Development of a Data Model for an Adaptive Multimedia Presentation System (AMPS)

David Newell Software Systems Research Group Bournemouth University Bournemouth, UK dnewell@bournemouth.ac.uk	Philip Davies Higher Education Bournemouth and Poole College Bournemouth, UK pdavies@bpc.ac.uk	Suzy Atfield-Cutts Software Systems Research Group Bournemouth University Bournemouth, UK scutts@bournemouth.ac.uk	Nick Rowe Faculty of Technology Bournemouth and Poole College Bournemouth, UK nrowe@bpc.ac.uk

Abstract

We investigate the requirements and nature of data models for a multimedia learning system that presents adaptable learning objects based on a range of stimuli provided by the student and tutor. A conceptual model is explored together with a proposal for an implementation using the well-known relational data model. We also investigate how to describe the learning objects in the form of hierarchical subject ontology. An ontological calculus is created to allow knowledge metrics to be constructed for evaluation within data models. We further consider the limitations of the relational abstract data model to accurately represent the meaning and understanding of learning objects and contrast this with less structured data models implicit in ontological hierarchies. Our findings indicate that more consideration is needed into how to match traditional data models with ontological structures, especially in the area of database integrity constraints.

Keywords – ***e-learning, adaptive, metadata, semantic, ontology.***

I – INTRODUCTION

In previous work [1], we proposed an Adaptive Multimedia Presentation (AMP) System to provide a semi-automated tool for learning that adapts to students' needs. A prototype was constructed and evaluated in a real class environment in the Cisco Academy at Bournemouth University [2]. This showed that undergraduate students liked using the AMPS, but would prefer more 'adaptability' in the presentation of materials. The results led the writers to conclude that more investigation was needed to find alternative, flexible methods of multimedia learning object creation, storage and retrieval. The principal aim of this paper it to look further at the conceptual, semantic, and ontological data modelling issues involved in the making a more rigorous AMP system implementation.

In section II, we set out our understanding of the learning object concept and its role in our AMP system. In section III, we look at the role of adaption and the staging of its implementation. In section IV, we present a conceptual model of AMPS and relate it to subject ontologies. In section V, we create the necessary ontology calculus to enable us to produce knowledge metrics that feed into our AMP system and use the structure of the ontology itself as a reference point for the construction of learning objects. Section VI indicates how all of this might be implemented in a relational data model, while section VII reflects on the appropriateness of using relational models for hierarchical structures. The paper concludes with Section VIII indicating future directions.

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, [3]. 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 supposed benefits is that these objects can be reused [4]. However, interoperability and reusability may have been overstated. McGreal, [5], points out the difficulties in reusing a learning object in a different environment. This is principally because it is difficult to create learning objects independent of context. The likelihood is that the object bears the imprint of the ideology and culture in which it was produced. Links between objects in different contexts may still be useful to students, because it provides another way to see a concept, and may well provide alternative applications and examples.

Boyle, [4], describes the learning object as a wrapper around content. The wrapper describes the structure of the object and includes the metadata about the object. The learning object is packaged in a standard container format which can be stored in a database. The included metadata permits fast effective searches to retrieve learning objects suitable for a particular purpose.

The Linking of Learning Objects

Breaking up knowledge into learning objects based on the content structure highlights the importance of two aspects of the presentation of materials. Firstly, a lesson can be considered to be a selected sequential set of segments and secondly, any segment presented may be connected to another segment in the database.

Authoring a lesson becomes a process of

- choosing related segments in the database
- creating new segments
- attaching metadata to the new segments to allow them to be linked, once published.

ADAPTING CONTENT

Adaptation can take many forms but it is important to realise that adaption, as in nature – so in computing, is always in response to a particular stimulus.

Stage	Stimulus	Adaption	Method
1	Student emails	production of new video segments	Manual
2	Student prior knowledge	selection of video segments	pre-lesson test
3	Student ability	selection of video segments	Real-time response

Figure 1: Staging of Adaptive Methods

The AMP system is at present only adaptive at stage 1 in responding to manually produced additional video segments to the stimulus of student emails. This is considered a low level of adaption and is not automatic. The adaption is performed by the tutor rather than the AMP system and thus requires a huge manual effort to respond to requests for further information. We plan to increase the number of stimuli which produce automatic responses. Possible stimuli will include student prior knowledge and student ability, which we call the "student signature" and will be developed further in another paper.

In order to introduce adaptation into AMPS, segments are presented with different levels of detail for each student according to the

1. level set by the original author of the segment in deciding a preferred presentation level.
2. tutor model in the AMPS can override the author level by using test information about the student's level of knowledge.
3. student is allowed to alter the level of detail presented.
4. selections of level can be made persistent.

A typical lesson segment will be 2-5 minutes long. The presentation system plays an AV file in real-time leaving the original segment intact. Metadata carried with the segment is used to cue synchronized events such as the display of an incremental HTML file. Here the file is formatted as a normal HTML file presented in paragraphs by adding it to a display box at a time specified by an adaptive descriptor.

CONCEPTUAL MODEL FOR AN ADAPTIVE PRESENTATION SYSTEM

AMPS Data Schema

The aim of the AMP system is to link together learning objects as segments so students can explore by following links, regardless of the lesson or course on which the student begins their journey. The AMP system needs to respond to the meaning of segments to enable the automation of data link creation.

The aim of this section is to carefully define terms and concepts used in the AMP system model. Textual definitions are given and then a representation is derived of the system as an ordered graph.

AMP System -Textual Description of Terms

Administrator
The administrator is a role which completes any task not related to the courses or their content. The role of adding, editing or removing students would be considered an administration task. The role of adding, deleting and editing courses, lessons and resources may be completed by the Tutors or Authors. This role may be given to human effort or automated.

Answer
An answer is the answer to a single question available on a test for the student. Questions and answers are to be determined and designed by the author. This may become another task the system can automatically undertake.

Author
The author creates the segments, lessons and/or courses by means of implementation and editing. The author may be the same person as the tutor.

Tutor
The tutor determines the intended content of courses, lessons and resources and may instruct the author on the construction of materials for delivery.
This role may be partly replaced by automation in the future.

Student
The student is the course subscriber, or person learning the course content and committed to completing a course. Once all courses to which the student has subscribed are complete the student ceases to be a student. The student may be subscribing to many courses at any one time. Subscription may be limited or prevented by the delivering institution or the tutor.

Course
The content is delivered as a set of lessons related by the sequence in which they are to be presented to the student. The content can be referred to by a single attribute known as the course title. The set of content related learning segments the student is committed to complete, or is given access to, by completing an enrolment or subscription process.

Lesson
A lesson is a set of learning segments related by the sequence in which they are to be delivered. As a course is normally made of several lessons so a lesson is normally constructed from several segments of media and the sequence related to those resources. A single lesson can be referred to by its title. It forms part of a course or a number of courses at any one time.

Segment
A segment is the description of the timeline of a single piece of media, or part thereof. Each segment conveys to the student a single point of learning. The granularity of what constitutes a single point of learning is to be determined by the tutor and constructed into a resource by the author. A segment is part of a lesson and a set of segments can be identified by the lesson title and the sequence identifier within the lesson. The delivery of the entire sequence of segments may vary if student knowledge has been proven and tested to deem a particular resource need not be presented to the student. This is part of the personalisation process.

Test
A test tests the students' knowledge of the content of a section of the curriculum. That section may be based on the course or lesson level.

Question
This is a question available on a test. There may be many answers for each question where the MCQ format is used. A set of questions is formed to become a test for a lesson or course. Questions and answers are to be determined and designed by the author. This may become another task the system can automatically undertake in future.

Class Hierarchy of Terms in Protégé Software

This can be expressed more compactly in ontological form in terms of classes and entities. The relationships between terms in the class hierarchy are shown in Figure 1 modelled in Protégé [6].

ONTOLOGICAL CALCULUS

Since the storage of information needs to be indexed in order for it to be retrievable, the segmentation of individual learning objects will need to reference the ontology of the subject knowledge area in order for it to be retrieved and structured into lessons. It will be essential therefore to construct full subject ontology [7] to which all the learning segments are related.

An ontology can be represented as a tree network where there is one and only one path between two nodes. While an ontology specifies the structure and relationships within a body of knowledge it is also necessary to determine metrics in the structure which can be used to provide measures of attributes such as complexity, level of detail or closeness of subject areas.

The first step to defining these metrics is to provide each node with a unique address which defines its location on the ordered tree.

We use an ordered tree for this description where the branches from each node are ordered so that the sub-nodes have an order of preference. [8] This structure is then used to label an ontology where fragments of knowledge have an order determined by their pre-requisites. Thus a body of knowledge is divided into section, sub-section, sub-sub-section etc. and so we adopt an addressing system which corresponds to this knowledge hierarchy where each address is correspondingly specified by sections, sub-sections, sub-sub-sections etc.

Node Address Notation

Our unique addressing system for each node is in the form of an array which has entries providing positional representation for each node level. For simplicity we use the matrix notation of Bras and Kets borrowed from quantum formalism where <X| represents the left ideal (row) and |X> represents the right ideal (column) of the matrix array.

Thus from Figure 2 we find node |X> = [1, 2, 2] and node |Y> = [2,1,1]

The first stage in producing a calculus which can be used for the determination of knowledge metrics is the mathematical representation of unique addresses for nodes within the tree network. We also define the following representations for specific elements:

The unity ideal <U| = [1, 1, 1…]   and correspondingly |U> = the equivalent column vector extendable to n dimensions

The level ideal <L| = [1, 2, 3,...] and correspondingly |L> = the equivalent column vector extendable to n dimensions

We will also have cause to make use of the Level Order of Magnitude ideal <LOM| = [1, 0.1, 0.01, …]. In addition we make the following definitions:

Segment: a segment is defined as a node together with all its sub-nodes. The total number of nodes in a segment is a measure of the amount of detail contained within a segment of knowledge and can be

Complexity: we define complexity of a knowledge node to be equal to the degree centrality minus 1which is the measure of the number of sub-nodes that are connected to a given node. Thus a knowledge node composed of many sub-nodes or subdivisions is deemed to be more difficult than one with fewer subdivisions and is a measure of **difficulty** of the knowledge node.

Level: We designate the term level applied to each node by the position it occupies in the representation. Thus the level of node |X> is 3 while the level of node |Z> 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: the distance or separation of one node from another is a measure of how close two knowledge segments are related to the subject ontology. For a tree network this is a unique value determined by the number of steps between the nodes. However the separation of knowledge segments is dependent also upon the level of the nodes traversed (i.e. 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 etc. Distance is a measure of the **strength of connection** between two nodes.

Thus to obtain the distance between two node we use an algorithm which takes the modulus of the difference between the nodes and multiplies it by the level order of magnitude vector.

Distance |X>|Y> = <LOM|[|X> - |Y>]

Thus for the nodes in Figure 2 we have the following assigned addresses

|W> = [3, 2, 0]
|X> = [1, 2, 2]
|Y> = [2, 1, 1]
|Z> = [1, 3, 0]

Hence the distance D between various nodes is

D[|X> - |W>] = [1, 0.1, 0.01] (|[1, 2, 2] - [3, 2, 0]|)
= [1, 0.1, 0.01]|[2, 0, 2]
= 2.02

While

D[|X> - |Y>] = 1.11
D[|Y> - |W>] = 1.11
D[|Y> - |Z>] = 1.21
D[|X> - |Z>] = 0.12

If we have a general node |A> = |a.b.c> then distance of |X> = |i.j.k> from |A> is given by the algorithm:

D |A> - |X> = (|a-i| * 1) + (|b-j| * 0.1) + (|c-k| * 0.01)

This set 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.

IMPLEMENTING THE DATA MODEL

Our AMP system has been structured on a relational data model, where the user data together with the knowledge content data (in the form of learning object segments) is held in a relational database

Figure 4 depicts the rudimentary data model of the AMP system which has been derived using Chen's ERA method. We expect the segment entity to hold such attributes as **Level** (a measure of the importance of the segment) and **Complexity** (a measure of the difficulty of a knowledge node) as well as **Strength** of nodel links (a measure of the ontological proximity of the knowledge areas). Each of these three metrics are determined through the ontology calculus.

Adaption is performed by additional metrics attributed to the student entity and the tutor entity. A student signature will contain a measure of the prior knowledge of the student to enable adaption at level 2, and student ability to enable adaption at level 3 in real time as indicated in Figure 1.

[pic]

However it should be noted that this data model requires that the knowledge tree (or subject ontology) is contained within the relational structure along the content-backbone of

COURSE-UNIT-LECTURE-SEGMENT

However the appropriateness of using a relational model to represent the semantics of the learning system is potentially problematic and we turn now to a consideration of the issues involved in using a relational data model to hold an essentially hierarchical ontological structure.

EVALUATION OF DATA MODEL LIMITATIONS

The object of this paper has been to produce a robust architecture and design independent of any given implementation model. A key issue has emerged that concerns the AMPS architecture. Specifically, the suitability is in question, of a relational database to store and retrieve learning objects in real time to dynamically assemble learning objects in an multi-media presentation that adapts objects to the user's learning abilities and needs.

Ted Codd introduced the relational data model with 12 rules (actually 13) of relational database management in 1969 [9]. The 'Relational Model' altered data management systems at the time because it imposes strict rules of formal logic. Previously ad hoc methods were used to stored and retrieve data items held in network or hierarchical data models [10]. These abstract data models had arisen informally from contemporary data storage structures, such as storing pointers to files that connect records. Some designers had realised that rigorous approach was needed, and were using forms of relational algebra before Codd formalised these into the abstract relational data model. The new rules ensured that stored data conformed to integrity constraints, so that a well-structured data bank only stored 'true' data and could be relied upon to only allow correct data that conformed to the integrity constraints to be stored. Other data was rejected as false.

Recently, extensions to the relational model have been suggested that create novel data models and some have been used in commercial products such as the object database called ObjectStore [11] [12]. More or less, these data models allow semi-structured or unstructured data to be stored in 'relational' databases. However, such extensions do not adhere strictly to the relational model and are considered to be ad hoc.

Whilst modelling the AMP system, the writers have found it necessary to strictly define the use of database concept by rigorously defining terms. For example, it has become of vital importance to distinguish the terms 'abstract data model' from the usage of 'data model' as implicit in database design methods.

Furthermore, we need to carefully consider structures that describe learning objects, or segments, in a subject hierarchy– which we have identified as an ordered tree structure or ontology – and contrast it with storing of that data in a relational database. The first of these is essentially hierarchical, while the abstract data model used for storing the data is relational.

While designing the AMP system data model, an attempt was made to model learning objects consisting of lesson 'segments' in a relational database. The writers have encountered e-learning application data that is structured in different ways. This is typically blocks of text, for example HTML, or multimedia data types such as animations, that are linked in network hierarchies such as tree structures, rather than simple data types normally stored using the abstract relational model. This needs further investigation.

'Semantic Modelling 'is the attempted representation of 'meaning' to allow systems to interact 'intelligently' with users [13]. However, relational databases understand little about the data they store, and what it actually means to humans. Relational database management systems 'understand' only simple data types and certain integrity constraints. Understanding or meaning is left to the user of a database when using the relational model. For the writers, semantic modelling is about the structure of meaning rather than the structure of data. The relaxation of integrity constraints in still controlled ways is required to maintain a rigorous logic to data stored, and this where most change to data management methods is occurred. The modifications result in changes to the abstract data model, in addition to the ways data itself is viewed.

Recent developments are essentially partial reversals to less rigorous, more ad hoc abstract data models, similar to the pre-relational ones, and their use is unhelpful in the context of the pure relational model.

Semantic Modelling is often referred to as a form of 'Data Modelling' (e.g. applying Chen's ERA modelling to a problem domain[14] [15]) to capture persistent data. This is useful as an aid to database design, but is distinct from the writers' interpretation of semantic modelling used in this paper.

Semantic modelling in a rigorous sense ought to relate to capturing the 'meaning' or 'intelligent interpretation' of data. The other commonly used meaning of semantic modelling relates to data modelling to design an implementation on a DBMS which is an implementation of the (abstract) relational model. Further research is needed into the nature of rigorous models, tools and techniques for semantic modelling, tools. It is a growing research area for multimedia systems in general, and the complexities of interpretation of meaning, semantics and data are compounded by the adaptive features of the AMP system.

CONCLUSION

Investigations into semantic models and semantic modelling should be strictly logical explorations into how data models and integrity constraints can be modified without rendering the database contents (facts, meanings, and intelligent interpretations) uncertain or meaningless.

Meta-learning by the AMP system requires awareness that it is participating in a learning process and therefore needs an explicit, built in 'tutor model'. The current AMP system implicitly assumes there is a real-life tutor who will perform the role of the tutor model, which involves intelligent and experienced selection of learning objects appropriate to the student.

In future, we need to construct a full, robust tutor model to automate the segmentation process, which needs detailed investigation of the nature of meta-learning []. Our vision is to build this into a novel abstract conceptual data model encompassing all the properties that are needed to make explicit the qualities of an effective 'tutor model'.

Finally, although work discussed in this paper answered research questions posed in previous papers, it has indicated further questions with a different emphasis:

1. What is the usability level of the user interface and how can this be further improved?
2. What further adaptation features are required and how are they to be evaluated?
3. What model is best employed to define the interaction between the interface and the adaptation engine?
4. What is the full specification of the ontology required and how is it captured?
5. How should database schemas be constructed for the AMPS for real-time extension at data and meta-levels?
6. How should the ontology engine structure be modelled and evaluated? Can fuzzy logic or data mining techniques be candidates for a useful algorithm?
7. How do we determine the appropriate definition of an API, possibly by means of an IDL, between the ontology, the adaptation engine and the AMP system's user interface?

We leave these questions to further papers.

References

[1] 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.

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

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

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

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

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

[7] Gruber, T., "A Translation Approach to Portable Ontology Specifications", Knowledge Acquisition, 5(2), 199-220, 1993.

[8] Newman, M., E., J., "Networks, An Introduction", Oxford University Press, 2010

[9] Codd, E.(1970). 'Data Models in Database Management, 'ACM SIGMOD Record 11, No. 2

[10] Date C.J. (2000). 'WHAT not HOW: The Business Rules Approach to Application Development' Addison-Wesley. And Date, C. (2004). 'Introduction to Database Systems', 8th Ed., Pearson.

[11] Progress (2010) Objectstore, http://documentation.progress.com/output/ostore/7.2.0/pdf/user1/basicug.pdf (Last Accessed Dec 2010)

[12] Lamb, Charles, Landis, Gordon, Orenstein, Jack, Weinreb, Dan., (1991). 'The Objectstore Database System', *Communications of the ACM* 34 (10): 50–63.

[13] Date, C., Darwen, H. & Mcgoveran, D. (1998). 'Relational Database Writings 1994-1997', Addison Wesley.

[14] Chen, P. 'The Entity-Relationship Model-Toward a Unified View of Data' (1976), *ACM Transactions on Database Systems* 1/1/1976, ACM-Press.

[15] Chen, P. (2007). 'Active Conceptual Modelling of Learning: Next Generation Learning-Base System Development', with Leah Y. Wong (Eds.). Springer.

Figure 2: Ontological Structure represented in Protégé

Figure 3: Knowledge hierarchy corresponding to an ordered tree

1
- 1.1
- 1.2
  - 1.2.1
  - 1.2.2
  - 1.2.3
- 1.3

2
- 2.1
  - 2.1.1
  - 2.1.2
- 2.2
- 2.3

3
- 3.1
- 3.2
- 3.3

Tutor

Student

Course

Unit

Lecture

Segment

Figure 4: Schematic representation of basic ERD