

Implementation and Evaluation of an Adaptive Multimedia Presentation System (AMPS) with Contextual Supplemental Support Media

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
satfield-cutts@bournemouth.ac.uk

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

Abstract – *Requirements have been captured for a multimedia presentation learning system that adapts content through interactive interventions between the student and tutor [5]. With the addition of contextual supplementary learning materials selected by a tutor responding to a series of email questions, supplementary video segments that personalise learning are added. A prototype has been developed using HTML, Flash and XML. Evaluation in this paper shows that adaptation was achieved but with some drawbacks. An analysis model at semantic and data level, needed to process an adaptive multimedia presentation system) in real-time is described, raising several research questions. Our results show that the addition of context-based rules to process and recommend descriptions of segmented multimedia components according to a bounded ontology can potentially produce dynamic adaptation of learning material in real-time. A new demonstrator application is under development.*

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

I. INTRODUCTION

Teaching programmes are replacing traditional lectures and seminars with multimedia presentation systems. However, this shift to on-line learning suffers from a number of defects due to the lack of immediacy of contact with tutors which inhibit the teaching-learning feedback loop. As discussed in a previous paper [5], a system has been proposed which supplements traditional video training materials with an adaptive multimedia interface, allowing learners to view the video tutorial and ask questions of the material presented. This system has been developed to the stage where evaluation by users has been undertaken. Feedback has been used to make the improvements presented in this paper. We envisage a system operating in real time and acting as a front-end for an Adaptive Multimedia Presentation System (AMPS).

Research Questions Addressed

This paper takes earlier work and attempts to answer the following research questions:

1. Is the student /tutor user experience improved by using the AMPS ?

2. What are the best tools to generate an executable ontology model to achieve adaptation? What form do the output files need to take?
3. How far can the current implementation of the AMPS be considered adaptive and how are adaptations improved?

The structure of the paper is as follows: Section 2 describes the initial prototype adaptive multimedia presentation system. Section 3 is a suggested approach to automating AMPS while Section 4 considers the architecture, including a brief case study that demonstrates potential adaptation techniques. Section 5 contains the student user interface evaluation findings and Section 6 is a conclusion and discussion of future work.

II. THE PROTOTYPE

A stage one prototype of AMPS was developed based on the tutors' understanding about how students would be expected to learn. This was felt to be a valuable first step in personalization [6]. We are now looking to develop the personalization further through a new level of automated adaption where the next stage involved working with student end-users to gain their direct feedback of AMPS. The prototype system shown in Figure 1 is composed of five principal parts: the main presentation panel (A), the table of contents panel (B), the supplementary text panel (E), the questions panel (D) and submit button, and timeline controls (C) for the running of the audio/video presentations. The information displayed in the table of contents is a hyperlink to a position on the timeline, so that it is possible to jump between places within the same video/animation or sequence of them.

Additional supporting notes appear on the right of the screen which may contain hyperlinks to other timelines. The words 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.

The interface also provides the student with the ability to ask questions. The student invokes a text dialogue box with a tutor triggered by a button. 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.

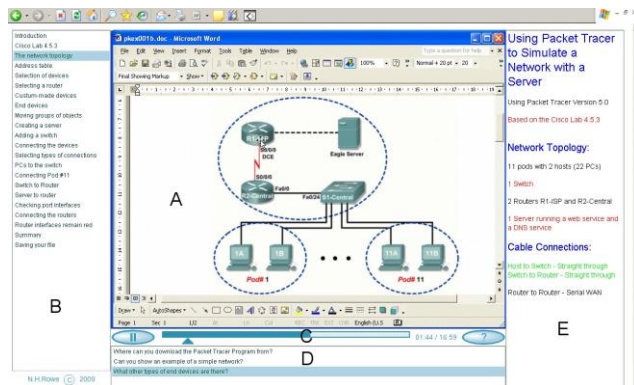


Figure 1: The AMPS prototype

An audio/video segment containing the answer can then be uploaded and added as a supplementary segment. In this way the content of the presentation grows dynamically in response to student needs. The next stage of development is to introduce automation into this very labour intensive system. We now consider how this might be approached.

III. 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 index 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 2 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:

Step 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.

Step 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's 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.

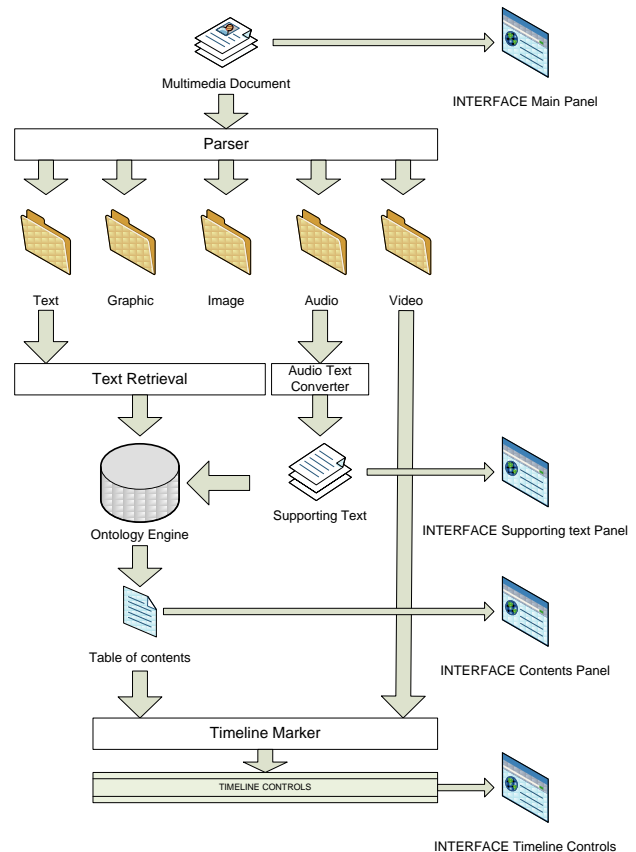


Figure 2: Automation stages in content generation

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.

Step 3: The index elements will be passed through a timeline marker to set up the timeline controls.

This deals with the first three stages in the automation programme. Stages 4 and 5 will be considered in a later paper

Additional working assumptions for the AMPS are -

- 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.

IV. ONTOLOGIES, ADAPTION ENGINES AND THE API

Developing a Networking Ontology

The writers have started work on the construction of an ontology representation in an executable form based on a sample of the case study described in Section 3. 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 [14]. 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. 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. 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. [17] 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 [15]. 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 [15]. 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.

Figure 3 shows the contents of the Protégé ontology modelling tool. 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.

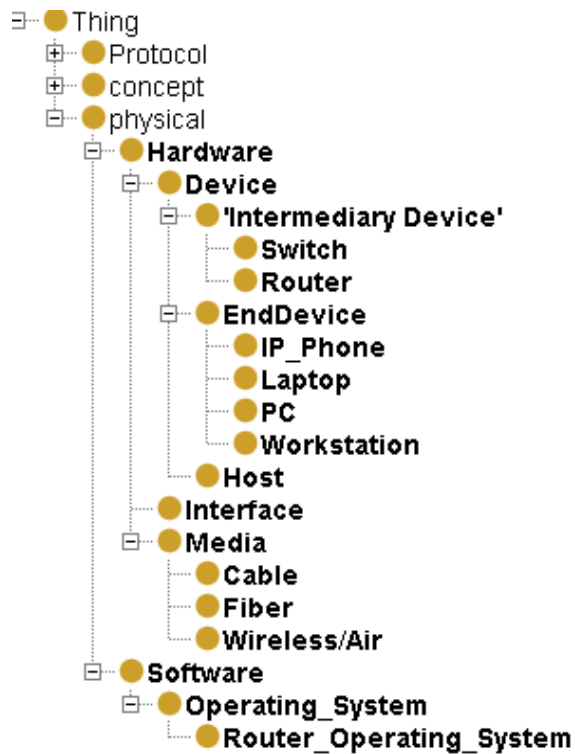


Figure 3: Sample Class Hierarchy of Digital Network Ontology Model

The ontology was extracted from Protégé as an .owl file using the Manchester OWL Syntax [2], developed by the CO-ODE project for writing OWL class expressions, or as an XML file as shown below. This new information format is expected to be useful for analytic computational purposes as an input to the ontology engine.

```

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

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

```

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

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.

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. 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 of -

- 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
- Implementation of methods
- Determination of evaluation approach will validate the effectiveness of adaptations.

Figure 4 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

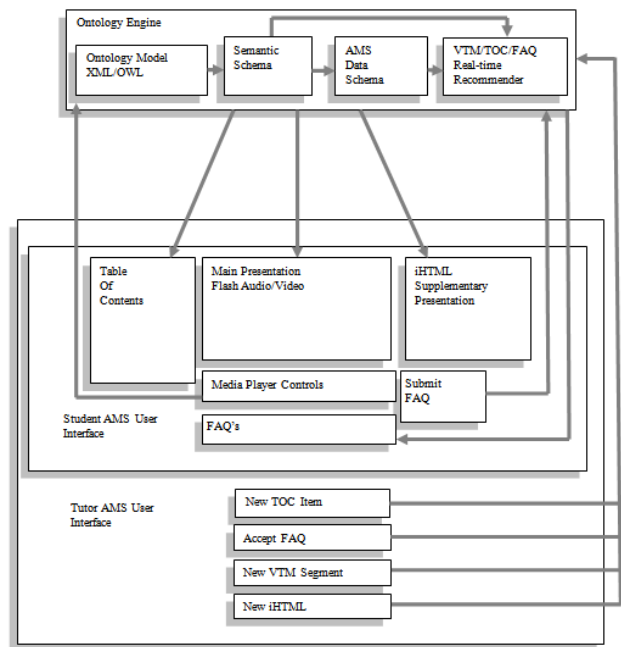


Figure 4 - A Collaboration Graph of the AMPS

: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.

OntologyEngine::AdaptiveSegment - Describes sections of multiple components or segmented learning material, e.g. VTM 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.

V. 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 [3] 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 learners assessed themselves as have good or excellent knowledge in the following areas:

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

Less than half of learners (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 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: 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 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. Second, concerning usability, a number of students commented that the audio segment was too long at 30mins and requested shorter teaching modules. This will be implemented in the next version and evaluated with students to determine the ideal duration for a presentation.

Media Segmentation

As a consequence of this feedback, a number of changes to the interface will be implemented. 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 stages 4 and 5 of the automation of AMPS. Different segments may be linked together in different ways to produce different VTMs each with their own 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. Furthermore, in order to respond to the differing needs of learners, the linking of the media segments will involve more than just a linear arrangement of segments. The response to student interaction requires branching capabilities within a "segmentation architecture". Segmentation allows the selection of material according to learning need. Students may choose to view only those segments they need to see. Additionally, the system will have the ability to respond to new learners' needs not already met, or even envisioned, by currently available material.

VI. CONCLUSIONS AND FUTURE WORK

An investigation has been undertaken into the requirements, underlying techniques and technologies needed for an adaptive multimedia presentation system (AMPS). Research issues associated with this knowledge based approach to personalisation of learning have been outlined for future exploration. A generic framework for adapting multimedia presentations through adding new content segments requested by student interaction, e.g. email, using a tree-branching sequencing system rather than the usual linear sequencing system for multimedia segments has been implemented and evaluated. Evaluation has shown that these adaptations were liked by students but do not achieve adaptation in the traditional sense because of time delays. A much more real-time approach to adaptation has been described and the foundation of an analysis model has been described.

Research Questions Addressed

We began this paper with a series of research questions, some implicitly posed in the previous paper [5] and some developed explicitly as a result of this paper. We are able to provide some complete answers and some partial ones.

Our first question asked was, "how far can the current implementation of the AMPS be considered adaptive and how are adaptations improved?" Adaptation can take many forms of response to many 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.

Our second question considered “*whether the student /tutor user experience is improved by using the AMPS presentation system?*” Feedback from students indicates the learning experience has been enhanced as evidenced by the results of the online survey presented above.

Our third question considered “*what are the best tools to generate an executable ontology model that achieves adaptation and what form do the output files need to take?*” These have been discussed at length in section IV and it has been argued that XML is the most appropriate tool to carry this metadata.

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:

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 user interface and the adaptation engine?
4. What is the full specification of the ontologies that are required and how is it best 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? Which possible data mining, or other ‘smart’ techniques are considered candidates for the algorithm or protocol?
7. 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?

We will address these questions in a future paper.

REFERENCES

[1] 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]

[2] 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]

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

[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.

[5] 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.

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

[7] Frensel, D., van Harmelen, F., Horrocks, I., McGuinness, D., Patel-Schneider, P. (2001) 'OIL: An Ontology Infrastructure for the Semantic Web', IEEE Intelligent Systems.

[8] 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]

[9] 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.

[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 (submitted).

[11] 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.

[12] Ketter, W. Batchu, A., Berosik, G., McCready, D. (2008) 'A Semantic Web Architecture for Advocate Agents to Determine Preferences and Facilitate Decision Making', *ACM*.

[13] 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]

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

[15] 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.

[16] 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)*.

[17] 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/inciis/article/viewFile/241/253> [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] 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

[20] 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]

[21] 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