other controllers to its descendent nodes.

This division of user roles helps to control the integrity of the course. At the same time, the mechanism of applying authority control to every node of the course structure increases the flexibility of concurrent, multi-person development of the course.

Check in and check out mechanism

TILE LCMS uses version control method to manage learning contents. The system allows a controller to check-out and check-in course units. While course reviewer can only check out a course unit for reference or re-use. For example a controller may develop a complete course locally and then check it in to the repository. The controller may also check out that course unit, edit it and check it back in again. Learning objects may be shared between courses and it is possible to author without creating any content at all by simply reusing existing learning objects. In this case only structure will be created and even this structure may be a copy of structure found in another course. During the editing process, as material is checked-out and back in again, the server database will keep track of all changes made to course material with an audit trail to identify who has made what change and to what data and when. The implication of this requirement is that the metadata for a course structure being developed may contain multiple instances of a node if that node has been changed. The system must therefore keep track of a version number, associated with each node, automatically

updates it whenever a change is made.

Since one course unit might have multiple controllers, a locking mechanism is used to ensure a consistent update. Checking out is divided into two types: lock then check out and check out without locking. Any LCMS user may check out a course unit without locking. But only unit controller can lock the unit and then check out. Locking a unit implied locking all sub-nodes below it. After that, only the single controller holding the lock will be able to perform a check-in. The protocol for checking-in, locking and checking-out is as follows:

Any TILE LCMS user may check out a unit of a given course.

Only one of the course controllers for a unit node may lock that node.

The lock is propagated down the structure tree but not to the learning objects. A separate locking mechanism is provided for the learning objects

Only one controller can hold a lock for node at any given time. Any other controller attempting to lock this entry will be notified the identification of the person currently holding the lock.

Only the lock holder may check in updated contents.

An author, who has changed some checked-out structure that is subsequently updated by someone else, by locking and checking-in, is responsible for checking out the new material and transferring their changes to the new structure. Warnings will be given by email to anybody who has structure checked out that is updated by somebody else checking in.

Labeling

The TILE LCMS repository keeps multiple version for each course structure node. By default the user will only see the latest version in the sequence of changes, although an interface will be provided to view previous versions. The system supports storing author-defined labels for given versions of the structure node. This label is used for defining an author-specified version of the course; it may be a development version, it may be a test version or it may be a version that has been identified for publication to the course delivery system.

Open Environment for Learning Content Authoring

The TILE LCMS client does not put any restriction on what kind of learning content and raw media segment authoring tool the user should use. The user can choose their favorite tool and only need to let the TILE LCMS client know the tool when it is used for the first time. After that the system will automatically launch the authoring tool for the user, whenever they wish to edit the raw media or learning objects that has been created with it. There must also be a mechanism for uploading any files involved, this may include both source and raw media data. If an application is not scriptable, the user will have to locate the information locally for the LCMS client to store it in the repository, when it is checked in.

Course Structure and Learning Object Authoring Tool

Course units are organised into a tree like structure, with the root node being a that unit that is identified by the learning institution as a component of its programs. It is at this level that the administrator passes over control to the unit controller. Course units below this root node are created by the course controller to structure the course pedagogically or administratively, for example the unit may be shared between many instructors, each having control over their own sub-units and each having permission to edit their own unit. In this way a course unit is recursively defined.

There may also be precedence relationships established between the various units, even though they not be a part of the same root unit. These relationships form a set of prerequisite constraints and may be used to control the capabilities of students to view material.

Metadata is used in TILE for indexing, delivery, management and re-use purposes. It is used to describe both course units and the learning objects. The authoring application and the delivery client both use metadata searches for locating content.

Course structures are managed separately from the learning objects. This allows the user to insert and remove learning objects to/from a course unit's structure quite freely. It also means that a learning object can be shared by many course structures without duplicating the learning object. Indeed a learning object may be an external URL. The update of a learning object therefore, will be visible in all the courses that reference it.

The course structure and metadata in the TILE LCMS are SCORM compatible.

XML and SCORM and Cross-platform issues

When the TILE LCMS user constructs a new course structure, they will first construct it locally on their own computer's disc. Alternatively the user may check out an existing structure from the LCMS server, in this case, all the related information needed is downloaded from the LCMS server and also stored locally for browsing and updating. XML files are used on the client side to manage the course structure. This decision allows a standard application to access the data without the requirement for installing a database server. The XML data structure is, in any case, the preferred method of communicating the data between the LCMS server and the authoring application. XMLs characteristics of being totally text based and conforming to a well established standard provide simplicity in dealing with both firewall and cross platform issues. SCORM also uses an XML course structure format to move a course from one LMS to another. The XML file will be compliant with the SCORM XML DTD.

The TILE LCMS is completely implemented in Java. So it can be supported across many platforms. Careful design of the system has also made it firewall friendly.. Communication between the client and the server will use HTTP protocol, which may embed XML data. Because most firewalls pass traffic on port 80, which is used for the http protocol, then this allows a means by which a client server application can work through the firewall.

Conclusion

We have described a learning content management system to be used in conjunction with the TILE learning management system. This paper has described the general requirements for such a system, borrowing from the area of software development we have added further issues to those generally regarded as being desirable in such a system, such as version control and concurrent development of material. We have described our planned implementation and described standards issues, which will determine the schemas used. A prototype implementation of the TILE Authoring client has already been undertaken (Wang 2002).

ACKNOWLEDGEMENTS

We gratefully acknowledge the support from the New Zealand New Economy Research Fund, NERF, for support of the Technology Integrated Learning Environments, TILE, project.

References

- EML (2001) EML downloads, <http://eml.ou.nl/>
- Gehne, R., Jesshope, C.R. and Zhang, J. (2001), Technology Integrated Learning Environment - A Web-based Distance Learning System. Proceedings of IASTED International Conference 2001, Internet and Multimedia Systems and Applications. Hawaii, USA. ISBN 0-88986-299-0. pp1-6.
- IEEE (2002) IEEE Learning Technology Standards Committee (LTSC) IEEE P1484.12 Learning Object Metadata Working Group home page, <http://ltsc.ieee.org/wg12/>
- IMS (2001) IMS specifications, <http://www.imsproject.org/specifications.html>
- Koolen, R (2001), Knowledge Mechanics , Learning Content Management System; the 2nd Wave of e-Learning!, <http://www.internetttime.com/itimegroup/lcms/lcms2ndwave.pdf>
- Lennox, D (2001) , Managing Knowledge with Learning Objects, The Role of an e-Learning Content Management System in Speeding Time to Performance, http://www.internetttime.com/itimegroup/lcms/wbt_Mngknw.pdf
- Raghavan R (2001) , LCMS and LMS, Taking Advantage of Tight Integration, http://home.click2learn.com/en/downloads/lcms_and_lms.pdf
- Rosenberg, M (2001), E-learning Basics: A guide to the e-learning industry, <http://www.elearningmag.com/elearning/article/articleDetail.jsp?id=2779> retrieved date January 26, 2002.
- Rover, R (2002), Shareable Content Object Reference Model Initiative (SCORM),

<http://xml.coverpages.org/scorm.html>

Ryann K. Ellis (2001), LCMS

Roundup<http://www.learningcircuits.org/2001/aug2001/ttools.html>

Singh, H (2000), Achieving Interoperability in e-Learning, <http://www.learningcircuits.org/mar2000/singh.html>

Süss, C (2000) A Meta-Modeling Adaptive Knowledge Management: Approach and its Binding to XML (2000) (<http://daisy.fmi.uni-passau.de/db/literatur.php3?key=S00>)

Wang, Y (2002) *An authoring tool for structuring and annotating on-line educational courses*, M.Sc. Thesis, Massey University.

Zhang, Z (2001) *A Feasibility Study for the Design of a Web-based Course Delivery System*, M.Sc. Thesis, Massey University.

Interactive Multimedia for Dummies

Regina Gehne and Chris Jesshope

Hull University

r.gehne@dcs.hull.ac.uk and c.r.jesshope@dcs.hull.ac.uk

ABSTRACT

This paper will describe some significant enhancements to a multimedia, authoring tool that has been developed for use by lecturers and teachers, rather than multimedia professionals. The tool uses a novel paradigm that eliminates time and hence any synchronisation in the development and editing of multimedia documents. Instead it uses the notion of a strict sequence of media elements. The media is packaged and compressed and is delivered by a browser plug-in to users. The media can be streamed over low-bandwidth modems making it suitable for delivery in any environment. Feedback from a large base of users of the existing tool has led us to develop a new version of this software that adds new capabilities, while maintaining the goals of a broad user base and an easy-to-use interface. The new capabilities include the introduction of interactive media and the incorporation of a richer set of media elements. The original tool produced linear presentations which are rather like a video on playback and we have added a hyperlink capability. The design and implementation of the interactive media has been a big challenge and the constraints that we had to meet were a requirement to maintain the streaming property of our presentations, in the presence of user choice, and to maintain the ease of use of the tools. The paper will discuss the choices made in our design. It will discuss the user interfaces and the presentation of the now more complex multimedia presentation and it will demonstrate the use of the tools and give examples of the new pedagogical techniques that can be used with the new tool.

Keywords

Multimedia authoring, interactive multimedia, low-bandwidth streaming, easy-to-use user interface

INTRODUCTION

The AudioGraph toolset has been developed to enable on-line teaching by providing the equivalent of face-to-face lectures as web-based multimedia documents. The goal of this project has been two-fold. To develop the tools and to experiment and evaluate their use in on-line teaching scenarios in a conventional university environment. That goal has been achieved and there are a number of publications on both our own and others' research using this tool, for example: Segal, (1997), Pearson and Jesshope (1988) and Jesshope(1999, 2000a, 2000b, 2001). This paper focuses on further developments to these tools based on our own evaluation and also feedback that we have had from upwards of 1000 registered users of the tool.

The AudioGraph software itself is the result of some 5 years of research on three campuses, Surrey University, Hull University and Massey University, see: Jesshope and Shafarenko (1997), Jesshope Shafarenko and Slusanschi (1998), Jesshope (1999), and Gehne and Jesshope (2000). AudioGraph can be downloaded from the NZEdSoft web site (<http://www.nzsoft.com/>) and the tools are available free of charge to anybody wishing to use them.

The methodology of teaching that the tools supports is very similar to what has been named Just in Time Teaching (JITT), by Novak and Patterson (1998). We also believe that the use of on-line multimedia in education not only liberates the students from time and geographical location, but also addresses issues of learning style. A growing body of literature in the impact of learning styles, e.g. Montgomery. (1998) and Felder (1987,1993), as well as end-user feedback, has driven the developments described in this paper. A summary of the results from Montgomery is still highly relevant and is reproduced below.

67%of the students learn best actively, yet lectures are typically passive;

57%of the students are sensors, yet we teach them intuitively;

69%of the students are visual, yet lectures are primarily verbal;

28%of the students are global, yet we seldom focus on the ``big picture."

Multimedia, especially interactive multimedia can overcome these barriers.

The AudioGraph tools

First we will give a general introduction to the AudioGraph tools, which comprises two pieces of software, an authoring tool and a player, which is a plug-in that enhances a web browser's functionality, enabling it to play the AudioGraph presentations. The authoring tool, the AudioGraph Recorder, is used for producing the multimedia content, which is a web-ready recreation of a teaching session. Unlike some other tools, such as Tegrity (<http://www.tegrity.com/>), AudioGraph records the presentations in the teacher's own time and not by capturing a live class. The results are very similar but usually more polished. The AudioGraph Recorder is available on both Macintosh and Windows platforms, but the authoring tool for the latter, is still at version 1 whereas the Macintosh authoring tool has been enhanced as described in this paper.

The key feature of the tools is their ease of use. They have been designed from the outset to be simple, with a clear and intuitive interface. This has meant reducing the number of concepts that the user is confronted with and of course this limits the capabilities of the tool, when compared to tools that are used by multimedia professionals. Thus AudioGraph, unlike other tools, can be learnt in an afternoon and requires little time to produce professional looking multimedia web sites. Typical preparation to presentation time vary from 2:1 to perhaps 10:1 for a complex, animated presentation(ref). This compares very favourably with professional authoring ratios, which are between 100:1 to 200:1.

AudioGraph makes use of a range of media elements, such as images, direct voice recording, vector graphics and pen annotation. These tools provide analogues of tools used in face-to-face education and hence the teacher immediately feels at home with the use of the tool. Images replace the slides used in a face-to-face class; the vector graphics and pen input the various drawing devices, such as whiteboard, blackboard, flipchart etc. and of course, spoken voice is the essence of a face-to-face presentation. Video is not supported as we believe it to be poor pedagogical value for the bandwidth required. Still or quasi-moving images can be used at the author's discretion but do increase storage and download requirements.

A key issue in the debate of enabling the lecturer with this technology, is that the experts who would otherwise produce the multimedia are often too far removed from the teaching area, as well as good teaching practice. This results in professional CDs or web sites that favour gloss, such as animation for its own sake, rather than sound educational content and the use good pedagogical techniques. If the teacher and media expert work closely together they can ameliorate this problem but this only increases the cost of the educational outputs, as there are now two people working on the production, an educator and a multimedia specialist. Again the cost of the multimedia becomes prohibitively expensive for all but the largest of audiences.

Another requirement of this project was that the media should occupy only a small space on the web server and, more importantly, be accessible to the students by modem.

AudioGraph has been used in a variety of ways and some examples of its use are illustrated below:

- the presentation of on-line material to both internal and extramural university students;
- the facilitation of on-line training of equipment and software;
- the replacement of video instructional courses;
- as a means of asynchronous technical communication in virtual organisations;
- as a means by which school children can present their study to their peers and to their teachers.
- as a means of teacher evaluation; and
- as a means of sending electronic greetings.

The AudioGraph Principle and Realisation

This tool is based on what we call the AudioGraph principle, which states that the media elements be arranged as an ordered collection and are played in strict sequence, regardless of playback timing. This is a simplifying principle that effects both authoring and playback. In authoring, there is no concept of time and synchronisation to worry about and in playback the presentation quality is independent of download speed and speed of the computer. No loss of synchronisation is ever seen, even in the presence of a slow internet connection or very old computers.

Normally, complex programming over time is required when using a multimedia-authoring tool. There are tools that provide an exception to this, but these can not really considered to be multimedia authoring tools. For example PowerPoint allows different media elements to be placed on slides and these may be output to make web presentations. However, PowerPoint provides no functions or display for the sequencing or editing of these media components.

In the AudioGraph this is not the case, we provide a window, which controls precisely the sequence of components. Each media component is represented as an icon and the sequence of icons is displayed in a window called the *Edit console*, which is shown in figure 1. This is linked to the display window so that when an icon is selected in the control panel, everything up to and including that media element in time is displayed. Thus the *Slide window* can easily display different snap-shots of the presentation at different stages of its progress. The Edit console represents the sequence in time, from left to right. You can see the tape-recorder-style controls in the bottom left corner of this window for previewing the presentation. The slide window on the other hand, shows exactly what the learner will see in the presentation when viewed in the web browser. Thus the unique difference between the AudioGraph recorder and other presentation packages, is the ability to see and edit any slide in the presentation at any point in its delivery, between its start and end.

IMPLementing Interactive Multimedia

This paper is primarily concerned with how we have implemented interactive media in the AudioGraph. Current AudioGraph lectures can be interactive, but only because of the environment that they are presented in. AudioGraph lectures comprise html pages with the AudioGraph slides embedded within the web pages as links. The AudioGraph slide itself is a strict sequence of media elements that has a start and a finish. The standard output from the tool is a single index page, with one link to each slide in the presentation. The problem with this is that to generate anything more than a linear sequence of material requires the author to also edit the html pages, which is possible but adds to the design time.

Our goal therefore was to create AudioGraph presentations with non-linear characteristics, so that with different input from the person viewing it, the output would follow different paths through the presentation. This means that there are certain choice points and that at these points there are links to different parts of the material, which will depend on the user's input. There are a number of design issues in creating a tool that would do this. A number of questions have to be asked:

what is it that is being linked to? Html links are to pages or to anchors within a page.

When and how should the links be active? Remember a multimedia presentation has a time element.

How can we deal with streaming in the presence of the user's choice?

Finally, how are these links represented and displayed in the authoring tool?

Editing and file model for links

In AudioGraph a presentation is a set of linked files, in version 1 the linking was in html. Each file is a sequential presentation, which was played by the AudioGraph plug-in. Similarly, each episode or slide in the presentation is an editable unit represented by the combination of edit console and slide window, see figure 1. On export to the web, each slide produces one .html file and one embedded .aep file of MIME type application/vnd.audiograph. It was important that we did not move too far away from our current editing model, which is simple and intuitive. Keeping this same file and editing model therefore was our first approach and we found that it provided answers to both points 1 and 3 above. That is links would be to episodes only and a users choices selects the next file to stream to the them.

In version 2 therefore we maintain the episode as the unit of presentation and an episode is still sequential. What we add are links in the presentation and those links can be to any other episode (including itself), but only to the start of those episodes. The user's choice by mouse clicking on a linked area in the presentation, will determine which of potentially many links are taken. Streaming is solved because there are no non-linear paths within a presentation, that are non-deterministically chosen. The presentation is linear and has choice points anywhere within it that exit that episode and link to another.

In fact a linked presentation with AudioGraph links is still a linear presentation and is shown in the plug-ins progress bar as such. It is more accurately the union of a set of non-deterministically chosen episodes, committed by the user's choice. It is also dynamic, in that the user can go back and re-commit the choice and produce a completely different linear sequence. The is only wasted bandwidth when a choice point is made early in an episode and the user exits before of it is all downloaded. This can be avoided however, by splitting the episode at the choice point. The linking strategy for both version 1 and version 2 links is illustrated schematically in figure 2. Notice that in the new version we have a much richer potential for linking and moreover it can all be completed within a single tool, the AudioGraph recorder. Notice that a link may be to itself, and may be taken automatically, which allows for loops in presentations. A model of an AudioGraph presentation is therefore a directed graph (which may contain loops) of AudioGraph episodes, which are the nodes in the graph and choice is made in each episode by selecting one of the arcs that starts at that node.

Authoring model for links

By design, therefore there is no significant change in the authoring model, as each episode is still sequential and is represented by the combination of Edit console and Slide window. To answer questions 2 and 4 above however, we need to consider more carefully what a link is. To that end we define a link to be an attribute of any graphical object, or collection of objects that have been grouped together. The activation area of the link is the drawn area of the object or the union of all of the drawn areas of all objects in a group. The default link tool creates a transparent rectangle and allows the author to add a link to it; this has no visual representation on the screen but simply defines an area, which activates the link. The link itself can be either another AudioGraph episode within the same presentation, or a standard URL, which could be an external AudioGraph episode or anything that can be represented in an html page. Links may be opened in the same window, in which case the episodes are concatenated together, or can be opened in a new window, rather like a pop-up window. This provides for a number of different pedagogical styles of authoring. Pop-up presentation can be used, for example to give more detail on a topic and links in the same window can be used to provide alternate pathways based on user preference or ability, as testing may be an element of the choice.

An html page which has no sequencer, without scripting and hence links are active as soon as the page is displayed. AudioGraph presentations however, have a default sequencer built into the plug-in. The issue of when and how links are activated must therefore be considered. We decided that links would only be activated when the presentations is stopped. The presentation can be stopped by clicking anywhere on the screen or the start stop button. When stopped, a link's active area is shown a change of cursor (it becomes a finger pointer) and mouse clicking on it will continue the presentation with the episode it links to. Mouse clicking in a non-active area will restart the presentation; at the beginning, if the presentation has already reached its end. We have also introduced a stop tool, whose action on playback is to stop the presentation and hence activate any links that may have been placed at that point in the presentation.

details of Audiograph recorder version 2

Links display at the Lecture level

The AudioGraph recorder has a lecture window, which provides a view of the complete presentation. Two views are available in version 1, a text view and a thumbnail view, the former giving a textual view of what will be presented on the index page, complete with episode durations and the latter giving a graphical index to the different slides. In version 2, a "Links" view has been added to the Lecture Window. This currently just displays the links between episodes as a matrix of directed arrows, although our intention has been to provide link destination editing from this view as well. This will be added later. The links view of the Lecture window is shown in figure 3, it is now a resizable window.

The colour coding of the cells is to help seeing which link belongs to which slide. Slide titles and links list can be independently scrolled horizontally. If the slide links to an URL, the URL gets displayed when the cursor is over the URL field, as illustrated.

Finally the arrows in the index column to the left of the slide titles indicate which slides should be placed on the HTML output page. In this case there will be just two entries, as the other episodes are all reachable by AudioGraph links.

Text Tool

In version 1, text could only be created by copying and pasting from another application. Version 2 enhances the media AudioGraph produces by providing text component. It is a hybrid between a rectangle and an image. It has the visual attributes of a rectangle, i.e. colour, line width, transparency, filled/unfilled etc., but gets drawn and erased in the background or image layer, whereas rectangles are drawn and erased in the foreground or annotation layer. Additionally, the text itself has its own attributes like size, font style, colour and justification. Text font, size, style, colour and justification within its rectangle (left, centred, right) can be edited in the Visual Attributes Editing Dialogue.

Text gets exported as a PNG image for display by the plug-in. This solution is chosen to mitigate any problems with a user's installed fonts. As a bitmap, the text is displayed and aligned exactly as it is in the Recorder, regardless of whether the fonts used for creating the text are present on the viewer's machine or not. The only other way to ensure this alignment would be to export fonts, which is expensive in file size even with scalable fonts. PNG compression does a good job of optimising the image files as we support the full standard, which supports bit depths, from 1 to 32 bits, with transparency in both indexed and alpha channel images.

On the Macintosh, there is an option in the Appearance Control Panel to turn anti-aliasing on for fonts bigger than a specified size. This gives the letters in a font a smoother appearance by gradually blending the text colour with the background colour. For big fonts this makes a real difference to the text. We have a similar option in the Visual Attributes Editing Dialogue now, called "Smooth text drawing", which turns anti-aliasing on before drawing, no matter if this is turned on in the Appearance Control Panel. The catch is that this results in bigger exported text, because the PNG image will need to store more transparency information. A non anti-aliased text component requires at most 2 bits per pixel, this represents black, white, the text colour and background colour. An anti-aliased text component would have a requirement for perhaps 16 levels of transparency in the colour index table for the background colour, to create the smooth effect on the jagged edges, requiring at least 5 bits to represent each pixel and more than doubling the file size.

To get a feeling for file sizes, a typical bullet point on a slide requires about 2.5 KBytes for non aliased text and 6.5Kbytes for anti-aliased text, which is just seconds of download time on a modem. For comparison, a minute of speech requires 100Kbyte.

Both images and text can be persistent across slide boundaries during a link and again this feature is implemented in order to minimise download time. A large detailed image may require up to 100KBytes, This can also be switched on in the Visual Attributes Editing Dialogue.

Recording Sound

Sound quality

The sound recording interface and algorithms have been changed. Version 1 of AudioGraph used only GSM compressed sound for exported presentations and only PCM encoded sound in the lecture document. GSM requires 13 Kbps or about 100Kbytes per minute of recorded sound. PCM requires much more (depending on the sampling rate). For example at 44KHz, a mono recording requires 2.6 Mbytes per minute of recorded sound. This feature had two negative effects:

File sizes were very large for lecture documents

Exporting presentations was very slow because every sound component had to be converted from PCM to GSM, every time a file was exported.

This has been changed in version 2 and it is now possible to set the export quality and choose speech quality (GSM), music quality (Ogg) or linear samples (PCM). This setting will determine how the sound is stored in both the lecture document and the exported presentation. In the first release of version 2, exported presentations will only support speech quality sound but the lecture document will support linear samples and speech quality.

Sound qualities supported or to be supported

PCM quality

PCM stands for pulse code modulation. The sound is sampled at a specified rate, say 44 thousand times a second, which is CD quality. What is stored then is one sample for each sampling period. The samples are normally 2 bytes each. This is called PCM. PCM is currently supported in the lecture document and will soon be supported in the exported presentations.

Music quality

There are a number of music quality compression algorithms, perhaps the most popular is MP3. These compression techniques use the human hearing characteristics to remove redundant information from the data. Although not yet implemented we intend to implement an Ogg-Vorbis compression scheme (<http://www.ogg-vorbis.com/>). This supports a variable-rate, music-quality compression of sound. At high stream rates 64 to 128Kbps there is little qualitative difference between this and CD-quality PCM, and yet the data required is 5-10 less. It can be downloaded on a high-speed modem connection. Ogg is not yet supported.

GSM

GSM is a sound quality compression scheme. It is based on a model of the human vocal tract and compresses speech well but not music. It uses only a limited sampling rate 13KHz and requires 50 times less data than CD quality PCM. It can be downloaded on a low-speed modem connection. GSM is now supported in both lecture documents and exported presentations.

Conversion between sound qualities

The sound parameters, such as quality, sampling rate, etc. are set in the Preference dialogue, and these apply to all sounds recorded until the parameters are changed. It is also possible to edit the quality of an individual sound annotation in the Edit Console, by double-clicking a sound. This editing will determine the sound quality for any subsequent saving of that annotation. It will also allow the annotation to be re-recorded with the new parameters.

Without re-recording, a change from a low to a higher quality is ignored when saving, because there is not point in using more data to store a lower quality sound annotation. However a change from a high quality to a lower quality will take effect on saving, even without re-recording the sound. This will reduce the sound's quality and also reduce all file sizes. The key point is that compression reduces the data required to store sound and also reduces the quality. Once that quality is lost, it can not be recovered by re-encoding the data, only by re-recording the clip.

Volume activated detection of sound

Volume activated detection (VAD) of sound has also been introduced in version 2. What this means is that it is now possible to select the recording tool and to compose your speech with both natural and inadvertent pauses and to have only the active speech recorded as sound and the pauses encoded as pause annotations of a given duration. Before, this would have all been encoded as one sound annotation, including the silence, which would require between 100K bytes to 2.5 Mbytes per minute to encode. A pause annotation requires just a few bytes for whatever length of pause. VAD works with both speech quality sound and linear samples.

On-the-fly compression

The recording tool can now compress sound as it is being recorded. This feature uses the time you spend speaking to convert the PCM samples to speech quality sound if speech quality is selected. Because of this, there is no conversion when the lecture document is saved or when the document is exported to a presentation.

VAD sensitivity adjustment

A great deal of empirical evaluation effort has been put into simplifying the VAD parameters. On initialisation the VAD recorder measures the ambient noise in the environment, it uses this to decide when to start recording sound or when to express the silence as a pause. The sensitivity of silence detection can be adjusted. The VAD recorder has a number of parameters that determine how it works, these include triggering threshold, pre and post buffers to gain continuity before and after triggering and these have all determine the quality and sensitivity of detection. These have been linked to the single slider control labeled "VAD sensitivity". The sensitivity adjustment procedure therefore, is to start with the setting to the right, with high sensitivity, and to reduce the sensitivity until all of the sound you want to record is captured. Once set it is very stable for the same equipment.

CONclusions

We present in this paper, extensions to an existing tool that has already been proven to be effective at providing on-line teaching content. The enhancements implemented have been suggested by users of the tool but their implementation has been designed so as not to lose the original ease-of-use of the AudioGraph. One new concept has been introduced, which is one that most users will have

familiarity with, that is the hyperlink. Hyperlinks have been introduced quite naturally into the Recorder tool, by allowing any visual object to have a link attached to it. That visual object's drawn region then becomes the activation area for the link. Links are only active on playback, when the presentation is in the stop state and links can only be to the start of AudioGraph episodes or slides. In this way presentations with an arbitrary non-linear flow can be streamed file by file from the server with the user only downloading those components of a presentation that they select by their choices in taking links. Thus we retain the second key feature of AudioGraph on-line teaching, low-bandwidth downloads.

The paper also describes some other features introduced into the AudioGraph, including volume activated detection of sound, which records only active speech and encodes silence much more efficiently as parameterised pauses. One of the challenges in this implementation was defining a user interface and set-up procedures to make the use of this sophisticated user programmable for environmental and technical parameters, such as microphone sensitivity etc. This has been achieved with a simple environment test and a slider control, which has been set up to change the three major parameters based on an empirical evaluation of a range of systems.

ACKNOWLEDGMENTS

We gratefully acknowledge the support for this work under the TILE project funded under New Zealand's NERF funding scheme (<http://www-tile.massey.ac.nz>).

REFERENCES

Felder, R.M. and L.K. Silverman (1988). "Learning and Teaching Styles in Engineering Education," *Engineering Education*, 78 (7), 674-681, April 1988.

Felder, R.M., K.D. Forrest, L. Baker-Ward, E.J. Dietz, and P.H. Mohr (1993). "A Longitudinal Study of Engineering Student Performance and Retention: I. Success and Failure in the Introductory Course." *Journal of Engineering Education*, pp. 15-21, Jan. 1993.

Jesshope C. R. and Shafarenko A. (1997) Web Based Teaching: a minimalist approach, *Proc. Second Australasian Conference on Computer Science Education*, ISBN: 0-89791-958-0, pp16-23, (Association for Computing machinery Inc.).

Jesshope, C. R. Shafarenko A. and Slusanschi H. (1998) Low-bandwidth multimedia tools for web-based lecture publishing, *IEE Engineering Science and Educational Journal*, 7 (4), pp148-154., also published in: *IEE Computing and Control Engineering Journal*, 9 (4), pp156-162 and on-line at: C. R.

Jesshope, A. Shafarenko and H. Slusanschi (1998) Low-bandwidth multimedia tools for web-based lecture publishing, *IEE Computing Forum*, <http://forum.iee.org.uk/forum/library/>, September 1989.

Jesshope C. R. (1999) Web-based Teaching - Tools and Experience, *Australian Computer Science Communications*, 21, (1), pp27-38, ISBN 981-4021-54-7, Proc Australasian Computer Science Conference, ACSC99, Auckland, Jan 1999, (Springer).

Jesshope C. R. (2000a) The use of streaming multi-media in microelectronic education, *Microelectronics Education*, Kluwer Academic (London), ISBN 0 7923 6456 2, pp45-48.

Jesshope C. R. (2000b) The use of multi-media in internal and extramural teaching, *Proc Lifelong Learning Conference*, Central University of Queensland (Brisbane, Australia), ISBN 187 6674 06 7, pp257-262.

Gehne R. and Jesshope C. R. (2000) Tools for the production of small-footprint, low-bandwidth, streaming multi-media for distance education, *Proc Lifelong Learning Conference*, Central University of Queensland (Brisbane, Australia), ISBN 187 6674 06 7, pp240-244.

Jesshope C. R. (2001) Cost-Effective Multimedia in On-line Teaching, in *Educational Technology & Society* 4 (3) 2001ISSN 1436-4522 (http://ifets.ieee.org/periodical/vol_3_2001/jesshope.html)

Montgomery, S. (1998) Addressing Diverse Learning Styles Through the Use of Multimedia, (1998). Available: <http://www.vpaa.uillinois.edu/tid/resources/montgomery.html>

Pearson M. and Jesshope C. R. (1988) Multi-campus teaching using computer networks, *Proc. of the Third Australasian Conference on Computer Science Education* pp 106 - 111, July 1998 (Association for Computing machinery Inc.).

Segal, J. (1997). An evaluation of a teaching package constructed using a web-based lecture recorder. *ALT-J*, 5 (3), 32-42.

Novak, G. & Patterson, E. (1998). Just-In-Time Teaching: Active Learner Pedagogy with the WWW. Paper presented at the IASTED International Conference on Computers and Advanced Technology in Education, 27-30 May 1998, Cancun, Mexico,

Beyond Just Replay: Multimedia Support for Online Learning

Eva Heinrich

Massey University

E.Heinrich@massey.ac.nz

ABSTRACT

This paper proposes the use of multimedia technology for the development of communication tools in support of online learning. The aim of these tools is to provide a virtual discussion environment that closely mirrors the features of a physical environment. This means specifically to provide access to learning documents in support of discussion and to utilise multimedia technology to its full potential. In a number of scenarios the paper describes how these tools can be used to support learning. The key features and requirements for the tools are extracted and the Multi-Modal Description Framework that forms the basis for the implementation of the tools is introduced.

Keywords

Multimedia tools, online communication, document description

INTRODUCTION

In this paper we are advocating the use of multimedia technology for the development of innovative tools in support of online learning. Our work is motivated by two observations. Firstly, with the change to online as compared to face-to-face interaction we move from a very rich to a quite restricted form of communication. Secondly, today's online learning environments are a long way from fully capitalising on the technical possibilities provided by multimedia computing.

In our own teaching area of computer science a typical example of a teacher – learning interaction would be as follows. A student enters the lecturer's office with a printout of a computer program s/he has written and that does not work as expected. The student points to a program section on the printout, explains his/her assumptions and asks questions. The lecturer responds by identifying an error in the program, explaining some concepts or referring to a section of the lecture notes. In this face-to-face interaction multiple ways of communication are involved: exchange of questions and answers, direct reference to specific sections of documents, spoken words, writing and possibly drawing.

Transferring this example into a current online learning environment shows the restricted nature of this type of interaction, where a dialogue would most likely be established by exchanging text messages. In our opinion, the most severe restriction stems from the fact that the communication between the participants is removed from the object of discussion, in our example the printout (or online

version) of the computer program. The communication can be established via interactive online tools and the document to be discussed can be transmitted electronically but there is no way to link discussion and document as in the face-to-face situation.

Based on these introductory comments it is clear that the innovation we want to provide with our tools will not stem from the introduction of new teaching methods, but from the novel use of multimedia (and more general computing) technology to allow us to transfer well established teaching practise into the online environment. We want to overcome some of the restrictions to utilising multimedia inherent in current online interaction and add further strength in terms of building repositories or adding search and retrieval functionality.

In this paper we introduce several examples to support our general ideas on the construction of innovative online teaching tools. Firstly, we talk about two different forms of communication tools and then about cross-referencing tools that allow both teachers and learners to outline relationships between multiple documents. We then extract the key requirements for the tools and describe the implementation platform we are using to develop the tools introduced.

The next section provides a brief review of existing applications. We look at the accessibility of multimedia documents in current online learning environments, at online communication tools in general and at qualitative data analysis using multimedia documents in educational research (the motivation for looking at qualitative data analysis will become apparent once we discuss our cross-referencing ideas).

EXISTING APPLICATIONS

Virtual learning environments

In the current virtual learning environments like WebCT (2001) or Blackboard (2001) learning materials of various media types can be provided. The learning environments administer the learning material yet they do not provide any specific facilities tailored to the various media types. A learning object that is for example of media type video can be accessed by a student via the interface of the learning environment but the actual replay of the video takes place using a video player external to the learning environment.

Text-based communication

The virtual learning environments provide the expected range of asynchronous and synchronous communication tools like email, threaded discussions or chat rooms. WebCT, for example, provides an electronic whiteboard that can be used in parallel to a chat room conversation. There are a number of specialised communication tools from both within and outside the virtual learning community. ICQ (2002) is a popular text-based general-purpose chat tool. WebBoard (2002) provides chat facilities and threaded discussions. Inline graphics can be added to the text-based messages. These types of tools provide textual communication supported by a visual channel showing graphics or drawings. While this is useful it does not provide a way to integrate a learning object into the communication.

Video conferencing

The above mentioned text-based communication tools facilitate the personal communication between a small number of individuals or a discussion within a group. The technical requirements to participate are very low and a computer with low bandwidth internet connection and a web browser is in general sufficient. This is different for video and audio conferencing tools like NetMeeting (2002) or Interactive Video Network (2002). As the names suggest, these tools transfer video pictures and audio via a network. Video cameras, microphones and a much higher network capacity are required to participate. These tools can be used for one-to-one conversations, to conduct group meetings or to broadcast a lecture to remote locations. Having a visual channel does allow focusing a conversation on a document as in our introductory example. Yet, it still does not fully provide what we think is important for the following. Firstly, to transmit a clear picture of a document a high-resolution document camera is required. We cannot expect every learner to have access to the necessary technical equipment. Secondly, the tracking of references made to document sections is not possible in the way we are going to suggest for our document-based discussion tools.

Qualitative data analysis

We now want to have a brief look at a very different area, the area of qualitative data analysis in educational research. In educational research it is common to video-record, for example, classroom situations. The video recordings are then analysed together with their textual transcripts or with other related data (like transcripts originating from interviews conducted with the teachers involved in the lessons). Qualitative data analysis programs like VPrism (2002) facilitate this type of analysis. VPrism helps a researcher to link transcripts and video recordings via time counters and to attach descriptions. The relationship to our proposed tools is established in the following way: the video recordings and transcripts become our documents in the centre of the communication; the descriptions are the discussion contributions that refer closely to sections of our documents.

MULTIMEDIA TOOLS

After discussion the motivation for our research in the introduction and providing a brief literature review we now want to introduce our concepts for innovative tools to support communication and cross-referencing. We do this by presenting three example scenarios. We then analyse these scenarios to extract their requirements.

Document review

In this scenario we present a tool that will allow a lecturer or tutor to provide online feedback for a draft document (like a thesis chapter) to a student. Outside an electronic learning environment we regard the following interaction as typical:

The student submits a printout of the document to the lecturer;

The lecturer writes notes directly on the document printout; this will be corrections for spelling or grammar, comments on format or layout, conceptual remarks that are just single words or several sentences long;

Lecturer and student meet face-to-face; the lecturer talks the student through the major comments on the draft document; a discussion evolves;

The student takes the document away, works through the comments in detail and addresses the required modifications.

In our virtual environment we want, in principle, provide the same type of interaction.

The student submits the document electronically to the lecturer: This is done via a virtual learning environment that automatically performs the following tasks. The document the student submits is converted into a standard, image-based format (like a pdf-type format for all printable originals). This step will make the correction process independent of the software program a student uses and will allow dealing with images, spreadsheets or text documents in the same way. It will further prevent the lecturer from modifying the document the student has submitted (that will be of less importance while reviewing a thesis draft but of high importance for the online evaluation of assignments). The document is stored and registered in a central database. The access rights for the document are set to restrict access to the student and the lecturer.

The lecturer performs the corrections on the document: Like on a physical desktop the lecturer has tools like highlighters and pens in different colours available. These are used to write or draw directly onto the electronic document, basically in the same way as this would be done on a physical piece of paper (of course, input via the keyboard is possible as well). Yet, the electronic version will have some advantages. The colours used, for example, can be regarded as codes (red for spelling, blue for format, green for conceptual comments, ...) and later on used by the student for selective viewing.

The lecturer will not have to choose between an 'input' mode and a 'comment-mode' (similar to using the 'Comment' tool in the Microsoft Word application). This makes commenting on the document more direct. The absence of an 'input' mode has the further advantage that the document cannot be changed inadvertently.

All comments, both of graphical and textual nature, are stored in a central database. The database entries include the comments themselves (including their shape and colour parameters), the exact location of the reference in the document, the date and time the comment was made and the author of the comment.

None of the comments actually change the document. The comments are put on top of the document (much like writing on a transparency put on the document). While viewing the document one can choose in which intensity original document and comments are displayed. This gives the advantage that the document can be viewed in its original form and that the comments can be viewed selectively.

The lecturer adds some verbal messages: This step can be seen as replacing the first part of a face-to-face conversation in which the lecturer would make some introductory general comments before talking the student through the major issues with the document. The lecturer might want to make these statements verbally to add some personal touch to the feedback. Another reason could be that some kinds of information can be more easily expressed in spoken form (like a sound pattern or intonation in a language). These verbal comments are attached to document sections and are stored in the database, in a way similar to the textual annotations.

Discussion between student and lecturer: This online discussion could be conducted either by conventional online means as outlined in the application review section or by document-based discussion as described in the next scenario.

The student works through the comments: The student can look at the document in different modes by displaying only comments of a specific type (using the colour codes the system can present selectively comments of grammatical, formatting, structural or conceptual nature) or status. The student can mark a comment as 'being addressed', which facilitates working through the comments step by step. If questions occur the student can add his/her own statements to the lecturer's comment. This can lead into a discussion with the lecturer or can simply be a reminder to the student to follow up on this issue.

Document discussion

To outline our ideas for the support of virtual discussion we again first want to present a 'physical-world' scenario. As part of a psychology class students are attending tutorials in which they are required to view and discuss video clips (that, for example, display the interaction of a group of people in a conflict situation). The tutor provides some introductory comments, screens the video, and then starts the discussion by asking some key questions referring to specific episodes in the video. In the discussion, the participants draw on examples from the video to support their arguments.

As in our document review scenario, we want to create a virtual environment that preserves key features of the face-to-face communication. This leads us to two main requirements for our discussion tool:

We need the thread of discussion to capture sequence of contributions and arguments;

We need a very direct way to create a link to a specific sequence in a discussion document (like a dialogue in the video or a gesture captured across a sequence of frames in the video).

Currently available online tools fulfil the first requirement but not the second. While it is possible to refer to a whole document, it is not possible (in the tools we are aware of), to create a direct reference to a specific sequence within a document like a video recording or a text document. What we want to have is a discussion tool that displays side-by-side the discussion thread and the discussed document. The tool needs easy to use features that allow every discussion participant to define sequences in the document (a few seconds of a video or audio clip, a specific area of a image or video frame) and to link these into their discussion arguments. Such a tool would support a tutorial situation where the participants exchange arguments and back up their claims by precise evidence presented in the provided sample material.

Depending on where we set the emphasis in using the two main parts of the discussion tool, the discussion thread and the discussed document, we can facilitate different types of discussion:

If we focus on the discussion thread we support a higher-level, more abstract discussion that uses document sections to underpin arguments;

If we focus on annotations attached to document sections we can develop a discussion moving from concrete examples (e.g., a raised voice by one of the persons shown in the video clip) to abstract concepts (e.g., the expression of anger).

This second point can lead us to an asynchronous form of communication. The tutor annotates key sequences in the document to focus the students' attention. The students analyse these key sequences and attach their own observations and thoughts to the sequences. As all contributions are stored in a database every participant can see all annotations made for a particular sequence. In this mode the primary focus lies on studying the document, the secondary focus on exchanging information with co-students and the tutor.

Considering these different forms of conversation we see that there is a continuum from thread-based discussion with little focus on a discussion document to the exchange of annotations attached to document sections. From a technical point of view, the whole spectrum can be served by an application that manages documents, sections of documents and annotations of these sections. Conceptually, we can view these annotations as *discussion contributions that are supported by evidence in document sections* or as *descriptions of events displayed in the document that are exchanged by a number of 'discussion' participants*.

The latter viewpoint leads us into the area of qualitative data analysis where the focus lies on the data (our discussion document) to be analysed. In the next section we will provide an example for cross-referencing of various documents that are closely linked to this type of analysis.

Cross-referencing

In current virtual learning environments it is easy to make multiple learning documents available. Only in a very restricted way is it possible to create links to show relationships *within* these documents. In current systems links can only be included for web page

documents. Additionally, the way documents are protected only the person submitting the document (the lecturer or tutor) can modify the document. That means, students cannot add their own links.

We envisage a cross-referencing tool supporting situations similar to the following scenario. A tutor prepares an exercise for the students of a law class that have been studying the use of language and communication protocols in courtrooms. Material available online are video recordings showing court sessions, lecture notes on language in courtrooms and a dictionary of law terms. The tutor focuses on the communication between judge and prosecutor in one of the video recordings. S/he annotates this communication regarding terms used, gestures and intonation. The tutor further links the annotations of terms to the appropriate entries in the dictionary and provides links to relevant sections in the lecture notes. The learning goal for the students is to re-enforce what they have heard in lectures about the procedures in court. They do this by first studying the annotations provided by the tutor and by then annotating several video sequences on their own following the same annotation criteria as the tutor. Depending on the requirements set by the tutor, the students will work on their own or in groups (using discussion features as outlined for the earlier scenarios). To evaluate the students' work the tutor can use a comparison feature of the tool to check for overlap of the students' annotations with his/her own sample solutions.

The annotations entered by tutor and students (after some quality control by the tutor) are stored in a repository. Over time, a sizeable collection of sample data develops. This collection is divided into various sets of data that can be released to future students in a controlled fashion.

KEY FEATURES AND REQUIREMENTS

In the previous sections we have presented various scenarios to indicate the types of tools we want to develop to support online learning. There are a number of factors that can be adjusted depending on the learning situation:

Focus on discussion thread or document annotation;

Discussion documents supplied solely by teacher or as well by learners;

Cooperative discussion environment or dissemination of information through teacher;

Synchronous or asynchronous communication;

Use of contributions as they are generated or collection of contributions in a repository for later use.

Across these variations we can identify several key factors our tools need to provide:

Access to documents of multiple media types: The learning documents that are the focus of discussion or description can be of any media type (video clips, audio recordings, text documents, images, spreadsheets, or AudioGraph (2002) recordings).

Support of multiple forms of information input: To create a 'natural' virtual environment we need to give users various ways to interact with the system and to allow them to choose the input mechanism most appropriate for a given situation.

Sequencing of contributions: Regardless of synchronous or asynchronous mode of communication our tools need to capture the sequence of contributions.

Storage of contributions: We need to store who has made the contribution, the actual content of the contribution and the precise link of the contribution to a learning document.

To implement the tools we need to satisfy a number of key requirements:

Multimedia capability: Most modern personal computers already have the hardware and software capabilities to replay video and audio documents. Our tools will utilise these existing capabilities and extend existing programs, like video players, according to our needs. For video and audio documents it is common to refer to 'player' software. Characteristic for these players is that they can replay files of different video or audio formats and that they only reproduce the file content but do not allow modification of the files. We suggest using a similar mechanism for all printable documents. When a printable document, a text document, spreadsheet, image or graphic, is registered with the virtual learning environment it should be converted into a generic format. A file in the generic format is then displayed by a 'player' software. The advantages of this approach would be that, like with video or audio players, the

user deals with a consistent interface for the software that displays all printable documents and that the content of the files cannot be changed inadvertently.

The multimedia capabilities we need for providing multiple forms of information input again are already available in today's computer systems. The underlying graphics routines to implement a software highlighter pen using mouse or pen/graphics tablet input, for example, are available – our task is to synthesise conceptually useful features from these underlying technologies.

User interface research: Closely linked to the previous point is the need for careful design of the user interface for our tools. As we have outlined in the introduction to this paper we want to create a virtual environment that has all the good features of a physical environment. In a physical environment, we simply grab a highlighter pen to mark a document section or we start speaking to communicate with someone else present. For our tools to be accepted we need to create a virtual environment that is similarly easy to use. The features of the tool have to be transparent to the user and have to be immediately accessible (this means among other things to limit the number preparatory steps that are necessary to perform an activity).

Central database: We are looking at a networked environment in which multiple users submit documents and contribute to discussions. We need a central database that stores all data generated. This database needs to be combined with the data storage already existing in virtual learning environments that contains the student and course management data.

Management of ownership and access rights: We need to manage the ownership and access rights for learning documents and associated meta data (annotations or discussion contributions). This has to be done in a way to ensure that everyone can access all information they require, to protect the privacy of users and to prevent unauthorised release of information. A lecturer, for example, should be allowed to disseminate information to a whole class of students yet individual students might only be allowed to do so if their contributions are clearly identifiable as student opinions. A group of students might want to discuss work leading to an assignment submission without other students being able to follow their thoughts. Assignments submitted should be confidential to the submitting students and the tutor.

OUR IMPLEMENTATION PLATFORM

Within the TILE project (2001) we have developed a platform, called the Multi-Modal Description Framework, MMDF (Heinrich and Chen, 2001) that forms the basis for implementing the tools outlined in this paper. MMDF satisfies most of the main requirements established earlier:

Access to documents of multiple media types;

Description input using text and audio; time- and area-based definition of document sections;

Storage of description data in central database;

Management of ownership and access rights for documents and meta data.

With MMDF in its current form we are very close to what we need to provide the document review and cross-referencing tools. Missing at this stage is a generic format for printable documents. To implement document-based discussion we need to add a new interface to MMDF that displays description contributions in a sequential format.

CONCLUSION

In this paper we have introduced online multimedia tools for the support of document-based discussion and the cross-referencing of learning material. The general idea behind these tools is to provide a virtual learning environment that preserves some of the rich interaction possibilities of physical environments and makes use of multimedia technology to achieve this.

There are a number of ways in which we want to use multimedia for more than 'just replay'. The tools will access learning documents with program features adjusted to the media properties of documents. We will allow for multiple forms of input to mirror more closely a natural interaction environment. The tools will support the establishing and displaying of links in documents of multiple media types.

We suggest that our multimedia tools will make contributions to learning by facilitating:

One-to-one and group discussions;
Close reference to learning documents;
Construction of repositories of information;
Design of exercises.

We are currently in the specification phase for the tools suggested. We have implemented a framework that will form the basis for the development of the tools. While we cannot yet report on experiences of using the tools we think it is important to discuss our ideas with the research community at this early stage. We are hoping for constructive feedback, many new ideas and contacts to research partners who are willing to test our tools in their teaching.

ACKNOWLEDGMENTS

We would like to acknowledge the support for this project from the New Zealand government's New Economy Research Fund (NERF) under contract MAUX9911. Without this support, this project would not have been possible.

REFERENCES

AudioGraph (2002). <http://www.nzedsoft.com/audiographhomepa.html>; accessed 15/02/2002.

Blackboard (2001). <http://www.blackboard.com>; accessed 07/12/2001.

Heinrich, E., Chen, J. (2001). A Framework for the Multi-modal Description of Learning Objects. Proceedings of International Conference on Dublin Core and Metadata Applications 2001. Keizo Oyama and Hironobu Gotoda (Eds.), pp 32-37. Tokyo, Japan.

ICQ (2002). <http://icq.com/>; accessed 15/02/2002.

NetMeeting (2002). <http://www.microsoft.com/windows/NetMeeting/>; accessed 15/02/2002.

TILE (2001). <http://www-tile.massey.ac.nz>; accessed 07/12/2001.

VideoNetwork (2002). <http://www.umats.ums.edu/IVN/>; accessed 15/02/2002.

VPPrism (2002). <http://www.lessonlab.com/vprism/>; accessed 15/02/2002.

WebBoard (2002). <http://www.chatspace.com/products/webboard/>; accessed 07/12/2001.

WebCT (2001). <http://www.webct.com>; accessed 07/12/2001.

Teaching Mathematics with Audiograph

Maureen Loomes, Alex Shafarenko, Martin Loomes

University of Hertfordshire

m.j.loomes@herts.ac.uk

ABSTRACT

Mathematics teachers are charged with the task of teaching children not only how to solve problems, but also how to explain their solutions. There is, however, very little pedagogical support for this enterprise, and very few tools that support the process. We argue that mathematical explanation is essentially a multi-modal activity, requiring the integration of several modes of discourse, and that learners need tools capable of capturing such discourse in ways that can be edited and developed. The Audiograph system is proposed as a possible candidate for this, some experiences of using this with students of mathematics are introduced, and the potential for future work discussed.

Keywords

Mathematical explanation, Audiograph, multimodal interfaces, pedagogical issues, teaching/learning strategies

INTRODUCTION and motivation

The research outlined in this paper is motivated by the observation that students of all ages seem to find great difficulty in producing clear explanations of mathematical activity. In some cases, of course, this can be directly attributed to a basic lack of subject knowledge, but often students who are able to "get the right answer" seem unable to reconstruct their solution processes into a rational mathematical explanation. It might be argued that this is a symptom of the reduction of formality within the mathematics curriculum generally: without proof how can we have explanation, but this is a simplistic view. Many of the more formally presented topics in mathematics (such as Euclidean Geometry and the solution of algebraic equations) are traditionally accompanied by mysterious frames into which the student is required to force the written representation of the solution. Magic incantations such as "Q.E.D." or "We assume that the result holds for x , and show that it holds for $x+1$ " are commonly copied from exemplars and re-used without understanding. These frames, we would argue, are meaningful only to those who already understand the subject. Cobb et al. refer to these as "sociomathematical norms ... [which] establish... what counts as an acceptable mathematical explanation and justification" (Cobb et al. 2001, p.126). Once a student has mastered the technique for forcing a solution into the appropriate frame, the teacher usually provides the required "tick", indicating success. If the question required only an answer this seems quite reasonable, but the tick is usually forthcoming even if the question required an "explanation" of the answer. Thus, even if the student has no idea how to produce such an explanation, the frame provided has produced an illusion of understanding. We would argue, therefore, that the presence of formality may obscure the problems a student has in explanation, but it is not sufficient to overcome it.

An important point to note is that mathematical explanation is a discourse *about* mathematics, generally requiring a fair degree of fluency *within* various mathematical domains. Whilst there has been much debate about, and research into, the processes that students use in finding the solutions within domains, there seems to have been very little discussion of the problems associated with the problems of mathematical explanation itself. Indeed, explanation *per se* seems to be a much-neglected topic in both educational research and psychology: as Donaldson notes

"This brings up the question ... how and when does the ability to explain develop? Despite the considerable educational relevance of this question, there is a dearth of research which addresses it directly." (Donaldson 86, p1).

Of course, this might not matter: after all mathematics has progressed in spite of this handicap for many centuries, so why address it now? One answer is simply that teachers are now *required* to do so. The U.K. National Curriculum (DFE, 1995) at Key Stage 2 (i.e. children aged 7-11) specified that "Pupils should be taught to ... explain their reasoning" and this has remained within the revised National Curriculum in Mathematics (DfEE, 1999a) which requires children at Key Stage 2 to be taught to "explain their methods and reasoning". The preliminary report of the Numeracy Task Force (DfEE, 1998) also suggested that "numeracy pupils should explain their methods and reasoning" and teachers should "collect information about... the clarity of explanations given in oral and written responses." The OFSTED (1999) review of Primary schools in England (1994-8) also suggests that schools should monitor and review how time is managed in lessons to ensure that pupils have the opportunity to *learn how to explain* their methods and present a reasoned argument. Thus mathematical explanation is no longer a higher-level task, which somehow transcends the substantive curriculum and just "happens", but it needs to be addressed in its own right: it has become a topic, just like algebra or geometry. Moreover, teachers must be able to identify and record progression within this, and keep records to provide evidence of this. Unlike topics like algebra and geometry, however, there is currently a complete lack of support for teachers in this area of the curriculum. The lack of current pedagogical support might suggest that the problem is actually too hard to be tackled, in spite of the National Curriculum requirements, but a classroom-based study carried out by one of the authors suggests that "mathematical explanation" can be taught (Loomes, 1999, 2001). It has become clear, however, that traditional paper-based approaches to recording mathematical artefacts are not sufficient upon which to build a pedagogically sound approach to the teaching of explanation across a wide range of ages and abilities.

A Role for Audiography?

We would argue that teachers need considerable support if a curriculum addressing the teaching of mathematical explanation is to be implemented. One aspect of this support concerns the artefacts of mathematics. Currently teachers use models of the application domain (blocks, shapes, counters etc.), but the actual "mathematics" produced by the students is usually thought of as being a paper-based artefact, typically comprising handwritten text, mathematical notation and diagrams. Whilst there will frequently be spoken discourse, this is usually seen as a means to an end, and is restricted to part of the learning process, rather than being seen as a valid component of the explanation. Paradoxically, a significant part of the explanation *experienced* by the students in the form of teaching will be spoken and informal. It is also likely to be multi-modal, making use of speech, pre-printed materials, handwritten incremental materials (such as the building up of a solution on a board alongside other modes of discourse), gestures relating components, reference to physical models, and anything else that a teacher thinks might help. Thus, at one level, a student sees explanation as an evolving thing, developing in response to feedback from an audience, making use of a wide variety of resources. There will be multiple strands developing, some of which are fragmented as lemmas are developed off-line, several variants of some parts will be provided to meet the needs of different students, and partial results will be stored and retrieved as needed. At another level, however, explanation seems to be reduced to a somewhat pointless template for producing written answers to questions. If we are to bridge the gap between this unstructured, rich environment and the sociomathematical norms of traditional mathematics, we believe it is necessary to allow students better access to the components on route. This problem has been addressed by Cole and Engestrom (1993) in the context of educational research

"Audio and video tape recording, films, and computers have all, in their own way, enabled us to interact with the phenomena of mind in a more sophisticated way. We can now not only talk about the mutual constitution of human activities, but display it in scientifically produced artefacts" (p. 43).

Whereas researchers want to capture snapshots of the world for analysis, teachers and students need systems that are dynamic, so that the artefacts can be viewed statically, for purposes of summative and formative assessment, but also as objects of development and evolution. From this viewpoint, we see mathematical explanation not as pre-conceived frames to be filled, but as narrative structures to be developed.

This is essentially a constructivist view, in Papert's interventionalist sense (Papert, 1991), where we aim to improve our understanding of how knowledge is constructed so that we can structure learning activities to support the development of the individual student. In the words of Bers and Best (1999),

"Computational tools become computational construction kits (Resnick et al., 1996) when they support users as designers of their own projects by making both personal and epistemological connections"

If we analyse some of the modes of expression frequently encountered whilst explanations are being developed we can draw up an informal list of requirements for such a system.

- Many problems start their life (as far as the student is concerned) on paper. This typically includes diagrams, typed text and mathematical formulae. Thus the ability to integrate printed materials into the developing discourse seems important as students (and more experienced mathematicians) often annotate the expression of the problem at the early stages of devising a solution (for example, underlining key components, adding angles to diagrams or linking elements in a sequence).
- Students often work together in solving problems, and they do so using speech. Similarly students often work "out loud", talking themselves through problems. Teachers often provide spoken help and verbal feedback during problem solving tasks. Questions may be posed in spoken form (or spoken form may be provided alongside a written question to provide context and hints).
- Drawings, both carefully constructed and jottings, also play an important part in mathematical explanations, not only in the final presentation but also during the development.
- Mathematical notation needs to be written. This may be "correct", conforming to all the sociomathematical norms, but could also be private language, understood only by the writer at the outset, typically translated into more conventional forms when the presentation is tidied up for "publication".
- Rubbing out is important, both completely (where its presence may confuse either the writer or the intended audience later) or semi-transparently (for example, when cancelling fractions, where absence of the initial form may be equally confusing later).

- Gesture is vital. The importance of gesture in scientific explanation has recently been discussed by Roth, who observes "the analysis of an individual's gesture and talk over and about inscriptions shows how deeply integrated these are. Furthermore, the changing relation of gesture and talk over time also suggests that, for the individual, there is a change in the nature of the display" (Roth 2001, p. 55).

This rather simplistic analysis suggests that we need a system capable of integrating speech and pen-based developments, with interleaving of these also allowing gestures (such as highlighting with a pen) and erasures of various types. In essence, we want a flexible, multimodal system that can be adapted by the individual as the construction of an explanation develops. Cohen and Oviatt (1995) & Oviatt and Cohen (1991), have noted that users tend to prefer speech for descriptive purposes, including giving properties of objects, describing what needs to be done and discussing past and future events. This is perhaps not surprising, given that most people can speak clearly faster than they can write legibly, and with less physical effort. Handwritten input, however, is often preferred where the descriptions suggest the use of numbers, iconic representations and diagrams (Oviatt, 1997 & Suhm, 1998). Thus, there is no preferred single mode for all tasks: users like to have the power to select the mode according to details of the task. As Oviatt et al. (2000) have noted:

"Taken together, the speech and pen modes easily can be used to provide flexible descriptions of objects, events, spatial layouts, and their interrelation. This is largely because spoken and pen-based inputs provide complementary capabilities. For example, analysis of the linguistic content of users' integrated pen-voice constructions has revealed that basic subject, verb and object constituents are almost always spoken, whereas those describing locative information invariably are written or gestured (Oviatt, DeAngeli, & Kuhn, 1997). This complementarity of spoken and gestural input also has been identified as a theme during interpersonal communication (McNeill, 1992)." (Oviatt et al. 2000, p.268).

The Audiograph Tool

The Audiograph tool was originally conceived as a means of allowing university lecturers to prepare materials for distribution over the WWW which fitted naturally into their normal modes of working, and required no additional specialist training or skills in the use of technology (Jesshope, Shafarenko and Slusanschi, 1998). The basic model replicates the normal lecture producer-consumer model in part, which may be caricatured as a lecturer preparing slides and a "script" to deliver as an accompaniment, the two being linked in delivery by components such as referential speech ("as you can see at the bottom of the slide...."), gesture (highlighting accompanied by a phrase such as "look here...") and the handwritten annotation of slides. A significant feature of Audiograph is that it is multimodal, supporting background text and graphics, speech, simple gesture, real-time handwriting, and real-time drawing, but at the same time very simple to use. There have recently been several studies of technology to provide such multimodal interfaces. As noted by Oviatt et al. (2000),

"The growing interest in multimodal interface design is inspired largely by the goal of supporting more transparent, flexible, efficient, and powerfully expressive means of human-computer interaction. Multimodal interfaces also are expected to be easier to learn to use, and they are preferred by users for many applications "(p. 265).

It is important to stress, however, that most studies of multimodal interfaces refer to systems with multimodal *control* interfaces. For example, controlling equipment by speech or movement. Audiograph is not multimodal in this sense, but in its data interface. Thus many of the often-cited limitations of multimodal systems are not pertinent here (for example, the need to train the system to recognise particular users, or the high error rates associated with speech recognition). The actual authoring interface is shown in figure 1.

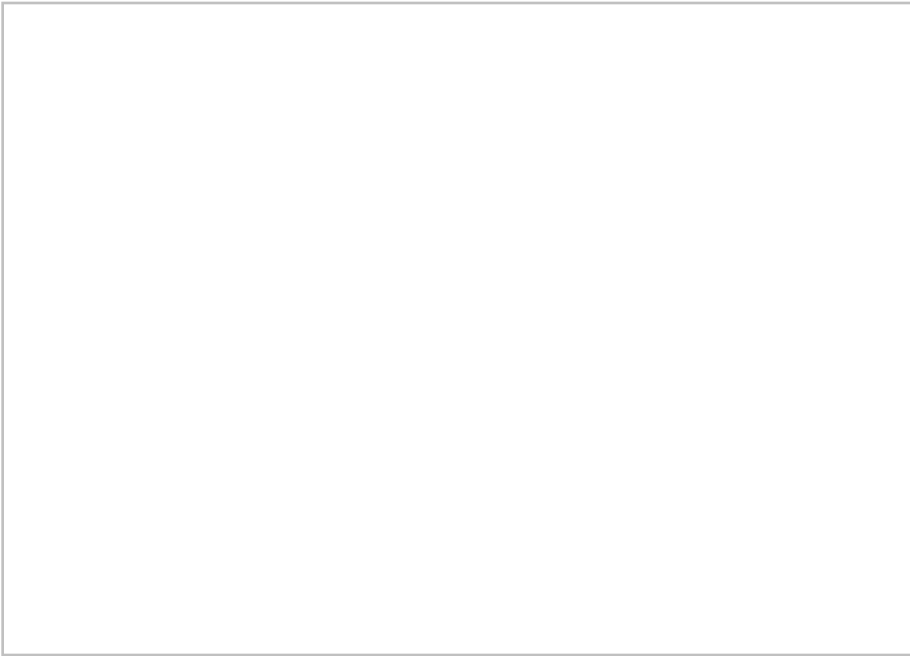


Figure 1: Audiograph in use for authoring.

The main window shows an imported slide presenting a pair of simultaneous algebraic equations handwritten by a teacher (the solution is developed by a student, and is referred to below). The tool palette to the right of the main window provides click-on controls for selecting a pen for handwriting or gesture, an eraser, a highlighting tool, a microphone for speech input and a pause. The other tools need not concern us here. The edit console below allows the sequence of actions comprising pen strokes, speech fragments, erasures, pauses, etc. to be edited. Presentations can also be optimised for delivery over the WWW, but that is not relevant to this paper.

The ease of use, public-domain nature and multi-modal features of the Audiograph make it an excellent tool for exploring the teaching of mathematical explanation. The fact that recording is sequential, with no multiple streams (unlike most complex multimedia authoring packages) is not a problem, indeed it is a virtue. Because the background slide (or worksheet), voice, handwritten text and gestures all result in independent events, it is simple to amend elements of a presentation, or move content from one mode to another. For example, a student who knows what needs saying ("this angle is the same as that angle") but does not know, or has forgotten, the conventional ways of naming angles, can use speech and gesture to achieve an acceptable explanation. With support from peers or a teacher, this can subsequently be amended to include a more formal line of mathematics to replace or supplement the speech fragment. There are also cases where students know exactly what they are doing, and even say the right thing, but the audience is tuned in to hearing something rather different. For example, in explaining an answer to a Key Stage 2 National Curriculum Test question which required pupils to take number patterns presented in a series of triangular shapes and fill in the missing numbers in the final triangle, one child wrote "The numbers are going up in threes" which suggests to most teachers of mathematics that somewhere there is a sequence $n, n+3, n+6, \dots$. In fact, there are three numbers $\langle 2,3,4 \rangle$ which go up together to become $\langle 4,5,6 \rangle$ and $\langle 6,7,8 \rangle$. Thus, just as children may "go up in twos" to collect their books, so these numbers are "going up in threes". The existence of a soundtrack and gestures, indicating "the numbers" grouped in threes would make this absolutely clear, whereas the purely written form looks "wrong".

Achieving this development of multimodal explanations with more traditional representational devices such as video recording would be very problematic, as a recording of a student standing in front of a whiteboard combines all of these events into a single data stream which can only be untangled by complex and expensive editing tools. For example, editing a video so that a spoken phrase is replaced by something written on a board is non-trivial.

The pen-based interface lends itself readily to mathematics, allowing symbols and simple diagrams to be produced to accompany spoken and written text. Mathematical word processors are not so simple to use, and would constrain the pupil to the mathematical norms as most only allow for well-formed expressions, and so pupils could not create their own notations as intermediate steps or present mistakes indicating they need help.

using audiograph in teaching mathematical explanation

We have explored two possible uses of this with students. First, individuals have used the system in the production of solutions to simple problems, and the work submitted was marked and returned by the teacher using the same technology. Second, groups of

students were asked to develop audiograph presentations to be placed on the Web as resources for other pupils to use. Several interesting features emerged from these trials.

First, the learning required to use the technology effectively took just a few minutes. All of the students were familiar with basic computer use, and the only "strange" feature was a graphics tablet. A headset microphone was used, enabling good quality sound to be achieved in a typical classroom setting. Typically, students mastered the technology within a few minutes. The graphics tablet presented no problems, except for the drawing of geometric diagrams, where some students wanted to draw accurately, even when a sketch was clearly sufficient. One solution to this was to use a sheet of paper placed over the tablet, and a pen that actually drew, but then the erasure model of the system is compromised. A second "solution", which unfortunately places the cost of the system well above most educational establishments, is to use an interactive screen which can be drawn upon: experiments with this system are currently underway.

Second, there were interesting features that emerged when the spoken modes were analysed alongside the written ones. Sometimes this discrepancy indicated trivial slips, which could be easily spotted (for example, saying 5 and writing down 6). Very often students said the one thing, but wrote something that meant something different to more experienced mathematicians. Sometimes they said something that was not really correct but, interestingly, wrote down the correct thing (as if the template they already knew took over when they moved to a written form). For example, in the solution presented in Figure 1, there were two such instances. The symbol "=" is frequently overloaded in written mode, denoting concepts such as numerical equality, implication and identity, leading to mathematically incorrect statements. The audio strand associated with this Figure contains "If we take equation two star from one star *we get...*" The written form contains an equal sign to indicate "we get". This particular student is a very competent mathematician, who often uses algebra from choice, and hence tends to use symbols rather than words. There was, however, a lack of knowledge as to how to represent "we get", so the nearest approximation was used. When the two equations were transformed (to give 1* and 2*) the student actually continued to refer to them as 1 and 2, but wrote the correct form. This suggests that the formal need to introduce a correct name was appreciated, but the student realised that no confusion would arise if a short form was used informally. The ability to mark this assignment multimodally meant that the discrepancies between the audio and written presentations could be highlighted as they occurred, simply pointing out what was happening. There was no need to cover the page in red ink to provide feedback (which would have been required to spell out in detail what the overloading had led to, and suggested that the solution was "wrong" whereas in fact it was simply the grasp of as-yet untaught issues that were being addressed), but equally appropriate feedback could be provided to a pupil who was clearly ready to move on. The ability to separate the meta-level commentary on the explanation (the teacher's voice) from the annotations on the pupils work seemed helpful.

A third situation that arose, typically amongst older students, was a desire to reduce the amount of spoken explanation, replacing it with more conventional mathematics. In the extreme, one group of students who were preparing solutions for the WWW actually insisted on removing all speech, claiming that "mathematics can't include talking". It is interesting to note that the English curriculum within the U.K. has moved firmly towards a system where speaking, listening and writing are firmly established with both teaching and assessment. Mathematics has made similar changes to the curriculum, with provision for "oral and mental" activities included in the National Numeracy Strategy (DfEE, 1999b), but this is, as yet, not reflected the assessment mechanisms.

Future work

These explorations are, as yet, embryonic and small scale. They have convinced us, however, that audiography has a valuable role to play in the mathematics curriculum. There are several avenues that we would like to explore, and these are briefly described below.

First, the ability to allow teachers to assess and provide feedback on drafts of explanations (like English teachers do on narratives) seems very valuable. We would like to explore this for particular classes of students, such as those with learning difficulties or problems with motor skills. Our discussions with teachers suggest that the transition from spoken to written forms is often the place where many difficulties seem to reside, and a systematic study of these which teachers can carry out for themselves might prove useful.

Second, we would like to mirror some of the work on narrative group work carried out in the English curriculum within mathematics. For example, the activity of older students producing resources (typically books) for younger ones is well-established, and brings the benefit of making the authors consider and discuss the intended audience, which in turn is reflected in their choice of expressive devices. We believe that similar activities (such as the production of web pages for younger pupils or parents) would bring similar benefits, and audiograph makes this very simple to do. The issue of providing an audience for mathematical explanations is a complex one, and space does not permit a detailed discussion, but evidence suggests that students consider mathematical explanations simply as a device relevant to assessment tasks, rather than communication. This might explain the willingness of more able students to accept the socio-mathematical frames so willingly and uncritically (in contrast, they seem very willing to criticise social norms for narrative in literature).

REFERENCES

- Bers, M & Best, M (1999) Rural Connected Communities: A Project in Online Collaborative Journalism, *Proceedings of the Computer Support for Collaborative Learning (CSCL) 1999 Conference*, Ed: C.Hoadley & J.Roschelle, Lawrence Erlbaum Associates.
- Cobb, P., Stephan, M., McClain, K. & Gravemeijer, K. (2001) Participating in classroom mathematical practices. *The Journal of the Learning Sciences*, **10**, (1&2) 113-163.
- Cohen, P. R. & Oviatt, S.L. (1995) The role of voice input for human-machine communication. *Proceedings of the National Academy of Sciences*, **92**, 9921-9927. Washington DC: National Academy of Sciences Press.
- Cole, M. & Engestrom, Y. (1993) A cultural-historical approach to distributed cognition. In *Distributed cognitions: Psychological and educational considerations*, ed. G. Salomon (1-46). Cambridge, England: Cambridge University Press.
- Department for Education. (1995) *The National Curriculum for Key Stages 1 & 2*. London: HMSO.
- Department for Education and Employment. (1998) *Numeracy Matters: The Preliminary Report of the Numeracy Task Force*. London: Crown Copyright.
- Department for Education and Employment. (1999a) *The National Curriculum for England*. London: HMSO.
- Department for Education and Employment. (1999b) *The National Numeracy Strategy: Framework for Teaching Mathematics*, London: Crown Copyright.
- Donaldson, M. (1986) *Children's explanations: A psycholinguistic study*. Cambridge: Cambridge University Press.
- Jesshope, C., Shafaranko, A. & Slusanschi, H. (1998) Low-bandwidth multimedia tools for Web-based lecture publishing. *Engineering Science and Education Journal* **7**,(4), 148-154
- Loomes, M.C. (1999) Developing Skills in Mathematical Explanation. *TTA publication 63 / 8-99*
- Loomes, M.C. (2001) Developing children's skills in mathematical explanation. *Topic:Practical Applications of Research in Education*, **26**, 6 , 1-6 (Autumn, 2001)
- McNeill, D. (1992) *Hand and mind: What gestures reveal about thought*. Chicago: University of Chicago Press.
- OFSTED (1999) *A review of Primary schools in England 1994-1998*. London: HMSO
- Oviatt, S. L. (1997). Multimodal interactive maps: designing for human performance. *Human-Computer Interaction*, **12**, 93-129.
- Oviatt, S. L. & Cohen, P. R. (1991) Discourse structure and performance efficiency in interactive and non interactive spoken modalities. *Computer Speech and Language* **5**, 4, 297-326.
- Oviatt, S., Cohen, P., Wu, L., Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J. & Ferro, D. (2000) Designing the user interface for multimodal speech and pen-based gesture applications: State-of-the-art systems and future research directions, *Human-Computer Interaction*, **15**, 263-322.
- Oviatt, S.L., DeAngeli, A. & Kuhn, K. (1997). Integration and synchronisation of input modes during multimodal human-computer interaction. *Proceedings of Conference on Human Factors in Computing Systems (CHI'97)*, 415-422. New York: ACM Press.
- Papert, S. & Harel, I. (1991) *Constructionism*. Norwood, NJ: Ablex Publishing
- Resnick. M, Bruckman. A & Martin.F (1996) Planos not Stereos: Creating Computational Construction Kits, *Interactions*, **3**, 6.
- Roth, W. (2001). Situating Cognition. *The Journal of the Learning Sciences* **10** (1&2) 27-61.
- Suhm, B. (1998). Multimodal interactive error recovery for non-conversational speech user interfaces. *Doctoral thesis*, Fredericiana University, Germany.
- Vygotsky, L.S. (1987) *The Collected Works of L. S. Vygotsky, Vol I*, New York: Plenum

CHALLENGE – An intuitive, goal-based scenario authoring system for electronic learning

Terry Stewart and Paul Bartrum

College of Sciences, Massey University, New Zealand

T.Stewart@massey.ac.nz

ABSTRACT

CHALLENGE is an authoring tool designed for constructing and presenting goal-based scenarios to students in any subject domain, either locally or across the Internet. Using a Scenario Player, students can explore the scenario, examine objects and move them from location to location, interview people, conduct tests and undertake research. Information is provided by hypertext, images, sound or video either local or external. The exercise can be used as a basis for class discussion, or students may submit an analysis for marking by the tutor. The program tracks student activity and provides a tutor-supplied tailored debriefing dependant on that activity. This paper describes the features of the software, and the mechanisms used to develop a Player to use scenarios over the Internet.

Keywords

Authoring tool, goal-based scenarios, computer-assisted learning, problem-based learning, case-based learning

INTRODUCTION

Transferring knowledge and skills from those who know, to those who don't can take many forms. There is broad agreement that the style of teaching should reflect the subject being taught, and the depth of knowledge required.

At the introductory level, dynamic written, oral or audio-visual presentations along with "drills" can aid the retention of simple facts. However, retention can be further improved when students, by way of example, are presented with a real-life problem that actually requires knowing these facts! Suddenly, motivation to learn increases, as those facts become relevant information (Schank, 1997, Merrill, 2000).

At a more advanced level, a student may be required to use previously learned facts (or learn new ones) to assess situations and make decisions. For example a farm advisor, recommending a treatment for the disease of a farmer's crop must have a high level of knowledge of plant pathology, entomology, nematology, weed science, soil science, plant physiology and meteorology and how these relate to the growth of healthy plants. Of course it is not enough to just know facts about these disciplines. The practitioner must integrate them into the decision-making process.

These integrative skills are not always easy to learn in the classroom environment. They are learned best through simulations and role-playing, where a task or problem is presented to the student and they must work through it. This can be accomplished in many ways. A simple way is to outline the problem or task, some of the factors related to it, and then ask for an assessment. However, the more the student can be immersed in problem solving, the greater the opportunities for integrative learning. For example, if the student has to gather the necessary facts through interviews, tests and observations, then the problems or tasks can seem so much more "real" and the learning experience richer. These kinds of problem-based simulations have been termed "Goal-Based Scenarios" (Schank, 1993). The modern computer, with its integration of all forms of media, can provide a "virtual reality" in which to work through such scenarios.

Goal-based scenario teaching is an approach, rather than a particular computer-based package. However, using computers to deliver

such scenarios can provide many benefits. Firstly, media (text, sound, video and pictures) can be integrated into one package. Secondly, students and their investigations can be monitored. Thirdly, a navigation structure can be enforced, so students can be "stepped-through" a pathway if required. Fourthly, in a networked environment, students can work their way through a scenario in teams even at a distance.

Third generation languages or multi-media authorware can be used to construct goal-based scenarios. However, unless software has been specifically designed to construct, present and manage goal-based scenarios, creating one requires much programming skill and effort. Examples of a few specific programs for this very purpose do appear in the literature (Riesbeck, 1988), but seem to be confined to "in-house" or contract training applications, rather than being available for teachers generally.

CHALLENGE is a goal-based authoring and delivery tool, which can be used to deliver these particular type of learning components across networks within a structured learning framework such as that supplied by TILE (see below) or, alone without such frameworks. It can also be used as a non-networked, stand-alone application.

Scenarios developed in CHALLENGE are flexible and have many uses. For example, one could be used with children simply to illustrate a problem with pollution, which could then lead on to a teacher-mediated classroom discussion. Another could be used to step a veterinary student through the diagnosis and treatment of a sick dog. A scenario could be used to test the diagnostic skills of a doctor attending a refresher course by presenting a "mystery" case. Another example may find use in training sales staff as to the appropriate response to an irate customer. Students taking a distance course could explore a scenario as a team project over the Internet. There are many possibilities.

Some History

CHALLENGE grew out of an earlier stand-alone program "DIAGNOSIS for CROP PROTECTION" (Stewart *et al*, 1995, Stewart, 2002). This program used an "Adventure game" metaphor to train students in the diagnosis of plant diseases. Students were placed the middle of a grower's field, orchard or greenhouse where they were then required to make observations, interview the grower and collect specimens for later analysis. In the simulated laboratory, they could test these samples for various things or try to extract a pathogen from the tissue. Once the investigations were complete, the student had to provide a diagnosis, a justification for their diagnosis and a recommendation. The tutor marked these. Furthermore, an associated program allowed tutors to easily construct scenarios.

In 2000, DIAGNOSIS for CROP PROTECTION was rewritten from scratch for the Win32 platform and renamed DIAGNOSIS for CROP PROBLEMS (Stewart *et al*, 2001). In DIAGNOSIS for CROP PROBLEMS, as with its predecessors, scenarios are constructed by the teacher and/or domain expert in a *scenario builder* and saved as a special data file. A *scenario player*, operated by a student or teams of students, reads these files to explore the scenario. New features for this version include a totally revamped user interface, a much more flexible template system providing support for hyperlinks and rich formatting options.

CHALLENGE takes the program one step further, allowing the construction of scenarios in any domain, and usability across the web. Part of the work was done in the context of the Technology Integrated Learning Environments (TILE) project (Gehne *et al*, 2001). The aim of the TILE project is to develop an integrated system for the management, authoring, delivery and (student) monitoring of education at a distance. Essentially, a CHALLENGE scenario would fit as a learning object in the TILE framework, although it could also be used alone or any other virtual learning environment.

Both the builder and player use part of Microsoft Internet Explorer v5 and above, and this program must be present on the computer for the program to work.

THE SCENARIO PLAYER

Figure 1 shows a screen from the CHALLENGE player, during a plant disease scenario. Using a familiar frames-based navigation structure common on web pages, a student can click on icons and hyperlinks representing objects, tasks or locations on the left-hand pane. Once clicked, these links expand to reveal further links, enabling other tasks to be undertaken relating to the activities, locations or objects presently under investigation. Using this navigation structure, students can examine objects, interview people, conduct tests and undertake research. Some menus may actively guide students through a process and conduct multi-choice tests.

The main pane on the right hand side shows the results of these activities. It displays hypertext, and so can include pictures or links to other disk-based or Internet resources. It may also show a fill-in form or the results of submitting such a form. Video and sound is currently achieved using hyperlinks, which when clicked activate whatever media players are installed on the system.

The top right hand side pane is reserved for special tasks and backwards navigation. A "more info" link is also present, which can present tutor-supplied clues related to that particular screen.

Objects can be collected in the scenario and transported to different locations. One might do this in a crime or diagnostic scenario for example, where evidence must be transported back to the lab for "forensics".

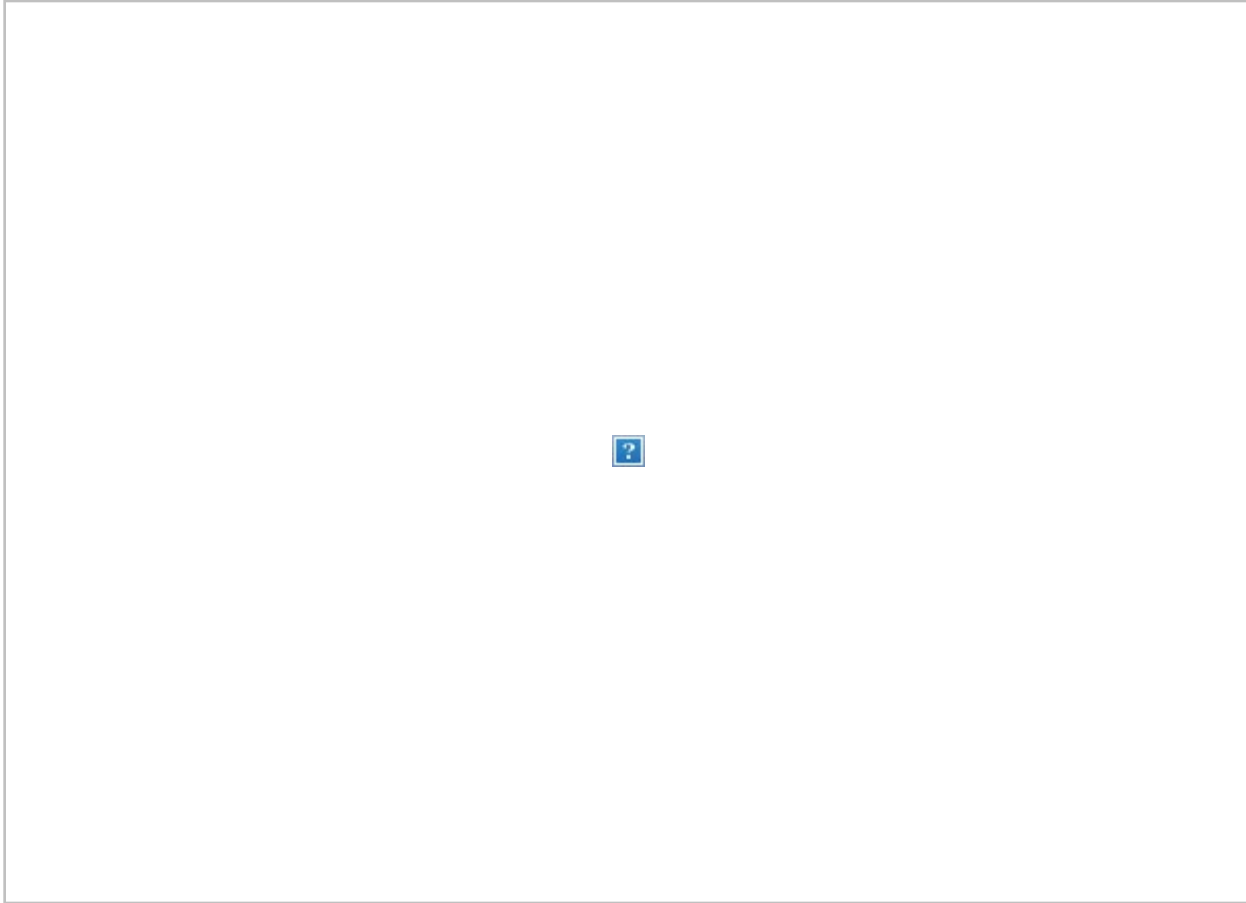


Figure 1. The scenario player showing a plant disease exercise.

Most scenarios will have some kind of "Report" icon on the top navigation pane (in the example scenario shown in Figure 1, this is called "Final Diagnosis"). When clicked, students will be asked to supply an assessment or diagnosis of the situation, perhaps with a justification and recommendation. Their input will get saved as an encoded file either locally, or on a server. At a later date, the teacher will access this file, perhaps for grading or comment. As well as the student input, the file will contain a time-stamped log showing the student's navigation through the scenarios. This can be used for analysis the steps taken to solve the scenario problem. Also, the student file includes a debriefing, pointing out the significance of certain observations and giving the domain expert's assessment of the scenario. Furthermore, the *Scenario Builder* (see below) allows the tutor to tailor the debriefing according to what the student did (or didn't) do. This debriefing can be available immediately on the screen after the student completes the exercise, or it can be delivered appended to their report, once the teacher has assessed it.

The CHALLENGE player takes two forms. One is a stand-alone application written in C++, which can be installed on whatever machine (or machines) a student might use. The other is a web-based Java servlet, which can take a CHALLENGE scenario, and serve it to a web browser.

the scenario builder

The user interface

The authoring window is split into three parts (Fig. 2). The left hand side contains a series of *nodes*. It is these nodes that are represented as navigation links in the player. The main right pane is an *edit screen*, and this would normally hold the HTML content viewed in the main screen of the player when the node representation is clicked. The top left-hand screen contains selection and input boxes, which determine the properties and behaviour of each node.

When populating a scenario, the author first creates a number of hierarchical nodes in the left-hand pane, similar to the way one might construct a directory tree in Windows Explorer. These nodes represent places (usually the top nodes), objects, information sources (e.g. people), or perhaps tests of some sort or another. Special design templates applied to nodes (see below) allow students

to collect objects and transport them from room to room, sequence activities, submit a final report, receive a de-briefing or take a multiple choice test.

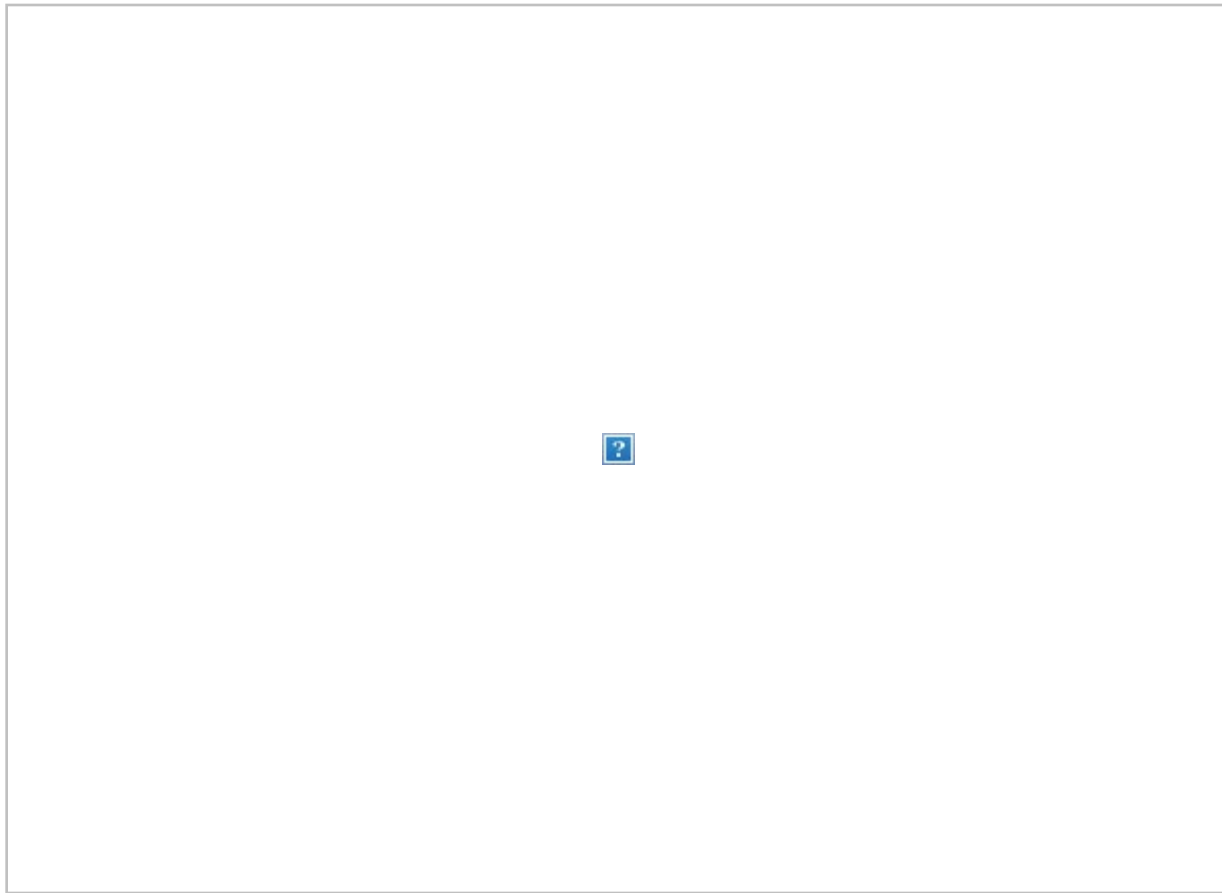


Figure 2. The scenario builder showing a plant disease exercise

Node behaviours and properties

Design templates

The behaviour of a node, both in the player and builder, relies on whatever *design template* (Table 1) is applied to it. Templates are written in HTML and a proprietary markup language. The design template can also affect the number and nature of the edit screens available. Child nodes generally inherit the design template of their parent unless set otherwise.

Design templates are a powerful concept as they determine behaviour of nodes and so make the program quite generic. In the Player, the interface is determined entirely by the design template of whatever node is being shown.

Table 1: Design Templates currently available in CHALLENGE

Template	Purpose
Main	The default template, used on the root node and for most of the nodes. In the Player, its role is to simply present a screen when its icon is clicked on the left-hand side. It contains two edit screens, Main (where the hypertext the student sees goes) and More Info. The More Info screen can provide optional guidance and assistance to the students. It is shown to students when they click on the More Info icon at the right hand of the top frame in the Player screen.
Collected Items	Provides the functionality to show collected items within the player.
Collect	One node with this design template applied should appear immediately under a

	collectable object in the scenario. When this node is present, users will be able to collect the object immediately above it, and all its child nodes (which may include tests etc). It has one edit screen; the message to be shown to students.
Referral	Used where objects are taken (or referred) for specific tests and a result is returned. The things being tested FOR appear as nodes underneath the node with this design template applied. The referral node has five edit pages, dealing with the heading, the heading when the results are returned, the type of substances, any discount being applied and More Info. Any node created underneath a referral node is assumed to be a subject of the test. They are automatically given a child referral design template.
Focus	Used when a student is required to undertake a multiple-choice test. The suitability of each choice is explained once the test is taken. During a scenario, these tests can be used to "focus" a student down to a short list of suitable options. The focus node allows a heading to be written, specifies the maximum choices a student can have and allows for a results heading. Each focus child node allows an explanation (which differs depending on whether it was selected or not selected) of the option concerned.
Sequence	Allows sequencing. It is used when the tutor requires the student to "step through" a series of nodes in order. Essentially, all sub-nodes under a sequence node are stepped-through from top to bottom by the use of a "continue" button in the main screen when the scenario is being played. Nodes do not appear on the left-hand side until they are visited.
Report	Provides a fill-in box the student can use to submit a report. The author can supply a debriefing and determine whether or not the student should see it immediately after the exercise, or whether it simply should be appended to the student disk file.

Prerequisites

Nodes have a prerequisite property, which can make the entities they represent invisible in the scenario player, unless they are under a particular node. This is useful when objects can be collected, and are carried from room to room (essentially from node to node) and tests applied to at one location only (e.g. a laboratory).

Icons

These can be applied to individual nodes. Nodes are represented by these icons in both the Player and Builder.

Visible on Left

Nodes with this property set are shown by default in the left hand navigation pane of the player.

Important Clue

There are two tabs associated with each node, which can be filled with text to be shown at the debriefing. One tab, "important clue found" contains the text to be shown if the node is visited, the text in the other "important clue not found" to be shown if it is not. If the author feels that this action or observation reveals something significant, then they can explain it here and congratulate (or not) the player for undertaking that task.

Cost

All nodes can have a cost associated with them. A test might cost money to undertake for example. In the player, the cumulative costs are shown in the top left-hand corner.

CHALLENGE and the Web

Why a Web-based Scenario Player?

It was decided to extend the versatility of CHALLENGE by creating a Scenario Player which would be allow scenarios to be used

across intranets and the Internet. Only the browser would be needed on the client machine. This would make it easier for distance students to use the scenarios, as no extra software would need to be installed on their computers. The scenarios would be located on a server hence ensuring the latest version was available (and all students were using the same version). Also, student input and logs could be held on the server, so teachers had easy access to them for marking and analysis respectively.

Finally, a web-version would allow groups of students at a distance to collaborate in problem-solving by working through a scenario together.

Client-Server Architecture

The CHALLENGE web-based Scenario Player is implemented as a Java servlet. Servlets are programs that run on a web server and build web pages. Requests and responses are both in XML format, thus enabling the leveraging of existing XML parsers. Requests (sent from the client) contain user action information (clicking a link, typing in text, etc) while responses (sent from the server) contain information about the scenario (text, images, etc). Requests are logged on the server in order to track the user's progress (Fig. 3).

At the time of writing, the web-based Scenario Player is still under development. Currently, our design calls for the client to periodically request updates from the server. It is unknown what kind of performance impact this will have in a group session.

CONCLUSIONS AND Future developments

CHALLENGE is an authoring package for constructing and presenting goal-based scenarios. As such, it is a useful tool for facilitating problem-based learning. The Scenario Builder program and stand-alone Scenario Player are fully functional and have been beta-tested. A web-version of the Scenario Player is being developed which will add features to assist the use of the product in distance education.

Once the web-based Scenario player is complete, it is hoped that the software could be further developed in two areas. Firstly, it would be useful if the software could respond to student behaviour as they worked through a scenario. Although hints can be supplied, these are passive. Some kind of active persona, which can comment on a student's direction, might be helpful. Secondly, it would be useful to build a "time" element in the scenarios. Presently, the scenarios are very much like a snapshot. Although the user can explore the situation in order to assess it, they can't actually alter it by their decisions. Some kind of adaptive behaviour within the scenarios would take the system to another level.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support of the New Zealand Government's New Economy Research Fund (NERF) for its contribution to this work. The authors would also like to thank members of the TILE team, for constructive criticism of this document.

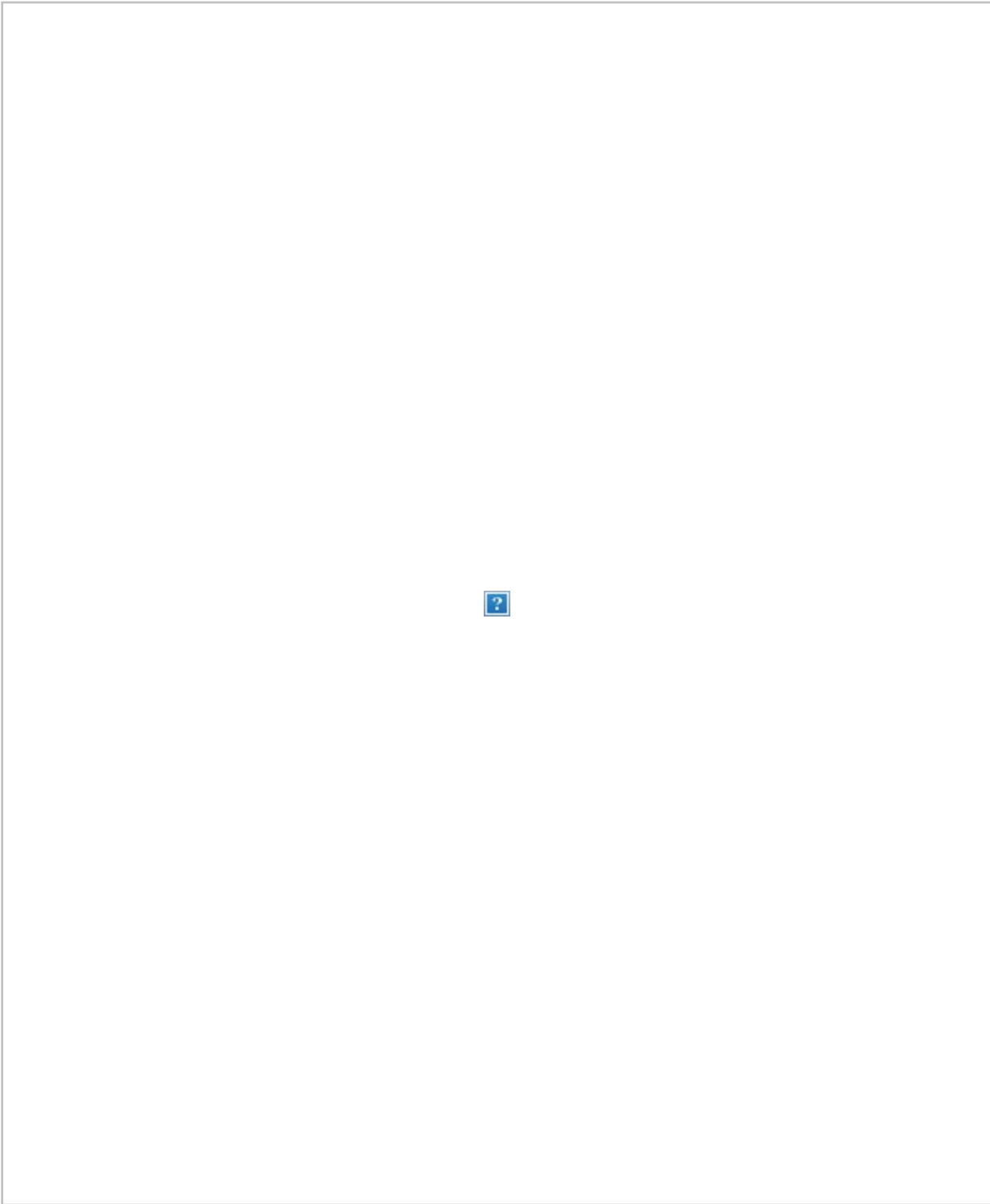


Figure 3. The Client-Server Architecture of the Web-based Scenario Player

REFERENCES

Gehne, R., Jesshope, C.R. and Zhang, J. (2001) Technology Integrated Learning Environment - A Web-based Distance Learning System. *Proceedings of IASTED International Conference 2001, Internet and Multimedia Systems and Applications*. Hawaii, USA. 1-6.

Merrill, M.D. (2000) Does Your Instruction Rate 5 Stars? *Proceedings of IWALT 2000: International Workshop on Advanced Learning Technologies*. IEEE Computer Society, U.S.A. 8-11

Riesbeck, C. (1998) INDIE: An Authoring Tool for Goal-Based Scenarios, [Online] Available

<http://www.ils.nwu.edu/~riesbeck/indie/> . (Feb 1 , 2002)

Schank, R. C. (1997) *Virtual Learning: A Revolutionary Approach to Building a Highly Skilled Workforce*. MacGraw-Hill.

Schank, R., A. Fano, B. Bell, and M. Jona. (1993) The Design of Goal-Based Scenarios. *Journal of the Learning Sciences* 3:4. 305-345.

Stewart, T.M. (2002) Diagnosis for Crop Protection [Online] Available
<http://www.diagnosis.co.nz> , (March 1st 2002)

Stewart, T.M., Kemp, R. and Bartrum, P. (2001) Computerised Problem-Based Scenarios in Practice – A Decade of DIAGNOSIS. *Proceedings of ICALT 2001: International Conference on Advanced Learning Technologies*. IEEE Computer Society, U.S.A. 153-156

Stewart, T.M., Blackshaw B.P., Duncan, S., Dale, M.L., Zalucki M.P, and Norton G.A. (1995) Diagnosis: a novel, multimedia, computer-based approach to training crop protection practitioners *Crop Protection* 14: (3), Elsevier Science Ltd, U.K. 241-246