An Adaptive Multimedia Presentation System

Philip Davies
Faculty of Technology
Bournemouth and Poole College
Bournemouth, UK
pdavies@bpc.ac.uk

David Newell
Software Systems Research Group
Bournemouth University
Bournemouth, UK
dnewell@bournemouth.ac.uk

Nick Rowe
Faculty of Technology
Bournemouth and Poole College
Bournemouth, UK
nrowe@bpc.ac.uk

Suzy Atfield-Cutts
Software Systems Research Group
Bournemouth University
Bournemouth, UK
satfieldcutts@bournemouth.ac.uk

Abstract - **Requirements elicitation for a multimedia presentation system for e-learning led the writers to propose a video segmentation process that adapts learning materials through online interventions between the student and tutor. The tutor tailors audio/visual segments by dynamically inserting new fragments that provide supplementary updates in response to questions from students. A survey of advanced adaptive approaches revealed that processing of manually or automatically generated metadata would provide better adaptation. Automated use of metadata requires storage and processing of context dependent ontology hierarchies that describe the semantics of the curriculum. Data and semantic models needed to adaptively process multimedia presentations in real-time are derived. The design models are implemented using HTML, XML and Flash. The authors conclude that the use of context-based rules that process meta-level descriptions of segmented multimedia components stored according to a bounded ontology can produce a system that dynamically adapts learning materials.**

Keywords – e-learning, adaption, metadata, semantic, ontology.

I. INTRODUCTION

Traditional lectures and seminars are being supplemented or replaced by multimedia presentation systems. However, this movement towards on-line learning suffers from a number of drawbacks, such as reductions in contact with real tutors and changes to the traditional teaching-learning feedback loop. The Adaptive Multimedia Presentation System (AMPS) is an attempt to overcomes some of these drawbacks [1].

A brief survey of prevailing approaches to adaptive multimedia learning [2],[3],[4] has shown that systems with personalisation requirements have begun to be designed and developed. For example, Yang and Yang discuss the development of SMILAuthor [5] a tool based on the Synchronised Multimedia Integration Language, SMIL. SMILAuthor generates SMIL code to spatially place objects on a presentation panel using a drag-and-drop interface. It claims benefits over other multimedia authoring tools because the use of visual representation of a timeline for the placement of events making generation of SMIL referring to temporal events much simpler and less error prone than the alternative manual coding of an SMIL document. Reducing the complexity of the content creation process helps reduce the incidence of coding errors. The novel approach introduced by this paper provides features of the dynamic fragmentation of learning materials, which the SMILAuthor does not. Fragmentation facilitates the formation of better multimedia materials because the tutor supplements materials when responding to online questions from students. It also provides a future platform for a multimedia presentation system that is adaptive in real-time. The future development of HTML 5 may address some of these shortcomings [5].

Evaluation of an initial prototype provided evidence for the need to add efficient navigation for student users, so that they can access relevant learning at any point in the audio/video segment. This requires user controls and the structure of the presentation to be manifest to the student in the form of a table of contents. An evaluation is made to determine how the student users' experience is genuinely improved by using adaptation, what models are needed theoretically and what are the best practical tools to generate executable models to achieve dynamic adaptation - for example, the ontology and the student/tutor model - what form do the input and output files need to take, what is the nature of adaptation, to what extent can the current prototype interface be considered adaptive and how can the adaptations be evaluated and improved. The structure of the paper is as follows: Section 2 gives brief requirements specification for the proposed adaptive multimedia presentation system, Section 3 introduces the prototype AMPS while Section 4 looks in more detail at the media segmentation process used within AMPS. Section 5 looks at the adaptive authoring tool and its architecture. Section 6 discusses the prototype AMPS interface evaluation findings in a pilot study with degree level students and their implications. Section 7 discusses the question of automating AMPS and presents a staged implementation plan. Section 8 looks at the issues surrounding the use of ontology and develops a particular instance of network ontology and its application to AMPS. Finally, section 9 is a conclusion and discussion of future work.

II. REQUIREMENTS FOR INTERFACE DESIGN

An initial use case diagram in Figure 1 shows essential requirements for the tutor and the student. The tutor requires the minimum amount of time and effort to input learning material. Initially, this is limited to producing and uploading the audio/video segments and being able to put them into an appropriate order. An adaptive engine within the system could extract appropriate text and timeline data from these and distribute this to the display panes of the interface to present the table of contents and supplementary text.

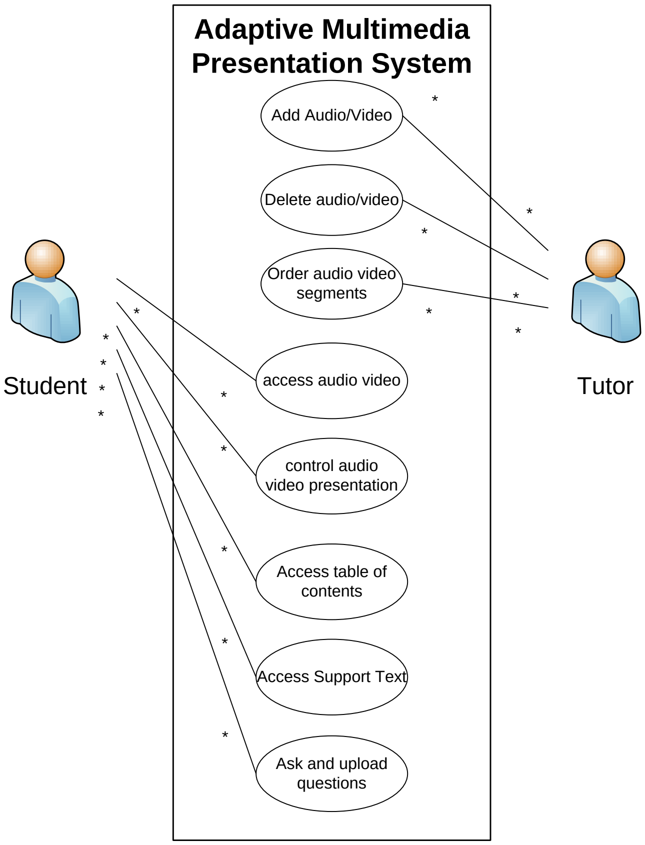

Figure 1: Use Case Diagram

The student requires access to the audio/video segments and some measure of control over their delivery. Being able to select and re-run segments is important for learning at the student's own pace. To enable this, an intuitive navigation system is required, which sequences and orders the significant points in the presentation and displays them in a table of contents with the associated supporting text. The ability to gain clarification on points not understood is also an essential requirement to effective learning. It is intended to fulfil this requirement with supplemental text and by providing access to other materials at any point in time during the presentation, as well as the ability to stop, start and jump to other points on the presentation timeline.

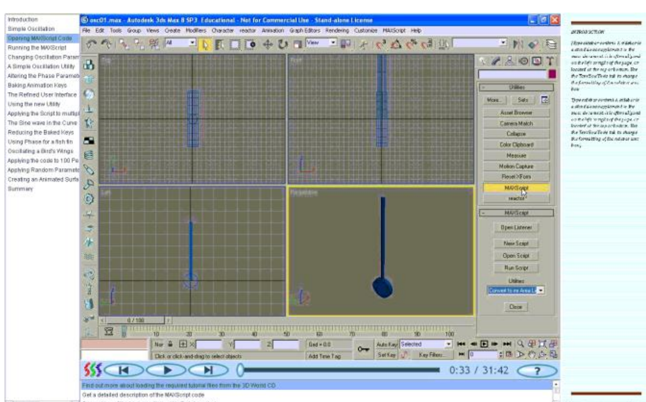

Figure 2: A proposed prototype system

A proposed prototype system shown in Figure 2 is composed of five principal parts: the main presentation panel, the table of contents panel, the supplementary text panel, the questions panel and submit button, and timeline controls for the running of the audio/video presentations.

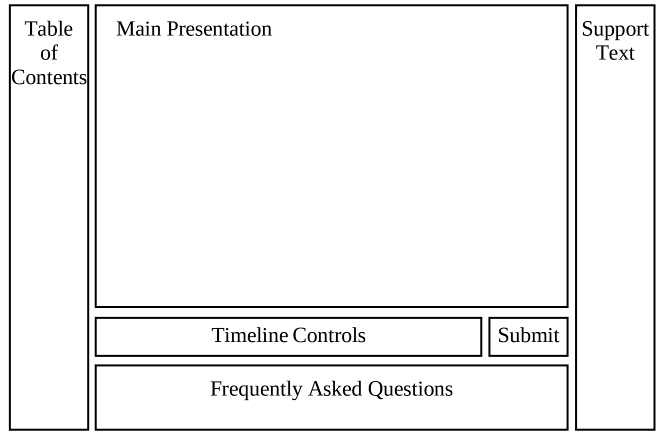

Figure 3: Schematic of the prototype system

A. *Main Presentation Area*

This contains the multimedia document which may display any combination of text, graphic, image, audio and video. It is also the primary data display area from which all supplemental information will be retrieved.

B. *Table of Contents*

The information displayed in the table of contents is automatically retrieved from the support text pane. This will require the use of intelligent knowledge storage and retrieval techniques that can structure, select and display the most useful learning material. The table of contents is presented in a hierarchical structure with a breakdown of sections. Each section title is a link to a position on the timeline, so that it is possible to jump between places within the same video/animation, or sequence of them. In later developments, additional supplementary information may be provided from the main presentation area using a variety of knowledge engineering techniques including text-based retrieval, image retrieval, video retrieval, and audio retrieval to construct a more adaptable multimedia presentation. Content-based retrieval techniques vary from one element of multimedia to another, ranging from keywords for texts, colour and texture for images and spoken words for audio, for example.

C. *Supporting Text*

Additional supporting notes will appear in this portion of the screen. This is intended to be text that assists the user's accessibility of the learning material. It may contain links to other timelines, e.g. open a new window with a duplicate set of components and its own timeline. The text displayed here may be a simple transcription of the audio part of the presentation displayed in the main area which could be retrieved by voice recognition techniques but at present are manually produced by the multimedia author.

D. FAQ and Submit Button

A facility is needed to answer with questions raised by students during a class or lecture. This external interaction requires the tutor to respond to questions put by students using the system. A proposed solution is to allow the user to invoke a text dialogue with a tutor triggered by a button.

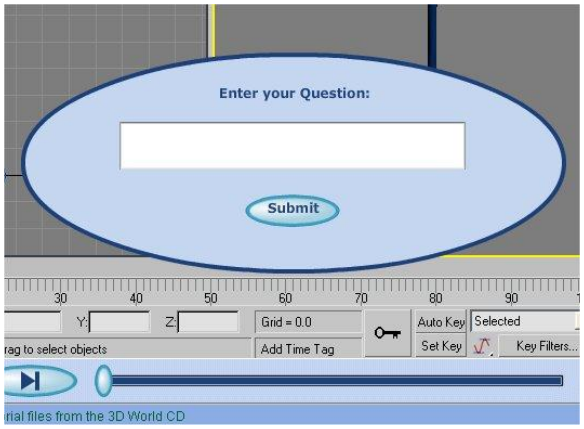

Figure 4: The submit question dialogue box

Questions are typed into the text area and submitted to the tutor. From this, an e-mail might be generated, additional automated data is added including a unique identifier for the presentation module and a timestamp. The timestamp isolates the precise time in the timeline when the question was asked, allowing the tutor to track into the presentation to see the context of the question.

The user's specific question forms the basis of feedback to alert the tutor of possible clarifications in the presentation that need additional explanation. The student's specific question is normally answered by the tutor through the creation of new video segments designed to provide clarification, which is made available to all students by insertion into the original presentation. The text question is displayed in a FAQ region when the presentation timeline reaches the point when it was asked. The audio/video segment containing the answer can then be optionally activated by selecting the question, and pausing the main presentation until the supplementary segment has been played. As more students view the modules, ask questions and gain answers, the presentation evolves by dynamically enhancing the learning resources.

E. Media Time Line with Function Buttons

The system offers temporal interaction that allows students to move through the presentation using the time bar, offering the ability to pause a presentation, to select another point in the timeline and restart the presentation, or by clicking on the table of contents to move to a different specific area. The current topic in the table of contents is highlighted in real-time so students can determine the position within the presentation, enabling students to manage their study time effectively. This type of interaction allows students to adjust the delivery of the presentation to suit their own learning style.

A graphical representation of a time line is provided, similar to a media player, representing the temporal state of the currently playing video or animation. A standard set of buttons for controlling playback will be provided. The total duration of the video/animation, or set of videos/animations which run in sequence, determines the maximum duration of the media time line.

III. THE PROTOTYPE AND ARCHITECURE

The first prototype of AMPS was developed based on the authors' understanding about how students would be expected to learn. This was felt to be a valuable initial step in personalisation [7]. The next stage is to develop the personalisation further through a new level of automated adaption and work with student end-users to gain their direct feedback of AMPS.

The prototype system shown in Figure 3 is composed of five principal parts: the main presentation panel (A), the table of contents panel (B), the supplementary text panel, (C) for the running of the audio/video presentations (D) submit button, and timeline controls and (E) the questions panel.

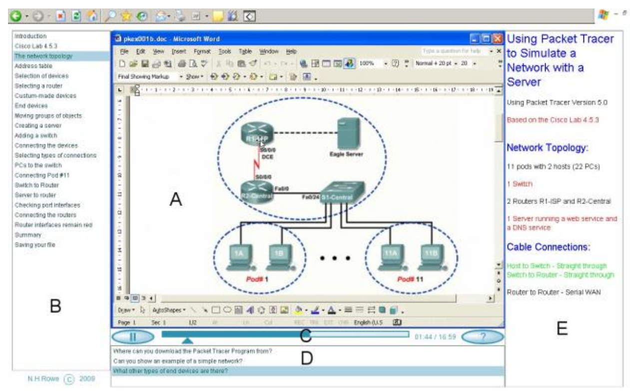

Figure 5: The AMPS prototype showing an adaptive CISCO™ learning object

The tutor builds the e-learning modules by using the segmentation architecture, which provides flexible delivery. The presentation is broken down as required into multiple segments each corresponding to an individual learning object. The selection, arrangement and linking of segments will constitute the delivery of a particular learning obect with a learning approach. In this way many segments could be played one after the other to view different aspects of the content. For example, screen shots within on-line learning materials may be followed by a video of a practical laboratory example.

IV. MEDIA SEGMENTATION

Re-segmentation of the video into smaller sections with each section carrying a single learning objective will be a direct consequence of the new user requirements. Smaller segments will further allow the personalization of the learning packages in a highly customized way and lead towards the better adaptation of AMPS.

Furthermore, in order to respond to the differing needs of students, the linking of the media segments will involve more than just a linear arrangement. The response to student interaction requires branching capabilities within the segmentation architecture [8],[9]. Segmentation allows the selection of material according to learning objectives. Students may choose to view only those segments they need to see. Additionally, the system will have the ability to respond to new students' needs not already met, or even envisioned, by currently available material. Hence the system will record and insert new media segments as required. For example, in response to a student's question for more information on a particular topic, the tutor can record a new segment and make the new segment available to all students.

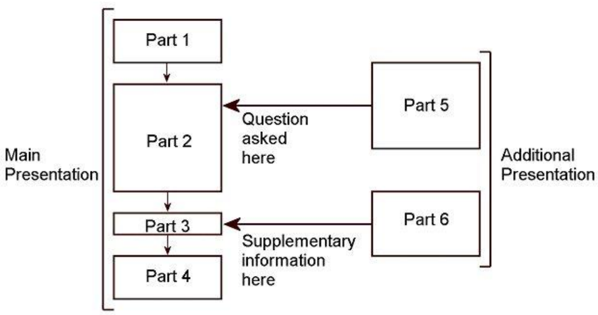

Figure 6: The timing of presentation segments

Figure 6 shows a main presentation sequence of four media segments making up a learning object. Questions asked by students at points in segment 2 and segment 3 have led to the generation of new segments 5 and 6 by the tutor which link to the main sequence at the correct points shown in the diagram.

<SEGMENT 1>
<SEGMENT 2>
      <SEGMENT 5>
<SEGMENT 3>
      <SEGMENT 6>
<SEGMENT 4>

Figure 7: Multi-level list of media segments with a hierarchical architecture

This is equivalent to a multi-level list with a hierarchical architecture. Each new segment is simply added as a subsection at the appropriate place in the list which is constructed in XML. This is rendered by the system to produce a new table of contents entry and FAQ entry. When either of these is selected, a new window opens containing the video or animation explaining the answer to the query. Each term listed needs to be linked back to a point or points in the video when the term was used and is marked as a point on the timeline. Clicking the link moves the current timeline to the associated video or animation.

B. Media Player Configuration

As a single player is required to play any module, configuration is required to activate the required resources and also to give the temporal information needed to activate the table of contents entries and the FAQs. Figure 8 shows the original XML file used for configuring the system. The file has an outer main tag. The children within this are frame rate, module ID, filename, tocInfo and questions.

The filename tag contains the files to play in sequence in the main presentation area. In this case a small presentation was played before the start, ploadv2.swf. This allowed the main presentation to be preloaded. While this was playing there was no loading delay for the main presentation.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<main>
   <framerate>8</framerate>
   <moduleid>V200134234</moduleid>
   <filename>
      <node name="ploadv2.swf"/>
      <node name="art02.swf"/>
   </filename>
   <tocInfo>
      <node label="Introduction" fileset="0" time="0.00" />
      <node label="Simple Oscillation" fileset="0" time="11.50" />
      <node label="Opening MAXScript Code" fileset="0" time="24.75" />
      <node label="Running the MAXScript" fileset="0" time="64.75" />
      <node label="Changing Oscillation Parameters" fileset="0"
time="109.38" />
      <node label="A Simple Oscillation Utility" fileset="0"
time="183.25" />

      ...

      <node label="Creating an Animated Surface" fileset="0"
time="1563.25" />
      <node label="Summary" fileset="0" time="1802.25" />
   </tocInfo>
   <questions>
      <node name="Find out more... " file="art01.swf" frame="88"/>
      <node name="Get a detailed... " file="art05.swf" frame="552"/>
      <node name="See a video of..." file="art06.swf" frame="10416"/>
      <node name="See a video of..." file="art02c.swf" frame="12416"/>
      <node name="How can this..." file="art03.swf" frame="12560"/>
      <node name="How can the oscillation..." file="art04.swf"
frame="12640"/>
   </questions>
</main>
```

Figure 8 : The XML configuration file

A prototype design architecture satisfying these initial requirements has undergone implementation and evaluation by the writers.

V. ADAPTIVE AUTHORING & RETRIEVAL TOOLS

A. Development Stages

A prototype development with staged design and implementation with increasing levels of adaptation uses two Virtual Learning Environments (VLE). One VLE is at Bournemouth and Poole College, using the open source VLE Moodle. Bournemouth University uses a localised version of the Blackboard VLE. Both VLEs have been in use for a number of years at these institutions to support peer assisted learning [10].

The development stages are:

1. Presentation player to display learning object content from VLEs
2. Authoring integration tool with manually entered meta data to create segmented learning objects
3. Authoring tool with automatic generation of meta data using adaptation/ontology techniques

4. Authoring tool with adaptive retrieval engine to automatically create multimedia content for presentations from generated ontology/metadata
5. Personalised adaptive multimedia presentation system based on students' assessment test results.

B. User Types

The user types we model are student users and academic tutors.

C. Authoring Tool

The authoring tool is shown in Figure 9. This can be evaluated by the widespread use of the system by lecturing staff and students. Success amongst staff will only occur if authoring is easy and will continue where feedback from students is widespread and positive. An authoring tool for multimedia presentations must be easy to use by non-technical teaching staff for speedy development of content [12].

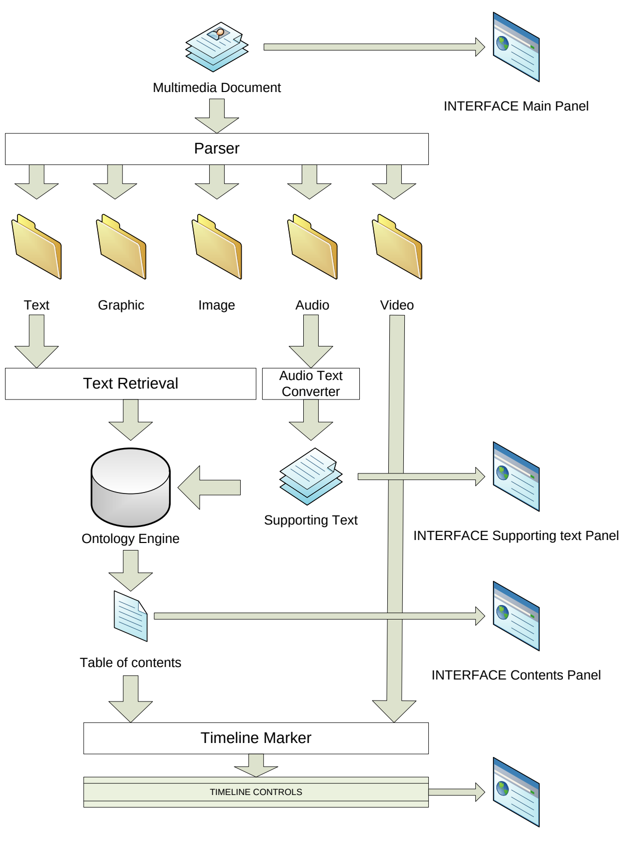

Figure 9: Architecture of an Authoring Tool

VI. Interface Evaluation Findings

An online survey was used for the evaluation of the AMPS. A simple online training session teaching students how to configure a Cisco wireless router, was set up in the AMPS using the Cisco Packet Tracer [10] network simulation tool Fifty-five first year undergraduates on the honours level computing degree at Bournemouth University were recruited during normal lab classes to undertake the training through the AMPS.

Three areas of examination were covered by the questions. The first is the current level of prior knowledge of online learning environments and the subject area. The second is their experience of using the AMPS with the focus on finding out what users are trying to achieve and whether that could be made easier using new technology. And the third is the level of knowledge attained through the AMPS. Opportunity was provided for additional comments the user wished to confide.

In terms of prior knowledge, the majority of students assessed themselves as have good or excellent knowledge in the following areas:

Computer Networking 53%
Using Visual Training programmes 60%
Using VLEs 57%

Approximately a third of students (34.5%) had prior knowledge of the Cisco Packet Tracer programme and none claimed excellent knowledge.

In the area of interface use, the following features of the AMPS were rated as the most useful:

The ability to pause and rewind the presentation (83.6%)
The index list on the left of the screen (83.3%)
The ability to click on the index link to move along the video (81.4%)
The video panel in the centre (70.9%)
The time line below the video panel (70.9%).

Ease of use of the same features was rated as follows with percentages showing responses rated as very easy or easy:

The index list on the left of the screen (83.7%)
The overall interface (83.6%)
The ability to click on the index link to move along the video (81.8%)
The teaching panel in the centre (77.8%)
The time line below the video panel (76.3%)

The content of the teaching package was rated as good or excellent as follows:

How well explained was the content of the video? (83.3%)
How good was info in the index on the left? (83.4%)
How good was info in the text on the right? (49.1%)
How good was the email response (if used)? (17%) N/A (64.2%)
How good were the FAQs? (15.1%) N/A (49.1%)

Asking students to rate the most important feedback features gave the following results for very important and quite important:

Ask a question during the presentation? (68.5%)
See other student's questions and their replies? (50%)
Create your own FAQ entries? (38.9%)

We also asked what would be an acceptable response rate time for feedback enquiries:

10 minutes 34.0%
1 hour 34.0%
4 hours 8.5%
24 hours 19.1%
2-3 Days 2.1%
1 week 2.1%

In the third section, we asked students how much they actually felt they learned from the experience. The rating for those who learned a substantial amount and those who learned quite a lot are as follows:

Networking (51%), Wireless (52.9%), Packet Tracer (62.2%)

As a result of this survey a number of findings emerged which have potential impact upon the redesign of the AMPS interface.

First, concerning the layout of the interface, not all users realized that there was a right-hand panel as this was just off the screen for some users. This issue needs to be addressed either by indicating the panel is off the screen, by reorganisation of the interface elements, or by automatic resizing of the application according to the size of the monitor used to view it. These, and possibly other options, need exploring, and user testing completed, to select the most appropriate.

Second, concerning usability, a number of students commented that the audio segment was too long at 30mins and requested shorter teaching modules. User testing will determine the ideal duration of each learning segment. In addition we have to consider if the acceptable duration of learning segments changes as the user becomes more familiar with the interface. Segmenting the video into smaller sections with each section carrying a single learning objective will be a direct consequence of the new user requirements.

VII. An Approach Towards Automating AMPS

A staged approach to the automation of AMPS is planned as a research programme:

1. The generation of additional video segments interweaved within the original presentation as a response to student feedback
2. The automatic generation of the content in the table of contents pane (B)
3. The automatic generation of the content in the supplementary text pane (E)
4. The segmentation of the video presentation (A) into learning objects
5. The presentation of the learning material adapted to the specific needs of the student and personalized to them.

At present only stage 1 has been realised. Figure 9 shows a model of a theoretical segmentation architecture containing a number of functions, including conversion of speech to text, a parser, the employment of an appropriate ontology engine and time line coordination to drive the AMPS.

The stages are as follows:

Stage 1: the audio component of the video clip will be parsed through a voice to text engine to transliterate the voice content of the presentation into text. This will be fed into the text panel at the right of the interface. While viewing a multimedia segment, for example, the audio of the presentation is separated and passed through a text retrieval engine which uses voice recognition principles to recover and provide text direct to the supporting text panel. Text may then be sent to the ontology engine. It uses a mixture of manual and automatically generated semantic structures that represent the conceptualisations meaningful within the context of the segment contents. The details of operation and application of ontology engines are current research areas [13] however the required outcome is the construction of the table of contents in the form of a hierarchy of terms. In the case of a 3D visualisation tool, a heading 'rendering' might be inserted into the table of contents referring to a combination of multimedia information available in the presentation system. The timeline controls links the term 'rendering' to relevant points in the multimedia content to mark the position on the timeline. The link provides a method to access the timeline of the relevant video segment or animation.

Stage 2: the generated text will be analysed by the ontology engine to construct the time-linked index. This will search the generated text for every token in the networking ontology to create a set of frequency distribution tables. Tables will be constructed for each token level within the ontology hierarchy. Level 1 tokens will form the primary analysis and will be ordered first. Level 2 will be performed within level 1, and so on. The frequency of level 1 tokens will determine how the index is structured. Boundaries of discussion will need to be detected in order to know when the topic has shifted from one domain to another. The frequency of tokens will be sufficient to name and label the domains of discussion but they will not be able to determine the boundaries. This will require a supplementary ontology dealing with concept boundary transitions and searches for the tokens that indicate these transitions.

Stage 3: The index elements will be passed through a timeline marker to set up the timeline controls. In an effort to further reduce authoring complexity, in the simplest case, metadata describing the content of segments could be created and entered manually by a domain expert at the time of media segment creation.

Stages 4 and 5 are more complex and will be considered in more detail in a later paper. However by analysing content dynamically in response to students needs in real time, the authoring tool itself would ideally be made capable of creating ontology information and using metadata. It is anticipated that the most difficult analysis would be looking for objects in videos and determining their type and meaning. However, the sports industry have analysis software for tracking the paths of moving objects such as balls on pitches and organisations involved in photography have workable face recognition systems in cameras already in use.

Beyond stages 4 and 5 we envisage a programme that will encompass the following considerations:

- The presentation system will be made adaptive through stages 2-5 and will attempt to approach real-time implementation.

- The scope of the application domain is the special case of 'Digital Networking' which will be defined through an example ontology
- The knowledge represented in the ontology will be in the form of a class diagram formatted in XML and processed in an ontology engine constructed for the purpose
- Inputs and outputs are used through a fully documented API to control input into the AMPS user interface and to personalise the learning experience
- There will need to be feedback from the user interface to the ontology engine; this will be via a fully documented API.

Future enhancements would add capability to 'see' the frames of the video, 'see' the contents of images, 'listen' to the audio, or 'read' text. The latter is the most feasible, for example by searching for key words in the text, building a semantic model of content or an ontology for the problem domain, and using this to dynamically classify and construct useful content based on the meaning of available materials.

Another challenging dimension is added when the dynamic assembly of learning objects, based on content descriptions is extended to distributed systems. An attempt is being made to apply knowledge engineering principles such as storage and retrieval of multimedia objects to the web. Practitioners are investigating these areas actively. Henze, Dolog, & Nejdl [14] have reported on the use of a logic description language, Resource Description Formats, RDF, to guide the formation of an ontology and metadata for three types of resource – domain knowledge, user knowledge and observer knowledge. These are used for personalisation of learning in a future semantic web, although the production of quality materials in an open system is problematic.

The theoretical foundations of logic languages and frameworks such as RDF hold the promise of producing practical tools and techniques for future adaptive multimedia presentation systems but they are not fully explored yet. Providing personalised on-line learning using an ontology engine to create adaptations in a closed system, let alone an open one such as the Web, is an active and complex research area [12]. Many writers are investigating competing methods and techniques to apply knowledge engineering based approaches to various application domains. This includes the use of multi-agent systems [15], neural networks or fuzzy logic filtering [16].

VIII. ONTOLOGIES, ADAPTION ENGINES AND THE API

Developing a Networking Ontology

There are a wide range of available ontology tools and models which attempt to describe knowledge domains using ontology capture and manipulation packages, e.g. Protégé Ontology Editor developed by Stanford California [17],[18]. Investigation into currently available ontology tools and models led to the decision to build our own prototype ontology of the digital computer networking knowledge domain so that it can be tightly customised to our students' particular learning domain.

However, we have tentatively concluded that these models are unlikely to contain the level of detail needed for digital networking [13].We are sceptical about the utility of constructing and executing, high-level, general-purpose ontology models in an adaptive multimedia system, especially if it is to operate in real-time [19]. This has also been supported by finding in other specialist areas such as the biomedical domain where formal ontologies can have clear limitations. Research by Shultz et al. [20] has taken the view that constructing large ontology models with many classes that range over wide topic-areas can be meaningful. More investigation is needed into this question.

Proposals to base real-time adaptation on feedback from students' responses to dynamically change the selection of menu links implies much closer integration between the ontology engine, the student's profile, or students' historical learned group profile, and the AMPS. Traditionally, two main components or sub-system types are identified in adaptive learning systems:

Case 1: Off-line recommender link mining engines, including web link miners that the tutor assists in generating adaptive presentations [12]. Output is in the form of candidate web links or menu items audited by the tutor that attempt to narrow the selections on offer to the student in the subject domain.

Case 2: Online engines that use pre-processed ontologies and combine them with individual or multiple student profiles that has been data mined, for example to find patterns that represent groups of students with given attainment levels. Outputs are recommendations for offering learning materials to these groups of students [12]. Materials presented are deemed appropriate to the student group as evaluated from outcome data such as Multiple Choice Question (MPQ) tests.

In addition to the problems already described, another drawback of Case 1 is that too many options can be presented to the tutor and the students. This makes the choices of learning materials presented to students even more problematic for a closed system such as ours. This is another reason why the writers decided to develop a restricted portion of an ontology of 'Digital Computer Networking' for use as a proof of concept model in the AMPS [21].

Figure 10 shows the contents of the Protégé ontology modelling tool [17]. This ontology was obtained using the writers' knowledge of the chosen 'Digital Computer Networking' problem domain. Knowledge of the curriculum in both academic and industrial certification courses that the writers have developed over many years of programme design and teaching of the topic to undergraduate and postgraduates at Bournemouth University was informally used to develop the ontology.

The ontology can be extracted from Protégé as an .owl file using the Manchester OWL Syntax [22], developed by the CO-

ODE project for writing OWL class expressions, or as an XML file as shown below in Figure 11 and Figure 12. This new information format is expected to be useful for analytic computational purposes as an input to the ontology engine.

A drawback of Case 2, making real-time adaptations hard to realise, is that the two sub-systems in the ontology and student model processes engine need to be combined and integrated for adaptations to be achieved in real-time, or in other words, without tutor assistance. The question therefore arises of how to model the functionality of these sub-systems and how to model the API between them to achieve close integration.

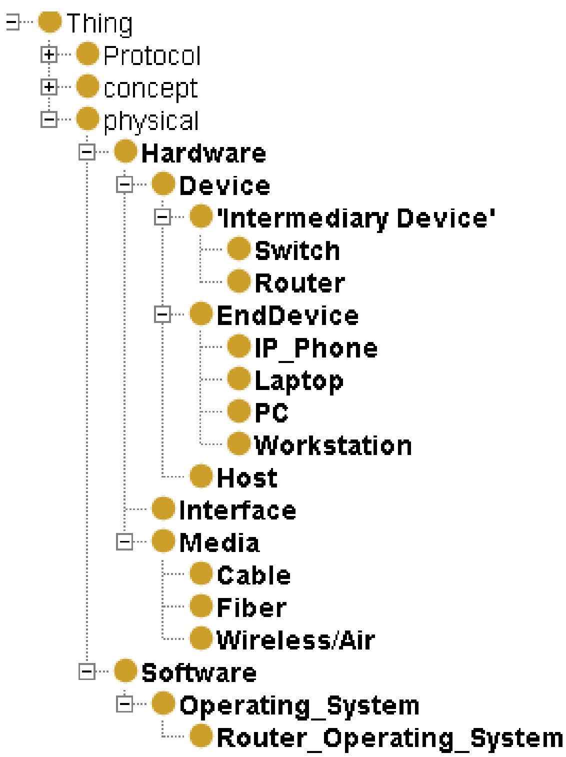

Figure 10: Sample Class Hierarchy of Digital Network Ontology Model

```
<!--
http://www.semanticweb.org/ontologies/2010/0
/OntologyOfDIgitalNetworking.owl#Device -->
<owl:Class rdf:about="#Device">
<rdfs:subClassOf rdf:resource="#Hardware"/>
</owl:Class>
```

Figure 11: Example fragment of a class from the owl file produced by Protégé

```
<SubClassOf>
<Class
URI="&OntologyOfDIgitalNetworking;Device"/>
<Class
URI="&OntologyOfDIgitalNetworking;Hardware"/>
</SubClassOf>
```

Figure 12: Example fragment of a class from the XML file produced by Protégé

The Adaptation Engine and AMPS API

Most adaptive systems contain a form of split architecture described above, but when considering the drawbacks mentioned, the writers have divided the future system into two sub-blocks and begun to develop an API between them. This allows separation and integration to be achieved simultaneously, so that the AMPS is able to perform adaptations closer to real-time.

Following Figure 13, firstly, there is an ontology engine-controller sub-block. Secondly, there is a user interface sub-block that uses standard object technology modelling methods such as model-view-controller notions, and a responsibility based class/object analysis method has been used to model the system. Messages can be bi-directional, providing feed-forward control and the feedback needed to be able to approach real-time adaptation. Thirdly, it is necessary to couple the ontology engine tightly to the user interface and to define the responsibilities of each sub-block. This requires detailed analysis and database design [23] including:

- Data about the inputs from the XML description of the ontology description tool that are processed by the ontology engine
- A diagram of user interface classes to be used to determine the optimal user interface behaviour
- Commands: these illustrate the input scenarios and can be described as a storyboard or state transition diagrams
- Messages: similarly, these explain possible output scenarios (e.g. menus, text, voice, and timeline)
- List of classes/object with functional requirements and an API will be modelled
- Choice of possible recommender algorithms [24]
- Implementation of methods
- Determination of evaluation approach will validate the effectiveness of adaptations.

Figure 13 is a first cut analysis output showing how sub-systems will collaborate and begins to locate functionality into sub-systems and conceptualise the API. The following classes have been included in the OntologyEngine sub–system:

:AdaptiveApp - Maintains abstract internal state of the UIApp object that normally would have one instance but could be many, this is so the engine takes control of the AMPS User Interface.
:ContextDependentMenuGenerator - Tells AdaptiveUIApp what to display
:OntologyEngine contains an Engine class that itself has a class structure. This will fundamentally consist of -
:OntologyEngine::Engine - The Engine class is responsible for the main control that drives the new AMPS system. The methods needed depend on the XML format (from/to the Protégé model) and the nature of the selected adaptation technique. These could be a data mining approach or a neural network approach. The effectiveness of adaptations will need to be evaluated to find the optimal choice.

OntologyEngine::AdaptiveApp - Maintains the state of the UIApp to make available to Engine. As explained above, this class is key and needed inside OntologyEngine to maintain state common to the engine and the User Interface.

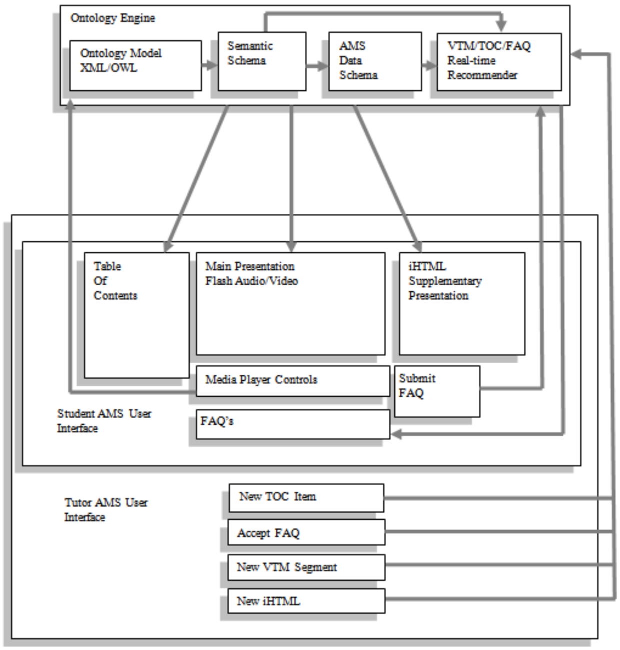

Figure 13: A Collaboration Graph of the AMPS

OntologyEngine::AdaptiveSegment – Describes sections of multiple components or segmented learning material, e.g. LEARNING OBJECT segments that can be enabled or disabled by the OntologyEngine::Engine to achieve adaptation.

Internally to the AMPS system, the OntologyEngine class itself has a structure that will need more detailed analysis than can be presented in this paper. Experiments with alternative class structures will be a critical determinant of feasibility, performance and usability. Methods and state will need to be further analysed as a guide to performance.

The design decision was taken to maintain the state of an AdaptiveUIApp class, which will mirror the AMPS state, internally to the Ontology Engine, rather than allow the User Interface to stand and operate alone as is the case with the current prototype implementation. This innovation will achieve the integration needed to approach runtime performance.

IX. Conclusions and Future Work

An investigation has been undertaken into the requirements, underlying techniques and technologies needed for an adaptive multimedia presentation system. Research issues associated with this knowledge based approach to personalisation of learning have been outlined and begun to be explored. A process for adapting multimedia presentations through adding new content segments requested by student interaction, e.g. email, using a tree-branching sequencing system for multimedia segments has been implemented and evaluated. Content selection can make use of a form of knowledge based analysis of semantic contents of multimedia segments, dynamic generation of ontology information about video segments is stored, and retrieval proceeds dynamically according to the use of the semantic data in future forms of such a system.

Adaptation can take many forms of response to different types of stimuli. The AMPS is at present only adaptive in responding with manually produced additional video segments to the stimulus of student emails. This is considered a low level of adaption and the programme plans to increase the number of stimuli which it will automatically respond to. These stimuli need to include student prior knowledge and student ability which we call the “student signature” and will be developed further in another paper.

Feedback from students indicates the learning experience has been enhanced as evidenced by the results of the online survey presented above. Evaluation has shown that these adaptations were liked by students but do not achieve real-time adaptation in the traditional sense because of time delays. A more interactive approach to adaptation has been described and the foundation of an analysis model has been described.

Further Questions and Continuing Research

Summing up, work discussed in this paper has answered some of the research questions posed at the start of this paper, but has also indicated further questions and directions for research. The unanswered questions are:

- What is the usability level of the user interface and how can this be further improved?
- What further adaptation features are required and how are they to be evaluated?
- What model is best employed to define the interaction between the user interface and the adaptation engine?
- What is the full specification of the ontologies that are required and how is it best captured?
- How should database schemas be constructed for the AMPS for real-time extension at data and meta levels?
- How should the ontology engine structure be modelled and evaluated? Which possible data mining, or other ‘smart’ techniques are considered candidates for the algorithm or protocol?
- How do we determine the appropriate definition of an API, possibly by means of an IDL, between the ontology engine and the AMPS user interface presentation system?

A carefully derived student and tutor model remains to be developed more fully to automate real-time adaptations. We will address these questions in a future paper.

References

[1] Cutts, S., Davies, P., Newell, D. and Rowe, N., (2009) Evaluation of an Adaptive Multimedia Presentation System (AMPS) with Contextual Supplemental Support Media Proceedings of the MMEDIA 2010 Conference, Athens, Greece.

[2] Adams G, Dolan M, Freed G, Hayes S, Hodge E, Kirby D, Michel T, Singer D (2009) Timed Text (TT) Authoring Format 1.0 – Distribution Format Exchange Profile (DFXP). Timed Text Working Group, W3C. Available at: http://www.w3.org/TR/2009/CR-ttaf1-dfxp-20090924/#intro [Accessed on 3 January 2010]

[3] Salton, G. & McGill, M.J. (1993) *'Introduction to Modern Information Retrieval'*. McGraw-Hill.

[4] Cristea, A.I., Smits, D., De Bra, P., (2005) 'Writing MOT, Reading AHA! - converting between an authoring and a delivery system for adaptive educational hypermedia'. A3EH Workshop, AIED'05. http://eprints.dcs.warwick.ac.uk/182/1/5-10-cristea4--.pdf pg no: 1

[5] Yang, C., & Yang Y. (2003) SMILAuthor: An Authoring System for SMIL-based Multimedia Presentations. *Multimedia Tools and Applications*, 21. 243-260 Kluwer Academic Publishers, Netherlands http://www.springerlink.com/content/t4853282j835k1g5/fulltext.pdf pg no: 245

[6] van Kesteren, A (2009) HTML 5 differences from HTML 4. W3C. Available at: http://www.w3.org/TR/html5-diff/ [Accessed on 3 January 2010]

[7] Evans, A., Fernandez, M., Vallet, D. and Castells, P., (2006) Adaptive Multimedia Access: From User Needs to Semantic Personalisation.

[8] Jun Yang, Q. L. (2007). 'Retrieval of Flash™ Movies by Semantic Content: Research Issues, Generic Framework, and Future Directions. *Multimedia Tools and Applications , 31*, 1-23.

[9] Ketter, W. Batchu, A., Berosik, G., McCready, D. (2008) 'A Semantic Web Architecture for Advocate Agents to Determine Preferences and Facilitate Decision Making', *ACM*. http://delivery.acm.org/10.1145/1410000/1409554/a10-ketter.pdf?ip=194.66.74.22&CFID=26634273&CFTOKEN=55467158&__acm__=1308058054_947195616d66cedefaa36c67a47df46d

[10] Jeary S, Atfield-Cutts S, Phalp K, Mayes H, Bates, N,. (2010). 'Using IT Support to improve the quality of Peer Assisted Learning'. 29-31 March Inspire 2010, London, UK. http://eprints.bournemouth.ac.uk/13247/1/Inspire2010.pdf pg no: 8, 10

[11] Cisco (2009) Cisco Packet Tracer, available at: http://www.cisco.com/web/learning/netacad/course_catalog/PacketTracer.html [Accessed 29 January 2010]

[12] Shankar Vembu, M. K. (2006). 'Towards Bridging the Semantic Gap in Multimedia Annotation and Retrieval'. *1st International Workshop on Semantic Web Annotations for Multimedia (SWAMM).* http://74.125.155.132/scholar?q=cache:DN_NGBiPjcgJ:scholar.google.com/+Towards+Bridging+the+Semantic+Gap+in+Multimedia+Annotation+and+Retrieval&hl=en&as_sdt=0,5 pg no:8

[13] Frensel, D., van Harmelen, F., Horrocks, I., McGuiness, D., Patel-Schneider, P. (2001) *'OIL: An Ontology Infrastructure for the Semantic Web*', IEEE Intelligent Systems. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=920598 pg no:38

[14] Henze, N., Dolog, P. & Nejdl, W. (2004) *'Reasoning and Ontologies for Personalised E-Learning in the Semantic Web'*, Educational Technology & Society, 7(4), 82-97. http://www.ifets.info/journals/7_4/10.pdf pg no:82

[15] Ketter, W. Batchu, A., Berosik, G., McCready, D. (2008) 'A Semantic Web Architecture for Advocate Agents to Determine Preferences and Facilitate Decision Making', *ACM*. http://delivery.acm.org/10.1145/1410000/1409554/a10-ketter.pdf?ip=194.66.74.22&CFID=26634273&CFTOKEN=55467158&__acm__=1308060281_afa4ea429e04c966c30ecc329cdb583d

[16] Teuteberg, F.(2003) 'Intelligent Agents for Documentation Categorisation and Adaptive Filtering Using a Neural Network Approach and Fuzzy Logic' in Knowledge-based Information Retrieval and Filtering from the Web (Ed. Abramowicz, W.), *Kluwer Academic*. 231-250

[17] Protégé (2009) Protégé Ontology Editor, Stanford University California, USA. http://protege.stanford.edu/ [Accessed online 28 January 2010]

[18] Stanford Centre for Biomedical Informatics Research, (2010). *Protégé - Ontology Editor and Knowledge Acquisition System*. Stanford, USA, Stanford Center for Biomedical Informatics Research supported by grant LM007885 from the United States National Library of Medicine Available from: http://protege.stanford.edu [Accessed 29 January 2010].

[19] Frensel, D., van Harmelen, F., Horrocks, I., McGuiness, D., Patel-Schneider, P. (2001) *'OIL: An Ontology Infrastructure for the Semantic Web*', IEEE Intelligent Systems. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=920598

[20] Shultz, S., Stenzhorn, H., Boeker, M., & Smith, B., (2009) Strengths and limitations of formal ontologies in the biomedical domain, RECIIS Electronic Journal of Communication, and Information and Innovation in Health, Rio de Janeiro, v.3, n.1, 31-45, Mar., 2009,

available from: http://www.reciis.cict.fiocruz.br/index.php/reciis/article/viewFile/241/253 [Accessed online 28 January 2010]

[21] Natalya F. Noy & Deborah L. McGuinness.(2009) *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford University, Stanford, CA, 94305 Available at: http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html [Accessed online 8 January 2010]

[22] Bio Health Informatics Group at The University of Manchester Department of Computer Science (2009) *The Manchester OWL Syntax developed by the CO-ODE project*. University of Manchester, UK. Available at: http://www.co-ode.org/about/ [Accessed 29 January 2010]

[23] Gruber,T. (2009) Encyclopedia of Database Systems, Ling Liu and M. Tamer Özsu (Eds.), Springer-Verlag, 2009. Available at: http://tomgruber.org/writing/ontology-definition-2007.htm [Accessed 8th January, 2010]

[24] Romero, C., Ventura, S., Delgado, J. A., De Bra, P., Salton, G. & McGill, M.J. (1993) Personalized Links Recommendation Based on Data Mining in Adaptive Educational Hypermedia Systems in *'Introduction to Modern Information Retrieval'*. McGraw-Hill.