TOWARDS THE DYNAMIC PUBLICATION OF MULTIMEDIA PRESENTATIONS – A STRATEGY FOR DEVELOPMENT

Professor Chris Jesshope
Department of Computer Science,
University of Hull, HU6 7RX, UK
C.R.Jesshope@dcs.hull.ac.uk

ABSTRACT

In this paper, interactive multimedia is used to transform a complex, design-oriented coursework assessment in Computer Science into on-line mode, using interactive multimedia. The media in question has already been used in this course in providing a didactic audio-visual presentation, which replace formal lectures. In this recent experiment, new features in the authoring tool have been exploited to give a more constructive approach to learning. The benefits of this approach are twofold; it provides automation to design-based assessment and it engages the learner in design-oriented work with adaptive delivery of additional information for those with difficulty in completing the assignment. What is shown in this paper is the feasibility of the solution and the formal requirements to provide interfaces and additional tools enabling the end user to adopt the principles of learning design to design-based assignments.

KEYWORDS

Interactive, multimedia, learning design, automatic assessment.

1. INTRODUCTION

It is important to put this paper in perspective, in order that the reader will know exactly what benefit may be gained from reading it. It is not a paper about models of learning; there are already a plethora of instructional design models (Ryder 2003), such that the selection of a suitable one must be extremely difficult for anyone wishing to develop pedagogically sound eLearning materials. It is not even a paper, which follows a particular instructional design model in developing eLearning materials, although it attempts, by way of example, to transform an exercise in generative learning (Osborne et. al. 1983), previously used in small group tutorials, to an individual assignment using hypermedia. Of course in doing so, some of the benefit of this approach is diluted. This paper is rather, an attempt to make some sense of the relationship between technology and pedagogy from the viewpoint of a designer of eLearning tools. We have an interest in developing tools that bring the power of technology to the educators in generating on-line media for teaching and learning. Notice the emphasis here, we believe that we must empower the educators rather than making them hostages to technologists, which means simplifying the tools and concepts embedded within them. The paper follows this ideology and presents its thesis in a form amenable to the spectrum of interested parties, involved in learning design.

On the one hand we have this plethora of pedagogical models and on the other we have languages for modeling instructional design, such as Educational Modeling Language (EML) from the Open University of the Netherlands (Koper 2001) that allow us to describe in a formal manner, the conceptual complexities embedded within any of these models. However, current tools do not give the educator support for such richness of content and at best we are stuck with a simple pedagogy built into the environment or tool we are using. This is not surprising, as in 1986, Doyle emphasised that a classroom is a highly complex environment with features of multi-dimensionality, simultaneity, immediacy, unpredictability, publicness and history. Using technology as a substitute for the classroom adds another dimension to an already complex problem and it is not surprising that the early adoption of technology did not attempt to create or allow for the same complexities found in the classroom to be transformed into on-line media. The implementation of the technological classroom has overwhelmed instructors with its steep learning curve and this has led to the development and adoption of simple learning environments, such as WebCT, Blackboard, Angel, Silicon Chalk, etc. Indeed there is also a plethora of simple learning environments, again making choice difficult but at an institutional level. These tools have provided some relief to the problems of display and even assessment functions but at the cost of constraining the user into a particular pedagogy or learning style, which is built-in. Although this may provide a shallow learning curve to the educator, it does ultimately limit what can be achieved.

Since the introduction of these tools, educational modeling languages have progressed significantly and are now being standardized, e.g. IMS learning design (IMS 2003a). It is now possible to model and capture some of the complex and rich interactions found in the classroom. The problem now, is that there are no simple tools, which support these models. Such tools must allow for **any** instructional model to be adopted and must be able to conceptual and populate material for the different roles and environments it comprises. In short, they must capture the complexities of these models and scenarios and then adaptively or dynamic publish this material to the actors involved.

In order to explore the problem described above, this paper describes an experiment undertaken to capture some complex interactions that previously occurred in a classroom environment and to transform these into a dynamically published, on-line multimedia presentation using an existing tool. The goal of this work was to develop an on-line assessment that enabled a complex design assignment to be assessed automatically. The assignment had previously been assessed by written report and the grading of this report was a very time-consuming task. As outlined in Section 2 below, this experiment exhibits some of the characteristics of dynamic publication in multimedia and was an aid to clarifying the ideas presented in this paper and in developing a strategy to enhance the tools and their future development. The assignment was taken from a course given by the author (Jesshope 2002) and has used the same tools to deliver the course content on-line. Previously, it used small-group tutorials to provide support for the design assignment in the time freed-up by on-line content delivery. This method of delivery is described in more detail in Section 3.1 and also in Jesshope (2001).

One of the major issues in small-class teaching is that the teacher is constantly assessing and adapting the presentation to meet the needs of the students. The objective here is to capture this and abstract from it an authoring process that can be adopted by the people developing and presenting the courses. Currently the most widely-used tools do not support this process, even though they may incorporate test authoring and assessment within their toolset. They simply do not have the capability to dynamically publish an environment. The work is motivated by emerging standards like EML (Koper 2001) and IMS learning design (IMS 2003a) and analyses how learning design can be incorporated into existing multimedia tools to provide dynamic publication of material.

2. DYNAMICALLY PUBLISHING MULTIMEDIA

2.1 Analysis of Dynamic Publication for eLearning Material

Dynamic publication can be defined as the publication of content according to a parameterized model, such that the presentation published to one individual may not be the same as that published to another. That model may describe prior knowledge or it may describe the preferences of the user. In a multimedia presentation, dynamic publication can also be defined as a presentation, which is defined by the choices made by the viewer during the presentation. In the sense that a story published on CD as a branching narrative will be published in many different forms according to the choices committed by the viewer. To differentiate this kind of publication from true dynamic publication we will refer to it as *Adaptive* publication.

One of the issues in a branching narrative structure is the constraint (or lack of it) in the possible stories covered by the dynamically unfolding of the storyline. On the one hand, in a true story telling environment, there is the desire for emergent structure to be present, e.g. in the Automatist storytelling system (Murtaugh 1996). On the other hand, a more deterministic model is proposed by Galyean (1995) in his doctoral thesis *Narrative Guidance of Interactivity.* He uses a river metaphor to describe the experience-guided interactivity.

In a river, Galyean notes, there is a "continuous flow," in which the viewer "steers" and generally never encounters "dead-ends". This dichotomy reflects the two extremes of the use of a branching narrative. In a storytelling device there is a desire to avoid closure for maintaining interest; "Closure is boredom" (Murtaugh 1996). On the other hand, in authoring learning material, particularly technical material, closure is a requirement of the process and the state space created by that closure can be critical in making the authoring tractable or not.

Perhaps one of the key questions is whether we can produce tools that support dynamic publication but do so without any requirement for programming on the part of the author so that authoring can be undertaken with some computer experience but no prior knowledge of either HTML, Java, scripting languages or XML. XML is one of a number of technologies that can be used to support dynamic publication and is used to define both EML and IMS learning design.

There is also a difference between publication in HTML and publication in multimedia frameworks. There is already a range of products available that allow the customized publication of standard web pages using static text and images only, through the use of personal web portals. For example, IBM's WebSphere Portal Server is one of many portals that let you build custom sites personalized to a particular user or class of users. Like all portal solutions the user must idfentify themselves, usually through log-in, and information stored about their preferences is used to customize their experience. For multimedia publication however this is a relatively new field and work in this area is of an exploratory nature.

2.2.The Tools and Strategy Adopted

The toolset being used in this work is one that has been developed over a number of years and in a number of institutions. It has been designed by educators in order to make multimedia authoring accessible to the educators. However, the AudioGraph (Jesshope et. al. 1997, 1998, Gehne et. al. 2000) is still able to provide a rich and interactive media framework for eLearning. Recently, the AudioGraph has been further developed within a collaborative research project in order to support interactive multimedia (Gehne et. al. 2002) and the resulting authoring tool has now been spun-off into an independent company, NZEdSoft Limited. This new tool is able to capture audio graphic presentations and adaptively deliver them in response to interaction from the student. The version used in this work is available as a web download (NZEdSoft 2003) and has been used to create a media-rich, interactive exercise that allows students to be assessed in a design-based assignment, which is taken entirely online.

AudioGraph supports dynamic publication using linked media presentations. It does not offer any higher-level tools that are required in order to support the assessment functionality nor the high-level planning and design of the pedagogical scenarios. Assessment can be achieved however, by a low-level analysis of web-logs and this is the method used in this work. Although this method is sufficient for the evaluation of the tool and the techniques used, it is not appropriate for the wider use of the tool in this context and one of the results of this work is a detailed specification of the enhancements to the tools in order to create dynamically published interactive assessments. Our collaboration and close relationship with NZEdSoft Limited give us an opportunity to provide input to the company in order to extend the tool to achieve these goals. We are also working collaboratively with NZEdSoft on further research into higher-level tools.

A strategy has to be adopted in order to until learning design within tools such as AudioGraph. A strategy is necessary in order to develop additional high-level tools that would enable learning scenarios to be developed in a conceptual form by pedagogical experts. This would enable the results of this scenario planning to be adopted by any tool that accepted the appropriate standards as input. These tools would develop templates in XML using schema based on IMS learning design or a subset of EML compatible with it. In this strategy, the template defines the structure of, and conditions for publication of, the components of the multimedia presentation. AudioGraph or other tool would then have to accept this template as input and construct a document according to that structure for population by the domain expert.

The major problem and difference alluded to in section 2.1 above concerning the dynamic publication of multimedia documents is that media tools usually comprise both authoring and delivery components, where the latter is usually a plug-in. For example the AudioGraph plug-in is used to deliver sequenced audio graphic presentations. We must therefore consider the publication as well as the authoring of the document in terms of the conceptual design. A delivery template must therefore capture the structure of the interaction between the learner and the presentation. For AudioGraph the structure is entirely defined by the links between what AudioGraph calls *Episodes*, where an episode is a file defined by a URL, which contains a linear component of the multimedia presentation. In addition to the structure of the presentation, this template must also capture any marking scheme, which assigns weights to each of the links taken and possibly any dynamic links required if the presentation is to be displayed according to any student preferences. This delivery template is generated by the authoring tool and, together with student preferences, will provide dynamic links to the various episodes (files) generated.

The authoring template captures the pedagogical scenario and can be reused in any tool that supports it. The delivery template is a part of the presentation published by a particular tool and defines the possible presentation pathways through the multimedia presentation. It also contains assessment information, in the form of a marking scheme. The media delivery plug-in or software uses the student's preferences and actual interactions to present an appropriate presentation and generates an assessment based on the interactions made. Test results would also need to be generated. This will provide for compatibility with existing learning environments. The workflow and dataflow for this process is given in Figure 1. This strategy facilitates the seamless integration of the material into the various systems, including authoring tools, web servers, learning management systems etc. One can imagine pedagogical experts creating a variety of templates, which would then be populated quite simply by domain experts using compatible tools, in order to provide the interactive multimedia assessment.

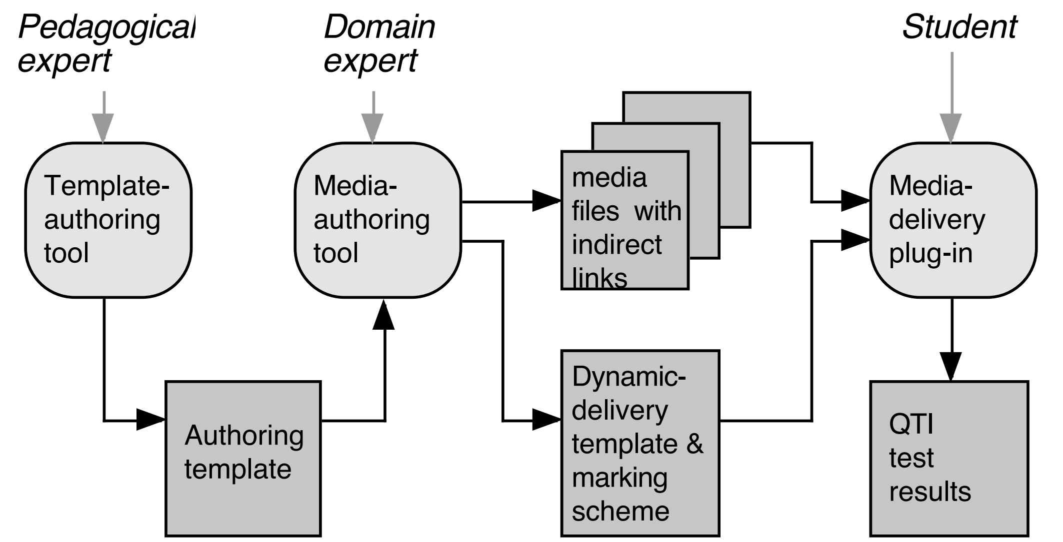

Figure 1. A diagram showing the generic workflow in creating media-rich, interactive assessments using multimedia tools, of which AudioGraph is just one such example.

In the experiment reported in Section 3, the template author and domain expert were one and the same person, namely the paper author. The authoring template was created within the AudioGraph tool and was not encoded in EML or IMS learning design, as the tools do not support the workflow specified above. The delivery was also statically linked using AudioGraph, where all possible paths were known a-priori. This means that the any adaptivity is restricted to the input provided by the student on delivery and not by their preferences or prior knowledge, which are assumed to be a known and constant.

The aim of the project was to gain knowledge through the implementation of a particular assessment and to encapsulate this expert into conceptual templates and template specifications for later reuse. One of the major goals was to generate specification of interfaces to both tools and standards in order to provide for future development of the AudioGraph and higher level tools that support pedagogical design and which conform to this model of dynamic multimedia publication.

In the assessment exercise described in Section 3, the features of the AudioGraph that were exploited include the ability to create audio visual presentations using images, vector-graphic annotations and voice clips, which were sequenced and published as media files. These files are played back using the AudioGraph plug-in in a web-browser environment. Unlike regular HTML browser content, AudioGraph has the significant advantage of reproducing the precise registration of all media elements in a cross-platform environment. This includes text, images and vector graphics, which are rendered precisely as authored, independent of playback platform. A second distinct advantage is derived from the AudioGraph authoring principle, which gives the precise synchronisation of the time-based media, again in a platform-independent manner. Synchronisation is achieved regardless of the speed of the playback computer and regardless of Internet connection speed. With both of these advantages, the author can guarantee delivery of the media exactly as authored to every student, which was extremely important in the design of this multimedia assessment, which requires precise registration of all media elements (see Figures 3 and 4 for examples).

It is necessary to understand the recent developments to AudioGraph that allow for user interaction through the introduction of hyperlinks within its sequential media. The units of an AudioGraph presentation are called episodes and each episode is a strict sequence of media elements that is decoded and displayed by the plug-in. Each episode is stored in a single file and because the file format is decomposed into small chunks, the plug-in can play the presentation as it is downloaded from the server and can do this without a specialized media server. This gives click-and-start characteristics from a regular web server. This is one of the advantages that the AudioGraph media has over media files, where often the complete presentation must be downloaded before it can be presented.

A hyperlink in AudioGraph media links the end of one episode to the start of another. The semantics of a link are subtly different from hypertext links, as the link can be considered as the dynamic concatenation of two sequential episodes, where the choice of which episode is made by the learner dynamically. Hyperlinks are activated only when the presentation has stopped playing, either under the student's or the author's control. The full episode concatenation occurs even in the case where the linking Episode has not yet been completely played (or downloaded). Therefore the result of linking episodes is a linear multimedia presentation that has been dynamically published according to the choices made by the learner at various link opportunities. This is illustrated in Figure 2 and is the basis for dynamic publication in Audio-Graph and the foundation to the experiment described here.

Hyperlinks within time-based media and in particularly the method adopted in AudioGraph, provide a very practical way of controlling access to information and presenting it on demand. It will enable us to support some of the complexity and structure of the tutorial scenario described in section 3 below, where parts of the assignment solution are constructed interactively, involving feedback and assessment from the tutee. In the work described here, the links are static, such that if a hotspot is selected it will always link to the same episode, regardless of student profile or background knowledge. Section 4.1 describes how this can be overcome with a small amount of further development of the tools.

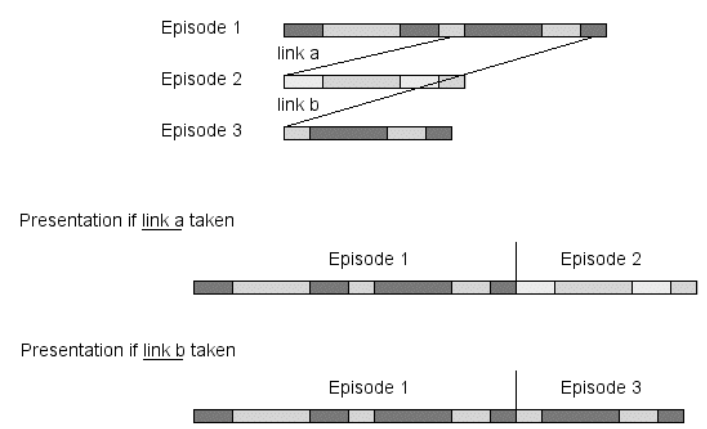

Figure 2. Link structure in an AudioGraph presentation showing dynamic publication of sequential multimedia presentations

3. THE MULTIMEDIA ASSIGNMENT DESIGN

This Section describes the experiment undertaken using the AudioGraph tool using the strategy outlined in Section 2 above. It demonstrates that at least some of the complex interactions that occur in a classroom environment can be captured and presented on-line. The experiment models a coursework assessment exercise, previously undertaken in small group tutorials and assessed by written report, which has been transformed into an on-line assignment.

3.1 Prior Teaching and Assessment Scenario

For two years now the author has taught an undergraduate course in computer architecture using a combination of on-line multimedia lectures, which replace the formal lectures, and small group tutorials to introduce and support a strong element of design into the course assessment (Jesshope 2002). Teaching design in computer architecture is notoriously difficult and time consuming, and recent, on-line activities in this area (Djordjevic et. al. 2000) have only provided a rather narrow, domain-specific solution to this problem, and without any automated assessment of the activity. There are many problems in completely automating the task of teaching and assessing design-based subjects. Design is, by its very nature, open-ended, and although there may be a single optimal solution, many student design decisions will cause the design space and assessment effort to increase rapidly. Most automated, online quizzes fall into one of a number of simple formats, such as multiple choice, ordering or matching, which do not provide a solution to the design assessment problem. Some alternative to existing approaches is required, which allows an adaptive approach and which steers the student through the design space providing feedback and support and which uses media-rich solutions to implement the on-line question and answer, in this case both graphical and tabular display and input.

In both years that this course has run, assessment has been made by written reports. The assignment being automated in this paper required the students to design a datapath for an instruction set architecture and to give a detailed implementation of a part of the control logic required for their datapath design, see assignment 2 (http://www2.dcs.hull.ac.uk/people/csscrj/web-site-08236/assignments.html). This assignment is quite demanding for the second year class taking this paper and support was given in tutorials, where partial solutions were built up using input from the students and guidance or steering from the tutor, which attempted to minimize the design space. The time for these tutorial sessions was freed-up by the use of on-line delivery multimedia presentations replacing the formal lectures.

Quite a significant amount of the solution was built up interactively during these tutorials, however in both years, this failed to restrict the design space in the assessment and many of the students failed to deliver even the partial solutions created in the tutorials. This was both disappointing and frustrating because the assessment effort in verifying a large number of divergent designs was considerable. It was against this background that an assessment was designed, which would more actively steer the students through the design space using on-line multimedia. The problem faced in this experiment therefore, was how to represent that interaction, in which both tutor and student contribute to a complex design and how to assess the students' design.

In automating the design assignment the students were required to construct a block diagram of the datapath that implements the instruction set architecture provided and then to design the logic required to control that datapath. Obviously divergence in the first part of the exercise resulted in divergence and additional marking load in the second part as well, which although logically deterministic, is dependent on the datapath designed. The second part was automated by requiring the students build a complex truth table for the control logic and to derive and minimize the Boolean equations derived from that truth table.

3.2 Design of the On-line Assignment

A number of Question/Answer systems were evaluated before it was decided to use AudioGraph to solve this problem, including some the most popular systems either embedded in current VLEs or as stand alone systems. However, none could be found that provided the kind of flexibility required that was required to assess this coursework. The requirement was to present a complex graphical interface with interactivity built into the graphical environment within which the student could construct a design comprising a block diagram with connections between the blocks. This was required to be presented as an assessment with feedback and support from the domain expert. The only prior work that we know about that has the kind of flexibility required was from Shafarenko on an automated examination system and reported in (Jesshope and Shafarenko 1997). This system required coding skills in both Java and a scripting language in order to author the questions and hence was not suitable for our strategy. Therefore AudioGraph was used, which provided the rich interactive environment required but we had to consider the two other stages of the workflow defined in Figure 1, namely the high-level template editing and the delivery mechanisms for assessment.

The on-line assignment is performed in three stages. Students are first given a definition of the instruction set, which defines the operation of the required design; they have access to this prior to the on-line assessment. Some time later, they are required to attend a laboratory session and complete the design of a datapath that implements these instructions. This involves an audio graphic interface that allows them to build up the datapath diagram using simple mouse clicks to make connections between components. Following this, and within the same session, students are asked to design the control logic as a truth table from which they extract Boolean equations. This part is quite deterministic. They are given precise definitions of each of the control points and are asked to build a truth table with the instruction operation code as its input and the control signals as its outputs and then to extract and minimize the Boolean equations defined by that truth table. More abstractly, what is required is the ability to present and incrementally develop a complex diagram and a large tabular structure based on a succession of mouse interactions.

In previous years, preparation for this exercise was undertaken in tutorial sessions. Partial solutions were given and developed interactively with input from both the students and the tutor, and with the datapath design being developed on the board in response to interaction between the tutor to the student group. During these sessions the design was constrained by the tutor and the intent was to develop a partial solution in order to limit the design space, so that the final assessment of their reports was easier. This tutorial contribution is now captured as a part of the on-line assessment. The template used provides a branching narrative, where the students who make choices, which are not on the optimal solution path are steered back to that design. This steer was provided by short audio graphic presentations, describing the choices, the benefits of each and the reasoning behind the optimal choice. These presentations capture, albeit in a limited manner, the interactions that had occurred in the classroom. Eventually the student is taken back to the main flow of the narrative. Of course a student making correct or optimal solutions would not get this additional instruction. The assessment therefore also acts in a constructive way making connections between the concepts that they already have and those required in producing an optimal design.

The Truth table is developed in a similar manner, although this is deterministic for a given design. Even so, the exercise reinforces the methods the students need to undertake in order to complete their design. The only component not fully automated is the extraction and minimisation of the Boolean equations, which are submitted as text document using our standard on-line submission procedure. Automatic assessment of this would be possible but more difficult with AudioGraph. A simple Java applet to implement an equation editor, such as developed in Jesshope et. al. (1997) would be more appropriate.

Datapath design

To design the datapath, a graphical presentation is used. The students are initially presented with a set of components and are required to complete the datapath by making connections between the components and control points. The components of the datapath are drawn graphically using the AudioGraph vector graphics tools and labeled with text. For each connection the learner is prompted with an audio description of the instruction and what is required of the datapath. They also have visual clues to identify the target of the data source that they must select. This is done by clicking on an appropriate hot spot in the diagram. If they are successful, the connection is drawn for them and they proceed to the next instruction group. If they are not correct or if their choice is not an optimal one, they will be taken to an explanatory episode which may give a detailed exposition of the various options their merit and why the option they selected was not either not correct or not optimal. These descriptions use audio graphic components and of course refer to the partially completed diagram they are constructing.

What is important, in terms of simplifying the design of the assignment, is that the state space covered by the various episodes is minimized. Design closure is accepted as a consequence of this optimization. Otherwise a complex programmed solution would be required, which was not our strategic goal. Students are therefore returned from an incorrect choice to the optimal solution after instruction on design appropriate to the choice they made. In practice, this is a multiple-choice exercise but it is very complex, as it uses a rich audio graphic interface both for the development of the design as well as the feedback given in the event of some choices. In relation to the AudioGraph presentation, the student is selecting from different episodes at each choice that they make. If they deviate from the main flow of the narrative, automatic links return them to an appropriate point. The links to these episodes are hotspots created by the author in the diagram being developed interactively.

For assessment purposes, all choice points are weighted according to a marking scheme. The action of selecting the correct data source for the target requested, results in the corresponding connection being drawn, a weight being added to their score and the next request being made. This is continued until the complete datapath is constructed by their actions. An example of a choice point is shown in Figure 3, which presents the output from the AudioGraph plug-in during the execution of the presentation. The datapath is built up from a large number of user choices in this way.

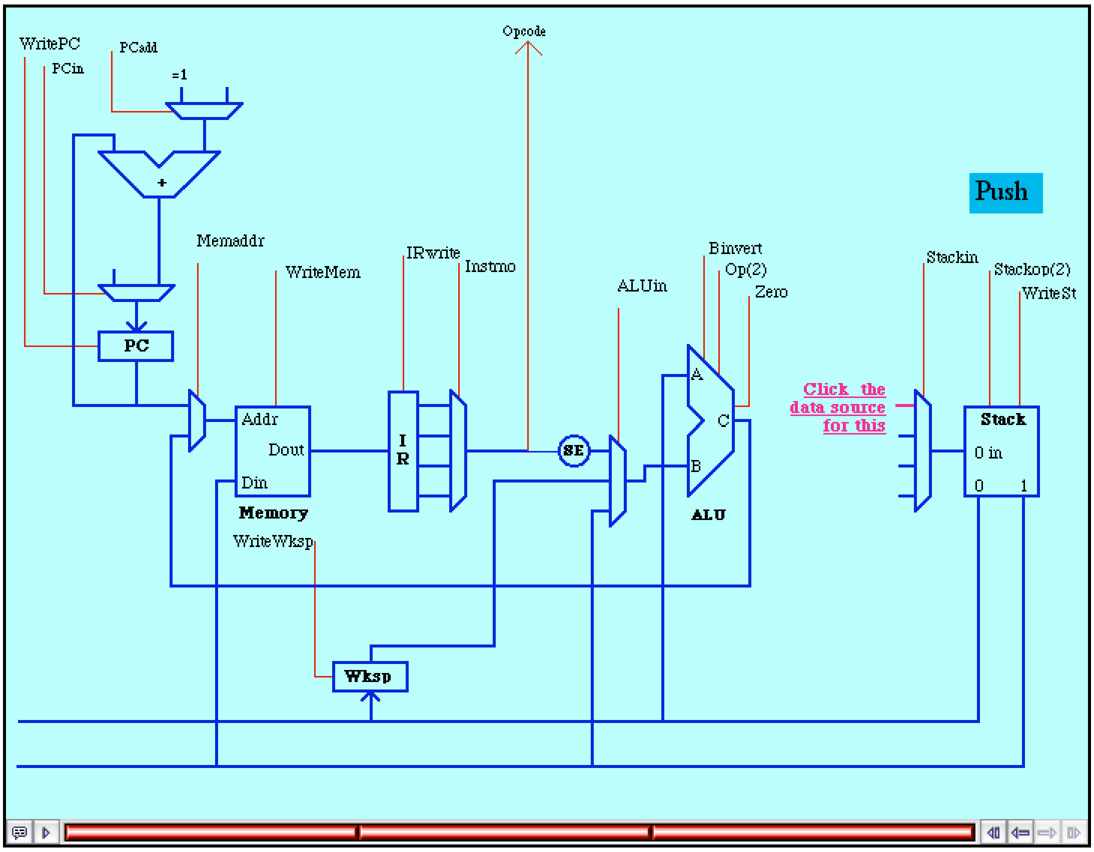

Figure 3. Construction of the datapath as a complex diagram during the students' assessment. The diagram is constructed from a large number of choices the one shown is for the "Push" instruction.

Of the wrong choices, some of these will be clearly wrong, some will be wrong but subtly so and some may be correct but not optimal and each has an appropriate weight, chosen by the author. The marking scheme thus recognises the spectrum of correct, incorrect or partially incorrect choices represented as links taken and the score for a given student, is the sum of the weights on all of the links taken in the dynamic publication of their presentation. In terms of our strategy, the marking scheme would be populated as a part of the domain-expert authoring within the AudioGraph tool and output as a part of the presentation. This marking scheme is read by the plug-in and the score accumulated in during the student's interaction. The plug-in accumulates the weights according to the marking scheme and generates a QTI compliant output file that is captured by the learning environment. This could be a managed learning environment or could be an open web-based environment.

The state space we wish to minimize is determined by the extent of possible choices in the presentation, with a new state resulting from each choice in the audio graphic presentation. Every state requires an episode to be authored by the domain expert. Although some state aggregation may in general occur, in this design-based presentation, each choice has unique characteristics. Our strategy described above is designed that stop this state space from exploding rapidly. A large number of choices have to be made in this scenario in order to make a rigorous assessment. In an open design, in which all possible solutions can be constructed, the state space would grew geometrically with the number of choices points. With just 10 choice points say and 4 choices at each choice point, an open solution would require 2^{20} or approximately 1 Million episodes to be authored! This is totally intractable. The approach adopted, which provides closure at each choice point constrains the state space to be linear in the number of choice points, i.e. at most 40 episodes must be authored. This clearly exposes the potential problems in such a design and explains the compromises made.

It should be pointed out that the correct path cannot be ascertained a-priori by the student, as the student sees just a single link from the HTML entry page and although the source of that page may provide the name of the first media file, they would be faced with a random choice of media files to select from even if they were aware of the naming scheme used. As the graphic is accumulated incrementally in each episode, no one episode can provide information that enables the student to cheat on the assessment.

Control unit design

Having designed the datapath, the students are presented with an overview of the control-unit design strategy and then asked to design a part of the control logic as a truth table, which is shown in figure 4. Students must complete the datapath design before they can continue to this part, in which they have to build the truth table and derive Boolean equations from it. The assessment again uses a multiple-choice selection and in this case, each choice adds just one cell to the truth table. Figure 4 shows a choice point in the partially completed truth table. It can be seen that at each stage, the contents of the required cell is selected from true (1), false (0) or don't care (--) states. Each control signal output is built up instruction by instruction and each wrong choice again provides feedback to the student, reinforcing the design methodology. While this presentation is being presented, they have access to two other presentations in different windows defining their task; one of these is the completed datapath from the first part of this assessment. This second part is does not have the potential divergence of state space found in the first part, as it is quite deterministic. Each choice has only one correct answer, although true and false can be substituted for the don't-care state at the expense of more work for the student in the minimization of their equations. The equations they generate however are quite deterministic due to the nature of logic. Over 200 choices are required to build this truth table in the assessment.

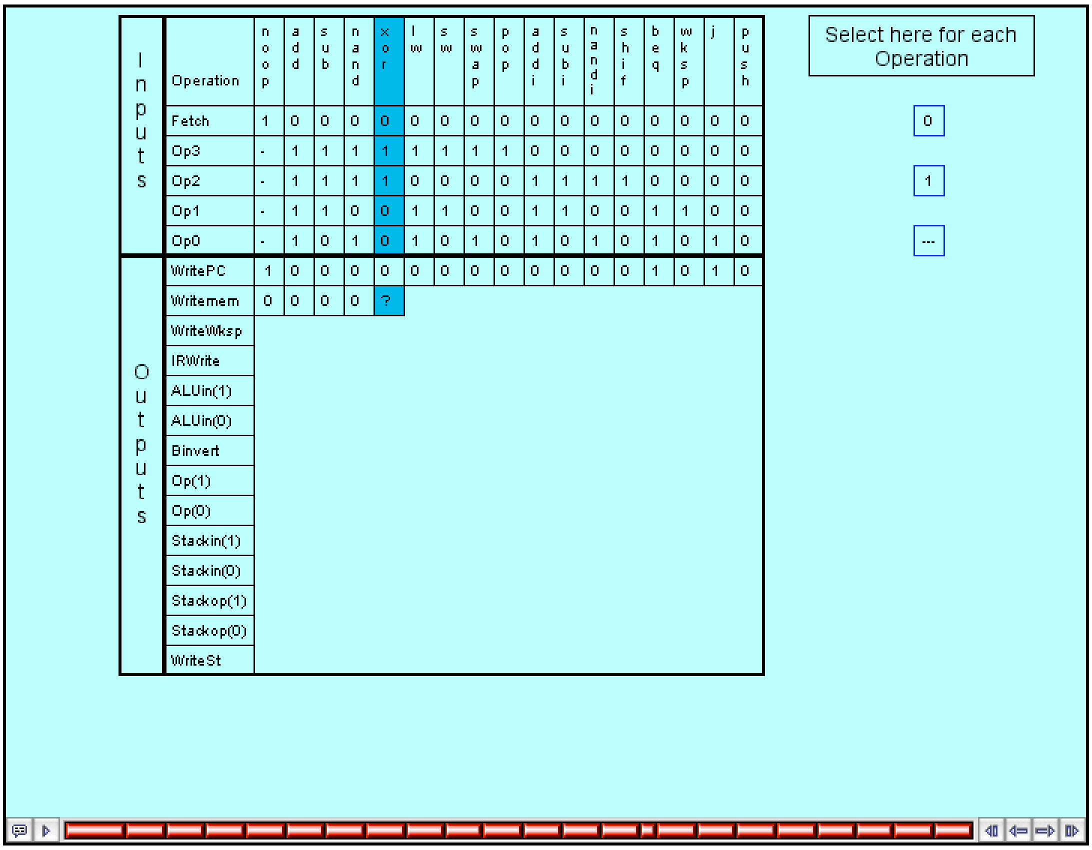

Figure 4. Construction of a truth table for the control logic from choices within an AudioGraph presentation

4. RESULTS OF ASSESSMENT AND INTERFACE DESIGN

4.1 Adaptive and Dynamic Publication

The prototype assignment has been evaluated by a small number of past students from the course that the assignment was prepared for. In this implementation, the assignment was designed with the current version of the AudioGraph authoring tool (Professional Version 2.1). The evaluation used the standard AudioGraph plug-in, which links between episodes statically and does not provide any test-log output. Assessment was achieved by analysis of links taken and recorded in the standard server log files. Although this solution is not ideal, there was a known mapping between the students taking the assessment and IP address of their computers. Hence all links taken by all students could be derived from the web log of the server, giving a score for the assignment using an analysis program, which had the marking scheme embedded within it.

The main results of this exercise however, is not the prototype but the insight it has given us in the specification of interfaces for higher level authoring tools for template editing and the integration with delivery engines including a range of learning environments. In order to understand the significance of this, we must describe the authoring and publication of AudioGraph presentations in more detail. It must be emphasized however, that although these interfaces were designed for AudioGraph, they are not specific and provide a generic interface for all such systems, which comprise an authoring tool capable of linking between or within its media files with delivery of that media via a plug-in or via another programmable delivery tool. Other examples might include Flash or Quicktime media or indeed any Java applet.

An AudioGraph presentation is held in a single file while it is being authored. This file contains all of the source media for each episode and the structure of the links between them. In this document, links are maintained and manipulated symbolically using the names given to each episode. The author creates and names episodes and links them using those names. This file is called the AudioGraph lecture document and it is the archive of a presentation. It can be opened, edited and saved. A web site can then be published from this document that creates a folder containing both html and compressed media files in the AudioGraph .aep format, which are read by the AudioGraph plug-in. The lecture document is the file format that has been used for the authoring template in this exercise. An example of a simple published quiz template can be found on the NZEdSoft web site. (http://www.nzedsoft.com/Demonstrations/Quizes.f/Index.htm). Such a template provides the linking structure and narrative flow and is populated by the domain expert.

In the future, the template will be created by a higher-level template editing tool (see Figure 1), and output as an XML file based on some predefined schema, such as that defining IMS learning design or EML. Such a template would contain symbolic links, which would allow dynamic publication using variables defined in the XML file. This would enable the dynamic publication of the populated presentation, based on learner preferences for example. It is relative easy to translate the XML template into an unpopulated AudioGraph template with a naming convention, which made each variable and value readily apparent. The author would then be prompted to populate each of the alternative choices to a particular style or preference. The schema produced would adopt one of a range of pedagogical styles offered by the template editor, which might be a subset of the schema supported in EML or learning design. These standards are very complex. For example the EML XML binding (Hermans et. al. 2000) offers a conceptual model for identifying metadata, roles, learning objectives, prerequisites, content (which include environments and activities), methods, plays and conditions. Conditions are the variables that define the options within a scenario. This specification runs to some 200 pages.

The AudioGraph authoring tool's symbolic links are only bound to a file name on publication of the media to a web site. Each episode is published as a separate .aep file, which is read by the plug-in only when linked to. The publication uses an automatic naming scheme for the files generated. HTML files and links within an episode are bound to these files using a local URL when the lecture document is published. The author is not aware of the naming convention and does not need to worry about it unless making a link from within another lecture document, in which case they must use an absolute link rather than a symbolic one. Static links are entered in the authoring tool as text strings defining a local or global URL.

AudioGraph, as it currently exists, only provides adaptive publication (as defined in Section 2.1) and is not capable of providing dynamic publication based on users preferences or prior knowledge. To enable dynamic publication the AudioGraph presentation must maintain its symbolic links when published to .aep files. These links would then be resolved at delivery time, when the person viewing the material instantiates the variables that resolve the choices in the dynamic publication. Thus the symbolic links defined within a template must be maintained through the authoring and publishing processes. To create symbolic links on publication requires that the publication not only produces the media and html files but also an XML file that defines the relationship between the symbolic links, the variables and the media files published. This flow of information is illustrated in Figure 1. The XML file used need not conform to an open standard, as it provides the interface between two proprietary tools. The mechanism used in publication is simple. The authoring tool would publish each episode as normal using the existing naming convention for media files. Links in the .aep files however, would remain symbolic. The binding between episodes is then made in the plug-in, which must translate the symbolic link tin o an actual file name based on the variables instantiated for a particular learner. This is stored in the dynamic publishing template illustrated in Figure 1. This file would also contain a marking scheme, which allows the plug-in to accumulate an assessment based on the weight associated with each link taken by a particular student. The results of the assessment would need to be output from the plug-in in some standard form to allow incorporation into existing learning systems. The only de-facto standard appropriate for this is the IMS QTI standard(IMS 2003b).

4.2 Student authentication

In order to assess a student we first have to know that a particular student is actually providing the interaction that we are monitoring in order to provide that assessment. This process is called authentication. This general area is still an open problem and we can only consider some aspects of it here, as it is outside of the scope of this paper. Authentication can be achieved in a number of ways. What is called passive authentication requires the user to enter a private user name and password, which are checked against stored information. It is assumed that the user will not divulge this information to another party and there are usually good reasons for the user not to do so where passwords are used. Alternatives to passive authentication use some form of biometric analysis to identify features of the user, which are constant and unique, such as fingerprint or retinal scan. Ideally biometric authentication would be required for formal assessment but it is not clear that even this is strong enough in a distributed environment. Just as there is no way to guard against the disclosure of a password. Biometric authentication could be provided prior to a third party performing the interaction with the on-line assessment. Unless and until some form of continuous biometric authentication can be guaranteed, it seems that a password-based authentication, perhaps combined with some other form of invigilation of the assessment is as good as can currently be achieved.

Let us consider the various scenarios under which automatic assessment may be undertaken. Under examination conditions, we may want the student to be physically authenticated by an invigilator, as is the practice in examinations everywhere. This would ensure that the authenticated student was the only one to have contributed to the solution assessed. To do this would require an invigilator to enter an id and password after physically authenticating the student's identity using a photo-card or similar document. In this scenario, the invigilator is a trusted intermediary. To be absolutely certain of security would also require some kind of security measures on the channel used between student and assessment engine. This might be provided by IP filtering or via digital fingerprinting of all messages. This guards against electronic impersonation.

For the pedagogical scenario described in this paper, the above solution is too strong, as in normal mode, the students submit a report and may work together, although not copy each other's work. The report is submitted, marked and checked for plagiarism. Plagiarism in the online assessment can be avoided simply by having a single synchronized session at which all students attend a supervised laboratory. They would prepare for the assessment beforehand, could take in books but would leave when they had completed the exercise and would be prevented from contributing to each other's design. A sufficient time would be allowed to ensure that even the slowest in the class could complete the exercise in the session.

The students taking the assessment must first authenticate themselves with the presentation environment using an id and password. This can be achieved in a number of ways. It could use standard web authentication but this is only an access control and would leave us with the problem of mapping the student id to the assessment files produced. Alternatively the presentation may be undertaken through some kind of learning environment or portal and authenticated user information can be inherited from that system by the plug-in. Finally and this is the method we have adopted, we can use authentication within the presentation delivery. This is probably the most secure and can use a one-time id and password specifically generated for the assessment. We have already solved this authentication problem (Zhang2002) and have a prototype plug-in that provides one-time authentication based on encrypted codes published within a presentation. The plug-in will not play a secure presentation unless it has been provided with a matching id/password pair based on the encrypted codes. To avoid repetitious input of that information, the id/password pair is cached in the user's file space for the session.

5. CONCLUSION

The AudioGraph toolset has been used to capture a complex design assignment and to transform it into online multimedia. That assignment has normally been assessed by written report with tutorial classes providing support and attempting to limit (unsuccessfully in the past) the design space, which must be assessed by the examiner. The assignment assesses the students design attempt using weighted links in the hypermedia. The automated assessment was achieved using web-log analysis that traced hyperlinks taken during the interactive presentation but this process can be designed, automated and authenticated. The exercise has attempted to replicate the same kind of interaction used in the face-to-face tutorials in supporting the students in the design process. It does this to some extent but limiting feedback to predefined presentations for each path on a choice point only allows one level of response. This could be extended with further levels of response but the technical nature of the assignment does not admit the more free-form discussion found in other subject domains. We were also restricted in our goal of creating dynamically published media without requiring programming experience, and this is about as good as we can achieve. Certainly the presentations capture the major pitfalls in design in this domain and support is given for these. It would however, miss the more subtle problems that involved deeper dialogue in face-to-face tutorials. The approach minims the state space of any possible solution and makes the design process amenable to non-media or programming experts.

This exercise has a number of benefits over the original exercise:

- Assessment, which is normally very time consuming, can be fully automated and hence scaled up;
- A wide range of discrimination can be obtained in the assessment using a marking scheme with a range of weights;
- Plagiarism can be avoided using the on-line assessment procedure; and finally
- The assessment load for the student is very much minimized using this tool rather than having to develop a written report on their design.

The assignment presentation has been tested using a small sample of students who have previously taken the same course. The results of this evaluation were small-scale and qualitative and results have been used to refine the design of the presentation. Generally the students were very enthusiastic about this development.

In addition to the production of this on-line assignment, the work has provided insight into the problems of authoring learning design, in particular in the relation to multimedia tools that have their own authoring and delivery engines. We have proposed a workflow, which decomposes the design process and defines templates between the different levels of the tools, pedagogical authoring, domain authoring and dynamic delivery. In this workflow, a higher-level tool would generate the templates and these could be translated into unpopulated AudioGraph presentations structures. The template editor would be a tool used by the pedagogical expert and would be based on EML or IMS learning design.

AudioGraph is currently able to publish adaptively and in order to publish dynamically both authoring and delivery tool must be able to manage symbolic links. AudioGraph already supports symbolic links in authoring so the structure of the program does not need to be changed significantly. The module publishing material would need to be modified to retain symbolic links and to generate dynamic-delivery template. The plugin would also need minor changes in its linking procedure by defining the binding between links and files based on other input, such as student preferences.

6. ACKNOWLEDGEMENTS

We acknowledge the support of NZEdSoft Limited. for their help in developing this work, in providing early releases of the AudioGraph Recorder for this experiment and in discussions on the potential means of assessment using the AudioGraph plug-in.

7. REFERENCES

J. Djordjevic, A. Milenkovic and N. Grbanovic (2000) An Integrated Educational Environment for Teaching Computer Architecture and Organization, *IEEE Micro*, 20, No. 3, pp. 66-74.

W. Doyle (1986) Classroom management. In Wittrock, M. C. (ed.) *Handbook of research on teaching, 3rd edition*. New York: Macmillan.

R. Gehne and C. R. Jesshope (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.

T. Galyean, (1995) *Narrative Guidance of Interactivity* MIT PhD Thesis, MIT Media Lab, pp. 58-59.

R. Gehne, R and C. R. Jesshope (2002) Interactive multimedia for dummies, *Proc Third International Conf. On Networked Learning*, (Banks, Goodyear, Hodgson and MCConnell Eds.), Sheffield University, March 2002, ISBN: 0902831 410, pp144-152.

H. J. H. Hermans, E. J. R. Koper, A. Loeffen, J. M. Manderveld, E. M. Rusman (2000) Edubox-EML Reference manual, Open Universiteit Nederland.

IMS Global Learning Consortium Inc (2003a) *IMS Learning Design Specification*, http://www.imsglobal.org/learningdesign/, accessed on 24/4/2003

IMS Global Learning Consortium Inc (2003b) *IMS Question & Test Interoperability Specification,* http://www.imsproject.org/question/index.cfm, accessed on 24/4/2003

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

C. R. Jesshope, A. Shafarenko and H. Slusanschi (1998) Low-bandwidth multimedia tools for web-based lecture publishing, *IEE Engineering Science and Educational Journal*, **7** (4), pp148-154

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

C. R. Jesshope (2002) *Computer Systems 3*, http://www2.dcs.hull.ac.uk/people/csscrj/web-site-08236/home.html, an on-line course, accessed on 24/4/2003

M. Murtaugh (1996) *The Automatist Storytelling System - Putting the Editor's Knowledge in Software*, Masters Thesis, MIT Media Lab, http://ic.media.mit.edu/Publications/Thesis/murtaughMS/HTML/ accessed on 28/4/2003.

NZEdSoft - New Zealand Educational Software (2003) *AudioGraph Recorder download page*, http://www.nzedsoft.com/audiographrecord.html, accessed on 24/4/2003.

R. Koper (2001) *Modeling units of study from a pedagogical perspective, the pedagogical meta model behind EML*, http://eml.ou/nl/introduction/articles.htm.

R. J. Osborne and M. C. Wittrock (1983). Learning science: A generative approach. Science Education, 67, 489-508.

M. Ryder (2003) *Instructional Design Models,* http://carbon.cudenver.edu/~mryder/itc_data/idmodels.html, accessed on 28/4/2003.

S. Zhang (2002) *Study and design of security system for AudioGraph multimedia teaching system*, Msc Thesis, Computer Science, Massey University, New Zealand.