The Anatomy of an Adaptive Multimedia Presentation System

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

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

***Abstract*—The use of multimedia presentations within learning environments is described and guidelines for the design of good E-Learning systems are identified. It is argued that a linear sequential presentation of knowledge segments is effective, but that the user is provided with optional links to relevant segments during the presentation. The synchronisation of multiple media is considered and the design of a prototype E-Learning system is discussed. The segmentation of material is then discussed and how the information can be stored in a data repository consider with respect to the requirement of accessing linked segments. Finally, the nature of adaptivity is discussed leading to a discussion of the salient parts of an adaptive multimedia presentation system.**

Keywords – multimedia, hypermedia, E-Learning, learning objects, adaptive, education.

I. MULTIMEDIA FOR LEARNING

Over the last fifteen years or so, there have many studies on using multimedia presentations to assist the learning process. Many applications have been designed to utilize the potential afforded by the use of computer-based learning systems. However, the early promise of these systems has not resulted in the widespread use of strong computer-based multimedia mechanisms within the learning environment. Instead, weak forms of multimedia have been favoured elevating form over content. It is perhaps hardly surprising that Craig, [6], shows that its use is not associated with a significant improvement in student grades.

This flexible 'one size fits all' approach to multimedia presentation makes it popular, but, Burke and James, [4], show within a business education environment, teaching abstract, conceptual and theoretical content with multimedia are more likely to be effective. However, for quantitative material requiring problem solving it may not be so effective. In these situations, they go on to say, the use of step-by-step instruction that allows students to see problems worked out in real time were more effective. This does not mean that multimedia applications cannot perform the latter tasks, it simply means that applications popularly used by teachers and lecturers generally do not do it.

So the dilemma here may be that in order to produce rich multimedia presentations which are inherently more complex, the authoring process will also need to be complex and therefore time consuming. But where is the starting point for the design of such systems? Gagne et al, [11], offers clear, if obvious, guidelines for the design of good E-Learning environments:

1. Gain the learner's attention (reception).
2. Inform the learner of the objectives (expectancy).
3. Stimulate recall of prior learning (retrieval).
4. Present the learning stimulus (selective perception).
5. Provide learning guidance (semantic encoding).
6. Elicit appropriate performance (responding).
7. Provide feedback (reinforcement).
8. Assess the learner's performance (retrieval).
9. Enhance retention and transfer (generalisation).

Also, if time and money is to be invested in the production of such materials the effect on learning outcomes needs to be clear. Krippel et al, [13], recently argued that this information is not readily available and that the true effect of multimedia technologies on learning outcomes remains unclear. More research is needed to examine educational environments where these new technologies are used to indentify improvements or underperformance over conventional pedagogies. It also needs to identify successful characteristics within certain contexts. Krippel argues that only with this evidence will educators be able to use multimedia technologies efficiently and effectively.

II. LESSON LAYOUT

If a multimedia presentation is to be designed to emulate a lesson or lecture, a good starting place would be to analyze the structure of a typical lesson and identify elements that will transfer well to these presentations. The difficulty here is that there no such thing as a typical lesson and very often delivery is adapted based on the content, teaching style and many other parameters.

One element that can be considered is the layout of a lesson and that it is usually planned. In other words, the content of the lesson has been identified by the teacher. This means that at its inception the lesson is rigid and linear. This is not to say the lesson itself is rigid, it will be adapted by the teacher based on an interaction with the learners. Deviation from the plan is acceptable; however, usually the main objectives learning outcomes will remain intact. Beasley and Smyth, [1], noted that despite multimedia learning environment giving an opportunity to explore their material in a more active, non-linear fashion, students exclusively studied the material linearly. They go on to say that this was possibly due to not being given any specific information on how to study in this way. Extending this slightly further it could be said that we are not taught to learn in this way.

Interestingly in this study, two features used in a non-linear manner were the hyperlinked glossary and the search facility.

In essence, a learning environment needs to bring together learning units and construct them into a linear form based on the learning objectives. Then at the delivery stage it needs to provide the learner with optional mechanisms to deviate from the planned path. These mechanisms can be extended to include elements seen in the classroom such as asking questions and requesting topics to be explained in more detail and providing optional links to allow the user to view related topics.

III. MULTIPLE MEDIA

Using multimedia for learning is not new and does not need to be computer-based. Teachers have used it for hundreds of years. Using more than one medium to relay information improves the efficiency of the communication. Ellis, [10], notes that the importance of multiple channels for the delivery of educational content can be found in the theory of multi-channel communication. This confirms that when information is presented by more than one channel, there will be additional reinforcement, resulting in greater retention and improved learning.

With computer-based systems the problem is not now having the computing power to present rich multimedia content as it was in the past. There may still be issues with network bandwidth and heavily hit servers, but the problems are now usually centered on the synchronization of the different media. Languages like SMIL, [5], seek to remedy this by providing a language to allow multimedia components to be synchronized and presented together. Although the presentations produced this way are impressive, authorship is complex.

IV. THE DEVELOPMENT OF A PROTOTYPE E-LEARNING SYSTEM

Using the principles of lesson delivery and synchronised multi-focus multimedia elements, a prototype was developed using Adobe Flash, [8].

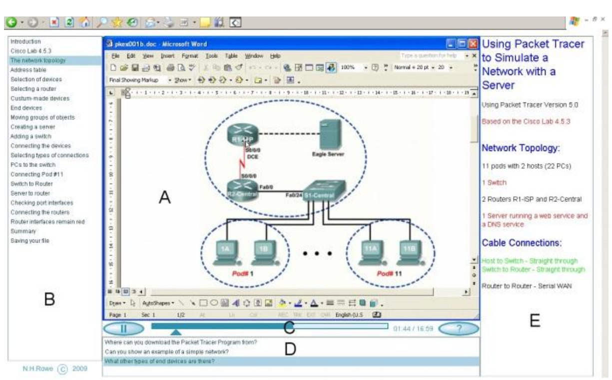


Figure 1: Screen Layout of a Multi-focus E-Learning System.

Figure 1 shows the screen layout of such a system. Here, five elements are synchronized to act from the same timeline. Element A is a traditional audio-visual presentation, B provides a table of contents that can be clicked to move within the presentation. C is a normal temporal control, D is a frequently asked questions section and E is incrementally loading HTML, (iHTML). Here the content, text and images, is displayed in real-time. Each segment of the HTML code is given a time-stamp and is not displayed until that time is reached in the presentation.

Authoring the table of contents and iHTML code is relatively easy and is carried out as a post-processing activity. The author watches the audio visual presentation through the system in the role of lecturer and is given access to additional functions that allow table of content titles and the segments of iHTML to be added to the system. These are then automatically entered into an XML configuration file and displayed during playback by users of the system accessing the system in the student role.

V. ASKING QUESTIONS WITHIN THE PRESENTATION

Panel C, in Figure 1, as well as allowing temporal control, contains a button that allows the user to ask the system a question. When the button is pressed it activates a question dialogue that allows the student user to enter a text-based question to be read by the author of the content: the lecturer. This question is marked with the time it was asked in the presentation. This question and time stamp are appended to a file on the server running the E-Learning system.

These questions that have been asked by any student user of the system are available to users of the system entering in a lecturer role. In this role, all questions that have been asked can be viewed and when selected the lecturer is taken to the part in the presentation where the question was asked. The opportunity is then given to the lecturer to answer the question with a short additional video. Once published, this video is available to all student users of the system and the question is displayed in the same manner as the table of contents being highlighted as it is relevant in questions panel, (D). However, the answer presentation is only played if the student selects the question. This allows the presentation to continue uninterrupted unless the student specifically wants to see the answer to that particular question. If a question is played the main presentation is paused while the answer video is played and resumed from the paused position when the answer video has ended. Thus, the student is given the option to view previously asked questions.

With the publishing of answers to asked questions, during the life of the presentation more questions are likely to be asked and therefore the presentation matures over time and provides more supplementary information useful to a learner viewing the presentation for the first time.

VI. ADAPTIVE E-LEARNING

Allowing learners to ask questions within the presentation and optionally view answers to previously asked questions is, in some measure, adapting the presentation to learner requirements. Generally, acknowledging the important relation between individual learners and education has along history. Shute and Towle, [15], note that the goal of aptitude-treatment interactions, (ATI), research is to provide information about learner characteristics that can be used to select the best learning environment for a particular student to optimise learning outcome. They go on to itemize four components of E-Learning:

- Content Model, including a knowledge map
- Learner Model, containing information about the user
- Instructional Model, concerned with the presentation of materials
- Adaptive Engine, which uses information from other models to drive the system

Systems can access the learner in terms of domain-dependent information and domain-independent information. The former gains knowledge of the learner through pre-tests and performance data. The latter keeps track of the cognitive abilities and personality traits of the individual. Systems concerned with adaptive instruction tend to base their *adaptivity* on assessments of emergent content knowledge or adjustments of material based on learner styles. The latter is a less suitable criterion than cognitive abilities for making adaptive instructional decisions.

It is true to say that research into adaptive hypermedia is at the crossroads of multimedia presentation and user modeling. Brusilovsky, [3], defines such systems as giving a presentation that is adapted specifically to the user's knowledge of the subject and suggest a set of most relevant links to proceed further. The second part of the definition is really a type of navigational adaptivity where the learner is given a level of control of over what content to see. So, two distinct areas of adaption are created: content level adaption, often called adaptive presentation, and link level adaption, called adaptive navigational support.

One interesting area that Brusilovsky identifies is the requirement to manipulate a presentation in certain ways according to the user needs. The information is offered in the context of *canned text adaption* and suggests applications can insert and remove text, alter fragments, stretch text, sort fragments and dim fragments. If the concept is extended to multimedia applications then these presentations can be manipulated in a similar manner. The fragments can be manipulated via some adaptive engine. The second implication leads on to another area. This is that the presentation needs to be reduced to fragments to allow these elements to be manipulated. These fragments are generally termed *learning objects* and much research has been done around their use.

A good example of adaptive navigational support offered by an application is AHA! an open source adaptive hypermedia platform, [9]. The system uses adaptive linking to suggest content for the user. It makes use of prerequisite relationships between the learning objects to link related references ensuring that the user has the required knowledge base to understand a given link. In this manner the user makes decisions about the content they wish to learn.

VII. LEARNING OBJECTS

The definition of a learning object is *any entity, digital or non-digital, which can be used, re-used and referenced during technology-supported learning*, [12]. Although the definition is easily understood and widely accepted, the advantages gained by splitting up a lesson into learning objects are somewhat controversial. One of the biggest benefits often sited are that these objects can be reused and repurposed, [2]. However, this interoperability and reusability may have been overstated in the past. McGreal, [14], points out the difficulties in taking a learning object and reusing it in a different environment. This is principally because it is difficult to create learning objects independent of the context it was made in. The likelihood is that the object bears the imprint of the ideology and culture it was produced in.

Consequently, it is difficult to standardize a learning object and an object-oriented approach, as applied to software environments. This is incongruous in the complex context of learning, especially when the learning material is based on narrow technical and specialized concepts. Despite the challenge, the concept persists driven by the joint goals of reuse and adaptivity.

Boyle, [2], describes the learning object as a wrapper around this object. This wrapper describes the component structure of the object, and includes the descriptive metadata. The learning object is thus packaged in a standard container format. This packaged object can be stored in digital repositories. The metadata permits fast effective searches to retrieve learning objects suitable for a particular purpose. A direct link can be made to the idea of learning objectives in pedagogical theory. This mapping suggests that each learning object should be based on one learning objective or clear learning goal, which links back to our original definition.

The design of the learning objects should be considered carefully to ensure they have minimal bindings to other units, (as well as being as context-free as possible). Even Boyle, [2], admits that this decoupling of learning objects is a considerable challenge and notes that this may be at odds with providing rich, integrated learning experiences. One way round this problem is to create a compound object consisting of two or more independent learning objects that are linked to try to achieve a richness not available to a single object, whilst maintaining a significant basis for re-use.

VIII. THE LINKING OF LEARNING OBJECTS

In fact, the linking of learning objects goes further than this and a particular syllabus may be defined as a linked series of these objects. Indeed, much of the research on developing E-Learning systems over the last five years has concentrated on these links. In the design of the open source adaptive hypermedia platform AHA!, (Adaptive Hypermedia Architecture), De Bra et al., [9], describe how the system has been designed to use adaptive linking to suggest content for the user. It uses, what they term, *prerequisite relationships* to link related references. The system is capable of selecting and presenting information content based on the user's previous actions which are processed and stored in a user model. The system selects and annotates the links in a way that guides the user towards the most relevant information. In this way, navigational adaptivity is provided and the system builds concept relationships between the objects.

Once the learning material has been segmented into individual learning objects, two aspects become important for the presentation of these materials. Firstly, a lesson can be considered to be a chosen sequential set of these segments and secondly that any segment presented may, to a lesser or greater degree, be connected to another segment in the learning repository. These two elements become essential to the development of any E-Learning system. Authoring a lesson to be presented becomes a process of choosing already available segments from the repository and creating new segments for areas not available. The presentation system then needs to be provided with a set of links to other relevant segments that the student may find useful and optional decide to view. The data in the repository needs to be mined to find the relevant links to each segment within the lesson.

To assist this process each segment is associated with a set of data relating to it. This data can contain simple information like name and description and also link to data used during its presentation like the iHTML text. Since this text is tightly bound with the original presentation it provides useful information to base decisions on linking one segment with another.

IX. THE STORAGE OF INFORMATION

The segmentation of individual learning objects has ultimately to be reference to the ontology of that subject area. The storage of information needs to be indexed in order for it to be retrievable. Each node is provided with a unique address which defines its location on the ordered tree. The addressing system is chosen in such a way that it corresponds a knowledge hierarchy that is specified by sections, sub-sections, sub-sub-sections etc. see Figure 2.

The ordered tree also provides the ability to define segmentation. Consider a video clip divided into 8 segments A to H. Each segment corresponds to a knowledge division or a set of knowledge divisions in the subject ontology. One typical association is seen in Figure 2.

1	1.1	1.1.1		A	0
		1.1.2			
	1.2	1.2.1	1.2.1.1	B	20
			1.2.1.2		
		1.2.2	1.2.2.1		
			1.2.2.2		
	1.3	1.3.1		C	60
		1.3.2			
	1.4	1.4.1		D	80
	1.5	1.5.1		E	90
		1.5.2			
		1.5.3			
	1.6	1.6.1		F	110
	1.7	1.7.1		G	120
		1.7.2			
		1.7.3			
		1.7.4			
		1.7.5			
		1.4.1	1.4.1.1	H	200
			1.4.1.2		
			1.4.1.3		
			1.4.1.4		
					250

Figure 2: Association of ontology divisions with video segments

X. ONTOLOGIES

According to Gruber, in a computing context, an ontology is "*an explicit specification of a conceptualisation"* [17]. This has been refined by Struder as "*a formal, explicit specification of a shared conceptualization"* where 'formal' indicates that the language of ontologies should be readable by machines as well as humans and where 'a shared conceptualization' indicates that this specification constitutes a community reference which allows the sharing of a consistent understanding of what information means and further makes possible interoperability between systems.

Usually ontologies are represented as knowledge hierarchies with the most general concepts at the top and more detailed and specific concepts at lower levels [16]. The structure of these knowledge hierarchies is naturally representable as networks, where each node on the network represents a unit of knowledge. Although many different network topologies are possible in theory such a linear, circular, hub/spoke, tree etc., the ontological model that we will be using here will be a simple ordered tree.

The ordered tree network is distinguished by 1. there is only one route from any node to any other node and 2. branches from any given node have an implicit order. These two properties ensure that the ordered tree network has the necessary properties to represent simple knowledge categorisation and sub-categorisation within an ontology. This structure will also enable a wide variety of knowledge maps to be represented.

Node addressing

The first step in building an operational structure is to reference the components of the ontology which we do by providing each node with a unique address. We adopt a positional system to delineate each sub-section within a knowledge hierarchy where each section, sub-section, sub-sub-section etc. is represented by series of numbers separated by points. This has the advantage of being scalable and universal in application. See Figure 3

Each node is represented by a unique vector. Thus

|X> = |1,2,1,1>
|Y> = |1,4,1,3>
|Z> = |1,3,2,0>

The knowledge tree network can alternatively be fully represented by the adjacency matrix A_{ij} where

$$|X_i\rangle = \sum_{j=1}^{n} A_{ij}$$

1 1.1 1.1.1
1.1.2
1.2
1.2.1
1.2.1.1 |X>
1.2.1.2
1.2.2
1.2.2.1
1.2.2.2
1.3
1.3.1
1.3.2 |Z>
1.4
1.4.1
1.4.1.1
1.4.1.2
1.4.1.3 |Y>
1.4.1.4
1.5
1.5.1
1.5.2
1.5.3
1.6
1.6.1
1.7
1.7.1
1.7.2
1.7.3
1.7.4
1.7.5
1.4.1

Figure 3: Example of unique address system for knowledge hierarchy

In the case of our example presented in Figure 3 it can be expressed in the adjacency matrix in Figure 4. This matrix is symmetric.

	1	1	1.1.1	1.1.2	1	1.2.1	1.2.1.1	1.2.1.2	1.2.2	1.2.2.1	1.2.2.2	1	1.3.1	1.3.2	1	1.4.1	1.4.1.1	1.4.1.2	1.4.1.3	1.4.1.4	2	1.5.1	1.5.2	1.5.3	2	1.6.1	2	1.7.1	1.7.2	1.7.3	1.7.4	1.7.5
1	0	1			1							1			1						1				1		1					
1.1	1	0	1	1																												
1.1.1		1	0																													
1.1.2		1		0																												
1.2	1				0	1			1																							
1.2.1					1	0																										
1.2.1.1							0																									
1.2.1.2								0																								
1.2.2					1				0	1	1																					
1.2.2.1									1	0																						
1.2.2.2									1		0																					
1.3	1											0	1	1																		
1.3.1												1	0																			
1.3.2												1		0																		
1.4	1														0	1																
1.4.1															1	0																
1.4.1.1																	0															
1.4.1.2																		0														
1.4.1.3																			0													
1.4.1.4																				0												
1.5	1																				0	1	1	1								
1.5.1																					1	0										
1.5.2																					1		0									
1.5.3																					1			0								
1.6	1																								0	1						
1.6.1																									1	0						
1.7	1																										0	1	1	1	1	1
1.7.1																											1	0				
1.7.2																											1		0			
1.7.3																											1			0		
1.7.4																											1				0	
1.7.5																											1					0

Figure 4: Adjacency matrix

Once a nodel address system is specified it then becomes possible to give quantitative values to terms such as ‘level of detail’, ‘difficulty’, ‘proximity’, ‘strength of links’ etc.

We define the following terms based on this nodel address system.

Difficulty: we define the difficulty of a knowledge node to be equal to the degree of centrality of the node – 1. In other words it is equal to the number of sub-nodes that are connected to a given node. Although it might be argued that this is a crude measure of ‘difficulty’ it has the advantage of being directly related to the complexity of the knowledge node and by association can be used as a measure of the difficulty.

Level: the level of a knowledge node as the same as the tree level of the node which is equal to the dimension of the representative vector of the node. Thus the level of node |X> = |1, 2, 1, 1> is 4 while the level of node |Z> = |1, 3, 2> is 2. We say that the **level** of a knowledge node is equal to its **importance** and represents the level of detail that a knowledge node contains.

Distance: this is a measure of how close two nodes are on the ontology. The degree of separation of knowledge segments is dependent upon the level of the nodes. Nodes at level 3 are an order of magnitude closer than nodes at level 2 and those at level 2 an order of magnitude closer than at level 1. We therefore define distance between nodes as the number of nodes traversed divided by the order of magnitude of their level. Thus two neighbouring nodes at level 1 will have a separation of 1, while two nodes at level 2 will have a separation of 0.1 and those at level 3 a separation of 0.01 Distance is therefore a measure of how close two knowledge segments are related to the subject ontology. For a tree network this is a unique value that indicates the **strength of connection** between two knowledge segments.

XI. Ontological Calculus

In order to determine the quantitative value of each of these terms it is required to define the algorithms or operations on the node addresses that will provide the appropriate values determined by the definitions. This set of operations will form a calculus enabling the manipulation of ontology.

A high level segment such as $|1.1\rangle$ contains less detail than a lower level segment such as $|1.2.1.1\rangle$ The level of a knowledge vector is given by multiplying the normalized vector by the unit covector. We define the unit covector of n dimensions $\langle U_n| = \langle 1\ldots 1,1,1|$ where there are n elements.

The normalization of a knowledge vector $|X\rangle$ we represent as $N|X\rangle$ where N is the normalization operator. Hence the level of the knowledge vector $|X\rangle$ is given by:

$$\text{Level} = \langle U_n|N|X\rangle$$

Thus for the case of $|X\rangle = |1,2,1,1\rangle$ we have

$$\begin{aligned}\text{Level } |X\rangle &= \langle U|N|1,2,1,1\rangle \\ &= \langle 1,1,1,1|1,1,1,1\rangle \\ &= 4\end{aligned}$$

Similarly

$$\begin{aligned}\text{Level } |Z\rangle &= \langle U|N|1,3,2,0\rangle \\ &= \langle 1,1,1,1|1,1,1,0\rangle \\ &= 3\end{aligned}$$

Distance algorithm

We define the n-dimensional Level covector

$\langle L_n| = \langle n,\ldots 3,2,1|$

The distance of two nodes is given by the modulus of the difference of their node addresses multiplied by the Level Order of Magnitude covector $\langle LOM|$ where

$\langle LOM_4| = \langle 1, 0.1, 0.01, 0.001|$

Thus for the two vector addresses $|X\rangle$ and $|Y\rangle$ their proximity is given by:

$$\begin{aligned}\text{Proximity}|Y\rangle|X\rangle &= \langle LOM_4|(|Y\rangle - |X\rangle) \\ &= \langle LOM_4|(|1,4,1,3\rangle - |1,2,1,1\rangle) \\ &= \langle LOM_4|0,2,0,2\rangle \\ &= \langle 1, 0.1, 0.01, 0.001\ |0,2,0,2\rangle \\ &= \langle 1\text{x}0 + 0.1\text{x}2 + 0.01\text{x}0 + 0.001\text{x}2\rangle \\ &= 0.202\end{aligned}$$

Similarly the proximity of $|Z\rangle$ to $|Y\rangle$ is

$$\begin{aligned}\text{Proximity}|Y\rangle|Z\rangle &= \langle L_4^2|(|Y\rangle - |Z\rangle) \\ &= \langle 1, 0.1, 0.01, 0.001\ |0,1,1,3\rangle \\ &= 0.113\end{aligned}$$

And similarly

$$\begin{aligned}\text{Proximity}|Z\rangle|X\rangle &= \langle L_4^2|(|Y\rangle - |X\rangle) \\ &= \langle 1, 0.1, 0.01, 0.001\ |0,1,1,1\rangle \\ &= 0.111\end{aligned}$$

It should be clear from these examples that proximity is not associative.

$$\text{Proximity}|Y\rangle|X\rangle \neq \text{Proximity}|Z\rangle|X\rangle + \text{Proximity}|Y\rangle|Z\rangle$$

Difficulty

The difficulty of a segment is defined to be equal to the degree of centrality of the node minus one. The degree of centrality is determined by the Adjacency matrix of the ontology A_{ij}

The degree of a node is the number of connections to it. We will denote the degree of knowledge vector $|X_i\rangle$ as

$$\text{Difficulty} = \langle D|X_i\rangle = \sum_{j=1}^{n} A_{ij}$$

These sets of algorithms form a calculus which enable clear metrics to be determined that can be calculated and fed into the AMPS system to facilitate adaption.

XII. The Practical Design of an E-Learning System

In practice, realization of all these concepts gives rise to two distinct functions of any E-Learning system. These are the authorship of materials and delivery of these materials. Cristea et al., [7], describe an attempt to combine two hypermedia systems, authoring with MOT, (My Online Teacher), and delivery with AHA. MOT uses domain mapping to structure and organize the resources. It uses adaption rules to build an *assembly language* of adaption. Concept weights, (meta-data), are then used to alter the presentation and make it adapt to a particular user. These weights can represent different measurable aspects of a learning fragment like difficulty or importance.

A Common Adaption Format, (CAF), sits between the two systems to convert data from MOT into a form understood by AHA. This is expressed as an XML document. Figure 4 shows both the assembly language and the CAF.

(a) if GM.Concept.weight > 10

```
        then ( PM.GM.Concept.show = true )

(b)     <CAF>
        <domainmodel>
        <concept>
                <name>Adaptive</name>
        <concept>
        <name>Adaptive HyperMedia</name>
        <attribute>
                <name>title</name>
                <contents>Adaptive HyperMedia</contents>
        </attribute>
        …
        </concept>
        …
        </domainmodel>
        </CAF>
```

Figure 5: (a) A typical fragment of assembly language, (b) A fragment of the CAF file in XML format

In this manner the systems attempt to establish a common platform and format for the representation of adaptive educational hypermedia: an extremely important goal if learning object re-use is to become a practical reality. The declaration and use of this intermediate language has another advantage. Each system can be developed and refined independently: one system generates the CAF, the other uses it. CAFs, specifically designed for testing, can be used by the presentation system.

XIII. ADAPTING MATERIALS IN AN E-LEARNING SYSTEM

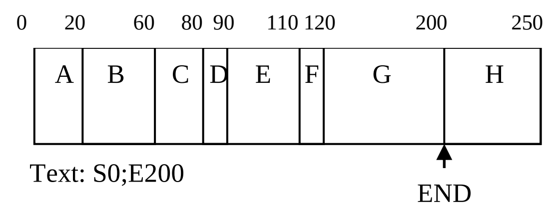
Once the decision to establishing the segment as the heart of an E-Learning system has been made, the rest of the system can be designed around it. Entities including the user and materials to test the user knowledge can be included in the E-Learning database.

In the development of the materials the educational concepts must be isolated from a unit of a course and developed into learning objects. The syllabus of a unit consists of an ordered set of concepts and a course is an ordered set of units. Each concept is formed into a segment. Initially a segment contains audio-visual resources required for its presentation. The authorship sequence continues by adding addition data to the segment including references to the AV file and the iHTML file used during presentation.

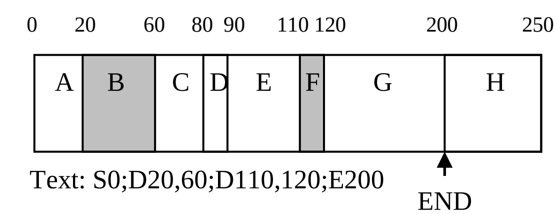
To make the segment adapt to the user's needs during presentation the author must also determine parts of the AV presentation that will be viewed at different levels of detail. By providing these different levels each segment becomes adaptable. During a presentation, the user can be presented with the segment information at a preferred level of detail. The user can then alter this level to provide more or less detail during the presentation. The system can record these levels and change these levels based on other information in the database including the results to tests linked to the segment. Thus, the system adapts to the user needs by presenting the material at the correct level of detail.

Authorship of such a system relies on the choosing fragments on a temporal basis and marking sections to be excluded or included at a particular level. Thus, more or less detail can be created to a standard form and adaptively chosen for the user. A textual code is used to allow the system to piece together the presented form for the level chosen and acts as an adaptive descriptor for the system.

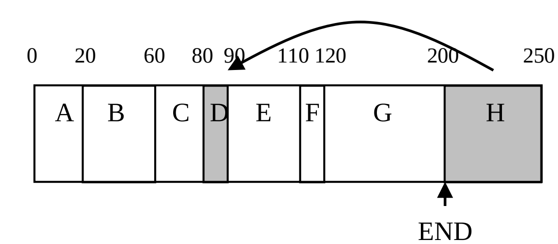
This is shown in Figure 5. Part (a) shows the media file being played as it was recorded from frame 0 to 200. The control text simply gives the end frame so that additional fragments are not played at the end of the file. Part (b) shows fragments of the media file being left out to create a less detailed presentation. Here, fragments B and C are left out of the presented sequence. The control text indicates which frames are to be removed. It also includes the end frame. Part (c) shows more detail being added to the presentation by substituting the larger fragment H in the place of the smaller fragment D. Here, more detail can be added to specific parts of the file and therefore particular concepts are elaborated within the segment. These additional fragments are added to the end of the media file and are additionally recorded at the time the presentation is made. The adaptive descriptor marks the frames to be removed and the frames to be substituted. Thus, a single media file is used for all levels of detail and adaptively presented by use of the set of descriptors at different levels.



(a) Normal level of detail, (as recorded). Segments A to G are played sequentially



(b) Less detail in presentation. Segments A, C, D, E and G are played sequentially



Text: S0;I80,200,250;E200

(c) More detail in presentation. Segments A, B, C, H, E, F and G are played sequentially

Figure 6: Three levels of detail from a single audio-visual fragment.

XIV. CONCLUSION

The E-Learning presentation system is driven from a sequential set of segments. Each of these segments has additional data connected to the AV file and an adaptive descriptor allows these additional elements to be synchronized with the original AV file. It also allows fragments to be added or removed from the segment as required adapting to the user requirements. At any stage in the presentation the detail can be manually increased or decreased. Questions can be asked, the answers published onto the system as a linked segment. Other segments within the data repository are displayed that may be relevant to the current segment. The algorithm to do this is contained in a separate system that has access to the same E-Learning database and acts independently from the presentation system. As this system discovers links between the segments in the repository they are added to the database by adding links to each segment. When the segment is presented to the user as part of a lesson these link are displayed giving the user the optional ability to display these linked segments. A strength variable keeps track of the relevance of the links and this can be displayed to the user.

The presentation side of the system runs from meta-data provided from an XML configuration file created at the time the presentation is requested by the user. Information on the user's progress is obtained from the database to pick the level of detail required for each segment. This information is obtained from the results of previously attempted tests and from changes made by the user if the segment has been previously viewed by the user.

The XML configuration file will consist of a number of essential elements for the presentation of the lesson:

- An ordered list of the segments contained in the lesson
- For each segment a list of allowed detail levels along with an adaptive descriptor for each detailing the way the content will be manipulated for that particular level and the synchronization information to present additional material, (for example iHTML blocks)
- For each segment, a list of other linked segments that are considered relevant, together with a metric indicating the strength of that relevance. The answers to previous questions asked by viewers of that segment can also form linked segments with a high value of relevance.

REFERENCES

[1] Beasley, N., Smyth, K., 2004. Expected and Actual Student Use of an Online Learning Environment: A Critical Analysis. *Electronic Journal on e-Learning.* 2(1). 43-50.

[2] Boyle, T., 2003. Design Principles for Authoring Dynamic, Reusable Learning Objects. *Australian Journal of Educational Technology*.

[3] Brusilovsky, P., 2001, Adaptive Hypermedia. *User Modeling and User-Adapted Interaction* 11. 87-110.

[4] Burke L.A., James, K.E., 2008. PowerPoint-Based lectures in business education of student-perceived novelty and effectiveness. *Business Communication Quarterly*, 71(3), 277-296.

[5] Bulterman,D.C.A, Rutledge, L., 2009. SMIL 3.0 Flexible Multimedia for Web, Mobile Devices and Daisy Talking Books. 2nd Ed. Berin:Springer-Verlog.

[6] Craig, R. J., Amernic, J.H., 2006. PowerPoint presentation technology and the dynamics of teaching. *Innovation in Higher Education*. 31(3), 147 - 168

[7] 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 (2005). Available from: citeseerx.ist.psu.edu/ viewdoc/download. [Accessed 10 March 2010]

[8] Cutts, S., Davies, P., Newell, D. and Rowe, N., 2009. *Requirements for an Adaptive Multimedia Presentation System with Contextual Supplemental Support Media*, Proceedings of the MMEDIA 2009 Conference, Colmar, France.

[9] De Bra, P., Smits, D., Stash, N., 2006. Creating and Delivering Adaptive Courses with AHA! *Proceedings of the first European Conference on Technology Enhanced Learning*, EC-TEL 2006, Springer LNCS 4227, 21-33, Available from: http://aha.win.tue.nl/ publications.html. [Accessed 10 March 2010].

[10] Ellis, T. 2004. Animating to build higher cognitive understanding: A model for studying multimedia effectiveness in education. *Journal of Engineering Education*.

[11] Gagne, R. M., Briggs, L.J., Wager, W.W. 1992. *Principles of Instructional Design*. Wadsworth Publishing Co.

[12] IEEE. 2001. *IEEE Learning Technology Standards Committee* (LTSC) IEEE P1484.12 Learning Object Metadata Working Group; WG12 Home page.

[13] Krippel, G., KcKee,A.J., Moody, J., 2010. Multimedia use in higher education: promises and pitfalls**.** *Journal of Instructional Pedagogies*, Vol 3. Available from: http://www.aabri.com/jip.html [Accessed 10 March 2010]

[14] McGreal, R. (Ed.), 2004. *Online Education Using Learning Objects*. London:Routledge, 59-70.

[15] Shute, V., Towle, B., 2003. Adaptive E-Learning. *Educational Psychologist*. 38(2), 105–114

[16] Novak, J.D., & Cañas, A.J. (2006). *The theory underlying concept maps and how to construct them*. Technical Report IHMC CmapTools 2006-01, Institute for Human and Machine Cognition.

[17] Gruber, T., “A Translation Approach to Portable Ontology Specifications”, *Knowledge Acquisition*, 5(2), 199-220, 1993.