The Multi-Model, Metadata Driven Approach to Personalised eLearning Services

A thesis submitted to the

University of Dublin, Trinity College

for the degree of

Doctor of Philosophy

Owen Conlan

Knowledge and Data Engineering Group,

Department of Computer Science,

Trinity College,

Dublin

Declaration

I, the undersigned, declare that this work has not been previously submitted as an exercise for a degree at this or any other University, and that, unless otherwise stated, it is entirely my own work.

Owen Conlan

Permission to lend or copy

I, the undersigned, agree that the Trinity College Library may lend or copy this thesis upon request.

Owen Conlan

ACKNOWLEDGEMENTS

There are a large number of people who have influenced my life and the path I have chosen in it. They have all, in their own way, made this thesis and work described herein possible.

Firstly, I thank my supervisor, Vincent Wade, whose expert and broad knowledge, coupled with his insightful guidance has made this work possible.

As importantly I thank my family - Aoife Brady whose love, kindness, encouragement and support has given me the motivation, reason and resolve to stay the course; my parents, Ann and Owen, for their loving assistance, unconditional support and continued belief in my education; my sister, Jennifer, for her expert English both in proofing this thesis and my undergraduate final year report; Brian, my brother, for his example of what it is to work hard for what you want; my grandparents for making all of this possible and finally Aoife's parents, Kay and Mike, for allowing an unkempt and (usually) hairy student into her life.

I would like to thank my friends many of whom I have made along the way. They have helped me, one way or another, in the completion of this thesis. Many thanks to Aoife Ahern, Paul Bouchier-Hayes, Peter Branagan, Katherine Dalton, Josephine Duggan, Niamh Flood, Elanor Foley, John Keeney, Mark Murnane, Andrew Nolan, Maire Redmond, Olivia Walsh and all of my friends who have supported me.

I would like to express my gratitude to the members (past and present) of the Knowledge and Data Engineering Group (KDEG) and the Centre for Learning Technology (CLT) in Trinity College, Dublin. I would like to especially thank David Lewis whose friendship, ability to decipher my cryptic writings and excellence as a researcher has had a significant effect on this work. Honourable mention must also go to Mark Gargan whose valuable contribution to this work cannot be overstated. KDEG is not a gathering of colleagues to me - it is one of my homes and the members of KDEG are one of my families. I also thank my friends and colleagues in the Department of Computer Science and the rest of the University.

Finally, I would like to thank the European Commission for continuing to fund the projects that I work on, such as EASEL and iClass. By gathering high quality researchers from around Europe into these R&D projects I have had the opportunity to work with and learn from many wonderful people.

Thank you all.

Before you become too entranced with gorgeous gadgets and mesmerizing video displays, let me remind you that information is not knowledge, knowledge is not wisdom, and wisdom is not foresight. Each grows out of the other and we need them all. – Arthur C. Clarke

ABSTRACT

One of the major obstacles in developing quality eLearning content is the substantial development costs involved and development time required [Marchionini, 95]. Educational providers, such as those in the university sector and corporate learning, are under increasing pressure to enhance the pedagogical quality and technical richness of their course offerings while at the same time achieving improved return on investment. One means of enhancing the educational impact of eLearning courses, while still optimising the return on investment, is to facilitate the personalisation and repurposing of learning objects across multiple related courses. However, eLearning courses typically differ strongly in ethos, learning goals and pedagogical approach whilst learners, even within the same course, may have different personal learning goals, motivations, prior knowledge and learning style preferences. This thesis proposes an innovative multi-model approach to the dynamic composition and delivery of personalised learning utilising reusable learning objects. The thesis describes a generic and extensible adaptive metadata driven engine that composes, at runtime, tailored educational experiences across a single educational content base. This thesis presents the theoretical models, design and implementation of an adaptive hypermedia educational service. It also describes how this multi-model, metadata driven approach, and the adaptive engine built in accordance with this approach, was trialled and evaluated from pedagogical, reusability and technical perspectives.

TABLE OF CONTENTS

TABLE OF FIGURES

ABBREVIATIONS

AH	Adaptive Hypermedia
AHS	Adaptive Hypermedia System
APeLS	Adaptive Personalised eLearning Service
CBT	Computer Based Training
CMI	Computer Managed Instruction
CSS	Cascade Style Sheet
DC	Dublin Core
DCEd	Dublin Core Education Working Group
DOM	Document Object Model
DTD	Document Type Definition
HCI	Human Computer Interaction
ITS	Intelligent Tutoring System
JDOM	Java Document Object Model
JESS	Java Expert System Shell
JSP	Java Server Pages
LIP	Learner Information Packaging
LLL	Lifelong Learning
LMS	Learning Management System
LO	Learning Object
LOM	Learning Object Model
SCORM	Sharable Content Object Reference Model
SQL	Structured Query Language
VARK	Visual, Auditory, Read/Write, Kinaesthetic
VLE	Virtual Learning Environment
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformation

1 INTRODUCTION

1.1 Motivation

Next generation eLearning systems need to provide greater dynamism, adaptability and flexibility to support today's increasingly pressurised learning requirements. Most current eLearning offerings deliver the same static content to all learners, irrespective of their prior knowledge, experience, preferences or goals. This typical problem, of adopting a *one size fits all* [Conklin, 87] approach to content delivery, may be addressed by tailoring the educational experience towards the characteristics of each individual learner. By tailoring the educational experience to a particular learner, next generation eLearning systems may increase the relevance, improve the intrinsic motivation and enhance the appropriateness of that educational experience. Such personalised eLearning systems should have a strong emphasis on learner empowerment and control, where the learner has an active role in scrutably shaping the educational experience.

To achieve higher levels of learner satisfaction Knowles' theory of andragogy [Knowles, 84] [Knowles, 98] indicates that learners, and specifically adult learners, need to know why they need to learn something before undertaking to learn it. It also states that they need to be responsible for their own decisions and to be treated as capable of self-direction [Atherton, 03b]. These needs are difficult to support in a traditional classroom as diverging interests, abilities and approaches to learning may not be compatible. Learning delivered online gives learners a self-controlled learning experience via a computer terminal. However, eLearning courses can also suffer from *one size fits all* if such courses are not supported with appropriate external mentoring. Such eLearning offerings have witnessed high drop out rates as learners become increasingly dissatisfied with courses that do not engage them [Meister, 02] [Frankola, 01].

A possible remedy for this dissatisfaction is to produce an adaptive eLearning course that will tailor itself to different learner needs. Intelligent Tutoring Systems (ITS) [Urban-Lurain, 02] and Adaptive Hypermedia (AH) [Brusilovsky, 01] solutions have been used as possible approaches to address this dissatisfaction by attempting to personalise the learning experience for the learner. Such systems may tailor the educational offerings to the learner’s objectives [Kaplan et al., 93] [Grunst, 93] [Vassileva, 96], prior knowledge [Milosavljevic, 97] [Hockemeyer et al., 98] [Kayama and Okamoto, 98], learning style [Gilbert and Han, 99] [Specht and Oppermann, 98], experience [Pérez et al., 95] [Vassileva, 96] and many more characteristics of the learner. Early Intelligent Tutoring Systems (ITS) traditionally embedded experts’ knowledge in the structure of its content and applied appropriate design models. However, such systems have continually been criticised for believing that this embedding of expert knowledge is sufficient for effective learning to occur [Stauffer, 96]. In reality, these early systems constrained the learner and limited the opportunities for the learner to investigate topics that the ITS deemed to be of little relevance.

Subsequent generations of ITSs used knowledge about the domain, the learner, and about teaching strategies to support flexible individualised learning and tutoring [Brusilovsky, 98c]. One of the goals of these ITSs was to adaptively deliver content. The majority of such ITSs operate on specific content that has adaptive navigational controls embedded. Similarly, the learner models utilised by such systems are highly specific to the adaptive engine at the core of the system. This gives rise to a system that may adapt effectively to a learner, yet is very difficult to repurpose. Repurposing becomes difficult as the strategy for adapting the content to the learner is an inextricable element of the adaptive engine. In this sense the early ITSs take a integrated approach to adaptivity, where none of the elements of adaptivity (learner, content or strategy) are represented discretely. Such systems include ELM-ART [Brusilovsky, 98a], Interbook [Brusilovsky et al., 98d] and early versions of AHA! [De Bra and Calvi, 98].

Adaptive Hypermedia is a newer research domain [Brusilovsky et al., 96] [De Bra, 96] [Brusilovsky, 99b] which combines some ideas from ITS and emergent WWW hypermedia concepts. Adaptive Hypermedia Systems (AHS) apply different types of learner models to achieve the adaptive selection of content and linkage of hypermedia pages to the learner [Brusilovsky, 98c]. While there tends to be a clearer separation of the learner model and content model in AHSs than that seen in the ITSs, the instructional approach utilised is rarely separated from the adaptive engine. This means that there is either no explicit and separate instructional model or that this model is embedded in the content, learner model or in the adaptive engine itself. This means that applying new or different pedagogical or instructional models, e.g. case based learning, didactic, enquiry-based learning, etc., to the content model is more difficult and would involve a re-authoring of the content model, learner model or adaptive engine. Development of traditional ITS and AHS solutions, while sometimes offering effective personalised eLearning to the learner, often embed the content sequencing logic within the educational content. This conglomerate approach makes it difficult to cater for different concerns by extending the axes of adaptation (the aspects of the learner or context that the system is adapting to) or to repurpose the educational material across many courses. Any approach used in producing personalised eLearning courses should support the learner in their learning with sound pedagogical structures and approaches. The technologies of adaptive systems cannot, in themselves, produce more effective learning results without appropriate pedagogical approaches. As such, the process of developing personalised courses or course sequences should facilitate input from not only content authors, but also instructional designers and knowledge domain experts.

The customisation of the learning experiences should enhance learner experience and increase their satisfaction with the learning offered. In particular, learner satisfaction may be heavily influenced by the appropriate tailoring of their experience of an instructional approach. It is not always easy for a learner to determine the preferences that may best influence this tailoring. Therefore, there needs to be flexibility in both how the learner controls the customisation and in their rapid, dynamic[1] re-composition of the customised courses. This influence the learner has on this customisation may take the form of explicit manipulation of their learner model or more subtle inferences made by a modelling engine, based on their interaction with the learning material. The challenge is, therefore, to provide a more flexible architecture for adaptive systems which can incorporate greater flexibility in the choice of instructional model, thus dynamically customising the learner experience of those models. Such an architecture

[1] By dynamic we emphasise the ability to immediately re-compose a learning experience based on up-to-date model information, i.e. each personalised experience is generated as the learner navigates through the content. This contrasts with static, or pre-compiled, adaptive composition, which involves explicit regeneration of an entire course prior to the learning experience.

should result in personalised eLearning offerings that provide sufficient flexibility in their adaptive axes to enable rich pedagogical courses which satisfy the learners.

1.2 Research Question

The research question posed in this thesis is whether it is feasible and beneficial to separate the elements of personalised eLearning, such as the content, sequencing logic and the learner model. With this separation the tailored courses produced should be pedagogically appropriate, coherent, satisfactory and effective for the learner.

1.3 Objective and Goals

In answering the research question posed the objective of this thesis is to propose and evaluate an architectural approach for implementing dynamic adaptive courseware which has sufficient flexibility to support multiple instructional models and multiple axes of adaptivity, whose selection is independent of the architectural implementation. The axes of adaptivity are the aspects of the learner, or their context, towards which an adaptive system may adapt. One of the primary outcomes of this objective is adaptive courseware that provides the learner with appropriate guidance in their learning based on these axes. The process of designing the adaptive courses should separate the different concerns in the development lifecycle of a personalised eLearning offering, thus enabling the course authors to utilise multiple adaptive axes and leverage appropriate pedagogical strategies to produce learning experiences that will be educationally satisfactory for the learner.

The main goals of a system produced, based on the architectural approach, should be that it –

- Delivers a satisfactory and pedagogically sound eLearning experience to the learner
- Provides different adaptive effects based on different sets of models
- Separates the different concerns and elements of learning
- Is flexible and extensible in the types of adaptivity offered
- Is easily integrated with existing Virtual Learning Environments

This thesis claims that through the separation and discrete modelling of the different elements of eLearning and adaptivity, a generic, flexible and extensible architecture for Adaptive Hypermedia Systems is possible. In particular, such a system should support the adaptive runtime creation of coherent and educationally effective experiences. Through genericity it is meant that the adaptive reconciliation engine, at the heart of the Adaptive System, should not contain the strategies and knowledge of a particular domain, but should be entirely model driven. The flexibility of the Adaptive System can be achieved by the ability to add and remove models that impact adaptivity, thus facilitating adaptivity across different properties, multiple axes of adaptivity and multiple contexts. By extensible it is meant that although the engine will be trialled and evolved using a specific set of models, new models may be added to the Adaptive Hypermedia System to produce different adaptive learner experiences without impacting the software engine implementation or architecture. In order to fulfil the objective of this thesis the following activities must be achieved:

- Researching and surveying the state of the art in Adaptive Hypermedia, with particular attention to flexibility, genericity of engine and influence of pedagogy
- Researching and surveying the pedagogical underpinnings of adult learning, which may be used in defining instructional models for adaptive courses
- Designing and implementing an architecture for producing Adaptive Hypermedia Systems and Services
- Trialling and evaluating this new technique for implementing Adaptive Hypermedia Services in the domain of personalised eLearning

The realisation of these activities involves the design and prototyping of a generic, flexible and extensible Adaptive Hypermedia System. The approach proposed is the multi-model, metadata driven approach. The multi-model, metadata driven approach seeks to enable the adaptive course author to implement multiple aspects of adaptivity, as well as diverse pedagogical approaches. The evaluation of the prototype will determine the satisfaction levels experienced by the learners through using a personalised Adaptive Hypermedia educational offering based on the approach proposed by this thesis. Also as part of this work the thesis will investigate the impact of the multi-model, metadata driven approach on the granularity of learning content[2].

1.4 Contribution to State of the Art

This thesis proposes a novel approach to developing Adaptive Hypermedia Systems called the Multi-Model, Metadata driven approach. In the proposed approach the knowledge representation of the different aspects of eLearning are partitioned into separate and discrete models to maximise flexibility, yet still produce satisfying and educationally sound personalised eLearning offerings. As part of the approach a generic and extensible adaptive engine design is proposed to allow the separate models to be reconciled dynamically at runtime into an eLearning offering personalised for the learner. One of the key contributions is the level of independence maintained between the engine and the actual models. This independence allows the engine to work with multiple adaptive course authors across heterogeneous content. The thesis also proposes a service-driven approach to integrating the Adaptive Hypermedia System with third party services such as Learner Management Systems.

1.5 Thesis Overview

This thesis proposes an innovative multi-model approach to the dynamic composition and delivery of personalised learning utilising reusable learning objects. The thesis describes the design and implementation of an adaptive metadata driven engine that composes, at runtime, tailored educational experiences across a single content base. Presented are the theoretical models, design and implementation of the adaptive hypermedia educational service prototype.

The thesis begins with two chapters reviewing the state of the art. The first is a state of the art review of adult learning theory and pedagogical approaches that support these theories. It is vital that any approach to developing Adaptive Hypermedia Systems for education should be pedagogically driven, rather than solely technically driven. This chapter is followed by a state of the art review of Adaptive Hypermedia, including the objectives and methods of adaptivity. This chapter also surveys the current state of the

[2] Digital learning content is usually referred to as Learning Objects and may be defined as a unit of educational content that is delivered via the internet.

art in Adaptive Hypermedia Systems. These two chapters provide the research background upon which this thesis is built.

The next chapter discusses the novel design approach to Adaptive Hypermedia Systems proposed in this thesis. The chapter discusses the multi-model approach, the importance of appropriate metadata and the issues concerning each of the core models. It includes an architecture for this approach and addresses the individual components needed to fulfil the multi-model, metadata driven approach, as well as the candidacy architecture used to support adaptivity. Also discussed in this chapter are the technologies supporting the multi-model, metadata driven approach, the design of the service architecture and the issues surrounding integration with third party services.

The design chapter is followed by a chapter describing a prototype implementation of the design called the Adaptive Personalised eLearning Service (APeLS). This chapter discusses how APeLS has been developed as a multi-model, metadata driven Adaptive Hypermedia Service. It includes a discussion of the component and technical architecture views of APeLS, individual sections detailing the implementation issues of the core models, the realisation of the candidacy architecture and rendering and delivery issues. Also discussed in this chapter are the implementation issues surrounding integration of APeLS with third party services. Finally, the instances of APeLS courses are outlined and discussed.

The penultimate chapter discusses the trial and evaluation of the approach taken in this thesis. It includes details of the user and technical trials. The most significant trial is that of the learners' satisfaction with using a personalised eLearning offering produced using APeLS.

Finally, this thesis is concluded with a summary of the key contributions of the multi-model, metadata driven approach proposed and a discussion of future work that may be carried forward from this thesis.

2 STATE OF THE ART – ADULT LEARNING THEORY

2.1 Introduction

Evaluations of new educational technologies tend to concentrate on the learning outcomes of instructional delivery methods rather than the pedagogical effectiveness or differences in learning outcomes [Parson, 98] [Russell, 98]. What is frequently forgotten is that the technology is just a medium used to attain a particular purpose; that it is a means to an end, and researchers feel strongly that those ends should be educational, not technological [Kirkwood, 98] [Bancroft, 01]. Technologies such as standards for representing educational material and learner information, however, cannot be ignored. It is through these specifications that information about the learning material and learner may be reused between different courses and different educational systems.

This chapter investigates the aspects of adult learning that influence the instructional design of adaptive courses. This includes an analysis of the fundamental learning theories of Behaviourism, Cognitivism and Constructivism, paying particular attention to Constructivism and the ways the learner and computer may be viewed using this approach. Also discussed in this chapter are the different properties of a learner, such as learning style and learning preference, which may be used to influence the personalisation of an adaptive course and how those properties may be modelled.

2.2 Learning Theory

This section discusses the fundamental learning theories of Behaviourism, Cognitivism and Constructivism and their potential impact on adaptive eLearning.

2.2.1 Behaviourism

Behaviourism stems from the view that psychology should concern itself with the behaviour of human beings rather than mental phenomena about which no concrete knowledge is possible [Bechtel et al., 98] [Skinner, 76] [Skinner, 77]. Behaviour theorists define learning as nothing more than the acquisition of new behaviour. From an educational perspective the behaviourist learner is viewed as a passive recipient of knowledge. Learning can then be viewed as the acquisition of this objective knowledge through rehearsal and correction [Tuckey, 92]. From a teaching perspective behaviourism maintains that the role of a teacher is to reinforce correct behaviour. Learning is measured by estimating the probability of a given stimulus producing the correct response by the frequency with which it produces the correct response. With reference to its possible application in Adaptive Hypermedia Systems behaviourism dictates that such systems should attempt to produce appropriate stimuli that elicit behaviour from the learner that may be correlated with the behaviour patterns associated with successful learning.

The behaviourist theory is, however, the least popular of the three theories presented in this section. The primary reason for the demise and lack of popularity for behaviourism is its commitment to the principle that behaviour can be explained without reference to mental activity [Cowie, 98]. This theory is relatively simple to understand because it relies only on observable behaviour and describes several universal laws of behaviour. Its positive and negative reinforcement techniques can be very effective – both in animals, and in treatments for human disorders such as autism and antisocial behaviour. Behaviourism is often used by teachers, who reward or punish student behaviours. Its benefits when used to support adult learning are, however, less clear as adult learners may be more resistant to positive or negative reinforcers to achieve the desired behavioural outcome.

The behavioural theorists believe the teacher’s job is to establish situations which reinforce desired behaviour from their students. The behaviourist expects the teacher to predetermine all the skills they believe are necessary for the students to learn and then present them to the group in a sequenced manner [Conway, 97]. The impact of this theory on adaptive eLearning is twofold – the learning system should reinforce student behaviour that it perceives to be correct and the learning system should also have a predetermined view as to the best order in which skills and knowledge should be presented.

2.2.2 Cognitivism

From the passive view of learning adopted by behaviourism, cognitivism developed quite a different view. Cognitivism makes mental processes the primary object of study and tries to discover and model the mental processes on the part of the learner during the learning process. In cognitive theories knowledge is viewed as symbolic, mental constructions in the minds of individuals, and learning becomes the process of committing these symbolic representations to memory where they may be processed.

Cognitivists maintain that there is an external reality and an internal representation of that reality. Cognitivism does recognise the existence of mental phenomena, but views the surrounding environment as an objective reality about which we may have knowledge. According to Bruner [Bruner, 60], information equals learning, so outward appearances to that effect are merely communications illuminating the result of learning rather than learning itself. As the mind seeks a view of the objective reality it goes through a number of processes when it receives information –

- Information is selectively received by **Attention**
- This information is then integrated into the inherent order of memory via a process of **Encoding**
- Information becomes knowledge when it is integrated into the existing cognitive structure
- The knowledge can then be remembered in the process of **Retrieval**

From the educational perspective the emphasis on teaching and learning strategies shifts to techniques to complement the attention, encoding and retrieval of knowledge This can be achieved by the careful organisation of content, and the use of analogies and mnemonics [Newby, 96]. Although cognitivism recognises the importance of the learner making sense of new information, the teacher maintains a central role as a transmitter of objectively correct information which the learner should absorb. The function of learning in cognitivism is to create an accurate representation of the external world in our minds.

The cognitive approach and cognitive theories emerged as a new perspective employing "information-processing ideas" rather than the behaviouristic assumptions that the learner is determined by his environments and so passively adapts to the circumstances. This cognitivistic view emphasises the active mental processing on the part of the learner. However, knowledge is still viewed as given and absolute just like in the behaviouristic school.

Computers process information in a similar fashion to how cognitive scientists believe humans process information: receive, store and retrieve. When viewed from an adaptive eLearning perspective the role of the computer in the education perspective would be to present a view of the information to be learned and drill the student until they understand it. This is similar to how behaviourism may impact adaptive eLearning except that cognitivism is also concerned with the active mental processes of the learner. Cognitive instructional designers are concerned with what was happening in the learners mind. This change in focus does not negate the design models of the behaviourist instructional designers, but builds on the existing models. A Cognitivist Instructional Design would include a lesson which bases itself on the learner's previous knowledge and then scaffolds towards the new learning objectives. It does not assume that all learners have the same past experiences or learn in the same manner. The goal is still to learn in the most efficient manner, but to break the lesson down into *chunks* that move from simple to complex to build on the learner's previous schema [Saettler, 90].

2.2.3 Constructivism

Constructivism subsumes the attention, encoding and retrieval of knowledge processes from cognitivism, but maintains that there is no single accurate representation of the world, only interpretations of experience. Knowledge is a collection of concepts which fit with the experience of the individual [Tuckey, 92]. Learning becomes a change in meaning constructed from experience [Atherton, 03a]. Learners actively take knowledge, connect it to previously assimilated knowledge and make it theirs by constructing their own interpretation [Cheek, 92].

Purist constructivism maintains that there is no reality. People acquire knowledge which is constructed and relative. The degree to which proponents of constructivism reject reality varies, but the view that knowledge is subjective and cannot be separated from the knower exists in all constructivist theories.

As far as teaching is concerned, the teacher or instructor should try and encourage students to discover principles by themselves. The instructor and student should engage in an active dialog (i.e. socratic learning). The role of the instructor is to translate and transform information to be learned into a format appropriate to the learner's current state of understanding. Curricula that are organised in a spiral manner, so that the student continually builds upon what they have already learned, are seen as particularly effective in the constructivist approach. There are a number of principles that should be adhered to to achieve effective instruction and learning within the scope of the constructivist theories –

- Instruction must be concerned with the experiences and contexts that make the student willing and able to learn (readiness).
- Instruction must be structured so that it can be easily grasped by the student (spiral organisation).
- Instruction should be designed to facilitate extrapolation and or fill in the gaps (going beyond the information given).

The teacher, or instructor, is removed from their central role of transmitting information to that of a guide to encourage learners to construct their own understanding. As there is no definitive solution to a problem, the guiding must be tentative [Atherton, 03a]. Collaboration, interaction, discussion and reflection all facilitate the acquisition and assimilation of knowledge.

From the constructivist perspective knowledge becomes an individual's interpretation of experience. Learning is the construction of a new or refined interpretation. Bruner [Bruner et al., 56] emphasised the importance of cognitive structures or mental models which must be in place to provide a structure in which the integration of new information can occur [Holcombe et al., 98]. In other words, the meaning assigned to new information is dependent on how it connects with the knowledge already present in the individual's mind. The individual constructs meaning in the context of the knowledge they already have. This construction is influenced by the individual's belief system and the culture of which they are a part.

New information is important for different reasons to the recipients of the information [Tuckey, 92]. From the perspective of adaptive eLearning the information being presented should be relevant to the learner in the framework of what they have previously learned [Henze et al., 99a] [Henze et al., 99b]. A concrete context should be established to demonstrate the authenticity of the knowledge. This aids the learner in identifying the link between the knowledge and real-world competencies. By transferring the information to other problem domains the significance of the knowledge and the generality of the concepts being applied may be demonstrated. The information should be presented in a way that is easy to assimilate [Wade and Power, 98b].

The constructivist approach implies that learners will learn more with a teacher than from a teacher [Newby, 96]. Similarly, learners will learn more with a computer than from a computer [Reeves, 98]. It is argued that traditional teacher-centric approaches to learning do not transfer successfully to technology and must be revolutionised [Wedekind et al., 98]. The philosophy must change from computers as teaching machines to computers as tools to empower learners and teachers [Oppenheimer, 97]. The full benefit of technology will not be achieved without making transformative changes in the teaching and learning strategies of students and teachers or in other words ‘what's wrong with education cannot be fixed with technology’ [Oppenheimer, 97]. While traditional teaching has required the learner to initiate cognitive tools [Katz, 96], the computer is a cognitive tool. As such, the computer can empower individuals to design their own representations of meaning during thinking, problem solving and learning [Reeves, 98] and encourage them to actively reflect on what is being learned, and actively integrate new and existing knowledge [Tuckey, 92].

2.2.4 Analysis of Learning Theory

Collins [Collins et al., 96] notes that the goal of instructional delivery is to design an educational system that transmits content and skills in a clear, well structured, and efficient manner. It is this view that dominates most curriculum and instruction, and is supported by standardised testing. The goal of the learner is to regurgitate the accepted explanation or methodology expostulated by the teacher [Caprio, 94].

The constructivist view, however, argues that the goal of education is to help learners construct their own understandings [Collins et al., 96]. Constructivism is concerned with learner creation of meaning and linking of new ideas to existing knowledge, and therefore involves a large degree of learner autonomy and initiative. The emphasis is on facilitative environments, rather than instructional goals, where the teacher assumes the role of mentor, or facilitator. As a mentor or facilitator the teacher's role becomes one of guiding mentor - stimulating initiative, play, experimentation, reasoning, and social collaboration [DeVries and Kohlberg, 87]. Dewey [Dewey, 66] argued that education depended on action, that children must actively construct knowledge by drawing it out of experiences that have meaning and importance to them.

Papert calls for further distinction of the constructivist views, by focusing on the involvement of the learner in the actual design, construction and erection of *external* products or artefacts [Papert, 80] [Papert, 93] [Harel and Papert, 91]. The idea behind using raw data, primary sources, physical, and interactive materials in real-world possibilities is to help learners generate the abstractions that bind phenomena together. Researchers at MIT use the word *constructionism* to describe the knowledge construction process that arises from the physical creation of objects [Harel, 91].

Constructivism has emerged in the last decade as an alternate pedagogy closely related to the advances in educational technology. Interest in constructivism has blossomed considerably while conventional instruction and assessment techniques have been criticised for their inflexibility [Ben-Ari, 98]. There is a turn to more flexible, open-ended, adaptive, and multi-dimensional instructional techniques as well as more qualitative, observation-based methods of evaluation. As a result, constructivism is embraced by many educational technologists and this is reflected in the plethora of multimedia and computer-based software that draw from the constructivist premises. As such, it makes an ideal basis for building a theory of learning for open, informal, and virtual learning environments.

There are, however, extreme implementations of constructivism, which have provoked reactions against it. Constructivism is considered by the behaviourists as "a label for fuzzy, unscientific thinking", as Cunningham notes in his dialogue assessing the two teaching approaches [Duffy and Jonassen, 92]. The breadth of applicability of constructivism led some to believe that it supports spontaneous, uncontrolled learning, in contrast to the systematic, organised instruction of knowledge employed by the objectivist tradition. The open-endedness of constructivist problems can be daunting for the entry-level learner. Similarly, it might be difficult for the teacher to incorporate constructivism into their teaching methods [Ben-Ari, 98]. Required course content and externally applied assessments are realities that teachers must accommodate. Changing to constructivist instruction will mean finding the appropriate balance between the existing instructional methods and this new educational practice. Finally, the efficiency and reliability of evaluation methods are questioned, as constructivist learning environments are difficult to evaluate.

In a constructivist setting, knowledge is not objective; mathematics and science are viewed as systems with models that describe how the world might be rather than how it is. These models derive their validity not from their accuracy in describing the real world, but from the accuracy of any predictions which might be based on them [Postlethwaite, 93]. The role of the teacher is to organise information around conceptual clusters of problems, questions and discrepant situations in order to engage the student's interest. Teachers assist the students in developing new insights and connecting them with their previous learning. Ideas are presented holistically as broad concepts and then broken down into parts. The activities are student centred and students are encouraged to ask their own questions, carry out their own experiments, make their own analogies and come to their own conclusions.

These theories of learning impact on the development of adaptive eLearning systems by providing insights into how learners may best learn using such systems. Behaviourism seems to provide minimal impact upon the instructional design of such systems as many of its ideas and ideals have been included and extended by Cognitivism. The criticism of Behaviourism for not referencing the internal mental processes of the learner and the documented difficulties the theory has explaining phenomena such as changes in reinforced behaviour towards related, but different challenges [Philips and Soltis, 03] seem to preclude it as a serious candidate. It seems to have been the basis of many Intelligent Tutoring Systems where the goal of computer-based learning was to present a view of the information to be learned and drill the learner until they understood it.

Cognitivism, on the other hand, is concerned not only with the observable behaviour of the learner, but also with the non-observable mental processes that occur. In this sense, cognitivist approaches may cater for aspects of the learner such as their prior knowledge and scaffold the learning experience accordingly. Cognitivism still strives to present an objective view of the knowledge domain and is ultimately aiming to present learners with that same view. Cognitivism has potential for application in the education of novice learners using adaptive eLearning systems as, like Behaviourism, consistency and drilling are reinforced, but, unlike Behaviourism, individual differences may be catered for.

Constructivism differs strongly from Cognitivism and Behaviourism as it postulates that each learner constructs their own version of a knowledge domain, thus integrating it into their current knowledge and experiences. This taking of ownership by the learner places more responsibility for the control of their learning experience onto them. This implies that constructivism may be better suited to more mature and self-motivated learners.

For adaptive eLearning systems there is a balance between Cognitivism and Constructivism that may be achieved as a learner moves from being a novice, where the learning experience may be more prescriptive, to becoming an expert, where they must take more control over their learning. Cognitivism may constrain the scope of knowledge that may be taught, but provides more explicit scaffolding in its teaching. Constructivism may facilitate learning in Open Corpus environments, but expects more responsibility from the learner.

2.3 Learning Styles

Mesick [Mesick, 76] defines learning styles as "characteristic modes of perceiving, remembering, thinking, problem solving and decision making." The learning preferences and styles of an individual form a distinctive and distinguishing feature of that individual. The ability to identify and adapt to a learner’s learning style may provide a powerful personalisation mechanism. By presenting learning material in a form that is best suited to the learner’s preferred style of learning it is hoped to improve the learner’s ability to understand and integrate that knowledge into their current knowledge framework. This improved ability to learn may manifest in both shorter learning times and in a deeper understanding of the material presented [Sadler-Smith, 96]. This section reviews a number of learning inventories and their applicability.

Kolb's [Kolb, 79] learning style model categorises learners on two continuums, *abstract/concrete* and *active/reflective*, based on personal preferences for the consumption of new information. It suggests that the learner should cycle through all four styles in order to gain a full understanding of the topic. The overview of this model is to present the learner firstly with their preferred style of learning, based on the two continuums and then cycle through the other 3 modes.

Bernice McCarthy [McCarthy, 87] [McCarthy, 96] has applied the original model set out by Kolb to categorise learners in accordance to the different cycles as follows –

- Innovative Learners: *concrete/reflective.* This learner prefers to be shown the practical applications of the new material being presented.
- Analytic Learners: *abstract/reflective.* This learner prefers to be presented with the results of research and well ordered sequential presentation of new material
- Common Sense Learners: *abstract/active.* This learner is a "try it and see" learner and would benefit from the provision of guided activities that keep them updated on where they are in the content.
- Dynamic Learners: *concrete/active.* This learner prefers the ability to explore other sources of information on the relevant topic, by providing hyperlinks to those resources.

The basis of the Myers-Briggs [Myers and McCaulley, 85] Inventory model stems from the theory of psychological types put forward by Carl Jung. The name comes from Katherine Briggs and Isabelle Briggs Myers who devised the model. Jung splits the learners into four types as follows –

- *Sensors/Intuitors* - The Sensors aspect refers to well-defined detailed procedures and practical applications of the new information. Intuitors are more concept-oriented and prefer the possibility to explore other related sites.
- *Extraverts/Introverts* - The Extravert aspect refers to the more out-focused learners, they prefer chat rooms and discussion forums to be included in the content. Introverts would prefer a more impersonal and private learning experience.
- *Thinkers/Feelers* - Thinkers prefer logical arguments and research results to be presented when consuming new information. Feelers prefer to be shown how this new information affects people, by using chat rooms and virtual teacher situations.
- *Judgers/Perceivers* - Judgers prefer a predefined agenda or structure to their learning experience. Perceivers, on the other hand, are more adaptive to changing circumstances.

With this the Myers-Briggs model constructs 16 different models of learning style, each a different compilation of the 4 types above. The behaviours and attitudes of a person, including their approach to learning, is said to be affected by this indicator.

The basis behind the Hermann Brain Dominance Instrument theory is that people can be categorised based on what quadrant of their brain is dominant. This quadrant model is based on Roger Sperry's [Sperry, 97] left/right theory and the triune brain model of Paul MacLean [MacLean, 90]. The model suggests that this dominance has a huge impact on all human behaviours and attitudes including their approach to learning. It divides the brain into four different sections as follows –

- *Left Cerebral* - This part of the brain processes logical sequencing, analysis of research results and factual information.
- *Left Limbic* - This part of the brain processes sequential and organised information (curriculum-sequencing best suits this dominance).
- *Right Cerebral* - This part of the brain processes emotions and the interpersonal skills (chat rooms and discussion forums would best suit this dominance).

- *Right Limbic* - This part of the brain processes the visual input (presentation of simulations would best suit this dominance).

For example, most lecturers will come under strong domination from the *left quadrants*. In this case their preference in this quadrant will lead to them designing a course curriculum that is *left quadrant* dominant. As a result some of the *right* quadrant activities, such as teamwork and communications, may be neglected.

The Solomon and Felder inventory [Solomon, 92] [Solomon and Felder] consists of 44 questions that attempt to classify a learning style across four dimensions –

- *Active and Reflective* - Active learners tend to retain and understand information best by doing something active with it, i.e. discussing or applying it or explaining it to others. Reflective learners prefer to think about it quietly first.
- *Sensing and Intuitive* - Sensing learners tend to like learning facts, while intuitive learners often prefer discovering possibilities and relationships.
- *Visual and Verbal* - Visual learners remember best when they see pictures, diagrams, flow charts, time lines, films, and demonstrations. Verbal learners appreciate words, either written or spoken explanations. Everyone learns more when information is presented both visually and verbally.
- *Sequential and Global* - Sequential learners tend to gain understanding in linear steps, with each step following logically from the previous one. Global learners tend to learn in large jumps, absorbing material almost randomly without seeing connections until they can see the inter-relationships.

The Solomon and Felder Learning Style Inventory views learners as preferentially taking in and processing information in different ways: by seeing and hearing, reflecting and acting, reasoning logically and intuitively, analyzing and visualizing, steadily and in fits and starts.

Although different modes of learning have been known for a long time, Fleming was the first to systematically present a series of questions with help-sheets for students and teachers to help identify these modes in the form of the VARK learning inventory [Fleming, 87]. This inventory seeks to be advisory, rather than diagnostic and predictive. Many researchers have focused on visual, aural and kinaesthetic characteristics, but Fleming subdivided the visual mode into iconic (symbolic) and text, creating four possibilities for preferences. A fifth category, multimodals, was added when it was found that approximately 60% of respondents had preferences for multiple modes.

[Fleming and Mills, 92] suggested four categories that seemed to reflect the experiences of their learners: Visual, Aural, Read/write and Kinaesthetic. The results indicate a 'rule of thumb' and should not be rigidly applied. The questionnaire is not intended to 'box' respondents into the mindset that they have been 'diagnosed', rather it is designed to initiate discussion about, and reflection upon, learning preferences.

It is not expected that any one preference will be dominant or that all participants will, to some degree, be multimodal. Initial data suggests that the number of multimodal students in a class can range from approximately 50% to 90%, depending upon the context and class make up. Approximately 50% of academic faculty seem to be multi-modal, although they usually show a preference for Read/Write as one choice [Bruen, 02]. Correspondingly, there will be some students or faculty that have a strong or very strong preference that stands out from all others. The most consistent finding from questionnaire results is that classrooms are very diverse. Faculty members cannot assume that students learn as they do. The power of VARK is that students and faculty understand it intuitively and it seems to correspond to practice.

Some multi-modal learners may need to process information in more than one mode in order to get effective understanding. Learners should be encouraged to try study strategies listed under their preferences that they may not have tried before. Experience tells us that many learners become much more successful if they develop a range of study strategies based upon their preferences. It also indicates that it is not helpful to use strategies that lie outside their preferences (e.g. mind-maps may not help if they do not have some Visual preference; mnemonics may not help if they have low scores for Read/Write). Westernised education systems place heavy emphasis upon the Read/Write mode in both instruction and in assessment. Since most teachers express a

Read/Write preference, which may constrain learners with different modalities, teachers should use a variety of modes in their presentations if they are to reach every student.

2.3.1 Analysis of Learning Styles

Both the Myers-Briggs Inventory and the Hermann Brain Dominance Instrument deal with psychological issues that, while they may be applied to learning, are not directly related to how an individual processes new learning material. They deal with the broader issue of how an individual interacts with and perceives the world. These theories and instruments have a definite application in learning. However, they possess features that enable them to be employed for learning differently. The Myers-Briggs inventory is rigorous and supported by appropriate instruments to aid in the modelling of an individual in accordance with this theory, defining the learner in accordance with its indicators. These instruments can yield complex combinations of the indicators and produce results that may not be intuitively understood by the learner.

The Hermann Brain Dominance Instrument is simplistic by comparison, yet can produce results that may be intuitively understood by the learner. The separation of behaviour and attitudes to learning into the four quadrants of the brain is a straight forward means of explaining not only the learner’s preferred mode of learning, but also how this mode may affect the learning material presented to them.

This balance between the learner’s understanding of the learning style theory being presented to them and the rigorousness of that theory is central to the ideal of a learner taking ownership and responsibility for their learning and is something that may be applied in adaptive eLearning. Any explicit modelling of the learner that a system attempts must have clear benefits for the learner. The balance between rigorousness and how understandable an instrument is may also be observed in the Kolb/McCarthy Learning Style Inventory and Solomon and Felder’s Inventory of Learning Styles. The dimensions of Solomon and Felder’s Inventory are easier and more intuitive for a learner to understand than the cycles of Kolb/McCarthy. Complexities arise in the realisation of the Kolb/McCarthy cycles in the learning material presented as the distinction and transition between the different cycles may be subtle. The distinctions between the dimensions of Solomon and Felder’s Inventory are easier to observe and may be clearer to the learner. This, again, responds to the learner taking ownership of their learning. If the learner perceives that the learning style instrument they are using accurately captures their preferences and that those preferences are reflected in the material presented to them, their ownership of the learning will be reinforced. This is dependent on two factors. The first factor is that the learner understands the learning style theory being presented to them. The more intuitive this understanding is the better. The second factor is the visibility of the influence of the chosen learning style on the material presented to the learner.

Both of these factors are strengths of the most simplistic of the learning styles presented in this section – VARK. The VARK Learning Inventory does not influence the sequence or structure of learning material. It merely influences the nature and form of the delivered learning material, i.e. whether it corresponds to a Visual, Auditory, Read/Write or Kinaesthetic aspect of the learner. This is something that the learner can intuitively understand and appreciate the effect of in the material delivered. As such, it supports the factors for reinforcing ownership of learning, but only does so in a limited area of the personalised courses presented, i.e. the presented learning material and not its sequencing or structure. From the perspective of adaptive eLearning systems, VARK is not only easy for learners to understand, but also relatively easy for systems to model and adapt towards.

2.4 Learner Properties and Modelling Learners

A learner/user model contains explicitly modelled assumptions that represent the characteristics of the learner which are pertinent to the system [Fink et al., 96]. The system can consult the learner model in order to adapt the performance of the system to each learner’s characteristics. Learner modelling allows the system to personalise the interaction between the learner and the content. To achieve effective learning this personalisation should put the content in a context that the learner can understand and to which they can relate. There are several techniques for modelling the learner and refining this model.

2.4.1 Learner Properties

There are many properties and characteristics of the learner that the designer of an adaptive educational system may utilise to produce a customised learning experience.

The properties chosen to represent the learner should be pertinent to the potential customisation by the system. The characteristics may be described in a binary, qualitative or quantitative manner. Learner characteristics which may influence how the learner interacts with an educational system are the learner's objectives, pre-knowledge, cognitive style, learning style, maturity, general ability, confidence, motivation, preferences and background. Some examples of learner features which have been used in educational systems are Novice/Expert, Adults/Children, Hurry/Browsing, First Time/Revision, Language, Technology and Disability.

The objective or goal of the learner is a description of what they are trying to achieve through a learning experience. This may be inferred by the context of the content and include learning goals and personal objectives. For example, the learner may be a novice wishing to aspire to intermediate level. Similarly, another learner may be an expert who is revising. The system may infer that when a learner follows a specific path it is because the immediate objective is to learn the piece of knowledge to which the path leads [Eklund, 95]. The global objective of a learner will be to learn the content of the course. The system can tailor the course to meet the differing objectives of both novice and expert learners. By specifying their objectives explicitly the learner is made more aware of their own learning objectives and is given more responsibility over their learning.

Adaptive systems need to gauge the level of prior knowledge of the learner as the rate and manner in which a learner assimilates knowledge is dependent on the learner's previous knowledge of the subject matter. The system must then monitor the learner's mastery of concepts and build upon the knowledge acquired by the learner as they progress through the course. Direct feedback or test results may be used to infer the knowledge of the learner at the start of the course. The system should then recognise changes in the learner's knowledge as they progress and update the learner model accordingly. Support can be gradually phased out as the learner's knowledge increases [Paolucci, 98].

The system should ensure that prerequisite concepts are known by them before they advance to a new topic. Technical terms should be avoided or explained until the learner is familiar with their use.

The computer can be used as a cognitive tool to develop higher order thinking skills. Learners who learn by associating and linking different ideas and information will be more effective at learning in a Hypermedia based system. Such learners think, perceive and solve problems in an active, exploratory manner. They exercise strategic analysis of the meaning of the subject matter [Dao and Parent, 98]. Active learners who are confident in their learning strategies regardless of the subject matter are called field independent learners. Cognitive styles which are critical and field independent are more conducive to the effective use of educational Hypermedia [Paolucci, 98]. Hypermedia systems need to allow for different cognitive styles and attempt to nurture a more analytic cognitive style in learners who adopt surface processing of the content [Dao and Parent, 98].

The learning style of a particular learner changes depending on the time, context and mood of the learner. The factors which may affect learning style include the learner's physiological and psychological state, the prevalent cognitive style of the learner and their prior experience of Hypermedia in general and the course content in particular [Paolucci, 98]. Learners may have a holistic learning style and wish to understand the context of the material they are learning. A learner with a serial learning style may tackle a new subject with a step by step approach [Dao and Parent, 98]. These learning styles correlate with the specific and generic mental models which are constructed in the mind of learners. It has been claimed that learners may be able to process information more effectively if it is presented in a manner that closely matches their learning style [Paolucci, 98].

The intelligence of the learner is a factor in their ability to assimilate information. An adaptive system can take the general ability of a learner as sourced from Grade Point Average or Psychometric testing and use this as a measure of how much guidance and support the learner may require [Paolucci, 98]. The aim should be to give the learner as much freedom and independence as they are capable of managing effectively.

Adaptive Hypermedia Systems need to take account of the sophistication of the learning skills of the learner. These skills tend to be more honed and varied in mature learners. Hypermedia systems do not teach but provide the learner with the means and opportunity to learn of their own accord [Eklund, 95] and therefore the learner needs to have acquired the skills to apply appropriate learning strategies to each learning task. The learner must be able to take responsibility for their own learning and organise the learning environment in a manner that suits their own learning style [Paolucci, 98].

Inexperienced learners in a Hyperspace environment or learners who are new to the subject matter are more likely to exhibit sequential behaviour i.e. navigating forward and back only [Eklund and Brusilovsky, 98]. These learners do not digress from a linear path because they are afraid of becoming disoriented and confused. Adaptive systems should provide security for these learners to embark on non-linear paths and be confident that they will be able to cope and find their way to the information they require.

A common feature of learners who are navigating in a non-linear way is the lack of a feeling of closure when a task or course is completed. Because there are so many possible paths, the learner is not confident that they have covered the entire information space. Adaptive Hypermedia systems can display the progress of individual learners so that they know what has been covered and where. Learners can then have the confidence to use greater initiative in their navigation [Eklund, 95].

Motivational and learning strategies are crucial aspects of self regulated learning. Motivational strategies are those strategies learners use to cope with stress and emotions that are sometimes generated when they try to overcome occasional failures and become good learners [Garcia, 95]. Sources of motivation are as varied as the learners themselves. There are extrinsic motivational factors such as exam results, degrees or certificates and improved career options. Intrinsic motivation is crucial to the Constructivist learning process and can be improved by maximising a learner's control and independence. However novices may lose motivation if they are given too much freedom and their confidence deteriorates. An adaptive system should maximise motivation by emphasising interactivity and providing feedback to the learner. It should also provide guidance and support when required.

The learner's background may include their profession, work experience, beliefs and hyperspace experience. The learner would ordinarily supply this information directly and the system may operate a stereotype model for targeted learners. Such information may be used to contextualise the information. For example, examples and case studies may be used that are appropriate to the learner's background.

2.4.2 Modelling Learners

Creating fixed stereotypes is one of the simplest ways of user modelling [Rich, 89]. New learners are categorised and the system will customise its performance based on the category that has been set for the learner. A common example would be the notion of novice, intermediate and expert users within a system. This approach is useful when a quick but not necessarily completely accurate assessment of the learner's background knowledge is required [Kobsa, 93].

The overlay model is widely used in the adaptive hypermedia systems in the educational domain. A model of the learner's knowledge is constructed on a concept-by-concept basis and updated as the learner progresses through the system. This allows for a flexible model of the learner's knowledge for each topic [Brusilovsky et al., 96].

For this model the knowledge domain must be modularised into specific topics or concepts. The complexity of the model depends on the granularity of the structure of this domain knowledge and the granularity of the estimation of the learner's knowledge. This estimation is built up by examining the sections the learner has read and the tests they have completed.

The stereotype and overlay techniques of user modelling are often combined in educational adaptive hypermedia systems. The learner may be categorised as a stereotype initially and then this model is gradually modified as the overlay model is built from information acquired from their interaction with the system. There are a number of implicit approaches that may be used in acquiring and refining the learner model. These include –

- The observation of the learner's direct-manipulative interaction with the software system.
- The analysis of the information which the learner retrieves from a database or repository [Kass and Stadnyk, 92].

The system can also explicitly ask the learner for information [Kobsa, 93] employing mechanisms such as questionnaires and tests.

There are a number of sources of information which may be used to construct a learner model. The system acquires data about the learner and infers user characteristics from this data. The validity of the assumptions depends on the technique used to acquire the information. Automatic modelling by the system may be unreliable. Any inferences made by the system about user characteristics are ultimately a guess [Espinoza and Hook, 95]. For this reason collaborative and cooperative modelling is frequently implemented. The learner describes pertinent characteristics directly and can provide feedback directly to the system by filling out questionnaires and forms. Indirect feedback is acquired from the results of exercises or problem solving tasks set by the system. The system may also track the mouse clicks and keyboard strokes of the learner to track their navigation path through the system.

2.4.3 Analysis of Learner Properties and Modelling Learners

There are many properties of a learner that may be modelled and used to produce personalised courses. The main limitation on using any learner property is the overhead involved in modelling that property. The properties of a learner, such as objectives, prior knowledge, cognitive style, learning style, maturity, general ability, confidence, motivation, preferences and background, may all be modelled, but they each impact upon the learner’s interactions with the learning course to varying degrees. Each property of the learner should, therefore, be examined for its potential impact upon the personalised course. Decisions as to what properties to model may also be based on the overhead involved in modelling the information required.

There are two fundamental approaches to modelling a learner – implicit and explicit modelling. These two approaches employ very different mechanisms for modifying a learner model and offer different levels of confidence in the information modelled. Implicit modelling does not intrude upon the learner while they are engaged in the learning process. This approach, however, has two drawbacks – the first being the lead time required before modelled information is assimilated into the learner model. As the learner is not giving information directly to the modelling instrument all information modelled implicitly must be inferred from the learner’s interactions with the system.

Secondly, as this information is inferred there is an inherent lack of confidence in it. This is often the root cause of the lead time for implicit modelling as only once the confidence in inferred information passes a threshold can it be assimilated in the learner model. The Combination approach to modelling a learner may help alleviate some of this lead time, by initially making stereotypical decisions about the learner and later refining those decisions using an Overlay Model.

Explicit modelling of the learner's properties can often yield more accurate results in a shorter time. However, explicit modelling directly impacts upon the learner's interactions with the system and removes them from their learning. A positive aspect of explicitly modelling the learner is that through this direct modelling the learner is given access to their learner model, even if it is indirectly through an instrument. The ability to manipulate how the system adapts towards the learner gives them direct control of their learning, thus supporting ownership of that learning experience.

The balance between explicit and implicit modelling is critical. The more properties we can model about a learner the more potential personalisation may be supported. However, the overhead of explicitly modelling this information from the learner is that they may become impatient with the modelling process. Conversely, if all learner properties are modelled implicitly the learner may become frustrated that they are not given control of their learning. Therefore, balance between the number of properties modelled and the approaches taken in modelling them must be achieved to maximise learner satisfaction.

2.5 Summary

In this chapter a number of learning theories were reviewed with particular emphasis on how they influence our view of the computer as an educational medium. Also reviewed were learning styles and properties of the learner that may be pertinent for personalisation along with the mechanisms for acquiring a model of the learner. These reviews are important as pedagogy should take precedence when implementing new technology [Parson, 98] and although this thesis proposes a new technical approach to implementing adaptive systems it must support and remain cognisant of a variety of pedagogical approaches if it is to be successfully adopted.

3 STATE OF THE ART – ADAPTIVE HYPERMEDIA

3.1 Introduction

Adaptive Hypermedia (AH) is one of the most promising application areas for user modelling and user-adapted interaction techniques [Brusilovsky, 94] [Brusilovsky, 01]. It has been suggested that AH systems can be useful in any situation where the system may be used by users with different goals and knowledge and where the hyperspace is reasonably large [Brusilovsky, 94]. Users with different goals and knowledge may be interested in different pieces of information presented on a Hypermedia page and/or may use different links to navigate to those pages. AH tries to overcome this problem by using knowledge about a particular user, represented in the user model, to adapt the information and links being presented to the given user.

This chapter discusses how adaptivity and hypermedia may be used to personalise to a learner's characteristics, by looking at the methods of adaptivity and how they may be applied to Hypermedia systems. This chapter also surveys some current Adaptive Hypermedia Systems and concludes with a review of current standards and specifications for representing, packaging and managing learning resources and also for representing the learner.

3.2 What is Hypermedia?

A Hypermedia system consists of Hyperdocuments. Hyperdocuments are nodes which have links through which the user can traverse to other nodes [Brusilovsky et al., 96] [De Bra and Calvi, 98]. This allows the information in the Hypermedia System to be accessed in a non-linear manner. The user can *jump* along a link from one page to another regardless of the physical location in Hyperspace of the linked page [De Bra, 98]. The content and link structure should be designed so that any possible path through the links is logically valid for the user.

The sufficient support of the variety of possible reading orders in a Hypermedia system is not a trivial task [De Bra and Calvi, 98]. The user may suffer from cognitive overload as they attempt to comprehend the expanse of unstructured information provided. Similarly, they may become disoriented within the system and not know where the current node is positioned in relation to the node which contains the information they require [Eklund, 95]. This is a phenomenon known as 'Lost in Hyperspace' [Conklin, 87] which is a frequent occurrence in users who are not experienced with Hypermedia navigation or with large Hyperspaces such as the World Wide Web. Even experienced users become spatially confused in systems whose information structure is random and incoherent.

One of the primary goals of Adaptive Hypermedia Systems is to alleviate the cognitive overload that may be experienced by users. This is achieved by deploying one or more adaptive techniques that tailor the system, or views of the system, to the user's requirements.

3.3 Objectives of Adaptivity in Educational Systems

One important aspect of adaptivity is the direction towards which the learning system shall be adaptive. Systems realising adaptivity are, in general, developed within the fields of intelligent tutoring systems [Cumming et al., 99] [Ottmann et al., 98] or adaptive hypertext and hypermedia [Brusilovsky and De Bra, 98a] [Brusilovsky et al., 98b] [Brusilovsky and De Bra, 99] [Tochtermann et al., 99]. The objectives of adaptivity are closely related to the characteristics of the learner as it is to the learner that the system wishes to adapt.

Adaptive Hypermedia Systems (AHS) bridge the gap between computer driven tutoring systems and learner driven educational environments. Two principle aims of AHSs are alleviating difficulties of content comprehension (cognitive overload) and of orientation. Adaptivity is primarily realised through content adaptation and through navigation support.

3.4 What is Adaptive Hypermedia?

Adaptive Hypermedia attempts to alleviate some of the difficulties encountered in Hypermedia systems by adapting the system individually for each user. The system collates information about each user into a user profile and this model is used to make assumptions about how best to change the system to benefit an individual user. The system may infer user objectives and help the user to discover the scope of information available or delineate a relevant path to get to the information required [De La Passardiere and Dufresne, 92].

Adaptive systems infer the requirements of a user and modify the system accordingly. This introduces the problem of balancing control between the user and the system and the issue of the extent to which a user should be made aware of system made changes, i.e. the transparency of the adaptivity. The correctness of assumptions made by the system cannot be guaranteed. This argument implies that users should be able to control the systems adaptivity. Adaptable changes are those which originate from and are controlled by the user. Adaptive changes originate from and are controlled by the system [Fink et al., 96].

The system adaptivity may be hidden entirely from the user so that the user is unaware of changes made by the system on their behalf. Alternatively the adaptivity may be negotiated with the user, allowing the user to accept or reject modifications suggested by the system. The modifications may be visible to the user, but the user may not be able to change them For example, a link which is visible as a link, but dimmed and inaccessible to the user.

Users should have some control over the adaptivity but should not have to control it continuously [Espinoza and Hook, 95]. System designers must attempt to strike a balance between the control allowed to the user and the ease of use of the system. It is imperative that users should not be surprised, disoriented or displeased by the changes made by the system [De La Passardiere and Dufresne, 92]. When the usability of the interface is in opposition to the potential effectiveness of the system, the designer must attempt to provide adequate balance.

So what features of the system are modifiable? The system may customise the link structure or format which is offered - this is known as adaptive navigation. Similarly the system may vary the content displayed - this is known as adaptive presentation. The system may adapt the modality of the content or the prominence of links or content. Orientation aids and search facilities may be included, omitted or highlighted depending on the information contained in the user model and the rules used to apply the changes [Kay and Kummerfeld, 95][Fink et al., 96].

3.4.1 What to adapt to?

There are many aspects of a user that an adaptive information system may adapt towards. They correspond in many instances to properties of the user. Highlighted in this section are a number of characteristics of the user that may be adapted to and some systems that adapt to these are referenced.

3.4.1.1 Cultural Background

This is a more classical approach to adaptivity allowing, for example, native language, familiar measures and weights, or specific ways of writing things (e.g. colloquial expressions). In the tutoring context, this may also be extended to cover other local references, e.g. by naming well-known brands, persons, or incidents. This has partly been realised in the ALEKS system [Doignon and Falmange, 99].

3.4.1.2 Preferences

This is another classical approach from the Human Computer Interaction (HCI) field. The systems user interface is adapted to the user’s preferences, generally determined through options or preferences menus [Helander et al., 97]. User preferences may be used to give the user a greater sense of familiarity and comfort with the interface with which they are interacting.

3.4.1.3 Communication Style and Needs

Users differ in their communication style, for example, in their preference for clear directives versus a broader freedom of choices. This topic also includes special communication needs, for example, in the case of handicapped learners who may need special input devices with different facilities, or who may be restricted in the selection of output devices. One example for this special type of adaptivity is the AVANTI system developed by Kobsa’s group [Brusilovsky and De Bra, 98a].

3.4.1.4 Cognitive and Learning Style

This aspect is, at least in its realisation, closely related to the former point of communication styles. Learners differ in their preferred way of learning material presentation. Examples for considering different cognitive styles are visual, textual, or auditory presentation of information. Different learning styles include the presentation of examples, presentation of theoretical knowledge, and practical exercises. An example for a tutoring system adapting according to learning styles is the CAMELEON system by Laroussi and Benahmed [Ottmann et al., 98].

3.4.1.5 Prior Knowledge

Depending on their knowledge, the learning objects made accessible to a learner are determined by applying meta information about prerequisite relationships between the available learning objects and the knowledge which is necessary for their comprehension. Systems providing such adaptivity based on the theory of knowledge spaces [Albert, 94] [Albert and Lukas, 99] [Doignon and Falmagne, 99] are the ALEKS system developed by Falmagne and the RATH [Hockemeyer et al., 98] system developed at the University of Graz [UoG].

3.4.1.6 Learning History

The learner’s learning history can be considered in two ways connected to their learning and communication style, and to their knowledge, respectively. The knowledge history does not only deal with the prerequisite relationships mentioned above. It also deals with already existing additional knowledge, including misconceptions. These misconceptions may need different explanations pointing to connections with this additional knowledge, or to differences to already known special cases of more general topics. These explanations aim to explicitly correct existing misconceptions. This approach has been realised in the AHA! system by De Bra and Calvi [De Bra and Calvi, 98]. Adaptivity to the learner’s communication style means adaptivity to the learner’s communication behaviour as observed by the system during their learning history.

3.4.1.7 Expertise

Psychological models of expertise show that novices and experts have quite different ways of acquiring knowledge within their domains. This includes not only different explanations of content, but also different approaches to navigation support. For example, more directives towards novices and more freedom towards experts.

3.4.1.8 Aims and Goals

Learners and/or teachers may differ in their conceptions about the aims and goals of the learning. A system could adapt by directing learners towards those contents they (or the teacher) have specified as a goal.

3.4.1.9 Requirements of Different Learning Cultures

Contents, structures, etc. must be adapted to requirements given from outside, often connected with formal or technical demands. Examples are existing curricula which may be predefined by public authorities, or the technical equipment available in a certain school which may depend on the responsible person's own preferences. In addition to that, material may need to provide adaptation to newly gained knowledge in the field.

3.5 Methods of Adaptive Hypermedia

There are several adaptive techniques that may be usefully employed in an educational environment. These methods include adaptive navigation, structural and historical adaptation and adaptive presentation.

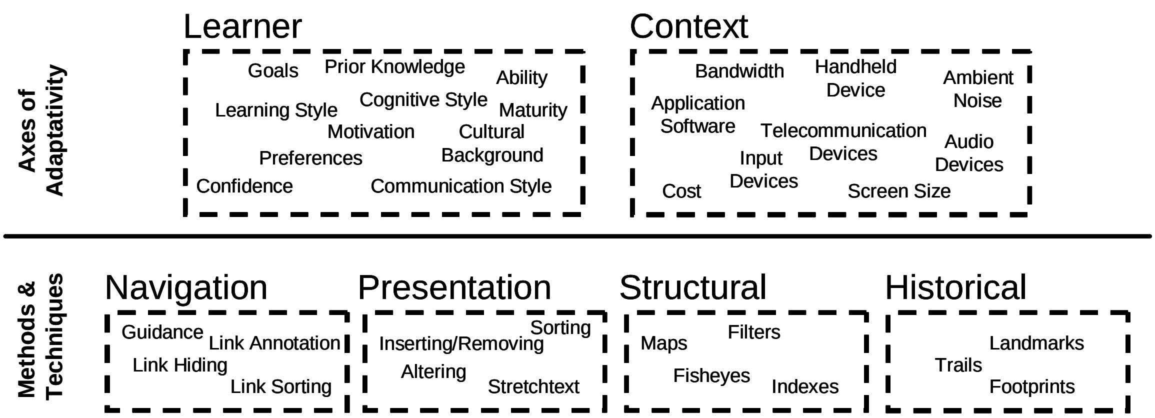


Figure 3.1, Axes and Methods of Adaptivity

Figure 3.1 shows the different axes of adaptivity, i.e. aspects of a learner or context that a system may adapt towards, and the adaptive methods that may be employed to realise adaptation towards the axes. For example, if an instructional designer wished to develop an adaptive course that adapts to the learner's prior knowledge and ability, they may use link hiding and link annotation on the navigation group of methods to display only links to content that the learner is currently capable of learning. When applied to characteristics of the learner the adaptivity may be considered as personalisation. In some circumstances adaptation towards contextual elements may also be characterised as personalisation. This is especially true if the learner has a preference for learning using a particular device and it is by using knowledge about this preference that the system chooses the device to deliver the learning experience.

3.5.1 Adaptive Navigation

Adaptive Navigation attempts to guide the learner through the system by customising the link structure or format according to a learner model. This form of adaptive navigation will determine the level of guidance and freedom granted to the learner within the system. Hypermedia experienced learners are known to be more likely to navigate in a non-linear way. Similarly learners who are familiar with the subject matter are more likely to navigate non-linearly and therefore reap the benefits of Hypermedia learning [Eklund, 95].

3.5.1.1 Relevance

Link adaptivity may require the system to decide on the relevance of certain sections of the course content to the user at a particular time. This decision is reached based on the information in the user model. As an example, the decision may be based on the current objective or goal the system has inferred for the user. If a link connects to information which is not required to meet the current goal, the link may be marked as irrelevant. Similarly, the concept to which a link is connected may require knowledge of concepts that the user has not yet covered. These links may be marked as irrelevant at this time.

3.5.1.2 Direct Guidance

Direct guidance is provided by the system by deciding where the user should link to next and presenting the user with this option. This is also called curriculum sequencing as the system enforces a path through the course. This path is customised for that particular user but the advantages of Hypermedia are lost when the user cannot organise their own learning through the Hyperspace. When link annotation and direct guidance are offered together, users who are not confident of their ability to work through the course independently are more likely to click on the next button and accept direct guidance [Eklund and Brusilovsky, 98].

3.5.1.3 Link Ordering

Link ordering is when the system sorts a list of links in accordance with their relevance to the user. The system filters the links on the basis of the user model and presents the list with the most relevant links displayed at the top. This type of link adaptivity is often used for indexes or table of contents. A user who is inexperienced with the content of the course or with Hyperspace generally can be disoriented by a link order which is unstable.

3.5.1.4 Link Hiding

Link hiding restricts the navigational choice offered to a user. The system decides what links are not relevant to the user and changes the format to that of regular text so that the link is not displayed as a link. The link may be removed completely so that the user cannot access it even accidentally [De Bra and Calvi, 98]. Link hiding can reduce the cognitive load on the user and conceal the complexity of a course while supporting the stable ordering of links. However, the usability of link hiding is questionable for a number of reasons. Users do not like to be restricted. There is a danger that the user will form an incorrect mental map of the Hyperspace. A sense of completion of the course will be difficult to attain when the user cannot be confident that all the links have been displayed.

3.5.1.5 Annotation

Annotation refers to adding information to a link so that the user has more of an idea of where the link will lead and whether it complies with the current objective of the user. Link annotation allows the user to be advised as to the degree of relevance the system applies to a link on the basis of their user model, and the user may then choose their own path. A link can have a number of different states, the values of which may be displayed to the user by colour, icons, or font formats.

World Wide Web browsers currently use link states with two values - visited links (the default for which is purple) and unvisited links (the default for which is blue). Adaptive Hypermedia Systems can extend this idea to show links with three states to signify concepts that are learned, well-learned or unknown. Links with numerous states have been implemented in Adaptive Hypermedia systems: visited, unvisited, current, suggested [Eklund and Sawers, 96]. The link may be changed to a light colour to suggest that the link is dimmed - this gives the effect of hiding without restricting the user. Annotation gives the user a degree of freedom and supports stable ordering and the formation of correct mental maps [Eklund et al., 97].

3.5.2 Adaptive Presentation

Adaptive Presentation is the customisation of course content to match learning characteristics specified by the user model. The granularity may vary from word replacement to the substitution of pages or the application of different media. Content may be customised to contain additional information, prerequisite information or comparative explanations.

This form of adaptivity may be implemented by fragmenting the constituent content components into discrete words, phrases or paragraphs. These components, or pagelets, constitute a discrete unit of information about a concept. The pagelet is displayed if the user model conforms to required conditions for the display of that pagelet. For example, if a learner has not covered a prerequisite concept for a given page the relevant pagelet may be included.

With this approach different pagelets may be displayed for different learners. An example would be a technical term or acronym with which the learner is unfamiliar. The system may substitute the unfamiliar content until the learner can be introduced to the technical term or acronym.

If the courseware is constructed dynamically each learner may potentially see an individually tailored course that is different to the course displayed for all other users.

3.5.3 Structural Adaptation

Structural Adaptation attempts to give the learner a spatial representation of the Hyperspace environment. This representation is based on the user model and is hoped to provide the learner with a sense of position within the environment and a sense of the size of the environment itself. Overview maps, local maps, fisheyes, filters and indexes are all structural aids which the system may adapt for the learner.

3.5.4 Historical Adaptation

Historical Adaptation attempts to give a time context to the learner by adapting representations of the learner's path through the system. History trails, footprints which are made by the system, landmarks which are made by the learner and progression cues may be customised by the system for the learner.

3.5.5 Analysis of the Methods of Adaptivity

Although there are four groupings of adaptivity methods listed in this section it is the Adaptive Navigation and Adaptive Presentation groups that are most prevalent in their use. Adaptive Navigation may be thought of as dealing with a conceptual level – through adapting the navigation links presented, and the structure they are provided in, the concepts presented to the learner and the way the learner browses to them are tailored. Conversely, Adaptive Presentation may be considered the *instantiation* of navigation concepts, i.e. the realisation of the concept as a learning object or learning event.

If this distinction is maintained then navigation structures created through Adaptive Navigation should not refer directly to learning objects. This distinction between content and concept could facilitate reuse of Adaptive Navigations. Similarly, the decoupling of content from the navigation structure means that it too may be reused. The overhead of this approach is that there needs to exist a mechanism to reconcile the concepts, which are the basic vocabulary of the Adaptive Navigations, and the learning content that will be used to present those concepts.

This separation facilitates the re-examination of user characteristics to which we wish to adapt. For example, does a learner's prior knowledge impact the presentation of a concept or merely its selection in the first case? In this example, prior knowledge will probably impact both, but will have a more significant impact at the concept selection stage. Thus, separating Adaptive Navigation and Adaptive Presentation in more than a theoretical sense facilitates separation of concerns within an adaptive system.

3.6 Survey of Adaptive Hypermedia Systems

The goal of this survey is to present a number of Adaptive Hypermedia systems, both current and a few years old, in order to highlight the differences between these systems and emphasise the benefits of the multi-model, metadata driven approach proposed by this thesis. This section contains an overview of each system under consideration followed by an analysis that examines each system under the areas of User Modelling, Adaptive Mechanism and Reusability of Components.

3.6.1 Adaptive Hypermedia Systems in Education

Much of the focus in adaptive hypermedia for educational courseware has attempted to alleviate the difficulties of content comprehension and orientation. Adaptive presentation techniques which affect changes to both the selection of different media depending on a user's preferences and adaptation of the content based on an individual's user model are beginning to show success. Also the use of adaptive navigation, which can affect changes to the link structure between elements of the hypermedia courseware based on an individual user's (mental) model, has proven effective since learners using such systems have demonstrated faster learning, more goal-oriented attitudes and take fewer steps to complete a course.

To achieve the maximum effectiveness from the use of non-adaptive Hypermedia in an educational context there are some features of learners that are particularly significant. These include pre-knowledge, cognitive style, maturity, general ability, confidence and motivation. These features influence the ability of learners to accept effectively the additional mental load caused by the need to monitor and self-evaluate as well as learn [Specht, 98a].

Although increasing learner control is thought to increase the learner's motivation and engagement, results in performance using adaptively controlled environments have been superior to systems within which the user is left to their own devices [Specht, 98b]. Studies have shown that users of educational Adaptive Hypermedia systems are faster, more goal-oriented and take fewer steps to complete the course. It is claimed that Adaptive Hypermedia learners are less likely to repeat the study of content they have already covered [Eklund and Brusilovsky, 98].

3.6.2 PEBA-II

PEBA-II is a text generation system that dynamically generates descriptions of animals in an interactive hypertext environment. The system uses frame based techniques to dynamically create hypertext nodes for an individual user. The fragments into which the data is divided are called slots which are assembled into a node. The system decides what slots to present and the order into which the slots should be organised on the page. Natural Language Generation aims to produce coherent language text from an underlying representation of knowledge. Natural Language techniques are used in PEBA-II to ensure that the resulting page of text sounds natural to a human reader. Comparison is used in the PEBA-II system to distinguish or liken new concepts to known concepts thus building on the knowledge that the user already has and building on the knowledge that the user is creating by following the course. The system identifies, compares and contrasts animals with other animals with which the user is familiar.

Technical words are not included for naive users. The user model is initialised by a stereotype of expert or naive user and an overlay model specifies what animals the user already knows from those indicated as known by the user and the animals and concepts covered in the course, i.e. the discourse history. The system only marks as known those animals or concepts that were recently encountered. The fact that a user may forget is therefore taken into account. PEBA-II uses a phrasal lexicon so that words to short phrases to long phrases can be decomposed as required [Milosavljevic, 97] [Milosavljevic et al., 97] [Milosavljevic and Oberlander, 98]. The use of comparison is interesting. For example, the length of the Echidna is compared to the length of the domestic cat for a naive user. Expert users would be given the length in centimetres. The implementation of Natural Language techniques gives greater flexibility to the granularity of text fragments. The cost of this, however, is complexity.

3.6.3 InterBook

Originally intended to break the paradigm of simply transferring day-to-day course content to the web (with an associated tangle of links), this system was designed to assist in creating a responsive and adaptable distance learning scheme based on web technologies. This system is based on the metaphor of an *electronic textbook*. The purpose of the Interbook [Brusilovsky, 96b] design is to simplify the process of creating adaptive electronic textbooks on the WWW. The precursor to InterBook was the WWW based LISP textbook called ELM-ART [Brusilovsky, 96a]. InterBook uses concept-based indexing and an overlay user model to provide adaptivity. A domain concept is an independent unit of knowledge in a given domain. Concept based indexing is used to provide additional information about the content of each hypermedia page by indexing it with related domain concepts.

The glossary, which also functions as an index, is a central part of an Interbook electronic book. Each concept is an outcome concept or a prerequisite concept. An outcome concept represents the piece of text or other media that explains that concept. A prerequisite concept is a concept on which the current node is dependent. Each concept has a glossary page and a text unit and the glossary exhibits the same structure as the domain knowledge. Links connect each unit of text to the glossary for each concept. Links from the glossary to the unit of text which explains it are generated dynamically depending on the user model. The table of contents is also comprised of clickable links. Prerequisite-based help gives support to the user when in difficulty by listing the prerequisite concepts required for that task or module.

Adaptive Navigation is implemented by link annotation using checkmarks and coloured balls. A green ball and bold text designates the state of the link as ready and recommended which implies that all prerequisites are well-learned or at least learned. A red ball and italic text designates a link that connects to content that is not ready to be learned which suggests that some prerequisites are not yet learned. A white ball is used when the link connects to information that is not new implying that all outcome concepts are learned or well-learned. Checkmarks by a link show that a link has already been visited. If the link is small then the content behind the link is known implying that learning has started. A medium checkmark designates content that is learned, while a large checkmark indicates content that is well-learned.

The overlay model is employed to represent the knowledge state of a user in the user model. However, in addition to the overlay model this system permits the definition of *learning goals*; sets of attributes that the user should acquire over the course of the systems operation. In addition to page visits, the Learner Model in this system takes account of factors such as quiz results and problem solving.

The Interbook Domain Model is based around a set of concepts, “elementary pieces of knowledge of the domain”. This concept space is arranged hierarchically within a glossary. Each concept constitutes a node within the glossary, and each node constitutes a hypermedia document. Concepts are arranged within books, which are arranged within bookshelves, consisting of several books on the same topic. Each book contains a spectrum, consisting of the prerequisite data for the book, along with its content. Thus, a book can be considered as a function with inputs (the prerequisites) and outputs in the form of an expression of the knowledge gained.

Adaptation Mechanism Once again, the overlay model is employed to map the Learner model to concept requirements. Content selection is based on the sequence of learning goals and their constituents. Two primary adaptation schemes are presented, which take some account of learning style. The first is to provide Direct Guidance, where the system rates and arranges nodes within the glossary. The overlaid model permits the system to rate links as ready to be learned, not ready (prerequisites not met) and known. In addition, the system can track whether or not a node has been visited. The second learning model is to present Prerequisite-based Help. This model can perhaps be most useful in assisting a learner with difficult or ill-understood topics. In this mode, the system lists and ranks the prerequisite links for a topic, presenting them for the learner's perusal. A topic rank is based on the number of relevant concepts addressed. This model lends support for a back propagated mode of learning.

The technology used includes HTML, LISP and the Common LISP Hypermedia Server (CL-HTTP). The adaptivity is implemented in a relatively cheap and useful way [Brusilovsky, 96b] [Brusilovsky and Schwarz, 97] [Eklund and Brusilovsky, 98]. Annotating links as undesirable or forbidden may be an unattractive option for confident users who wish to learn by exploring and do not wish to be guided. Link annotation should be intuitive and unobtrusive, these conditions are not fully achieved in the Interbook system.

Presentation The Interbook interface is arranged around two main windows –

- The Glossary Page: The glossary relationships are designed to resemble the semantic organisation of the concepts and this provides the primary navigation method.

- The Textbook Window: Consisting of three frames - the Navigation Bar, the Concept Bar and the Text Window.

In the process of Direct Guidance, the Concept Bar is employed to list both the prerequisite and resultant concepts of a particular topic, as presented in the Text Window. In addition, the Navigation Bar contains various standard buttons employed to navigate the text. This includes the portion of the glossary related to the topic, with links annotated for rating via colour (red for not ready, green for ready, white for topics which yield no new information). A checkmark icon indicates the visited status of the topic.

Content Authoring Content authoring in the Interbook system is performed in a number of stages –

1. The content for a concept is written and marked up in Microsoft Word, based on the style mechanism provided.
2. This annotation is augmented with conceptual relationships, which are generated via the use of pre-defined blocks that provide information on prerequisites and resultant knowledge.
3. These Rich Text Format files are converted to HTML, and interpreted automatically to create hypertext nodes and LISP rulesets.

Though the rule specification system of Interbook is basic, this system has a number of very desirable features with regard to the user interface. In particular, the explicit listing of prerequisites and requirements, along with a simple and direct visual representation of the adaptation state. There is no specific mention of user context within the Interbook system. In particular, the fact that rulesets are generated purely on the basis of input and output concepts makes the introduction of non-conceptual factors in adaptation difficult without major revision. However, the interface within the system does provide a useful clue for a more ‘intelligent’ adaptivity: the effect of any contextual input on the system must empower the user’s choices, rather than simply removing control. This is particularly true in deep cognitive tasks such as learning.

3.6.4 AHA!

In AHA!, the user model is constructed of user preferences indicated by the user directly, user knowledge initialised by stereotype and an overlay model which consists of mostly Boolean variables - true if a concept is known and false if the concept is not known. The concepts may also be typed as either String or Integer. The AHA! ‘User Model’ can, therefore, be considered as a vector of concepts with attributes. The act of accessing content will alter this vector, generally to increase a knowledge attribute. The system also allows a simple text export and import of the user model.

Adaptive Navigation is implemented by links with three possible states - desired, undesired and uninteresting. The standard colours of WWW browsers are used - blue links are desired, purple links are uninteresting which implies the information has been visited and does not represent new information to be learned and dark grey links indicate undesired information for which prerequisites have not been covered. This is similar to dimming the link and is a user friendly alternative to disabling. Links may also be hidden in the text, disabled or removed entirely.

Adaptive Presentation is implemented through conditional inclusion of fragments depending on the user model. Alternative presentation and hiding of text is also used depending on the inferred knowledge of the user. The system will switch to verbose mode for a novice user who has just started to use the system. A concept explanation is included in the current page if it is a prerequisite concept for that page which has not been covered. Similarly if a technical term has not been covered a substitute will be used. For example, the word page is used instead of the term node until the description of a node is encountered by the user and marked as encountered in the user model. Discrete portions of content are represented as concepts. These can be linked to any number of pages, objects or fragments. Each concept within the AHA! system approximates a single constituent unit of the Domain Model [De Bra and Ruiter, 01]. Examples given include a style of painting or a topic to be studied. The concepts and attributes contained by the Domain Model are reflected in the User Model. This is an example of the Overlay Model, i.e. for each domain model concept, an individual learner’s knowledge model stores some value which is an estimation of the knowledge level of this concept.

The adaptation mechanism of the AHA! model is related to the concept overlay. Upon visiting a page, a number of concept-linked attributes within the user model are altered according to embedded values within the concept ruleset. The two most common associated attributes are the visited and knowledge attributes. Each page is composed of fragments and objects which can, themselves, contain fragments which require adaptation. This creates a mechanism for recursively creating complex pages. AHA! pages can require certain attributes before a page is accessible, therefore embedded fragments can be used to enforce requirements in pages.

AHA! exposes two primary mechanisms of adaptive presentation. The first is through link hiding/annotating, the second is via fragment inclusion/selection. In both processes, as discussed above, the system evaluates the desirability of a page, relative to the requirement set of the page and the user's knowledge vector, particularly the visited attribute.

Given the low-level expression of the AHA! system, authoring tools are vital to the creation of useful content. Two principle tools are employed –

- The Concept Editor: Employed to annotate content with an attribute set.
- The Graph Author: Permits the high level creation of concepts, prerequisite sets and knowledge propagation.

In providing integrated rules and content, the AHA! system provides a very low level of control for the user. However, this model does not encourage reusability, nor does it readily accommodate expansion and alteration. A more modular method for content navigation control may be useful. The AHA! system provides significant control over its adaptation mechanisms. It is possible for the author to control the stability of the system at an object level. There are four levels of control: always adapted (default setting), always stable, session stable, expression stable.

It would, perhaps, be useful to empower the user to make a similar choice with regard to the reconfiguration rate of the system. This would give more control to the user rather than allowing potentially confusing dynamic changes. The authors of the AHA! system indicate that an 'important idea' would be to examine expanding beyond 'user behaviour' into the area of context 'such as device and network characteristics'. Integrating context into the AHA! system is assisted by the adaptable nature of the attribute vector. However, the combined rules and content would compound the difficulty of expressing context in a 'pluggable' manner, creating instead new axes of adaptation.

The technology employed is consciously limited to standard Web languages and tools - structured HTML comments and the CLASS attribute to links support conditional fragments. The early development of the AHA! system comprised CGI and Fast-CGI scripts. Subsequent versions have been implemented using Java-Servlets [De Bra and Calvi, 98] [De Bra and Calvi, 97].

The usefulness of the adaptivity in AHA! must be measured in the context of the complexity of creating and maintaining the system. AHA! offers substantial configuration options which allow users to register their preferences (for example the colour of link states) and even change the overlay model directly. This requires a considerable degree of trust in the competence of the user to monitor their progress and direct their own adaptivity. The configuration process also introduces a learning curve in the initial use of the system.

3.6.5 AHA! meets Interbook

The AHA! system is considered as an 'assembly language' [De Bra and Ruiter, 01] [De Bra et al., 02a] for Hypermedia. In the course of development, it was decided that the AHA! presentation system should be generalised to produce new functional possibilities. This includes, for example, the presentation of an Interbook Electronic Textbook as an AHA! course. The main feature of this process was to alter the method of presentation of the content representation scheme, in order to permit type classing of conceptual elements in an arbitrary fashion. In addition, normal integrated entity relationship information was exported to separate XML documents. The product of this research was a more generalised AHA! system, with an example Interbook to AHA! compiler [De Bra et al., 03].

3.6.6 KnowledgeTree

In the strictest sense KnowledgeTree is not an Adaptive Hypermedia System – it is a framework for adaptive eLearning based on distributed reusable learning activities [Brusilovsky, 04]. One of the most common failings of AHSs is their lack of popular adoption. Brusilovsky notes that often it is the authors of such systems that are the main users. Conversely, learning management systems such as Blackboard [Blackboard] and WebCT [WebCT] have gained significant market penetration; yet provide non-adaptive service. However, their success is due to learner and course management facilities features, such as enrolling users and assigning them to courses. KnowledgeTree describes the two main reasons why Adaptive Web-based Educational Systems (AWBES) have not been widely adopted –

- *Lack of Integration* – many AWBES can support features that are educationally superior to those offered by LMS. Typically, each AWBES only offers one such feature and cannot be integrated with other AWBES.
- *Lack of Re-use Support* – Modern AWBES are self-contained systems that cannot be re-used. For example, a teacher wishing to use some content from a system such as Interbook has no opportunity to do so – they must use all or none of it.

KnowledgeTree attempts to address these short-comings by providing an architecture which replaces the monolithic LMS with lightly coupled services. The architecture assumes the presence of at least four kinds of servers: activity servers, value-adding services, learning portals, and student model servers.

Through this architecture KnowledgeTree facilitates the combination of many different services to fulfil the teacher and student's needs. This view moves away from the view of Adaptive Hypermedia being provided by a system towards AH being offered as a service. This is very much in line with the perspective of this thesis. Indeed, it has resulted in a joint publication [Brusilovsky et al., 05] comparing and contrasting our approaches to Adaptive Services.

3.6.7 Analysis of Adaptive Hypermedia Systems

The goal of this analysis is to highlight the similarities and differences between the systems described in the survey. It also aims to point out deficiencies in the approaches taken in developing each system to ensure, where possible, that the product of this thesis does not repeat mistakes made in the past.

3.6.7.1 Genericity, Flexibility and Extensibility

Genericity, flexibility and extensibility are three areas in which AHS have traditionally performed poorly. For example, the adaptive mechanisms used at the core of AHS surveyed tend to have cores that are not generic. In other words they are tailored for the particular domain in which they are being applied. This is particularly the case with PEBA-II and Interbook. AHA!, on the other hand, utilises an Adaptation Model to describe its adaptation processes and this model remains separate to the engine that interprets it. This Adaptation Model, however, tends to be intertwined with the content fragments that are combined towards an adaptive outcome. This severely impacts the potential flexibility of AHA! as the adaptive logic is dispersed amongst the content and is not easily updated or changed.

Another area of flexibility in which the AHS surveyed fall down is in the addition of more models on which the system may adapt. Currently, each of the systems is limited to the notion of a user model being the only focus of adaptivity. This limitation does not easily facilitate adaptation across different properties, multiple axes or multiple contexts.

Of the AHS surveyed AHA! is the only one that has proved extensible into different domains. This is facilitated through its domain model which is again kept separate from the interpretation engine, thus facilitating adaptation in different domains. The adaptation mechanisms of both PEBA-II and Interbook are tied closely to the domains in which they operate.

3.6.7.2 User Modelling

Each of the systems surveyed employ varying degrees of the overlay model. PEBA-II uses the stereotypical classifications of naïve versus expert user as the basis upon which to overlay. By examining the discourse history of the user with the system and examining recently browsed animals PEBA-II overlays facts about those animals into the learner model. Interbook extends this basic paradigm by including learning goals in the learner model. In this sense the learner model does not just contain state information, but also gives some indication as to where the learner is going. Interbook uses page visits and quiz results as evidence to build the learner model. AHA! again employs an overlay model where concept-linked attributes are used to represent the learner's current knowledge state. Page visits containing different fragments fire rules that dynamically change these attributes as the learner browses content presented by AHA!.

One area in which each of these systems falls down is in providing the learner with mechanisms to examine and change their model. The lack of scrutability [Czarkowski and Kay, 03] can lead learners to feel like control of their learning has been taken away from them. This is especially apparent if the user modelling system employed by the AHS has made an incorrect decision.

3.6.7.3 Adaptive Techniques

While each of the AH systems surveyed utilise a similar user modelling paradigm the adaptive mechanism that operates on that model to produce a personalised effect differs quite a lot between the systems. PEBA-II uses natural language generation techniques to produce coherent text for the learner. This is achieved by dynamically combining fragments into hypertext nodes. The natural language techniques ensure that the resultant page sounds natural to a human reader. On top of this, PEBA-II compares new concepts in a node with those already known (according to the learner model) and draws comparisons between them, thus helping the learner to combine the new knowledge with what they already know.

Interbook utilises the additional learning goal information in the (overlaid) user model to select content based on a sequence of these goals. Using direct guidance the system rates and arranges nodes within the glossary. The rating of the nodes is based on information in the learner model and categorises the node as either known, ready to be learned or not ready. Interbook also employs prerequisite-based help to list and rank the prerequisite topics for a certain learner. Unlike PEBA-II, Interbook does not attempt to assemble Hypermedia pages for presentation and is primarily focused on structuring navigation.

AHA! employs both adaptive navigation and presentation techniques. Similar to Interbook it uses link annotation to indicate the suitability of a linked page. This is associated with the concepts in that page and whether the learner model indicates that the learner has met the prerequisite concepts for that page. The prerequisite structure is contained in the domain model. Adaptive presentation is achieved through conditional fragments being added or removed from the page. Again, the inclusion or exclusion of fragments is based on comparing the learner model with the domain model, which may be used to support techniques such as stretch text. The disadvantage of this approach to adaptive presentation is that pages a learner has already visited may change without their knowing. For example, if the learner has gained knowledge of a concept that was a prerequisite to the inclusion of a fragment on a page they have already visited then they may miss that fragment if they do not revisit that page.

Each of the Adaptive Hypermedia Systems surveyed employ either Adaptive Navigation or Adaptive Presentation techniques. Fragment exclusion/inclusion is a common approach between PEBA-II and AHA!. It is not clear, however, whether the adaptive mechanism deals directly with the content fragments or if there is conceptual mapping between the mechanism and the actual content. If it is the former case then the adaptive mechanism and the content fragments are entangled – combining adaptive logic and content. This is something that should be avoided in order to maximise reusability. The natural language processing carried out by PEBA-II on the assembled fragments is one means of ensuring human readability between fragments, but does not necessarily guarantee semantic coherence between them. For example, language may flow cleanly between two fragments, but if their subject matter is quite different, or aimed at learners with different experiences, then they may be readable, but not comprehensible.

Adaptive Navigation is employed by both Interbook and AHA!. Both systems use link annotation to denote the suitability of certain learning content. This approach seems to have user acceptance as it does not hide content from them; it simply advises which hyperlink is most suitable to follow. This form of user guidance is to be encouraged as it does not dictate what the learner can access. This approach, however, may still lead to certain degrees of cognitive overload if, in the initial stages of a course for example, the learner is presented with more links that are ill-advised than are advised. Learner control and freedom is important, but cognitive overload must be avoided/minimised where possible.

3.6.7.4 Reusability of Components

One of the downfalls of many Adaptive Hypermedia Systems, and the ones surveyed are no different, is that they do not explicitly consider the reusability of the components of adaptivity. For example, references to content fragments are often explicitly encoded in the adaptive mechanisms, making the mechanism itself specific to a content set.

Similarly, the user model is often proprietary to the system. This means that the information encoded in it, given the effort in attaining that information, is not reused. AHA! has a domain model which details the concepts, and their relationships, that may be used to infer information about the user model.

The content fragments used by each system have limited reusability without appropriate metadata. Currently, to include a fragment of content in one of the adaptive systems discussed, the author must have knowledge of that piece of content. Without appropriate metadata they cannot be discovered.

The Interbook meets AHA! experiment [De Bra et al., 03] shows that AHA! has a flexible layout model, but also demonstrates the significant effort in creating a compiler that will transform an Interbook course into AHA!. KnowledgeTree, on the other hand, does not try to transform adaptive courses – instead it wishes to combine them as services. Again flexibility in the layout of the individual services may be required to ensure a consistent look and feel to the environment the learner is learning through.

3.6.8 Related Work

Adaptive Hypermedia is an evolving field of research and many of the ideas and the systems surveyed above are being considered and extended in various research groups around the world. Derived from the AHA! system, a more conceptual model of its architecture, named AHAM [De Bra et al., 02b], is being developed based on the Dexter Reference Model [Halaz and Schwartz, 90]. Also associated with the development of AHAM are authoring tools for creating models for Adaptive Hypermedia. The authoring suite produced to support AHA!, namely the Form Editor, Concept Editor, Graph Authoring Tool and Test Editor, facilitates the creation of Adaptive Hypermedia offerings. My Online Teacher (MOT) [Cristea and de Mooij, 03] is a sub-set of these tools with a specific focus on creating adaptive educational offerings on top of AHA! Research into the authoring of models for APeLS is currently being conducted by [Dagger et al., 2004]. This research focuses on the rapid development and application of the multi-model approach to Adaptive Hypermedia. A central aspect of this research is the empowerment of educators to create pedagogically sound personalised eLearning offerings by developing personalised eLearning models in a graphically easy to use and non-technical way. This composition suite called the Adaptive Courseware Construction Toolkit (ACCT) significantly differs from other such adaptive authoring toolkits by actively supporting pedagogic models and influences within the course design and authoring process.

A second related area of research is the extension and application of ontologies to support concept models, learner models and some forms of pedagogical model. Typically, such ontology research focuses on the Semantic Web-based [Semantic Web] ontologies and languages. Specifically, the application of concept-based approaches [Houben et al., 05], the engineering of semantic web information systems [Vdovjak et al., 03] and the semantic meta-modelling of AHS [Seefelder and Schwabe, 04] are of interest. Ongoing research will explore the possibilities of utilising richer semantic models in the adaptation process. This research may offer many possibilities for the extension of adaptive systems as the adaptive logic in the engine, or encapsulated within narrative, may become more generic displacing domain specific knowledge to the semantics of the ontology model.

In the domain of characterising AHS and the use of Semantic Web in AHS there is significant research being carried out by the Learning Lab of Lower Saxony (L3S). Of particular significance is their work on the characterisation of adaptive educational hypermedia [Henze and Nejdl, 04], including ontologies in adaptive hypermedia

[Henze et al., 04] and relating Semantic Web to AH [Cheniti-Belcadhi et al. 04]. Also of note is their work in the area of model driven navigation design [Dolog, 04].

3.7 Metadata Standards and Specifications

Metadata Standards and Specifications provide mechanisms to describe elements of adaptivity such as Learning Resources, Learners, Enterprise Architectures and Content Packages. There are several specification and standardisation bodies developing metadata standards to describe these elements. Appropriate standards may –

- Make digital content more independent from services and hardware used to deliver it
- Provide more uniform and precise access to networked learning resources and services
- Help extend the life time of capital purchases and organisational changes
- Facilitate the integration and maintenance of system components and data resources

This section describes and analyses current and emerging standards in the areas of Learning Resources, Learner Models and Packaging and Managing Content.

3.7.1 Representing Learning Resources

Learning resources comprise a broad range of learning objects that may be used to assemble educational offerings to learners. They range in granularity from fine-grained assets used to compose individual Learning Objects to a complete eLearning course, comprising of hundreds of Learning Objects (LOs). With many learning institutions and publishing houses possessing large numbers of LOs it has become necessary to seek mechanisms to manage these resources. The most fundamental element of management is to associate metadata with learning resources. Such metadata facilitates their discovery and retrieval as well as information about how they may be used. This section describes candidate standards for describing LOs.

3.7.1.1 ADL SCORM

Since its launch in January 2000 SCORM has seen four releases and a change in name. Originally released as the Sharable Courseware Object Reference Model in Version 1.0

it was subsequently renamed to the Shareable Content Object Reference Model in Version 1.2. This was done to better show that the SCORM applies to various levels of content. Aligning with this shift, the Course Structure Format, one of the fundamental aspects of SCORM, was also changed to the Content Structure Format (CSF) to show that collections of learning content smaller and larger than an entire course could be represented through the SCORM. The CSF is a content structure format that defines all of the content elements, the content structure, and all external references necessary to represent content and its intended behaviour. This CSF is intended to promote reuse of entire courses and encourage the reuse of course components by exposing all the details of each course element. The CSF is intended to reduce or eliminate dependency of content on a particular LMS implementation.

With the release of SCORM Version 1.2 the addition of content packaging application profiles became available. Derived from the IMS Content Packaging specification, these profiles map the Content Structure Format (CSF) from the SCORM Version 1.1 into the general IMS specifications. This version of the SCORM also updated the metadata used to describe learning content.

SCORM 2004, released in January 2004, introduced Sequencing and Navigation as an aspect of the SCORM suite of specifications. This new version updated the other specifications to track evolution of standards and harmonised them with Sequencing and Navigation. With this version, the SCORM document is considered stable. SCORM is a collection of specifications and standards that can be viewed as separate "books" gathered together into a growing library. Nearly all of the specifications and guidelines are taken from other organisations (such as, AICC, IEEE and IMS). These technical "books" are presently grouped under three main topics: Content Aggregation Model (CAM), Run-Time Environment (RTE) and Sequencing and Navigation (SN).

SN has the most relevance to this thesis. There are several key concepts that are introduced in the SCORM SN. It covers the essential LMS responsibilities for sequencing content objects (Shareable Content Objects (SCOs) or Assets) during runtime and allowing SCOs to indicate navigation requests. In addition, guidance is offered for providing navigation controls to learners. General subjects discussed include –

- Sequencing Concepts and Terminology (e.g., Learning Activities, Activity Trees, Clusters)
- Sequencing Definition Model (i.e., detailed descriptions and requirements of the sequencing information that can be applied to learning activities)
- Sequencing Behavior Model (i.e., detailed descriptions of LMS behaviours to prescribed sequencing information and learner's experience with learning content)
- Navigation Controls and Requirements
- Navigation Data Model

Communication between content and LMSs facilitates use of SCORM Sequencing and Navigation to present content to learners based on learner choices and performance at runtime. This communication also enables LMSs to track learner progress and performance while content is presented to the learner.

The fundamental specification in SCORM SN is the Simple Sequencing (SS) Specification [IMS SS], which defines a method for representing the intended behaviour of an authored learning experience such that any LMS will sequence discrete learning activities in a consistent way. IMS SS is labelled as simple because it defines a limited number of widely used sequencing behaviours, not because the specification itself is simple. IMS SS is not all-inclusive. In particular, IMS SS does not address, but does not necessarily preclude, artificial intelligence-based sequencing, schedule-based sequencing, sequencing requiring data from closed external systems and services (e.g., sequencing of embedded simulations), collaborative learning, customized learning, or synchronization between multiple parallel learning activities [SCORM].

3.7.1.2 IMS Learning Resource Metadata and IEEE LOM

IMS Learning Resource Metadata Information Model Version 1.3 [IMS Metadata] is a standard derived from Version 3.5 of Learning Object Metadata Schema working document of the IEEE Learning Technology Standards Committee's [LTSC] Learning Object Metadata (LOM) Working Group. IEEE LOM aims to provide an extensive metadata description for Learning Objects. This standard specifies the syntax and semantics of Learning Object metadata, defined as the attributes required to fully/adequately describe a Learning Object. Learning Objects are defined as any entity, digital or non-digital, which can be used, re-used or referenced during technology-supported learning. IEEE LOM is divided into 9 categories – General, LifeCycle, MetaMetaData, Technical, Educational, Rights, Relation, Annotation and Classification.

IMS Learning Resource Metadata Information Model is an implementation of IEEE LOM containing a number of modifications approved unanimously by the IMS Technical Board. IMS have produced an XML binding of IMS Metadata Version 1.3. The EASEL [EASEL] IST project has developed an extension to IMS Metadata (see Section *4.4 Metadata for Describing Adaptivity*) to meet the metadata requirements of Adaptive Hypermedia Services.

3.7.1.3 Dublin Core

The Dublin Core [DC] Metadata Initiative (DCMI) was established to specify a common core of semantics for resource description to aid the emerging infrastructure of the Internet. The Dublin Core has attracted broad ranging international and interdisciplinary support and is being adopted by many disparate (internet-related) communities. It has a particular purpose for aiding resource discovery and facilitating interoperability.

There are 15 basic Dublin Core metadata elements namely, – Creator, Subject, Description, Publisher, Date, ResourceType, ResourceIdentifier, Language, OtherContributor, Format, Source, Relation, Coverage and RightsManagement. These 15 elements for describing resources have been augmented by the publishing of Dublin Core Qualifiers. These qualifiers attempt to further specify existing Dublin Core elements, thereby increasing precision of the encoded metadata. Two broad groups of qualifiers exist: Element Refinement and Encoding Scheme. The set published is an open set, which is expected to evolve over time.

The DCMI also creates working groups to tackle work areas as the need arises. One such working group, which would seem to have particular relevance to learning objects, has been the Education Working Group (DCEd) [DCEd]. DCEd has proposed the adoption of the following –

- Two new domain-specific elements (dc-ed:audience and dc-ed:standard) with accompanying element qualifiers for a dc-ed namespace.
- A new domain-specific qualifier to dc:relation for the dc-ed namespace (dc-ed:conformsto).
- The endorsement of three elements (lom:interactivitytype, lom:interactivitylevel and lom:typicallearningtime) from the IMS namespace.

One of the goals of DC is interoperability. To aid interoperability many non-DC metadata data models provide mappings of their elements to DC - e.g. IMS Learning Resource Metadata.

3.7.2 Representing the Learner

Currently, there are two candidate specifications for representing the learner – IMS Learner Information Packaging and the IEEE Public and Private Information specification.

3.7.2.1 IMS Learner Information Packaging

The IMS Learner Information Packaging [IMS LIP] specification addresses interoperability between internet-based learner information systems. Learner information concerns Learners (individual or group) or Producers (creators, providers or vendors). LIP includes facilities for the Learner to determine which aspects of their information are sharable with other systems. LIP has been designed with four basic requirements in mind –

- Distributed Information
- Scalability
- Privacy and Data Protection
- Flexibility and External References

The last requirement is described in LIP as Learner information and includes many constructs, such as learning objectives and learning history, which are in practice represented by different structures in different contexts. Learner information data models must be flexible enough to accommodate this need. An external reference may, in the future, be used by both Learning Environments and Adaptive Hypermedia Services to share learning objectives and learning style information. IMS LIP v1.0 is currently available as a public draft.

3.7.2.2 PAPI

PAPI [PAPI] is the IEEE Public and Private Information Specification which is a standard format for the representation and communication of student profiles. The purpose of the specification is to allow the creation of student records which can be communicated between educational systems over the lifetime of a learner.

The profile information for a learner is divided into four areas -

- Personal information which is for private consumption such as the learner's name, address and Social Security Number
- Preference information which may be for public consumption, such as the technology available to the learner, the learning style of the learner, physical limitations or disabilities. This information is collected with the cooperation of the learner, i.e. it is negotiated.
- Performance information which is for consumption by technology. This consists of the observable behaviour of the learner and may include grades, reports and logs
- Portfolio information, detailing the learner’s previous academic and design achievements.

The PAPI specification also incorporates the Dublin Core metadata element set. The information used to construct the learner profile is inferred by the system, directly input by the learner or is constructed by the learner and system in collaboration. PAPI also intends to address the privacy and security issues involved in the storage and communication of learner profiles.

3.7.2.3 GESTALT User Profile

GESTALT (Getting Educational Systems Talking across Leading Edge Technologies) [GESTALT, 99] which is funded by the ACTS project, is an educational environment for online learning which extends the IMS Learning Resource Metadata definitions. A user profile is constructed from information acquired from the user by asking the user to complete forms displayed by a wizard. The user model is created as an XML document which is then stored physically on the user's machine. The profile information stored includes the user's educational history and the technology available to them.

Some of the profile details acquired include personal details, contact details, qualification details, skill details, learning preferences and mode of delivery. The learning preference is stored as a boolean value of 'Yes' or 'No'.

3.7.3 Packaging and Managing Learning Resources

The management and distribution of learning objects is a key stage in facilitating their reuse. Described in this section is IMS Content Packaging and the evolution of the content interworking specification to facilitate the integration of learning objects into Learning Management Systems.

3.7.3.1 IMS Content Packaging

IMS Content Packaging is an interoperability specification to allow content creation tools, learning management systems and runtime environments to share content in a standardised set of structures. Version 1.1 of the specification is focused on defining interoperability between systems that wish to import, export, aggregate, and disaggregate packages of content [IMS Content Packaging].

The primary goal of the IMS Content Packaging specification is to provide a mechanism which, once implemented by producers and vendors, will allow content to be exported between systems with the minimum of effort. Version 1.1 of this specification is currently available as a public draft.

3.7.3.2 Content Interworking

The Content Interworking API was initially developed as part of the AICC CMI, more recently developed as part of ADL SCORM and the University for Industry Content Interworking Specification.

It is proposed that the IMS Content Management group will adopt the Content Interworking API as specified in the AICC CMI specification v3.0.1 and currently under implementation by the ADL [ADL] and by the University for Industry [UFI].

Aviation Industry CBT Committee Computer Managed Instruction Guidelines

The AICC guidelines [AICC] provide a method for seamless data flow between different computer based training (CBT) lessons and Computer Managed Instruction (CMI) systems. It also provides data flow between different CMI systems and from CBT lessons created with different authoring systems to a common data store and off-the-shelf analysis tools.

The driving force behind the development of the AICC CMI was to allow content developed by different authors to be used with any CMI system that supports the guidelines. To this end an API was designed that allowed the content to connect to the CMI system. The API principally facilitates the getting and setting of data in the CMI systems data model.

There are two aspects of the AICC approach to enabling interoperability of CMI systems with different CBT systems –

1. Lesson launch: The CMI should have a standard approach to CBT lesson initiation, and
2. Communication: The CMI should have a standard approach to providing information to the CBT lessons, and receiving information from the CBT lessons.

ADL SCORM

Advanced Distributed Learning SCORM [SCORM] (Sharable Content Object Reference Model) is based on the AICC CMI guidelines. ADL was established with the purpose of developing a (US) Department of Defense wide strategy for using learning and information technologies to modernise education and training. Versions 1.0 to 2004 of SCORM define a reference model to facilitate the interworking of Learning Management Systems (LMS) and content providers material. (LMS is used in the SCORM documentation in place of CMI). SCORM is based directly on the runtime environment.

The ADL collaborated with AICC members and participants to develop a common Launch and API specification.

University for Industry Content Interworking Specification

A definition for content interworking between UFI [UFI] endorsed learning materials and the Learning Support Environment (LSE) system is necessary for –

- Consistent launching and running of learning content within the LSE
- Consistent gathering and reporting of assessment data back to the LSE database for subsequent inclusion in the Lifelong Learning (LLL)
- Consistent storage of learner inserted bookmarks and annotations added to the content and subsequent display of bookmarks and annotations to the learner when requested
- Monitoring of the consumption and usage of the learning content

Data is transferred between the content and the LSE via a JavaScript API based on the API for Web implementation of AICC/IEEE CMI standards.

3.7.4 Analysis of Metadata Standards and Specifications

Metadata schemas for representing content and learners are becoming increasingly important in Learning Management Systems (LMS) such as Blackboard [Blackboard] and WebCT [WebCT] for facilitating exchange of courses and learner profiles. For Adaptive Hypermedia Systems, operating in the eLearning space, to gain acceptance, they too will need to be standards compliant. As the specifications are being developed primarily with LMSs in mind there will be occasions when a specification is not sufficient for use in an adaptive course. In these instances the specification should be extended (using whatever extension mechanisms it provides) and the results disseminated to the appropriate specification body. By supporting at least the baseline of a standard or specification AHS can begin consuming the large amount of content and/or learner models produced conforming to these standards. This can immediately open the scope of adaptive courses that one may produce with an AHS, as often the limiting factor is available content.

Through content interworking and initiatives like ADL SCORM adaptive eLearning services have a mechanism through which they may be integrated with current LMSs, thus producing a best of breeds solution. LMS provide collaboration tools, learner enrolment facilities and course management tools, while the adaptive eLearning service provides high quality personalised eLearning.

3.8 Summary

In this chapter the current state of the art of Adaptive Hypermedia was reviewed and analysed. It covered the objectives and methods of adaptivity and how they may be applied to Hypermedia. It also discussed early computerised education systems and surveyed current Adaptive Hypermedia Systems.

Also discussed and analysed were the current and emerging standards and specifications for representing digital learning resources and learners and for the packaging and managing of learning content.

4 DESIGN

4.1 Introduction

This chapter discusses the issues impacting on the design of an adaptive system towards fulfilling the objective and goals of this thesis. It describes how the state of the art, discussed in chapters 2 and 3, has impacted on the design of the Multi-model, Metadata driven approach to adaptive system development, including the issues concerning the design of the core models. In particular it examines how adult learning theories, adaptive hypermedia techniques and metadata standards have influenced the design of this approach.

The chapter continues by introducing the Multi-model, Metadata Driven approach for producing adaptive systems that is proposed in this thesis. This section describes the distinguishing aspects of this approach including the separation of models, the representation of models using metadata, the abstraction mechanisms employed within the engine to facilitate this separation, the principles of candidacy and candidate selection, and narrative execution.

Following the introduction of the Multi-model, Metadata Driven approach the chapter describes the design of the enabling elements for this approach. These enabling elements include the metadata utilised for describing adaptivity and the core models of the approach. The three core models of Content, Learner and Narrative are examined. In the section on Content Model Design Issues the content model and issues such as lifespan and granularity are discussed. The Designing the Learner Model and Modelling the Learner section examines how different aspects of the learner, which are pertinent to the adaptive process, are modelled.

The Describing the Narrative section discusses the most significant model in the multi-model approach. This section describes how the narrative may be viewed as not only an encapsulation of domain expertise, but also as a flexible mechanism for producing personalised courses at runtime through reconciliation of the other models.

The chapter concludes with a discussion of the technologies that may be used to support the multi-model, metadata driven approach and the design of a service architecture and an approach to offering an adaptive system, developed according to this approach, as a service.

4.2 Influences from the State of the Art

The state of the art discussed in chapters 2 and 3 impacts various aspects of this thesis. In particular these influences should be viewed from the perspective of how they affect the desired properties of the system being designed. The main properties of the system are that it –

- Delivers a satisfactory and pedagogically sound eLearning experience to the learner
- Provides different adaptive effects based on different sets of models
- Separates the different concerns and elements of learning
- Is flexible and extensible in the types of adaptivity offered

The various influences and how they impact the design are discussed under the headings of Adult Learning Theory, Methods of Adaptivity and Adaptive Hypermedia Systems, and Metadata Standards and Specifications.

4.2.1 Adult Learning Theory

In the Analysis of learning theory constructivism features heavily in practiced education, both traditional and eLearning. Constructivism gives responsibility over the learning process to the learner. Many early Computer Aided Instruction and Intelligent Tutoring Systems failed because the learner felt disenfranchised from the decisions the systems made about them [Wade and Power, 98b]. Constructivism may be supported by giving the learner appropriate controls over the learning provided by a system. To this end the design of the system should cater for learner control of their model, based on meta-cognition. Through self reflection and understanding of the adaptive mechanisms employed in a system the learner can change its behaviour to suit them. By meeting these criteria the eLearning experiences delivered to the learner should be more satisfactory and pedagogically sound.

Many authors have suggested that catering towards a learner's learning style or preference can yield benefits [Fleming and Mills, 92] [Parson, 98]. However, there is still much debate as to how best learning styles should be modelled and catered for. There is also some discussion as to whether one should cater only for the preferred learning style of a learner or expose them to many styles. To this end learning style does not directly impact the design of the adaptive system or its internal reconciliation engine, but rather impacts the mechanisms used to embody pedagogical concerns, e.g. the narrative model. A suitably flexible mechanism should be chosen that can facilitate the description and implementation of several different learning styles and preferences. This has significance for the desired system property of providing different adaptive effects based on different sets of models.

Related to the influence of constructivism, the mechanism chosen for modelling learners should empower them and not hide decisions over which they can exert no influence. Implicit modelling of a learner is suitable providing they are given an opportunity to change modelled information where/when appropriate. Again this corresponds to constructivist approaches.

4.2.2 Methods of Adaptivity and Adaptive Hypermedia Systems

Presentation and Navigation Adaptivity should be kept separate within the system, where possible. This separation can help clarify the responsibility of different elements of the system, such as the core models of narrative, content and learner. It also helps to ensure the reusability of different elements of the system as they do not have any explicit interdependencies. For example, if a course author wishes to reuse only the adaptive navigation paradigm for a course they may choose to only reuse the narrative models that implement it and not the other models. This may be the case in porting a course developed for desktop PCs to portable devices. By fulfilling this separation of navigation and presentation mechanisms, the desired system property of separating the different concerns and elements of learning is at least partially fulfilled.

Through observing the implementation of Adaptive Navigation in the Adaptive Hypermedia Systems surveyed, the paradigm of choice is link annotation, where the suitability of each node for the learner is indicated. This is suitable when the hyperspace is small and the learner can easily browse a list of links. When the hyperspace is larger, this browsing itself may cause cognitive overload. Link hiding, therefore, may be more suitable for larger hyperspaces. In this case the learner should have sufficient controls to increase/decrease the linked hyperspace as they wish.

The overlay approach to user modelling is the most common approach adopted by the Adaptive Hypermedia Systems surveyed. This is a suitable candidate for this thesis. As stated in the *Adult Learning Theory* section above scrutability of the learner model and learner control over that model, even if indirect, is important. The learner must be able to control the adaptive system (at least to some acceptable degree) and understand the effects the usage of this control will have on the resultant courses.

The Adaptive Hypermedia Systems surveyed, e.g. AHA! and Interbook, do not maintain separation between content and the adaptive logic that governs adaptivity. This separation is imperative as it facilitates reuse of not only the content, but also the adaptive mechanism itself. Such separation eases the replacement or substitution of an adaptive mechanism, potentially supporting more and varied pedagogical approaches. To this end the adaptive mechanisms should never refer directly to content, but should work at a conceptual level.

Towards reusability of the adaptive system as a whole the model described by the Interbook meets AHA! work is not feasible in the long run as it requires a customised compiler to enable Interbook texts to work within the AHA! system. The KnowledgeTree approach, where services are utilised, is more feasible though may not provide the same level of integration. Content interworking specifications should be examined to see if they may be used for reuse of adaptive eLearning services within other environments.

4.2.3 Metadata Standards and Specifications

There are three aspects of metadata standards and specifications that impact this work. The first is modelling of the learner information. This has significance for how the system interprets information about the learner and for how scrutable [Czarkowski and Kay, 03] the learner model is by the learner. Scrutability is important as it enables the learner to perform meta-cognitive reflection on what they have learned and how they have learned it. There is no clear candidate specification for representing the learner as neither IMS LIP nor PAPI cater for pedagogical information. Most of the schema in both specifications would also be unused in an adaptive eLearning system as information, such as the learner's address, is generally unnecessary for adaptive eLearning. From a scrutability perspective it makes sense to store only information that is of value to the adaptation process as then the learner has a model that is more accessible and focused on their current task.

The second metadata influenced decision is how metadata may be used to model content. Encouraging reuse of the content and providing appropriate representations of the adaptive uses of the content are the main reasons for looking at metadata to model content. Current specifications cater for discoverability and reuse of content quite well. The description of the adaptive uses of the content, in a similar manner to the description of pedagogical information in learner profiles, is not currently represented. Describing how a piece of content may be used in an adaptive setting may be important in differentiating it from other candidates. This deficiency may need to be addressed by extending the appropriate specification.

The third contribution from the state of the art in standards and specifications is the area of content interworking. Although the Content Interworking specifications are designed to enable learning objects to interact with Virtual Learning Environments (VLE) there is potential to enable Adaptive eLearning Systems to communicate with VLEs in a similar manner. The candidate specifications should be investigated for this potential.

4.3 The Multi-model, Metadata Driven Approach

The objective of this thesis is to propose and evaluate an architectural approach for implementing dynamic adaptive courseware with sufficient flexibility to support multiple axes of adaptivity and multiple instructional models. One of the technical goals of this thesis is to ensure that the proposed approach for producing Adaptive Hypermedia Systems facilitates the reuse of the key components of adaptivity, such as the content, narratives and learner models. This goal, however, is of secondary importance compared to the goal of providing the learner with guidance in their learning.

Section *3.6 Survey of Adaptive Hypermedia Systems* describes many diverse approaches to implementing AHSs, but these approaches have one common feature that limits the reusability of their content and instructional design. The interweaving of content and the adaptive logic that governs its adaptivity restricts not only the reusability of the content, but also the ability to repurpose or modify the courses. Such embedded logic restricts the course authors when adding new content to a course as they must have a complete knowledge of the possible outputs of the rule system and how this new content (and embedded logic) will impact on the course execution. This requirement for complete knowledge of the system and its execution logic also limits the possibilities for collaboration between course authors (knowledge domain experts), content designers and instructional designers (pedagogical experts), as if any modifications are to be made to the system they must each have a complete overview of how the content and sequencing logic interacts. Through separating logic and content, multiple strategies may be employed across a common body of content. The ability to employ different strategies leads towards learner empowerment, as course authors may create different pedagogically influenced adaptive strategies to cater for the needs of different groups of learners.

Considering the limitations on traditional approaches to designing AHSs the multi-model approach has been designed to deliver satisfactory and pedagogically sound eLearning experiences that –

- Provide different adaptive effects based on different sets of models
- Separate the different concerns and elements of learning
- Are flexible and extensible in the types of adaptivity offered

Towards meeting these properties the following technical requirements should be addressed in the design of the system –

- Separate the learning content from the sequencing logic into separate models
- Enable the inclusion of additional models to support the adaptive process
- Utilise metadata to aid reuse of models

In meeting these requirements the collaboration of course author, content designer and instructional designer at the system level should also be facilitated. This section describes how the different aspects of the multi-model approach have been designed to meet both the properties and requirements described above. It also describes how the aspects of this approach correspond to the objective and goals of this thesis.

4.3.1 Model Separation and Model Types

The content and sequencing logic (or more generally adaptive logic) are separated into two discrete models in the multi-model approach. The content is represented by the content model and the adaptive logic is represented by the narrative model. The content model does not include any sequencing or adaptive logic. It contains a metadata description of a piece of learning content, including a reference to the actual learning content. The narrative model contains a rule set and the metadata representing the features of that rules set. The narrative does not refer to pieces of content or content models directly; instead it uses an abstraction mechanism.

The learner model is also represented separately, but, unlike the content and narrative models, does not represent an internal feature (such as content or rules) of the system – it represents an external factor. Using this perspective there are four kinds of models utilised in the multi-model approach, those that –

1. Are an **input** to the adaptation process, such as the learner model
2. Are formed into a **composite** by the adaptation process, such as the content model
3. Are part of the **production** of the adaptation process, such as the narrative model
4. Are an **output** of the adaptation process (usually assembled from composite models)

The multi-model approach is not limited to the three models of learner, content and narrative, however, as other models may be added. These models tend to be input or composite models, but additional production models may also be used. For example, it may be desirable to represent the device the learner is using or the environment they are learning in as separate models. When the narrative is being interpreted the input from the device model and environment model may be used to select appropriate content for that device and environment. For example, if the device display is small you may only wish content that will fit it to be delivered or if the environment is noisy auditory content may not be appropriate. Both of these example models are input models and may be comprised of metadata only.

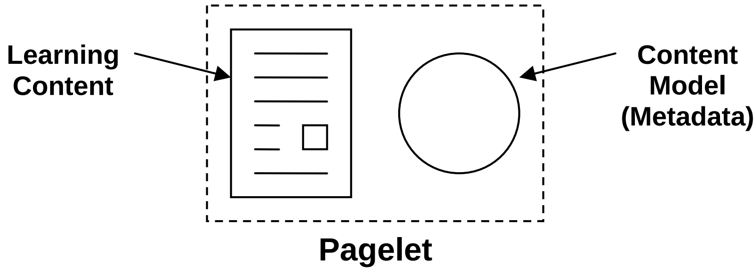


Figure 4.1, Conceptual view of a Pagelet

In the multi-model approach a pagelet is the term used to describe a piece of learning content and its associated content model. A narrative is a rule set and its associated narrative model.

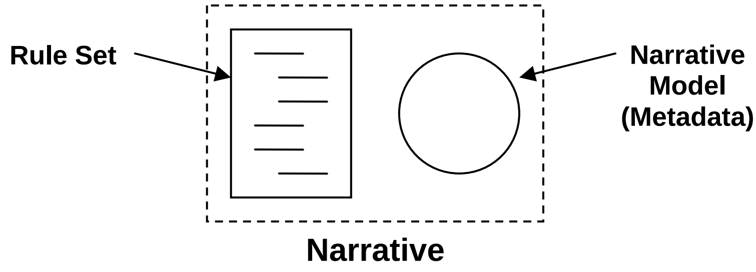


Figure 4.2, Conceptual view of a Narrative

As the learner is external to the system the learner model is comprised of the metadata describing that learner.

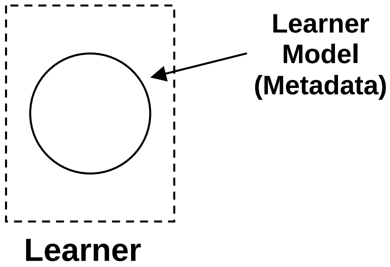


Figure 4.3, Conceptual view of a Learner

From the perspective of input, composite, production and output models an adaptive system produced using the multi-model, metadata driven approach should use at least one of each type of model. The most typical instances used are –

1. Input Model = Learner Model
2. Composite Model = Content Model
3. Production Model = Narrative Model
4. Output Model = Course Model (created at runtime)

4.3.2 Metadata for Model Representation

Appropriate and effective metadata is of key importance in the multi-model approach. The metadata representing the models is used as the basis for selecting content and narratives, or any composite or production model, and for supplying input to the adaptive process, as is the case with input models such as the learner model. The metadata used should be –

- Flexible, enabling the metadata author to describe a broad range of elements that impact upon the adaptive process for a model
- Compatible with current metadata standards, e.g. IEEE Learning Object Metadata, IMS Learning Resource Metadata, IMS Learner Information Package, ADL Shareable Content Object Reference Model.
- Extensible, allowing the metadata author to add more elements
- Capable of describing the adaptive system as a whole

To this end an extension to current metadata schemas was designed that would allow the metadata author to achieve these goals. As the schema extension needs to be compatible with current metadata standards it requires an information model and an XML representation. If it should also be capable of describing an adaptive system as a whole, then its syntax should openly support the description of any type of adaptivity. The information model for the metadata schema for describing adaptivity is included in *APPENDIX I – Metadata*.

4.3.3 Abstraction through Candidacy and Groups

To facilitate flexibility in the design and implementation of new course offerings the multi-model approach was designed to include an abstraction mechanism. It was envisaged that this mechanism would enable the collaboration of many people in the development of an adaptive course. For example, the abstraction mechanism enables the course author (knowledge domain expert) to develop a narrative describing the course sequencing, not in terms of the pagelets to be added, but rather using the learning concepts to be added to realisations of the course. This abstraction allows for greater independence in how the course author and instructional designer design the course. They can perform this design without necessarily being concerned with the individual pieces of content that will be used to populate the final course. Similarly, the content designer may develop instances of learning content without knowing how or where that content will be used. As independence is maintained between the content and the learning concepts, individual pieces of content may be replaced or substituted without a need to modify the narrative.

For this mechanism to be realised a layer of abstraction needs to be maintained between the narrative and content. This abstraction layer is facilitated through candidate groups. Candidate groups are used to group together *like* models. For example, a candidate content group concerned with a particular learning concept may contain several pagelets, each covering the learning concept in different ways. In the case of pagelets these differences may be pedagogical or technical – some pagelets may deal with the concept from different perspectives or render the material differently for different devices. For example, a candidate content group may consist of three pieces of content covering the Java object model – one LO is a textual description of the object model, another LO consists of a diagram of the object model and the third comprises a Flash animation allowing the learner to highlight different aspects of the model. The decision as to which LO to deliver can be made at runtime based on some information about the learner and environment. The requirement for different candidates can be determined by an educational instructional designer, although the task of generating the content for the candidates is generally a collaborative process between the domain expert, content designer and the instructional designer. Candidate Content Groups also enable the adaptive engine to deliver content that caters for different technical issues. For example, there may be alternative candidate LOs that require less bandwidth for delivery, allowing the learner to access the course over a low bandwidth connection (such as a 56k modem connection) as well as in a high bandwidth environment. Candidate groups are primarily a design-time entity, but may be updated after the initial design without impacting other aspects of a system, thus enabling incremental additions to an existing system.

Each candidate group has associated metadata that describes the role of the group and has the identifiers of the constituent models within the group. A simple candidate content group may be visualised as –

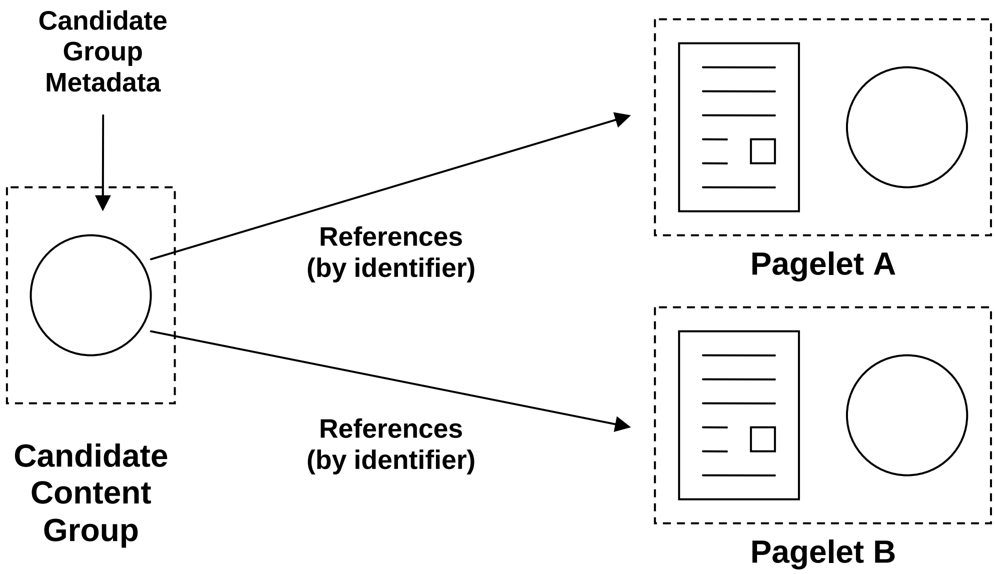


Figure 4.4, Candidate Content Group

In the figure above, the candidate content group has two candidates, pagelet A and pagelet B. Pagelets in the same candidate content group are equivalent on some axis, usually the learning concept they cover. The candidates are differentiated by their metadata and an individual candidate may be selected by reasoning over this metadata at runtime. Candidate groups can be formed across any set of models in the multi-model system. For example, there can be candidate narrative groups containing narratives that produce equivalent courses according to different instructional designs or pedagogical approaches.

As candidate groups refer to their candidates by identifier it is possible for any single model in the system to be included in multiple groups. Groups do not have to be homogeneous and may contain models of different types. For example, a candidate group may have some narrative and some pagelet candidates. In such cases the selection may choose either an adaptive sub-narrative or a non-adaptive pagelet. This enables course authors to enhance the adaptivity of a course by adding adaptive sub-narratives to candidate content groups that previously only contained pagelets.

4.3.4 Sub-narratives and Candidate Selection

Learner satisfaction and reusability are the main goals of the multi-model approach. This does not just refer to the reuse of learning content (pagelets), but also to the reuse of any element of adaptivity in the system. The abstraction mechanism, described above, enables narratives to be reused within the system. For example, if a narrative is developed to teach the fundamentals of mechanics it may be reused as part of larger courses covering applied mathematics or physics. To cater for this, narratives, within the multi-model approach, are designed to be capable of referencing other narratives. When the system reaches such a sub-narrative it should execute the rule set of that narrative as if it were a seamless part of the parent. When combined with candidate groups, learner satisfaction may be improved by providing the best personalisation mechanisms for each learner.

Closely related to narratives, or more precisely sub-narratives, are candidate selectors. Candidate selectors are the rule sets that choose a candidate from a candidate group. Candidate selectors are similar to sub-narratives in that they are constituted of a rule set and metadata describing the selector. When the execution of a narrative (or sub-narrative) meets a candidate group a candidate selector is invoked. Within the system there may be several candidate selectors available for choosing candidates for different purposes. For example, there may be a candidate selector for selecting candidates based on the screen size of the delivery device.

Candidate selectors may be designed to execute in three modes – **all**, **n** and **best**. The **all** mode returns a set of candidates that all meet the minimum requirements of the selector. **n** mode returns the top n (where n is an integer) results. The **best** mode returns a single candidate that best fits the requirements. Using these modes it is possible to call several candidate selectors on a candidate group – each refining the selection until the final candidate selector is asked to make a best selection. For example, a candidate group may contain pagelets covering a concept with different pedagogical approaches and for different devices. Two candidate selectors (one for each variable axis) may be used – one to select **all** appropriate candidates for the device, the second to select the **best** candidate from the subset according to the learner's learning preference, or vice versa if a different precedence order is desired. If the learner is a predominantly visual learner and is using a PDA to view the course the first candidate selector will select all pagelets suitable for a PDA and the second will choose the pagelet most appropriate for visual learners.

Each candidate selector may have descriptive metadata associated with it. This facilitates the reuse of the candidate selectors by other course authors. The metadata may describe the criteria the selector uses and the algorithmic basis for that selection. For example, the metadata of a candidate selector may state that it uses Pearson's Correlation to perform the numeric comparison of learner preference values to metadata values of pagelets.

4.3.5 The Multi-Model Architecture

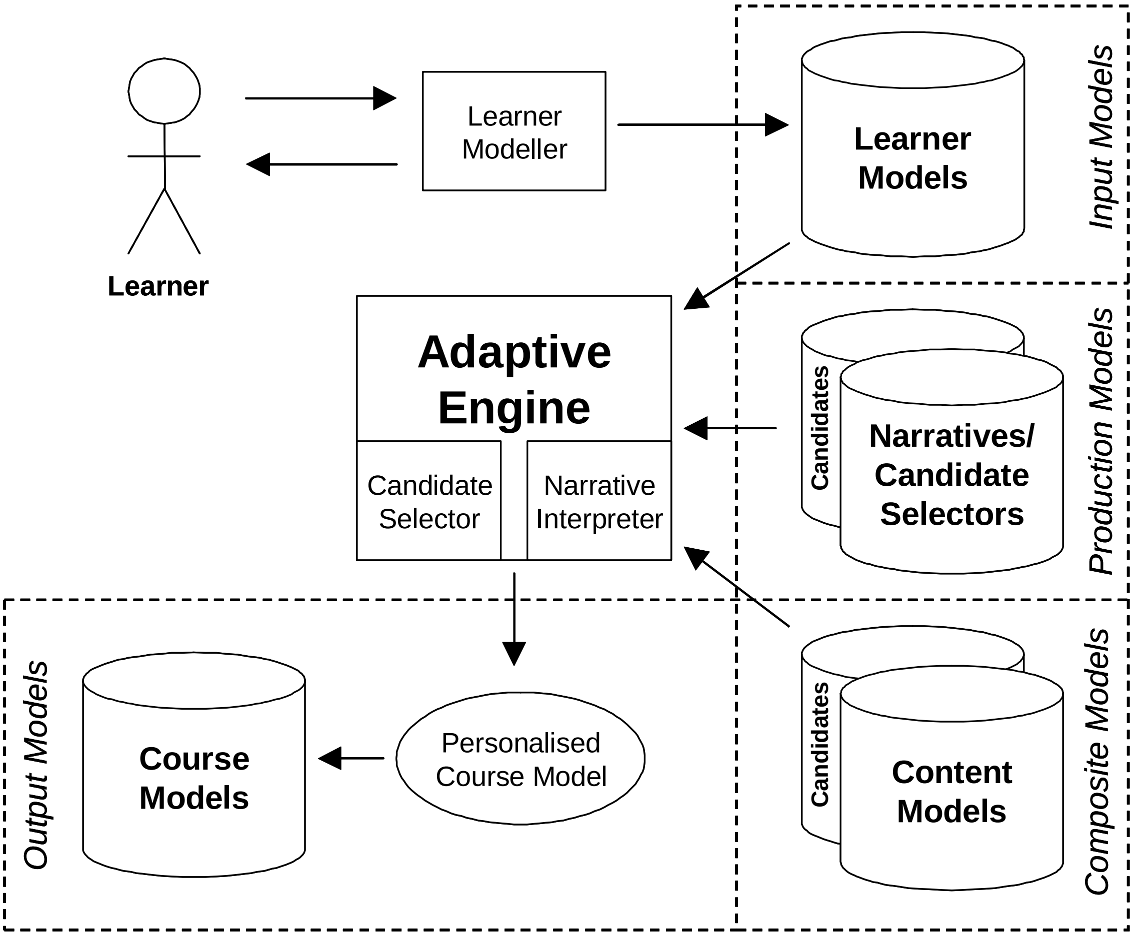


Figure 4.5, The Multi-Model Architecture

Figure 4.5 shows the design architecture of the multi-model approach for Adaptive Hypermedia Systems. At the core of the system is the Adaptive Engine which is responsible for generating personalised course offerings based on the input models, for example the learner model. It contains a candidate selector and a narrative interpreter. These components are rule engines that are responsible for interpreting and executing production models, such as narrative and sub-narrative rule sets, as well as interpreting any candidate selectors utilised by those narratives.

Surrounding the Adaptive Engine core are the repositories responsible for storing the models (learner, content and narratives/candidate selectors), appropriate candidate groups and the outputted course models. The separation of the models into separate repositories supports their reusability. For example, an academic institution may wish to share its educational content, but not its narratives or learner models.

If course authors wish to utilise other input models as part of the adaptation process, information relating to that model should be added, as metadata, to an appropriate input model repository. Such information may be modelled at runtime in a similar manner to the modelling of the learner model. For example, there may be a repository containing models pertaining to the current network conditions which may be used by narratives in the adaptive process. The Adaptive Engine, however, can only determine which model to use if it receives information from an external source about the network. The learner is the only external information source that has to be supported and modelled for personalised eLearning to occur.

4.3.6 Narrative Execution and Model Reconciliation

Narratives and sub-narratives are rule sets that are executed in the rule engine. Every adaptive course has at least one narrative. The main narrative is referred to as the root narrative. It is the role of the root narrative and any children narratives to produce a personalised course offering tailored to the learner model and any other models they wish to access. In this sense it is the narrative model(s) that reconcile the other model information to produce the personalised course. In eLearning applications narratives are the embodiment of pedagogy and may be used to describe different pedagogical approaches. For example, an author may create a course that has a basic *selector* narrative at the root. This *selector*, which may be implemented as a candidate selector, could then reason over a number of sub-narratives, each of which embodies a different pedagogical approach, therefore choosing the most appropriate pedagogical approach for the learner.

In order to provide as much functionality as possible to the course author/instructional designer the rule language expressed in the narratives and interpreted by the rule engine should be as rich as possible. It is envisaged that the course author would design new courses using a graphical interface with design support facilities [Dagger et al., 04], but if the author wished to create narrative structures/sequences manually based on more complex algorithms they should not be limited by the rule language.

As the narrative decisions are based on model information the rule engine must be able to access any model or candidate group information when executing narratives. This information is used to influence decisions made in the narratives. As candidate selectors are based upon the same rule language as narratives and executed by the rule engine they also have access to any model information they require to make a selection. For example, a narrative that adds concepts based on the learner's prior knowledge would, through the rule engine, look at the learner model repository to access the current learner's model. It would then query the learned competencies of that learner before adding the concept. If that concept was represented by a candidate content group with several possible pagelets the rule engine would execute a candidate selector to choose the appropriate pagelet. The candidate selector would use the rule engine to access the content model of each candidate pagelet before making its selection.

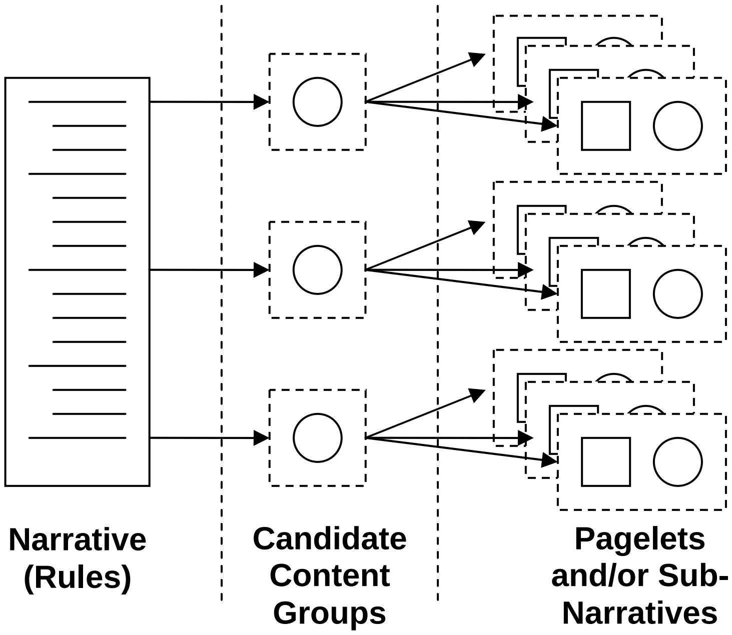


Figure 4.6, Candidate Content Groups as an Abstraction Mechanism

The output of the Adaptive Engine is a personalised course model. The structure of this course should, however, be open, allowing the narrative author to determine the structure. In this way the author does not have to comply with a restrictive course model dictated by the Adaptive Engine. For example, a course author may wish to have a course-section-unit-page structure in a course. The creation of this, or any other structure, is possible through the narrative. The rendering of the course model is not the direct responsibility of the adaptive engine and may be achieved through a separate transformation process.

4.4 Metadata for Describing Adaptivity

There exist a number of metadata standards and specifications issued by different communities and institutions. However, due to a large overlap in the activities of standards development bodies and working groups, these different standards are closely related to each other, for example IEEE LOM, IMS Learning Resource Metadata and ADL SCORM. This section details an extension to standard educational metadata to facilitate the description of the different ways a resource may be used adaptively. Due to the overlap between the different existing standards it should be easy to adopt the proposed extension to the other standards.

4.4.1 A Basic Generic Adaptivity Element

Proposed is the addition of an optional *adaptivity* element within the *education* block of metadata schemas such as IMS Learning Resource Metadata, IEEE LOM, IMS Content Packaging manifests and IMS Learner Resource Metadata [Conlan et al., 01]. Figure 4.7 shows the structure of this new element. It contains an arbitrary number of *adaptivitytype* elements, each of which describes one type or aspect of adaptivity available for a schema. The *adaptivitytype* element itself has two attributes (a mandatory *name* and an optional *ref*), and a *langstring* content. The *name* denotes the type or aspect of adaptivity for which information is provided while the *langstring* content contains the information itself. The second, optional attribute *ref* can be used to specify a URI where the vocabulary used in the *langstring* content is defined. The possible values for the *name* will be partially restricted by a best practice list.

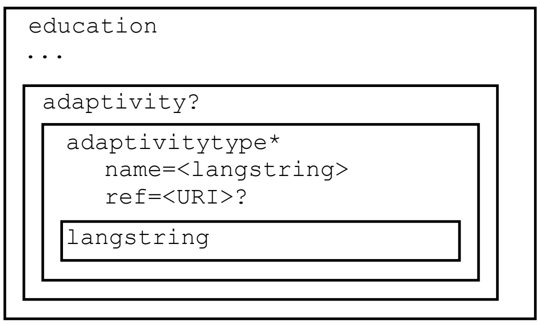


Figure 4.7, The proposed generic adaptivity metadata element

Below, an example *adaptivity* element for a learning resource is shown containing several different *adaptivitytype* entries. The *competencies* type elements in this example show that a hierarchical structure is possible. This demands that the same vocabulary should be used for the *langstring* content of all *competencies.XXXX* entries.

```
<adaptivity>
   <adaptivitytype name="learnerpreference.specialneed">
       Visual Disability
   </adaptivitytype>
   <adaptivitytype name="competencies.taught" ref="some-URI">
       RDBMS Management
   </adaptivitytype>
   <adaptivitytype name="competencies.required" ref="same-URI">
       Database concepts
   </adaptivitytype>
   <adaptivitytype name="learningstyle.vark.visual" ref="some-
otherURI">
       0
   </adaptivitytype>
   ...
</adaptivity>
```

This example shows a learning resource that offers four different types of adaptivity that may be reasoned about during a candidate selection. The *learnerpreference.specialneeds* entry declares the usability of this learning resource for learners with a visual impairment. The *competencies.taught* entry says that the learner acquires the competency to manage a relational database management system learning with this resource, i.e. the entry specifies the objectives of the learning object. A learning object may also have several possible objectives from which the learner (or learning environment) may choose. The content of this competency *RDBMS management* is described in more detail at the referred resource, pointed to by the values of *ref*. The next entry specifies that the learner should already have acquired the competency *Database concepts* before processing the current resource because that knowledge is needed for understanding it and the successful processing of it. Such information may be used for navigation support, i.e. for adaptive sequencing of learning objects. The exact meaning of this competency is defined in the same referred resource as the taught competency. This is important in order to ensure that taught and required competencies can be related to each. The final entry *learningstyle.vark.visual* claims that the resource is useful for learners without a strong visual preference for the display of learning content as the value contained within the element is 0.

4.4.2 A Structured Adaptivity Element

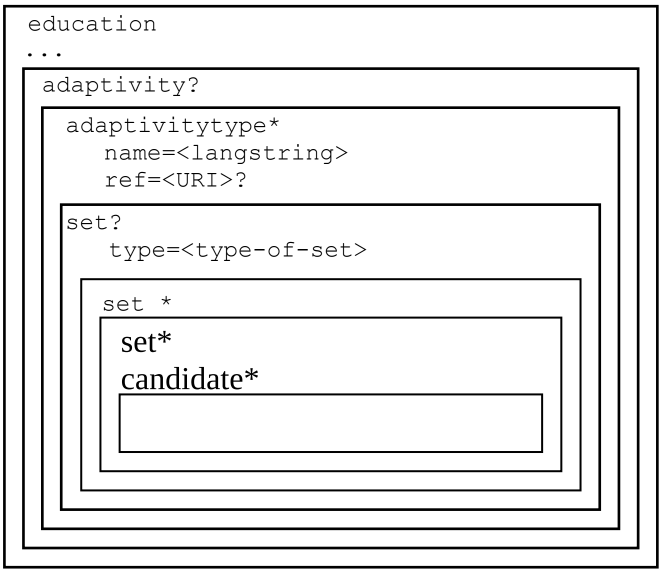


Figure 4.8, The structured generic adaptivity metadata element

The generic element described in the previous section provides, in principle, a means for describing many different approaches to adaptivity. However, for some approaches it would be advantageous to be able to structure the information specified within a certain *adaptivitytype* entry. For example, there may be several competencies necessary to be able to understand a learning resource, and there may be different approaches which make it possible to understand this resource, resulting in several alternative sets of prerequisites. This is especially true when the resource being described is itself adaptive. As a consequence, the need to structure the data provided within the *adaptivitytype* entry arises. Figure 4.8 shows the proposed extended *adaptivitytype* entry covering such structured data. This extension consists of two parts addressing possible problems in the interpretation of multiple *langstring* fields within one *adaptivitytype* entry. A *candidate* block may contain several *langstrings* specifying the same value in different languages. A *set* then can contain several *candidates* with different meanings. The optional *type* parameter of the *set* entry specifies how the candidates should be connected. Example values for this *type* are *all* (i.e. a conjunction) or *one-or-more* (i.e. a disjunction). On the top level, only one *set* is allowed in order to avoid ambiguities about the combination of *sets* at this level. In principle, *sets* may be nested arbitrarily, however, normally two *set* levels should be sufficient (any and-or structure can be represented by a disjunction of conjunctions or vice versa) in order to reduce complexity of the provided data. This structured approach is illustrated in the following example –

```
<adaptivitytype name="competencies.required">
   <set type="one-or-more">
      <set type"all">
         <candidate>
            <langstring lang="en">competency-A</langstring>
            <langstring lang="de">Kompetenz-A</langstring>
         </candidate>
         <candidate> ... </candidate> ...
      </set>
      <set type="all"> ... </set> ...
   </set>
</adaptivitytype>
```

At the top level, there is a *set* of type *one-or-more*, i.e. in order to understand the current learning resource, the learner should master at least one of the following *sets* of competencies. Both second-level *sets* are of type *all*, i.e. the learner should master all competencies within one of the *sets*. In the first *candidate*, we then also have an example for specifying information in different languages. The *competency A* is specified in English, while the German specification names it as *Kompetenz A.*

4.4.3 Best Practice Lists and Vocabularies

The generality of the proposed extension has the disadvantage that users have to identify domain specific terms in a unique, but machine-recognisable way. With respect to the *name* attribute of the *adaptivitytype* entry, this can be realised through a best practice list given that a simple but, nevertheless, moderated way for extending this list is provided. Regarding the *langstring* content, this is more difficult because that will often depend on the vocabulary of the knowledge domain under consideration, e.g. the *competencies* entries in the examples above. While there exist attempts to build a general ontology (e.g. [IEEE SUO]), these have not yet shown satisfactory results that could be applied for this purpose. The *ref* attribute is therefore proposed as a means to specify the vocabulary used in describing the learning object. For example, it may be used to point to an ontology expressed in the emerging Web Ontology Language [OWL]. It is important that widely accepted standard vocabularies for the different fields of knowledge are used because only standard vocabularies can ensure the interoperability and information exchange within learning objects' metadata.

Commonly used library classification systems of UDC [UDC] and DDC [DDC] are too coarse for this role, but may provide the hierarchical basis for such ontologies.

4.5 Content Model Design Issues

One of the main goals of the multi-model approach is to separate the learning content from the adaptive linking logic or narrative. This separation improves the possibilities of reusing a piece of learning content as the learning object (LO) is no longer specific to a given course, implementation or narrative. A second goal (with respect to content) of this approach is to allow course designers to easily discover appropriate learning content in the content repository. This is facilitated by two mechanisms –

- Free text searching, i.e. enabling the author to search the raw learning objects
- Appropriate descriptive metadata

The first mechanism is reasonably effective when the content is in text format (e.g. text/html), but is ineffective when the LOs are not text (e.g. image/jpeg). The LO alone does not contain any information about itself, such as what format it is, what software/plugin is required to render it effectively, how big it might be to download, how much screen real estate it will occupy, etc. These factors are not only useful to a course designer in selecting appropriate learning content, but can be used by the rule engine when executing candidate selectors to select appropriate content where there may be many candidate pagelets available to fulfil a learning or technical requirement.

4.5.1 The Content Model

The content model is the metadata describing an actual learning resource. The content model should be able to describe the aspects of a learning object that make it suitable for a particular learning need or technical requirement. This content model should be generic, scalable and extensible, thus facilitating its use in the description of both the fine-grained pieces of learning content and complete adaptive learning systems (or anything in between). In this regard the content model should conform to a metadata information model for adaptivity (see *APPENDIX I – Metadata*).

The adaptivity types used to describe a learning resource should correspond to entries in the learner model, for pedagogical or competency considerations, or entries in the environment, device, network, etc. models, for technical requirements. From the perspective of a single reusable learning object, the adaptivity types may include, but are not restricted to, types such as competencies.taught, competencies.required and learningstyle.vark.visual. Included in each adaptivity type is the ability to reference, using a URI, an external resource that enables the metadata author to describe that type and any vocabularies/requirements associated with it.

The competency elements of the content model are described in terms of a domain expert derived vocabulary. They can be seen as the pre-conditions and post-conditions to learning a piece of content. For example, a pedagogical approach, such as the Theory of Knowledge Spaces [Albert and Lukas, 99], may require the learner to have acquired all of the knowledge necessary to understand a piece of content before being presented with it. These pre-conditions would be characterised by competencies.required adaptivity types in the content model.

Aside from the adaptivity considerations of the content model it should also support the commonly used elements from metadata standards.

4.5.2 Content Lifespan and Granularity

The separation of content from sequencing logic increases the reusability of the learning objects and potentially increases their lifespan. For example, if a large scale learning object covering aspects of the Java programming language were developed three years ago then, if viewed as a complete course today, it may be out of date with respect to the current version of the language. However, if disaggregated into individual learning objects some of these learning objects (LO) may be reusable.

The potential reuse of LOs is directly related to the granularity, or the scope, of the learning object. From the example above, if the Java course existed as a single large LO, or a collection of coarse grained LOs then its potential reuse is greatly reduced. If, however, it was divided into finer grains of learning content its potential reusability is substantially increased. In Section *4.3.1 Model Separation and Model Types*, a pagelet is defined as a learning object and its associated metadata. This definition can be refined to include some comments on the scope and size of the pagelet. A pagelet should be fine grained enough to ease its reuse. Textual content on a paragraph level, single diagrams, thirty second snippets of audio content or a single interactive animation are considered fine grained.

One possible disadvantage of this approach is that if the pagelets are poorly sequenced then they may appear to be inconsistent or incoherent. It is a role of the narrative author to ensure that the customised courses produced from the narrative contain concept and pagelet sequences that maintain learning concept coherency and have a logical flow. There is research being carried out to monitor the coherency of personalised courses and make recommendations to narrative authors as to how the narratives may be improved [Funk and Conlan, 03]. This research is, however, outside of the scope of this thesis.

4.6 Designing the Learner Model and Modelling the Learner

4.6.1 Competencies, Objectives and Vocabulary Adoption

The learner model should be capable of storing any information about the learner that is required by the personalisation process. With respect to competencies such information may include –

- Prior knowledge or competencies already learned
- Learning goals or objectives

Using the metadata schema, described in *4.3.2 Metadata for Model Representation*, it is possible to create a syntactical structure in which to store this information. The semantics of how that information is organised and structured are not, however, defined in metadata. With respect to the competencies this raises a number of questions –

- What vocabulary should be used to describe the competencies?
- How should the competencies be structured?
- What level of detail does this vocabulary need to describe?

As the narrative is developed (in part or wholly) by an expert in the knowledge domain it is up to them to use whatever vocabulary they feel best describes the knowledge domain. This may be a vocabulary designed by them or extended from another source. Onthologies such as UDC tend to be good at high level organisation of knowledge domains, but lack the detail for fine grained description. If the knowledge domain expert wishes to make the vocabulary public they can do so by publishing it on the web and referencing it using the *ref* attribute of the *adaptivitytype* element. This enables other authors to discover, reuse and extend vocabularies published in this way and in turn publish their own extensions. This leads to an organic approach to ontology development, but it is driven by knowledge domain experts and will only gain acceptance if peers also use and develop the ontology.

The granularity to which the vocabulary exists and the scope of the pagelets determine the level at which the engine can adapt to learner competencies and goals. For example, if a course author decides there are ten learning objectives that may be modelled in a course then the finest level that the adaptive engine can personalise a course with respect to competencies is at the scale of one of those objectives. This is true even if the pagelets are finer grained, as each learning objective may require several pagelets to fulfil it. On the other hand, if there was a learning objective associated with each pagelet then the engine could personalise the course on a pagelet by pagelet basis. There is a balance between the vocabulary granularity and pagelet granularity that determines the level of content personalisation that is achievable by the adaptive engine.

4.6.2 Pedagogical Considerations

Learning style is a term used to describe the attitudes and behaviours that determine an individual preferred way of learning. Learning style preferences have implications for all types of learning, whether the learning is dedicated to the acquisition of knowledge through formal structured activities, such as lectures, case studies and books or through experiential learning, such as learning through experience [Honey, 86]. For the online paradigm, as in traditional classroom situations, there is no consensus on how best to model learners’ preferred approaches to learning. Therefore, the approach taken in the design of this system is to enable the instructional designers to impact the rendering of the personalised course at two levels –

- The structure/layout in which the content is placed
- The modality of content displayed

To this end the system enables many candidate narratives, supporting different pedagogical approaches to structuring the content, to be associated with a single course. This association and an appropriate candidate selector enables the system to deliver a personalised course that, while dealing with the same subject matter and learning goals, can be structured in a way that best engages the learner's preferred learning styles. Using candidate content groups, the narrative can refer to a concept that is represented by a group containing several pagelets, each designed to support a different learning preference.

Using this mechanism the instructional designer can implement several candidate pieces of content. Each candidate can be designed to best support a learner's preference for how the content is presented. Using an appropriate candidate selector (or developing a new one), the instructional designer can influence the candidate selection procedure.

One approach to producing alternative content candidates is the VARK (Visual, Auditory, Read/write and Kinaesthetic) [VARK] model that influences the individual candidate design. The learner's VARK preferences can be determined using an online questionnaire. These values can be used to populate the learner model. Individual pieces of content can also be designed to emphasise one of these aspects – visual, auditory, read/write and kinaesthetic [Bruen and Conlan, 02]. The system can reconcile the content model and the learner model to determine the appropriate candidate using a candidate selector.

4.6.3 Modelling Pedagogical Information

Within the system there are a number of models, not just the learner model, that represent pedagogical information. Section *4.6.2 Pedagogical Considerations,* mentioned two levels for describing pedagogical information, one that influences course structure and concept sequencing and one that influences candidate selection to fulfil concepts. In a similar way to how competency information across the models requires a common (or mapped) vocabulary base, the pedagogical information needs to be represented in an agreed manner. For example, if the system is modelling the VARK preferences of learners, how are the values quantified? Again, this is at the discretion of the course designers. The designers may use the explicit and implicit mechanisms for acquiring learner information (supplied by the system), but they should use caution as incorrect learning style and preference information can produce extreme results.

4.6.4 Describing the Learner – what to model?

The learner model employs a mechanism, similar to the content model, to enable an extensible framework where information impacting on the creation of a personalised course may be placed. There is an Adaptivity sub-section in the metadata model that enables the definition of new adaptivity types. For example, these types may include competencies.learned, competencies.required and learningstyle.vark.auditory, etc. As this is an extensible framework, the domain expert, who describes the learning content requirements of a course, and the instructional designer, who describes the pedagogy requirements of the course, can define new adaptivity types to facilitate other forms of adaptivity they may wish to implement in the narrative.

In the system design it is envisaged that the primary elements modelled in the learner model, with respect to competencies, are –

- Learned Competencies– Knowledge that the learner already has that relates to the course being generated
- Required Competencies– Knowledge that the course author has determined is the minimum knowledge the learner must acquire to complete the course
- Optional Competencies– Knowledge that the learner would like to acquire (aside from the required competencies), i.e. competencies that are outside of the core curriculum for the course

When the learner learns a required or optional competency it becomes a learned competency. The vocabulary used in describing these aspects of the model should correspond with that described by the *ref* attributes.

The learner model should also contain information pertaining to the learner's learning preferences and learning style, though, like all aspects of the multi-model approach, the course author and instructional designer may choose to model whatever elements they feel are required to implement effective narratives.

The learner model may also include information about the learner such as a unique identifier, forename, surname, date of birth, etc. These elements should be derived from relevant standards and specifications.

4.6.5 Acquiring and Developing the Learner Model

There are two primary mechanisms for building learner models – explicit querying of the learner and implicit observation of their interactions with the system. As the narrative development and the adaptive effects it produces are controlled by the domain expert and instructional designer the system should provide them with facilities to utilise both of these techniques. For example, if the author wishes to determine the prior knowledge of the learner before the narrative builds a personalised course they may use explicit techniques to query the learner for that information. If they wish to develop the learner's model while they are engaged in a personalised course they may use implicit techniques to examine what the learner is viewing and update their learner model accordingly.

Again the issue of vocabulary arises – what is the terminology used to populate the learner model? This should be the same as that describing the content model and referenced in the *ref* attributes of the competency adaptivity types. The author may, however, wish to employ a hierarchical structure for the competencies whereby if the learner's model states they have learned a higher node in the hierarchy it is assumed that they have learned all of the children of that node. This approach can save much duplication in the learner model, but if the model is to be used by other instances of adaptive courses this could cause difficulties if they do not understand this abbreviation.

The issue of vocabularies is analogous to that of describing learning styles and preferences. As there are no accepted best approaches to modelling these characteristics of the learner, again it is left to course designers to model these aspects however they wish. It is the flexibility of the metadata structure utilised in the multi-model approach that enables them to do this.

As these decisions rest with the course authors it is the role of the system to provide them with open mechanisms to manipulate the learner model as they see fit.

4.7 Describing the Narrative

The narrative for a course describes the sequencing rules and metadata, developed by domain and pedagogical experts, which govern the range and scope of personalised courses that the adaptive engine can produce for learners. The narrative enables the course author(s) to separate the rules which govern how the personalised course will be generated from the content that will be included in that course.

4.7.1 Encapsulation of Domain Expertise

The narrative rules are a representation of the expert's knowledge of a domain and are described at a conceptual level. Narratives can be used to generate courses that differ in ethos, learning goals, pedagogical approach and learner prior experience from a common content repository. The vocabulary used to describe the learning concepts embodied in the course is that of the domain expert. As the narrative does not refer directly to individual pagelets, but rather to candidate content groups using this vocabulary, the domain expert can create the narrative without being constrained by pedagogical or technical delivery issues at the content level. The author can simply refer to the candidate content group in the narrative and allow the adaptive engine, using a candidate selector, to determine which candidate from the group is most suitable for delivery.

4.7.2 Producing Coherent Personalised Courses

The primary goal of the narrative is to produce courses that are pedagogically sound, structured coherently and fulfil the learning goals for the course in a way that engages the learner. It is, therefore, the domain expert's task to ensure that each learning goal has sufficient learning concepts to fulfil that goal and that these concepts are sequenced appropriately.

From this perspective the domain expert must consider how the exclusion or inclusion of concepts or sequences of concepts (in the case of sub-narratives) will impact on the intelligibility of neighbouring concepts and on the personalised course as a whole. To this end it is often useful to determine, before designing a narrative, what is the granularity of personalisation that is to be achieved, i.e. personalisation on the section, page or paragraph level. This decision is influenced by both the granularity of the vocabulary describing the concepts and the granularity of the pagelets that will fulfil that concept. The granularity of personalisation cannot be smaller than the larger of the vocabulary or pagelet granularity. For example, if pagelets are of the paragraph level, but the vocabulary used to describe the course concepts is mode coarse then the granularity of personalisation is limited to that of the vocabulary. The reverse is also true, where the learning content is coarse grained and the vocabulary is fine grained, the granularity of personalisation is limited by the content.

When designing a narrative it is also useful to determine what content, if any, is considered core material and will always be presented in all personalised content. With granularity of personalisation and the core material determined, the expert has a framework in which to consider the impact of the inclusion or exclusion of concepts based on the learners' expertise and preferences.

The structure in which the content is placed should be completely open and defined by the course author in the narrative. This enables the course author to produce courses that correspond to any course model or structure they desire and not be constrained by a course model within the system.

4.7.3 Reconciling the Learner Model and the Content Model

The vocabularies used to describe the knowledge domain, for which the course is being developed, and to describe the learner's prior learned and required/optional competencies are determined by the expert. This expert must ensure that those vocabularies are one common vocabulary or that there is a translation mapping available between the two vocabularies. The system does not place any restrictions on what constitutes the vocabulary, only that the narrative and any elements of the learner model, or any other models that the expert wishes to reference share that vocabulary or that there exists an appropriate mapping between the model vocabularies. This enables the narrative to reconcile learner competencies (learned or required) with candidate content groups.

For example, if, through pre-testing, the engine has determined that the learner has learned the programming.language.c++.control_structures competency then, as part of a course on programming Java, the narrative may not include all of the learning concepts on Javas control structures. In order for the narrative to make this decision it must have some understanding of the vocabulary used to describe the learner's learned competencies.

The reconciliation between the Learner Model and the Content Model, via this common vocabulary, enables the narrative to select appropriate candidate content groups when building a personalised course offering. A narrative (root narrative or sub-narrative) is not directly concerned with reconciling learning style information that impacts content selection. It is only concerned that the learning concepts for the final course should meet the learner's learning objectives and that they should be structured in a pedagogically appropriate way for that learner. Learning style considerations are, therefore, catered for by the appropriate selection of narrative and content.

4.7.4 Candidate Narrative Groups

The candidate content group approach can be applied at a higher level to facilitate many instructional design approaches within the same course. Using the abstraction principles from candidate content groups, it is possible to have several candidate narratives for a single course grouped into candidate narrative groups. The candidates may have the same ethos, learning goals and require the same learner prior knowledge, but differ in pedagogical approach. Where these kinds of pedagogical issues, i.e. those which impact the course structure, are implemented the process of developing the narratives is often a collaborative effort between instructional designers and the domain expert. Using appropriate candidate selectors, which look at the narrative model metadata and learner model, the system is capable of selecting the most appropriate course structure for the learner. The factors affecting the course structure selected may not necessarily be derived from the learner model. For example, the learner may wish to undertake a revision course. In this case the candidate selector may use information from an input model representing the learner's task.

4.8 Technologies supporting the Multi-model Approach

The multi-model approach places a number of requirements on how implementations that comply with that approach are developed. These requirements, particularly with respect to how data is stored and manipulated, influence the technologies selected.

As information modelling and metadata representation are key aspects of the multi-model approach, a fundamental technology for the Adaptive Personalised eLearning Service (APeLS) implementation is the eXtensible Markup Language (XML) [XML]. Also of importance within the system are mechanisms that enable the modelling, storage, querying, parsing and transformation of such models. As XML is the basic means of representing the models these mechanisms should be capable of interpreting XML.

In order to produce adaptive effects, APeLS must manipulate these models in accordance with the design ethos and goals of the course authors. Therefore, APeLS must also have an expressive rules language and an engine to interpret these rules.

4.8.1 Data Representation using XML

XML, and its associated schema definition languages (DTD [DTD], XSD [XSD] and XDR [XDR]), provide a framework for representing information in a structured and machine understandable way. By defining appropriate schema definitions the structures of the different models used in the multi-model approach may be represented. The schema definitions enable the system to ensure that any information represented in a model conforms (at least syntactically) with the schema for that model. For example, in APeLS there are at least three schema definitions in a given instance of the service – pagelet metadata, narrative metadata and learner metadata.

By using XML, and associated schemas, the models used within APeLS may be exposed outside the system enabling their reuse. For example, the schema for pagelet metadata is compliant with IMS Learning Resource Metadata v1.2 and, therefore, may be reused by any system that recognises this metadata specification.

4.8.2 Data Modelling using JDOM

JDOM (Java Document Object Model) [JDOM] is a wrapper implementation of DOM (Document Object Model). JDOM and DOM enable compiled code to manipulate hierarchical data structures in memory. JDOM, in particular, was developed with the aim of manipulating XML structures (represented by a DOM) from Java code. There are a number of advantages to using JDOM rather than accessing XML files directly. The primary benefit is to performance – it is much quicker to read and modify a DOM in memory than to attempt the same with an XML file. The second benefit is that JDOM provides many ancillary features such as enabling DOMs to be parsed, queried and transformed as if they were XML. This again provides significant speed benefits, especially to traditionally slow processes such as transformations.

Within APeLS JDOM is used in two key areas of the system. The first is in the memory caching of commonly accessed XML models. For example, the Learner model is continually accessed during the adaptation process and runtime of the system. By reading the XML structure of the Learner model into a JDOM structure the system can read and update the model without the overhead of continuous file access. The second area in which JDOM is used is in the representation of the course model which is assembled by the execution of the narrative. As the course model is updated continuously during the execution of rule sets, it would be considerably slower to access XML files directly. Also, as the course author may wish to create any course structure from the execution of narratives, JDOM provides a flexible mechanism to create and modify any structured data (in this case the course model).

4.8.3 Data Storage using Xindice

Xindice [Xindice] (previously dbXML) is a database management system that uses XML as its basic unit of storage. The XML documents may be placed into conceptual collections for organisational reasons. Using collections, different models from APELS may be separated. For example, there are separate collections for –

- Learner Metadata
- Pagelet Metadata
- Narrative Metadata

- Candidate Content Groups
- Candidate Narrative Groups
- Course Models

Xindice can not only deliver and store XML documents, but can also use DOMs directly. This enables a technology such as JDOM to integrate with a Xindice database and retrieve DOMs from it. Xindice may also be optimised by defining indexes across different documents and document types. For example, if several different types of documents included an identifier element under different names (e.g. `<identifier>`, `<id>`, `<i>`) it would be possible to define an index that related the diverse forms of identifier into a single, searchable index. Xindice provides a number of Query Services to facilitate different means of querying the database. The queries may be executed across the entire set of collections, on a single collection or on a single document in a collection. One example of such a query service is the XPath Query Service.

4.8.4 Data Querying using XPath

XPath [XPath] is a language that enables the values of elements, groups of elements and hierarchical elements to be returned from XML documents and DOMs. Through the use of wildcards it can be used as a query language to retrieve elements and values based on partial matches of element names and/or the values of elements.

In APeLS XPath is used to access and query the collections in Xindice via the XPath Query Service. This is used in two ways in the narrative rule sets to enable the rules to–

1. Query and compare the metadata elements of different documents across different collections in the database and make decisions based on the return values, and
2. Retrieve specific information from elements within the metadata models for comparison with other information or for inclusion in course models

XPath syntax enables complex relationships to be queried. For example, a query may search for all the preceding ancestor nodes of a document where the name attribute of adaptivitytype is competencies.taught and the identifier is p-0001. The ability to return nodes based on relationships (axes in XPath) and not only on name (as SQL does) gives XPath great flexibility. XPath axes include child, parent, descendant, ancestor, following, preceding, following-sibling, preceding-sibling, descendent-or-self and ancestor-or-self. These axes along with the path resolution and function mechanisms of XPath enable any node of an XML document or DOM to be uniquely referenced of queried.

4.8.5 Data Parsing using Xerces

Xerces [Xerces] is an XML parser. This ensures that any XML (document or DOM) used by the system is well-formed. Xerces can examine XML that has an associated schema, such as the core models (see *APPENDIX I – Metadata - Schema for Content Model*), for validity with respect to that schema. For example, the schema for the pagelet metadata states that it must include an identifier element. Xerces will ensure that all XML that claims conformance with this schema has an identifier element.

Parsing is also an integral part to *navigating* XML structures as it builds the hierarchy of nodes from the tagged elements in the document. Other technologies used in APeLS, such as Xindice, XPath and Xalan require an XML parser to function.

4.8.6 Data Transformation using Xalan

Xalan [Xalan] is an XSLT (eXtensible Stylesheet Language Transformations) processor that enables well-formed tagged documents (e.g. XML, XHTML) to be transformed from one form to another. This transformation occurs in accordance with a style sheet defined in XSL.

Within APeLS Xalan is used to perform transformations from the course model(s) to Java Server Pages (JSP) [JSP] renderings of the course. Different style sheets may be developed to produce courses with a different look and feel or that are produced in different formats (e.g. XHTML, Adobe PDF, Microsoft Word etc.).

4.8.7 Rule Sets using JESS

JESS (Java Expert System Shell) [JESS] is a rules language and an extensible Java-based engine. JESS is based on CLIPS [CLIPS], which is a LISP-like language, and does not require compilation because it is interpreted at runtime. JESS extensibility is the primary reason for it being used as part of the APeLS rules engine. Using JESS it is possible to create new functions, written in Java, which can be accessed from the JESS rules. These custom functions can extend the functionality of JESS and access other components of APeLS, such as the data storage. It is also possible to author, in JESS, additional functions as part of the rule set. These additional functions can exist in the same rule file where they are called or may be in reusable rule files. In JESS rule files are referred to as batches and may be called from other batches. This facilitates the sub-narrative and candidate selector facets of the multi-model approach.

4.8.8 Content Delivery using Tomcat

Once the personalised course model has been produced and transformed into JSPs it must be delivered to the learner. This is facilitated by Jakarta Tomcat [Tomcat], which is a Java Servlet and Java Server Pages compliant web server. Tomcat may also be integrated with more robust web servers, such as Apache [Apache] and Microsoft Internet Information Server (IIS) [IIS], so that these may utilise its servlet and JSP handling facilities. Different instances of APeLS use Tomcat integrated with both Apache and IIS for robust delivery of the service. By using Tomcat and JSPs it is possible to access APeLS via a web interface. This facilitates both explicit and implicit feedback from the learner into the system which can be interpreted by the learner modeller.

4.9 Service Architecture

The system has been designed to reflect the course personalisation facility as a service to fit within a service architecture. This does not directly impact on the internal design of the Adaptive Engine, but impacts on –

- Learner access to the service
- Acquisition of the learner model
- Passing of learner progress information

As the system is designed as a service it can either be accessed through a virtual learning environment (VLE) or using a custom interface to provide the information and facilities associated with VLEs.

4.9.1 Service versus System

There are a number of advantages to offering personalised courses as a service –

- The learning environment will provide facilities such as learner management, access to tutors, online assessment, collaborative learning tools, etc., allowing the personalised course authors to concentrate on the development of personalised courses
- The adaptive service may be made accessible to many different VLEs, increasing the number of potential learners
- The adaptive service may use information about the VLE as the basis for a model in the multi-model architecture, thus offering different courses tailored to the requirements of the VLE vendor as well as to the learners

The last two points require the adaptive service and VLE to be able to pass information in order to build learner models. The adaptive service may also acquire information about the VLE, if a model of the environment is required by the adaptive service.

4.9.2 Information Flow and the Service Layer

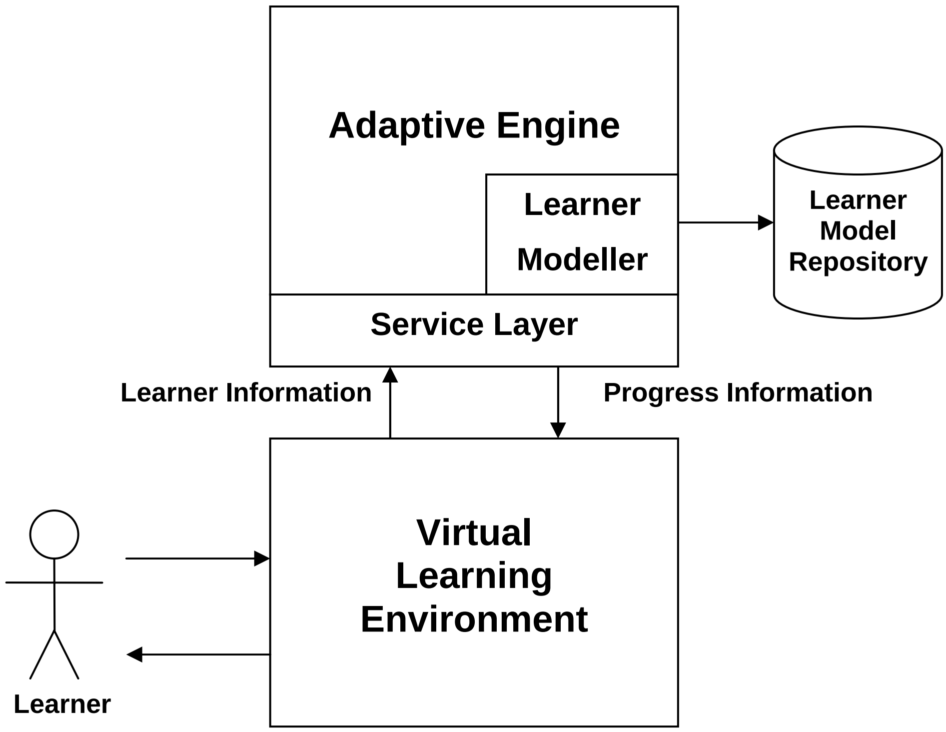


Figure 4.9, Information flow between the VLE and Adaptive Engine

Figure 4.9 shows the information flow between the learning environment and the Adaptive Engine (see *Figure 4.5, The Multi-model Architecture* for a more detailed diagram of the Adaptive Engine). The service layer is the only addition to the multi-model architecture required to interface between VLEs and the Adaptive Engine. It is the role of the service layer to translate between the VLE and the facilities in the engine, such as the Learner Modeller. For example, when a learner connects to the service the service layer may ask the VLE for information about the learner. It passes this information onto the Learner Modeller, which converts it into a learner model for the adaptive engine (e.g. the rule engine) to use.

The service layer is also responsible for passing any progress information required back to the VLE, this time translating from the internal learner model representation of the adaptive engine to a data model understood by the VLE. The service layer must, therefore, understand the data model of the VLE and have some means of communicating with it.

4.10 Summary

This chapter has discussed the design of the multi-model approach to developing adaptive systems. It has shown that the mechanisms required to ensure pedagogically sound narratives and content are produced that can be effectively reused (Abstraction and Candidacy). It has also discussed the factors that impact on the design and implementation of the core models. For the content model issues such as content lifespan, granularity and candidate content groups were discussed. With respect to the learner model competency vocabularies, pedagogical considerations and acquisition of the model were detailed. The narrative model was discussed from the perspectives of it being an embodiment of instructional design, expert knowledge and the means of reconciliation between the learner and content models.

A design approach for offering the adaptive system as a service was also discussed, suggesting a service layer component to act as an intermediary between the system and learning environments.

5 IMPLEMENTATION

5.1 Introduction

This chapter discusses the implementation of the multi-model, metadata driven approach described in the design chapter. APeLS, the Adaptive Personalised eLearning Service, is presented as a service that implements this approach. This chapter is divided into a number of sections each detailing an aspect of implementation; from the development of APeLS, through using it to create personalised eLearning offerings, to deploying it as a service. This chapter primarily looks at the technologies that may be used to support the multi-model, metadata driven approach and the architecture of APeLS, including how it accesses and collects the models used in this approach, how narrative is interpreted, how candidate abstraction is implemented and how personalised eLearning courses are produced and rendered. Also discussed in this chapter are the various instances of eLearning courses that have been created with APeLS.

5.2 Architecture of APeLS

Detailed in this section are two views of the APeLS architecture – a component view and a technological view. Both correspond to the architecture design of the multi-model approach described in Section *4.3 The Multi-model, Metadata Driven Approach*. The component view is used to illustrate the APeLS instantiation of the multi-model approach using three models and the technological view is used to show how the technologies described in Section *4.8 Technologies supporting the Multi-model Approach* have been used to implement APeLS.

5.2.1 Component Architecture View

Figure 5.1, below, shows the component architecture of APeLS, which corresponds to the Multi-Model Architecture shown in *Figure 4.5, The Multi-Model Architecture*. At the core of APeLS are three components – the learner modeller, the rule engine and the candidate selector. The architecture of APeLS utilises the three core models of the multi-model approach – learner, narrative and content.

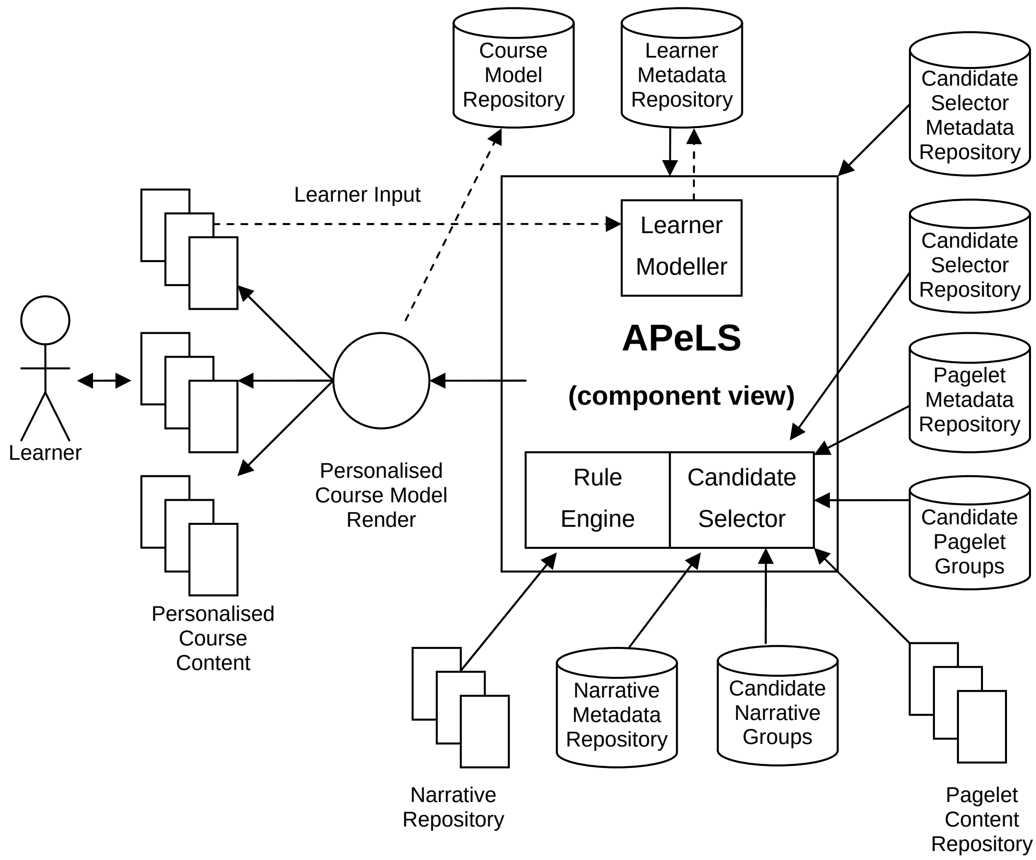


Figure 5.1, APeLS Component Architecture View

APeLS utilises a number of logical metadata and information repositories to store the models used in the system –

Learner Metadata Repository – This repository stores all of the metadata representing the individual learners in the system. This metadata conforms to the Learner Model.

Pagelet Metadata Repository – Stored in this repository are the metadata records, conforming to the Content Model, corresponding to each piece of learning content (or pagelet).

Pagelet Content Repository – All of the pagelet content referred to by the Pagelet Model Repository are contained in this repository.

Narrative Metadata Repository – Stored in this repository are the metadata records that describe the learning objectives and pedagogical approach for each narrative in the narrative repository.

Narrative Repository – The narrative repository stores all of the narratives used to construct personalised courses.

Course Model Repository – This repository stores the models of personalised courses previously created by the system.

Candidate Selector Metadata Repository – This repository contains descriptive metadata about the candidate selectors contained in the Candidate Selector Repository.

Candidate Selector Repository – Contained in this repository are rule snippets that may be used to select a single candidate from a candidate group.

For implementation it is not necessary that each of these repositories is physically separate. There are also two candidate group repositories –

Candidate Pagelet Groups – The groups in this repository reference metadata in the Pagelet Metadata Repository that fulfil the same learning goal. The content model metadata determines how the content differs, i.e. technically or in educational approach.

Candidate Narrative Groups – This repository contains groups of narratives that encapsulate the same knowledge, but employ different pedagogical approaches to structuring the content.

Learner Modeller – The role of the learner modeller is to provide learner model functionality to the learner interface (learner pre-test or delivered course). In APeLS this includes the capability to read and write information to/from the current learner model as well as the ability to create new elements in the model.

Rule Engine – The rule engine is responsible for interpreting narratives and sub-narratives in order to produce a personalised course model. It can access any information stored in metadata or candidate group repositories that the narrative execution may require.

Candidate Selector – The candidate selector is closely related to the rule engine (they may be two instances of the same engine or a single instance) and is responsible for executing candidate selectors. The rule engine will call upon the candidate selector when there are several candidates to choose from. The candidate selector is implemented as a separate instance of the rules engine when runtime candidate selection is required. For example, when selecting the most appropriate candidate for the delivery environment the candidate selector would be invoked at delivery time, whereas the course model may have been previously constructed.

APeLS utilises a candidate selector, from the candidate selector repository, for choosing the appropriate narrative (or sub-narrative). This is done by reconciling information in the learner model (stored in the learner metadata repository) with the metadata describing the narratives in the candidate narrative groups. After the narrative and sub-narratives have been executed in the rule engine a candidate selector is also used to choose the pagelets from candidate pagelet groups. This may either be done at the same time as the assembly of the personalised course model or may be done at runtime. The personalised course model is stored in the course model repository from where it may be retrieved for later delivery or modification. When the personalised course is being delivered to the learner the appropriate pagelet content is included in the course. A navigation structure is also derived from the course model to aid the learner's browsing of the course.

5.2.2 Technological Architecture View

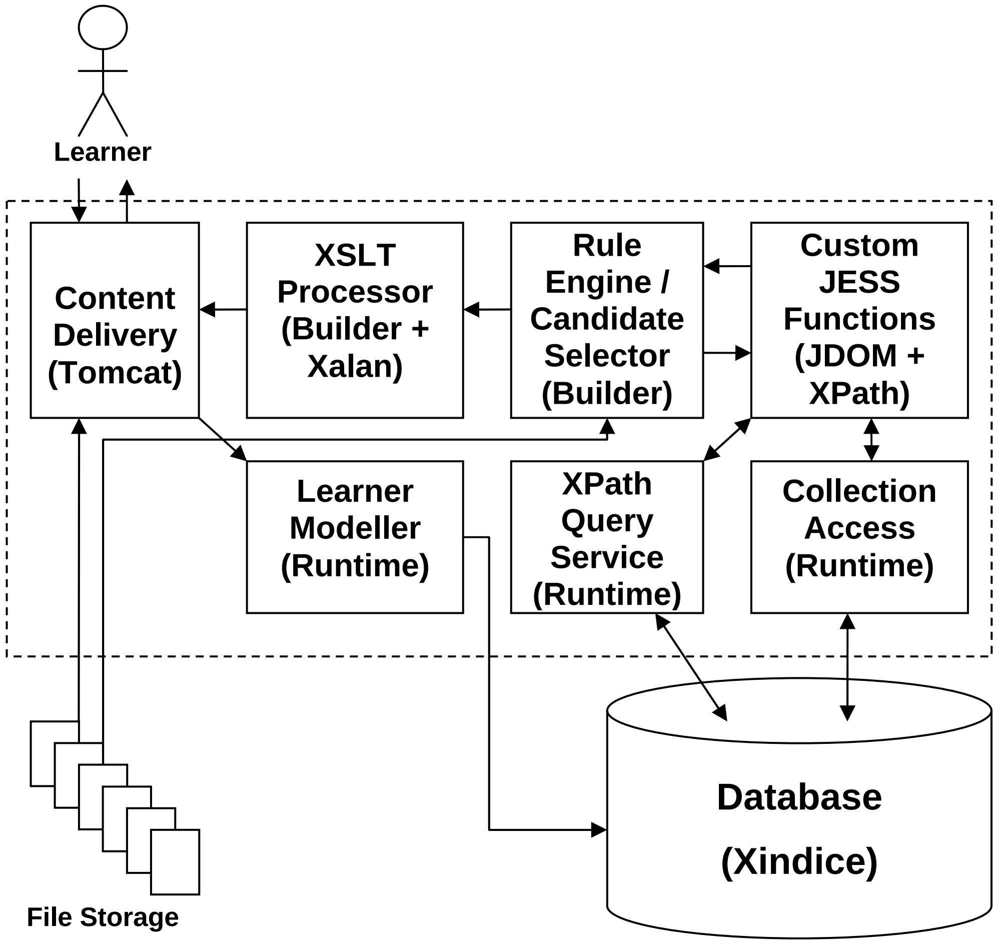


Figure 5.2, APeLS Technological Architecture View

From a technological perspective APeLS makes use of many open source technologies in its implementation. These technologies have been integrated and extended to implement the design requirements of the multi-model approach. Figure 5.2, above, shows the integration and interworking of the components of APeLS. The core of the system is implemented in Java and any functionality required from the open source technologies is imported into this core. A package and class hierarchy diagram is included in *APPENDIX IV – Implementation - Package and Class Hierarchy of APeLS*.

Figure 5.2, above, maps to the Component Architecture View of APeLS shown in Figure 5.1 in the following way –

Design Issue Addressed	Component Architecture View	Technological Architecture View
• Input/Composite Model Storage • Input/Composite Model Querying	Metadata and Candidate Repositories	• Database (Xindice) • XPath Query Service (Runtime) • Collection Access (Runtime)
Process Model and Content Storage	Narrative and Pagelet Content Repositories	File Storage
• Process Model Interpretation • Output Model Creation • Input/Composite Model Querying and Comparison	Rule Engine	• Rule Engine / Candidate Selector (Builder) • Custom JESS Functions (JDOM + XPath)
Candidacy and Abstraction Mechanism	Candidate Selector	Rule Engine / Candidate Selector (Builder)
Modelling Learner (Input) Model	Learner Modeller	Learner Modeller (Runtime)
Personalised Course Display and Delivery	Personalised Course Model Renderer	XSLT Processor (Builder + Xalan)

At the centre of the system is the `Runtime` class. This class provides static functionality for other classes in the system by giving them access to –

- The system definition and comparison tables in the resource bundles (see *APPENDIX IV – Implementation - Resource Bundles*)
- The learner repository for creating new learners, retrieving learner information and getting XML representations of the learner model
- The Learner Modeller functionality (i.e. updating the model, creating new elements, etc.)
- The collection query mechanism allowing other classes to execute XPath queries on any part of the collections

- Mechanisms for comparing learner attributes with those of documents in the collection

The second key Java class in the APeLS implementation is `Builder`. It is `Builders` role to –

- Initialise and run the rule engine and the candidate selector
- Provide access to the custom rule functions to the rule engine and candidate selector
- Provide a DOM structure for the rule engine to manipulate via the custom functions
- Provide mechanisms for transforming the DOM using XSLT

5.2.3 Collection Access and Metadata Comparisons

In APeLS the collection structure of Xindice is used to mirror the repository structure of the multi-model approach. Each repository required by the system is implemented as a separate collection in Xindice. These collections are defined by the APeLS.properties file. By using the collection architecture of Xindice the `Runtime` class, which has collection access mechanisms, can facilitate access to the collections from any part of the APeLS system.

Principally this enables the rules to compare elements and attributes from different collections using XPath as the query language for returning the values. These comparisons may be defined in the rules or in the APeLS_comparisons.properties file. A typical comparison from the properties file looks like –

```
competencies.learned=multiple
subcollection1 //*[name()='adaptivitytype'][@name='competencies.learned]
subcollection2
//*[name()='adaptivitytype'][@name='competencies.required']
```

where

- `competencies.learned` is the name of the comparison that may be referenced from the rules

- `multiple` is the type of comparison to be executed – all returned values from the second XPath query must exist in the first.
- `subcollection1` is the collection where the first document may be found, followed by the XPath expression to retrieve the appropriate values (in the above case it would return all values from the elements `adaptivitytype` where the attribute is `competencies.learned`) . `subcollection1` is defined in the APeLS.properties file.
- `subcollection2` is the collection where the second document may be found, followed by the XPath expression to retrieve the appropriate values (in the above case it would return all values from the elements `adaptivitytype` where the attribute is `competencies.required`). `subcollection2` is also defined in the APeLS.properties file.

From within the rules content may be retrieved directly from documents in a collection using the getMetadata custom JESS function. This function is called from the narrative rules using the following command –

```
(get-content "subcollection2" ?name "//*[name()='location']")
```

where

- `get-content` is the JESS name for the getMetadata function
- `subcollection2` is the collection from which the document should be queried
- `?name` is the identifier of the document to be queried
- `//*[name()='location']` is the XPath query to be executed on that document (in this case it is 'return the value of the location element')

These two approaches to information access and comparison enable the course author to write commonly used queries into the properties file and one off queries directly into the rules, both using the flexibility of XPath to resolve the query.

5.2.4 Pagelets

Pagelets in the multi-model approach are composed of both content and metadata describing the content. This section describes the nature of the pagelet content and the metadata that describes it.

5.2.4.1 Pagelet Content

As personalised courses generated by APeLS are delivered via the web their pagelet content may be any content that can be delivered via a web browser. Principally this includes XHTML, image files, Flash animations and possibly audio content. For regular textual content XHTML [XHTML] snippets are used for the pagelets. As textual pagelet content is at the paragraph level the XHTML does not need to include the markup for a complete document (such as header and body information), but only that markup required to render it. To comply with modern web authoring techniques the XHTML content is also tagged using CSS (Cascade Style Sheet) [CSS] identifiers. This enables the course author or designer to create style sheets for many different look and feels. (Indeed different look and feels could be represented as a different model in the multi-model approach and used to tailor the rendering of the content to the learner's preferences.) A typical piece of textual content would be marked up as –

```
<DIV class="Heading4"><B id="highlight">Selection</B></DIV>
<UL>
  <DIV class="Content">Is a set operation for constructing a <B>horizontal
  subset</B> of an existing relation.</DIV>
</UL>
<DIV class="Content">This example shows how the relational algebra <B>SELECT
</B>operator retrieves all flight entries where the departure location is
Dublin:
</DIV>
<DIV class="Code"><PRE><B id="code_highlight">SELECT</B> * <B
id="code_highlight"> FROM</B> <I>Flight</I>
<B id="code_highlight">WHERE</B> departure_location = 'Dublin'<B
id="code_highlight">;</B></PRE>
</DIV>
```

The class and id attributes are CSS values that may be used to change colour, position, size and other aspects of the rendering. Image and animation content is often associated with an XHTML snippet for CSS rendering. For example,

```
<DIV class="Diagram"><IMG src="images/sql/diagrams/table/air8.gif"></DIV>
```

defines the image as a diagram and the CSS may have a specific rendering style for diagrams.

5.2.4.2 Pagelet Metadata

The content model utilised in the adaptive hypermedia service is based on Information Model version 1.2 of the IMS Learning Resource Metadata [IMS Metadata]. The IMS Metadata specification was chosen as the basis for the content model schema as it was based on the IEEE LOM specification and an XML binding was available.

Adaptivity is not, however, directly addressed by the IMS Metadata Specification and in order for the adaptive engine to choose between several candidate pieces of content it may be necessary for it to have further information about the learning objects. The extension to the IMS Metadata schema consists of the addition of a sub-section called `adaptivity` to the `education` section within the schema. This sub-section caters for user definable adaptivity types allowing the metadata creator to create complex relationships and dependencies within the metadata description of the service [Conlan et al., 01]. This extension is described in terms of XML-Data Reduced Schema Language [XDR] and is included using the `apels` namespace. Below is an example of an XML pagelet metadata record –

```
<record>
    <general>
        <identifier>mod024-003</identifier>
        <language>en_GB</language>
        <title>
             <langstring xml:lang="en_GB">SQL Selection Example</langstring>
         </title>
          <description>
             <langstring xml:lang="en_GB">An example of the SELECT statement
             and WHERE clause using the Flight table
             </langstring>
         </description>
         <keyword>
             <langstring xml:lang="en_GB">SQL</langstring>
         </keyword>
         <keyword>
             <langstring xml:lang="en_GB">Select Statement</langstring>
         </keyword>
         <keyword>
             <langstring xml:lang="en_GB">Where Clause</langstring>
         </keyword>
         <keyword>
             <langstring xml:lang="en_GB">Example</langstring>
```

```
         </keyword>
    </general>
    <technical>
        <format>text/xhtml</format>
         <location type="URI">content/pagelets/mod024-003.html</location>
        <size>669</size>
    </technical>
    <educational>
    <apels:adaptivity>
     <apels:adaptivitytype name="learnerpreference.vark.visual" ref="">
                <apels:candidate>3</apels:candidate>
     </apels:adaptivitytype>
     <apels:adaptivitytype name="learnerpreference.vark.auditory" ref="">
                <apels:candidate>0</apels:candidate>
     </apelsadaptivitytype>
     <apels:adaptivitytype name="learnerpreference.vark.readwrite" ref="">
                <apels:candidate>2</apels:candidate>
     </adaptivitytype>
     <apels:adaptivitytype name="learnerpreference.vark.kinesthetic" ref="">
                <apels:candidate>1</apels:candidate>
     </apels:adaptivitytype>
    </apels:adaptivity>
    </educational>
</record>
```

This record contains three sections – general, technical and educational. The general section is used primarily for search and retrieval of the pagelet, technical stores the format, size and location of the pagelet, while educational is concerned with pedagogical issues such as competencies and support for learner preferences, using the adaptivity sub-section described above.

Competencies

Competencies may be represented in the pagelet metadata using the `adaptivitytype` elements from the `apels` namespace. For example, a pagelet teaching the mathematical concepts of terms in trigonometric functions and operations in elementary algebra might be described as –

```
<apels:adaptivitytype name="competencies.taught" ref="">
   <apels:set type="all">
            <apels:candidate>TrigonometricFunctions.Terms</apels:candidate>
            <apels:candidate>ElementaryAlgebra.Operations</apels:candidate>
   </apels:set>
</apels:adaptivitytype>
```

The subset of a vocabulary for describing mathematical, mechanics and physics material is included in *APPENDIX II – Vocabularies - Knowledge Domain Vocabulary for Physics / Mechanics*. Pagelets may also have competencies that are required to understand the material and may be represented using a competencies.required `adaptivitytype`.

Pedagogical Information

Again using `adaptivitytype` elements pedagogical information about the pagelet content may be represented. For example, the VARK learning preference may be represented as –

```
<apels:adaptivity>
   <apels:adaptivitytype name="learnerpreference.vark.visual" ref="">
             <apels:candidate>3</apels:candidate>
   </apels:adaptivitytype>
   <apels:adaptivitytype name="learnerpreference.vark.auditory" ref="">
             <apels:candidate>0</apels:candidate>
    </apelsadaptivitytype>
    <apels:adaptivitytype name="learnerpreference.vark.readwrite" ref="">
             <apels:candidate>2</apels:candidate>
    </adaptivitytype>
   <apels:adaptivitytype name="learnerpreference.vark.kinesthetic" ref="">
             <apels:candidate>1</apels:candidate>
    </apels:adaptivitytype>
</apels:adaptivity>
```

The scores for the different aspects of VARK are represented in the `candidate` elements.

5.2.4.3 Pagelet Selection

Using an appropriate candidate selector different pagelets may be compared based on their competencies, VARK or any other `adaptivitytype` values. Section *5.2.3 Collection Access and Metadata Comparisons* describes how Xindice and XPath may be used to retrieve information from metadata models and compare that information. When a course model is constructed for a learner by APeLS the pagelets may not have been added to the model – the candidate pagelet groups may be left as place holders.

Before the course can be delivered, however, the candidates must be selected. This may occur at runtime and involves the comparison of pagelet metadata and learner preferences by the candidate selector. As the candidate selectors are executed in an instance of the rule engine they have access to the custom JESS functions for accessing the `Runtime` class's collection access and XPath mechanisms. For example, a VARK candidate selector may compare the learner's top two VARK preferences with the potential candidates using Pearson's Correlation. This comparison requires the candidate selector to access all four of the learner's VARK scores, select the top two, and then select the corresponding VARK values from the potential candidates. If there are only two candidates the candidate selection involves eight accesses to metadata models.

5.2.5 Learners

Unlike the narratives and pagelets the learners in the system are represented by metadata only. This is because the learners themselves are what the metadata is trying to model. As learners are not static entities in the system the learner model is continually updated as they develop. Different aspects of a learner develop at different rates. For example, learning styles and preferences may change slowly, if at all, whereas competencies and goals may change rapidly as the learner learns and discovers new interests. As APeLS produces personalised courses for learners, and most personalisation decisions are based in some way on the learner model, the learner model is the most frequently accessed and updated metadata model in the system. APeLS has been developed to cache a permanent copy of the learner model in memory and update the copy in the learner repository (Xindice collection for learner metadata) periodically. This gives the system quick access to this frequently used model.

The Learner Modeller functionality provided by the `Runtime` class provides access to the learner metadata in a slightly different manner to how other models are accessed. The learner model is cached in memory as a DOM. The Learner Modeller provides methods that enable XPath querying on this model directly (rather than through the collections of Xindice). The XPath querying of the cached learner DOM provides a considerable performance increase, considering the frequency of access to the learner model.

5.2.5.1 Learner Metadata

The learner model in APeLS uses the adaptivity extension provided by the `apels` namespace. The adaptivity extension allows information that pertains to the personalisation process to be stored in the model. Here is an example learner record –

```
<learner>
   <general>
      <identifier>oconlan@cs.tcd.ie</identifier>
      <name>
         <surname>Conlan</surname>
         <forename>Owen</forename>
      </name>
      <version>17</version>
   </general>
   <educational>
      <apels:adaptivity>
        <apels:adaptivitytype name="learningstyle.vark.visual">
                <apels:candidate>3</apels:candidate>
        </apels:adaptivitytype>
        <apels:adaptivitytype name="learningstyle.vark.auditory">
```

```
                <apels:candidate>1</apels:candidate>
        </apels:adaptivitytype>
        <apels:adaptivitytype name="learningstyle.vark.kinesthetic">
                <apels:candidate>3</apels:candidate>
        </apels:adaptivitytype>
        <apels:adaptivitytype name="learningstyle.vark.readwrite">
                <apels:candidate>2</apels:candidate>
        </apels:adaptivitytype>
        <apels:adaptivitytype name="competencies.learned">
             <apels:set type="ALL">
                <apels:candidate>
                   <langstring lang="en_GB">Functions.Terms</langstring>
                </apels:candidate>
                <apels:candidate>
                   <langstring lang="en_GB">Functions.Concept</langstring>
                </apels:candidate>
                <apels:candidate>
                   <langstring
lang="en_GB">Functions.RateOfChange</langstring>
                </apels:candidate>
              </apels:set>
       </apels:adaptivitytype>
       <apels:adaptivitytype name="competencies.required">
             <apels:set type="all">
                <apels:candidate>
                   <langstring lang="en_GB">Newtons1Law.*</langstring>
                </apels:candidate>
                <apels:candidate>
                   <langstring lang="en_GB">Newtons2Law.*</langstring>
                </apels:candidate>
                <apels:candidate>
                   <langstring lang="en_GB">Newtons3Law.*</langstring>
                </apels:candidate>
              </apels:set>
       </apels:adaptivitytype>
   </educational>
</learner>
```

Aside from information relating to personalisation and a unique identifier the model contains only basic information about the learner (in the example above – forename and surname). APeLS does not attempt to store all the information a Learner Management System would. When it is deployed as a service additional learner information and functionality should be handled by the LMS. APeLS also keeps a version number for the learner model. Old learner models are not over-written, rather a new version is created. This leaves open the possibility for rolling back learner models.

The example model above has two forms of competencies represented – learned and required. Learned competencies represent the knowledge the learner has prior to entering the personalised course and the competencies they acquire while doing the course. The vocabulary used to describe these competencies should match, where possible, those used to describe the concepts taught/required by the course. If they do not match a mapping should be provided. *APPENDIX II – Vocabularies* describes the vocabularies used in the two courses implemented on APeLS.

In the example above the `competencies.required` uses the * (wildcard) symbol to indicate that all the sub-competencies of a competency are required. For example, if Newton2Law.* is required then that requirement is satisfied when all of the children of Newton2Law, such as Newton2Law.Concept and Newton2Law.Formula, are learned.

5.2.5.2 Building the Learner Model

Before APeLS can create a personalised course for a learner it must have some appropriate information about that learner. This information is obtained by asking the learner to complete an online instrument, typically a prior knowledge questionnaire, that determines their prior knowledge of the domain. An online instrument to determine their learning preferences, such as the VARK online questionnaire, may also be used to populate the pedagogical elements of the model.

These instruments use the learner modeller component to modify the learner model. For example, if an instrument determined that a learner already knew the syntax of the SQL INSERT statement then a competencies.learned `adaptivitytype` element with the value db.tables.populate.insert.syntax may be added to the learner model. Pedagogical elements may be added to the learner model in the same manner. The naming and vocabulary of these elements and how the narrative interprets them are determined by the course author expert and the instructional designer.

The design of the instruments that determine these or any other learner model elements is the responsibility of the course author. The learner is asked to interact with the instrument on their first visit to APeLS and they may re-access the instrument at any stage during their learning and modify their answers. This process gives the learner an indirect mechanism to change their learner model and rebuild their personalised course accordingly.

5.2.5.3 Updating and Maintaining the Learner Model

The memory-cached DOM representation of the learner model may be accessed via the `Runtime` class when the learner is partaking in a personalised course. There are a number of facilities provided by this class to enable information in the model to be updated, altered and removed or for new information to be added. This functionality is provided primarily by the `addAdaptivityType`, `removeAdaptivityType` and `containsAdaptivityType` methods. These methods are wrapper versions of more generalised methods for accessing the DOM representation of the learner mode which are also available via `Runtime`.

With respect to the interface to the personalised course, mechanisms for modifying the learner model may include explicit querying of the learner or implicit gathering of information by interpreting the learner's actions. Both techniques can use the `Runtime` class to update the learner model. The particular mechanisms employed are decided by the course author and instructional designer. This enables APeLS to support both adaptive and adaptable paradigms for personalised services. The Java Server Pages (JSP) interface, for an instance of APeLS, including implicit learner model updates is discussed in Section *5.5.2 Personalised Course on Physics / Mechanics* and the JSP code for the interface is included in *APPENDIX IV – Implementation*.

The DOM access to the learner model is analogous to how the narrative rule sets modify the course model to produce any course structure desired.

5.2.6 Narratives

Narratives in the multi-model approach are composed of both a rule set and metadata describing that rule set. The rule set governs every aspect of how the personalised course is assembled with respect to the learner model, including the concept sequencing and the adaptive/adaptable features of the course. This section describes the nature of the narrative and the metadata that describes it.

5.2.6.1 Narrative Metadata

The metadata describing the narrative rules may be used to discover appropriate pre-authored narratives in the narrative repository or to select an appropriate narrative where there is more than one candidate available. The primary information used to describe a narrative are the competencies that it requires/teaches and the way the course is structured from a pedagogical perspective.

Competencies

Using the same basic structure as the pagelet metadata the narrative metadata uses the `adaptivitytypes` of competencies.required and competencies.taught to describe the competencies required and taught by courses produced by the narrative rule set. The `set` qualifier attribute of type is particularly utilised in narrative metadata. For example –

```
<apels:adaptivity>
   <apels:adaptivitytype name="competencies.required">
      <apels:set type="one-or-more">
            <apels:candidate>
               <langstring lang="en_GB">RateOfChange.NonLinear</langstring>
            </apels:candidate>
            <apels:set type="all">
               <apels:candidate>
                 <langstring lang="en_GB">RateOfChange.Linear</langstring>
               </apels:candidate>
               <apels:candidate>
                 <langstring lang="en_GB">RateOfChange.Concept</langstring>
               </apels:candidate>
            </apels:set>
      </apels:set>
   </apels:adaptivitytype>
</apels:adaptivity>
```

By using the attribute value of *one-or-more* for type in conjunction with a set with the *all* value as a child, the piece of metadata states that the learner must know either non-linear rate of change or linear rate of change and the rate of change concept (or all three). In this way the metadata may be used to determine if a narrative is suitable for fulfilling a learner's requirements. In the example above the personalised course produced from the narrative may include a brief overview of non-linear rate of change, but not deal with it in depth.

Similarly, the metadata may be used to describe possible learning outcomes in terms of taught competencies. For example, a narrative rule set may not be designed to teach all learners the same final competencies. The metadata may be used to describe the possible learning outcomes.

It is feasible, although not implemented directly in APeLS, that an additional rule element could be developed to match narratives with learner needs based on competencies required. For example, if the learner was required to complete a course, but did not have the required competencies for that course, it is possible that a rule could be fired to search the repository for narratives that would bridge the gap between the learner's current competencies and the entry requirements for the course. Such a utility should only be required where the entry for the course is open to learners with abilities that vary strongly.

Pedagogical Information

The pedagogical information stored in the narrative metadata may include information about the instructional approach used in the narrative rule set. This information is particularly useful when there exists a number of narratives that teach the same competencies, but use different pedagogical approaches. In this case the pedagogical information in the narrative metadata may be used as the basis for candidate selection.

Narrative pedagogical information will usually concern the way the course has been structured. For example, there may exist two equivalent narratives – one that teaches basic theory first, then presents examples and finally a case study. Another course may present a case study first, followed by theory and finally includes self-test exercises. The two courses may teach the same competencies, but do so using very different course structures. In this case the two narrative identifiers may be added to the same candidate group.

5.2.6.2 Narrative Selection

The first step in creating a personalised course model is to select an appropriate narrative. For each course there may be several narratives available to achieve the same learning objectives. These narratives differ in the pedagogical approaches to structuring the concepts they describe. The selection of an appropriate narrative is based on the metadata describing that narrative and on the learner's metadata. For example, if the learner model has an `adaptivitytype` *preferredcoursestructure* with the value *abstract* a candidate selector may be used to select the narrative that matches this preference most closely. Narrative selection may also be required where there exists candidate sub-narratives to fulfil a candidate group included in another narrative.

5.2.6.3 Narrative Rules and Narrative Execution

The rule engine that processes the narrative rule sets is based on the Java Expert System Shell (JESS) [JESS]. JESS was chosen as the basis for the rule engine for a number of reasons –

- JESS is a flexible and expressive LISP-like language allowing the course author to develop simple courses or ones based on advanced AI adaptive techniques
- JESS can create and manipulate objects
- JESS is extensible, allowing custom functions to be created in Java and accessed from the JESS rules
- JESS is a Java-based engine, thus easing integration with other Java-based components of APeLS
- Several instances of the JESS rule engine may be created, enabling candidate selectors to be developed in JESS

These features, particularly the development of custom functions written for APeLS, give the course author great flexibility when creating the model of adaptivity they desire. The narrative rules are used to produce a course model for the personalised course. This model, created via the `dom` custom function, is a DOM (also available in XML) representation of the structure and sequence of the concepts or pagelets in the personalised course. It may be expressed in terms of either concepts or pagelets depending on whether the appropriate candidate selectors have been executed. If expressed in terms of concepts the candidate group and candidate selector identifiers are stored in the model to facilitate future resolution (possibly at runtime).

The narrative rules may also utilise sub-narrative rule sets. This is a feature of JESS that has been extended to comply with the candidacy paradigm. In a similar manner to how the concepts may be populated at runtime, sub-narrative execution may also be delayed to a later stage. In this way adaptivity may be provided just in time to fulfil the educational need, thus taking advantage of the most up to date learner model information.

5.2.6.4 Custom Narrative Functions

There are five custom functions accessible from the rules within APeLS, they are –

- Course model editing function
- Metadata model access function
- Metadata comparison function
- Compare candidate function
- Candidate selector execution function

Each custom function is implemented by extending the standard template for a function in JESS. Each new function may be called from JESS using the following syntax –

```
(function-name "parameter1" "parameter2" … "parameterN")
```

Each function call may return a single data type that is either a textual string, numeric value or multi-value field. New functions must be added to the instance of the rule engine so that they can be accessed.

Course model editing function

The course model editing function is used to change the course model based on the execution of rules within the narrative rule set. This function gives access to the DOM representing the course model and provides the course author with functionality to edit it. The basic syntax is –

```
(dom "command" "parameter1" "parameter2" … "parameterN")
```

The commands for the function are –

- `set-root`, this command sets the name of the root node of the DOM to that of its single parameter.
- `add-to-parent`, this command can accept one or two additional parameters. A single parameter indicates to add a node of the parameters name to the current parent node. Two parameters indicate add an element to the current parent of name `parameter1` with value `parameter2`.

- `set-parent`, this command has two modes, each accepting a single parameter. With a parameter value of ".." the function sets the parent to be the immediate ancestor of the current node. With a parameter value of a node name it sets the parent to be the closest predecessor matching that name to the current node.

For example, a (non-adaptive) narrative with the following function calls –

```
(dom "set-root" "course")
(dom "add-to-parent" "section")
(dom "add-to-parent" "identifier" "S-001")
(dom "add-to-parent" "pages")
(dom "add-to-parent" "pagelet" "P-001")
(dom "add-to-parent" "pagelet" "P-002")
(dom "add-to-parent" "pagelet" "P-003")
(dom "set-parent" "..")
(dom "set-parent" "..")
(dom "add-to-parent" "section")
(dom "add-to-parent" "identifier" "S-002")
(dom "add-to-parent" "pages")
(dom "add-to-parent" "pagelet" "P-004")
(dom "add-to-parent" "pagelet" "P-005")
(dom "add-to-parent" "pagelet" "P-006")
```

would create the following DOM structure (expressed as XML) –

```
<course>
	<section>
		<identifier>S-001</identifier>
		<pages>
			<pagelet>P-001</pagelet>
			<pagelet>P-002</pagelet>
			<pagelet>P-003</pagelet>
		</pages>
	</section>
	<section>
		<identifier>S-002</identifier>
		<pages>
			<pagelet>P-004</pagelet>
			<pagelet>P-005</pagelet>
			<pagelet>P-006</pagelet>
		</pages>
	</section>
</course>
```

Of course there are no rules above governing the conditions under which the pagelets get added to the course, but nonetheless the example illustrates the usage of the `dom` function.

Metadata model access function

Using this function metadata in any model in an APeLS repository may be accessed by the rules. The function is called `get-content` within the JESS rule engine and may accept one to three parameters, the final of which must be an XPath query. The other two optional parameters may specify the sub-collection and metadata document identifier on which the query is to be evaluated. An example call to the function may look like –

```
(get-content "subcollection2" "P-0034" "//*[name()='title']")
```

This function would return the string value of the title element value from the document identified as "P-0034" in "subcollection2" (which the resource bundle indicates is the pagelet repository). This function may be combined with the `dom` function to retrieve information from the repositories and add it directly to the course model –

```
(dom "add-to-parent" "name" (get-content "subcollection2" "P-0034"
"//*[name()='title']"))
```

This example would take the value returned by `get-content` and add it as the value of the name element to the current parent in the DOM. This function may be used to compare metadata from different metadata models –

```
(if    (eq

       (get-content "subcollection1" "oconlan"
"//*[name()='adaptivitytype'][@name='learningstyle.sequence']")

       (get-content "subcollection3" "N-002"
"//*[name()='adaptivitytype'][@name='learningstyle.sequence']")

       )
       then
              (#Do something)
       else
              (#Do something else)
   )
```

In the above example `get-content` is used to retrieve a value from a learner model (learner *oconlan* in subcollection1 – the learner repository) and a value within a narrative (narrative *N-002* in subcollection3 – the narrative metadata repository).

Metadata comparison function

The JESS code in the last example can be used to compare the values from metadata elements. A custom function has been implemented to enable the course author to represent common comparisons – those between a metadata document and current learner model. The function is called `compare-attrib` and accepts two parameters – the comparison name and document identifier. The comparison name is used to retrieve information about the comparison from the resource file. This information includes –

- The type of match required – single or multiple
- The sub-collection where the document is found
- XPath to the value(s) in the learner model
- XPath to the value(s) in the other document

For example –

```
(compare-attrib "competencies.learned" "P-0034")
```

would look into the resource file for APeLS looking for the comparison *competencies.learned*. This would return the following information –

```
competencies.learned=multiple subcollection2
//*[name()='adaptivitytype'][@name='competencies.learned']
//*[name()='adaptivitytype'][@name='competencies.required']
```

and would perform the comparison, returning a boolean result.

Compare candidate function

This function provides simple numerical caparison capabilities for comparing candidates. The function is called `compare-candidates` and accepts four parameters –

- Number comparison type – *high* or *low*
- List of candidate identifiers to compare
- Number of candidates to return
- XPath expression to the element to compare

This function enables the candidate selectors to compare many candidates based on numeric attributes, often expressed as adaptivitytypes. For example,

```
(compare-candidates "low" "P-0034 P-0067 P-0104" 1
"//*[name()='adaptivitytype'][@name='device.screenarea']")
```

would return the identifier of the candidate that had the smallest device.screenarea value. If more than one candidate was requested the function would return a space-separated list of the number requested in descending/ascending order depending on whether the comparison was *high* or *low*.

Candidate selector execution function

The final custom JESS function in APeLS is a function for calling a candidate selector. As candidate selectors are a set of rules, also written in JESS, they are executed in the rules engine. There are two modes under which a candidate selector may be executed – single mode or multiple mode. In single mode the candidate selector is asked to return a single result, i.e. a single candidate. In multiple mode the candidate selector is asked to return the top n candidates.

The candidate selector execution function called `select` can accept information about the candidates in two ways – either as a list of candidate identifiers or as an identifier to a candidate group. The most common way of passing this information is using the candidate group identifier. In multiple mode the function returns a comma separated list of the top candidates. This in turn may be used as the input to another candidate selector.

The final parameter accepted by the `select` function is the name of the candidate selector to use. This name is resolved using the resource file. For example,

```
(select "1" "CG-003" "vark")
```

calls the `select` function in single mode on the candidate group *CG-003* using the *vark* candidate selector. `select` functions may be called on the returned values of other `select` functions allowing the course author to compound multiple candidate selectors. For example,

```
(select "1" (select "4" "CG-004" "device") "vark")
```

calls `select` twice. The first (inner) call selects the 4 top ranking candidates according to the *device* candidate selector. The second chooses the best candidate from this group in accordance with the *vark* candidate selector.

5.2.7 Candidate Groups and Candidate Selectors

Candidate groups in APeLS are simply instances of metadata that list the identifiers of constituent candidates. Identifiers in APeLS are unique, even across separate metadata collections. The candidates of a candidate group do not have to be of the same type. For example, a candidate group may consist of pagelets and sub-narratives. For this group to be useful the pagelet metadata and sub-narrative metadata should contain elements that can be compared by a candidate selector. An example of such a candidate group might be one that contains candidates for different devices. The candidates for a device with a large screen might be regular pagelets while for smaller screen devices a sub-narrative may need to be executed.

In APeLS the candidate groups are separated into two repositories – candidate pagelet groups and candidate narrative groups. These are only logical collections as the actual candidate groups are equivalent in appearance and, as stated above, the constituent candidates of a group may be from any metadata collection. In general, however, candidate groups tend to be homogenous in composition, i.e. a candidate pagelet group generally only contains pagelets.

Candidate selectors in APeLS are implemented in JESS and utilise the metadata content retrieval and comparison custom functions. For example, a candidate selector for selecting between different candidates based on device screen area might look like –

```
;; *****************************
;; Declarations

(import java.util.StringTokenizer)
(import java.util.Vector)
```

```
;; *****************************
;; DEFGLOBALS

(defglobal ?*v* = 0)
(defglobal ?*s* = 0)
(defglobal ?*n* = 0)

;; *****************************
;; DEFFUNCTIONS
(deffunction is-identifier (?id)
  (bind ?*v* (new String))
  (if (= (?id indexOf ",") -1)
       then
              (bind ?*s* (new StringTokenizer ?id))
              (get-candidates ?id)
       else
              (bind ?*s* (new StringTokenizer ?id ", "))
              (parse-identifiers ?id)
  )
 )
(deffunction get-candidates (?id)
  (parse-identifiers (get-content "subcollection4" ?id
"//*[name()='candidates']"))
)

(deffunction parse-identifiers (?id)
  (while
       (= (?*s* hasMoreTokens) true)
   do
       (?*v* concat " ")
       (?*v* concat (?*s* nextToken))
  )
)
(defunction check-number (?num)
  (if (> ?num (?*v* size))
       then
              (bind ?*n* (?*v* size))
       else
              (bind ?*n* ?num)
  )
)
(deffunction device (?number ?id)
  (is-identifier ?id)
  (check-number ?number)
  (return (compare-candidates "low" ?*v* ?*n*
"//*[name()='adaptivitytype'][@name='device.screenarea']"))
)
```

The first function called in the candidate selector is *device* (the same name as the candidate selector) with the number of candidates required and either a comma separated list of candidate identifiers or the identifier of a candidate group. In this case the candidate selector performs a simple numerical comparison (looking for the lowest) of the candidates, using the `compare-candidates` custom function. The candidate selector example, above, illustrates how Java objects may be created and manipulated from within JESS code giving the course author access to all of the power and flexibility of the Java classes.

5.2.8 Course Rendering and Delivery

The output of a narrative execution is a personalised course model. This model is represented by a DOM which must pass through some form of rendering process to be viewed by the learner. A complete personalised course model does not include pagelet content, only the pagelet identifiers, so the rendering process must also include the appropriate content in the course. If there are any unresolved elements of the course model, such as candidate content groups or sub-narratives, the rendering process must call the rule engine to resolve these before rendering.

There are currently two instances of APeLS – a course on SQL and one on Physics / Mechanics, both utilising different approaches to rendering. The SQL course uses XSLT (see *APPENDIX III – Narratives and Style sheets - APeLS Style sheet for SQL course*) to transform the course model into multiple Java Server Pages (JSP) which embody the course structure. The JSPs use *server side includes* to add the pagelet content to the personalised course structure. Includes are also used to add appropriate header and footer information, including styling using Cascaded Style Sheets (CSS), to the pages. APeLS uses Jakarta Tomcat to deliver the personalised course. Using JSP the course model can add triggers into the course that can return information to APeLS. These triggers are used to update the learner model. In the SQL course the update procedure is triggered by the learner and they can give explicit feedback, producing an adaptable system.

The Physics / Mechanics course uses a different approach to rendering the course to that of the SQL course. There is only a single pre-authored JSP file in the course. This page integrates with APeLS to execute the rule engine. The rule engine returns a personalised course structure which is in turn passed through the XSLT processor. The processor returns a HTML table of contents for the personalised course which is added, using the JSP, to the rendering of the course. When the learner navigates this table of contents the content is added alongside the Table of Contents (TOC). After a learner has read the content the JSP (using the Learner Modeller component of APeLS) adds the competencies learned from that page to the learner model. On each page there is a *Rebuild TOC* button available to the learner, so at any stage they may update the table of contents to correspond to the latest learner model.

A new Cascaded Style Sheet may be developed for the personalised course model of each course. This enables course authors to have any look and feel they desire to be associated with each course. One of the elements of the personalised course model may define the style sheet to be used. This element may have been populated by referencing the learner's preferences in the learner model.

5.3 The APeLS Cycle – Creating a Personalised Course

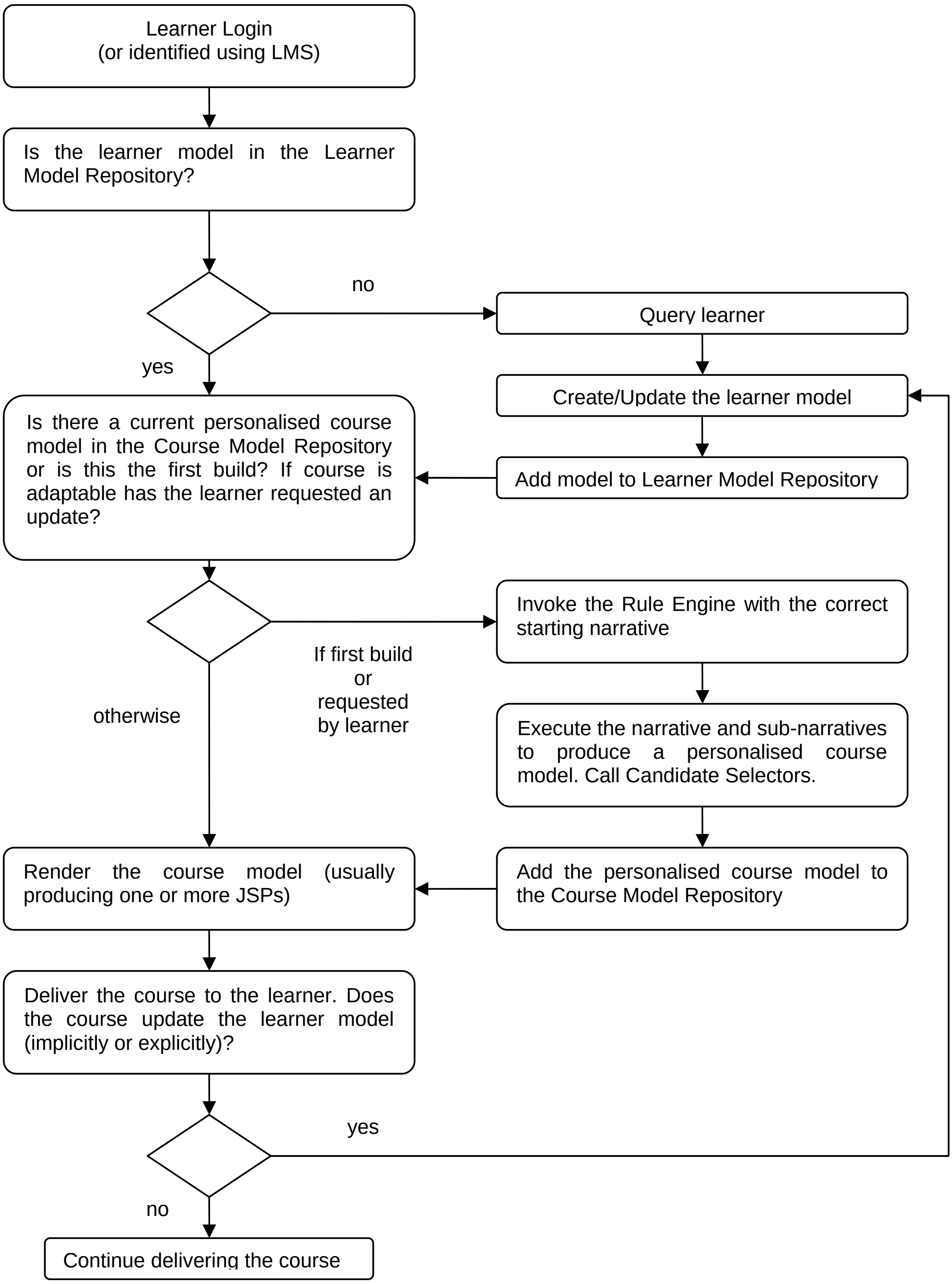


Figure 5.3, APeLS Personalised Course Creation Process

Figure 5.3 shows a basic view of the cycle APeLS follows when creating and updating personalised courses. The figure shows the cycle for an adaptable personalised course where the learner instigates the personalisation process, though the personalised course may combine adaptive and adaptable methods.

The personalised course model stored in the Course Model Repository does not need to be fully processed before the personalised course is delivered. For example, some candidate groups or sub-narratives may not need to be resolved until delivery time, at which point a candidate selector or the rule engine is invoked.

5.4 APeLS as a Service

APeLS has been designed to integrate with open Virtual Learning Environments (VLE) that support the content interworking API, compliant with the ADL SCORM data model, and that utilise any information about the learner they may have. For successful integration with a VLE there are a number of issues which must be addressed –

- APeLS, unlike regular static learning content cannot be imported directly into the VLE content repository. It, therefore, must be launched from its remote location. The VLE needs to know this location.
- When launching APeLS the VLE must identify itself as a Virtual Learning Environment supporting the Content Interworking API to allow APeLS to utilise that functionality.
- Both APeLS and the VLE must understand the same data model for communication.

The service architecture used in APeLS complies with that described in Section *4.9 Service Architecture*.

5.4.1 Metadata Describing the Service

The metadata structure described in Section *4.4 Metadata for Describing Adaptivity* may be used not only to describe the constituent models of an Adaptive Service, but may also be used to describe the service as a whole. For example, the metadata may describe the various prerequisite knowledge sets that the learner may require to understand the service. This is useful as Adaptive Services are, by their nature, not static and cannot be fully described by standard learning resource metadata. Using the `adaptivity` metadata sub-section developed as part of this thesis the metadata author may describe the possible start conditions and learning outcomes of the service.

By allowing the full description of the service, and its facilities, the metadata increases the potential reuse of the service, as anyone searching for a learning resource similar to an adaptive service will be able to determine if it fulfils their requirements. Using the `adaptivitytype` elements the metadata author may also describe any requirements the service may have. These requirements may be analysed and fulfilled automatically when the service location is imported into a Virtual Learning Environment.

When integrating the `adaptivity` metadata sub-section with specifications, such as IMS Learning Resource Metadata there are some limitations in describing Adaptive Services. For example, if an adaptive service can produce many potential learning courses what set of keywords should be used in the metadata to describe the service? This problem arises with many of the standard elements in learning resource metadata, e.g. title, description, keywords, etc.

There are a range of possible solutions. One is to use a single metadata record to describe all potential courses, i.e. the keywords would contain all possible keywords describing the different courses. This solution raises the problem that in a search scenario the metadata record becomes a searching wildcard. As the metadata fields describe many potential courses they may become overpopulated and any search tenuously related to the potential adaptive courses will return the adaptive service as a possibility. Another solution is to have several metadata records describing different courses achievable within the Adaptive Service. The metadata author may choose to describe the most popular potential courses in the Adaptive Service.

Both of these solutions would use the same sets of `adaptivity` metadata. This metadata describes all the possible variations of the system, while the static metadata describes either all variations or a single variant.

5.4.2 Importing the location of APeLS into a Virtual Learning Environment

IMS Content Packaging is a specification that provides mechanisms for static content to be imported and integrated into a Virtual Learning Environment. By importing content as part of an IMS Content Packaging archive, with an XML manifest included, content can be automatically installed into a VLEs content repository.

With APeLS the learning content is not shipped in discrete units of material. APeLS is a remote service that the VLE has a remote entry point to. As the VLE, such as Microsoft's LRN [Microsoft LRN], may already have an IMS Content Packaging import facility a reasonable approach would be to utilise this and enhance it to allow for the *importing* of the location of APeLS as part of an IMS Content Package. This could be imported into a VLE in a similar way to static content. The manifest would describe the adaptive service and also contain a URL that the VLE would use to launch the service.

5.4.3 Launching APeLS from a Virtual Learning Environment

When APeLS is launched from a Virtual Learning Environment the content it delivers will be coming from a different server than the Virtual Learning Environment. APeLS firstly initialises its connection with the VLE and then determines if the learner has been engaged in this course component before. If they have then it can continue the personalised course component for the learner, using the course model stored in the Course Model Repository to continue delivery.

If the learner has not accessed this adaptive course component before then APeLS must build a new personalised course for the learner. It may be necessary to pre-test the learner for prior-knowledge and learning style preferences to populate their learner model.

5.4.4 Communication between the AHS and the VLE

Communication between the Adaptive Hypermedia Service and the Virtual Learning Environment can be achieved using the SCORM Runtime Communication API as used in SCORM v1.1. A subtle modification to the HTML frame layout used by APeLS is required to enable calls to API functions residing on the VLE from content in APeLS.

The actual API calls used are the same as those used in SCORM v1.1 as the API is designed to get and set values that are separately defined by an external data model [SCORM].

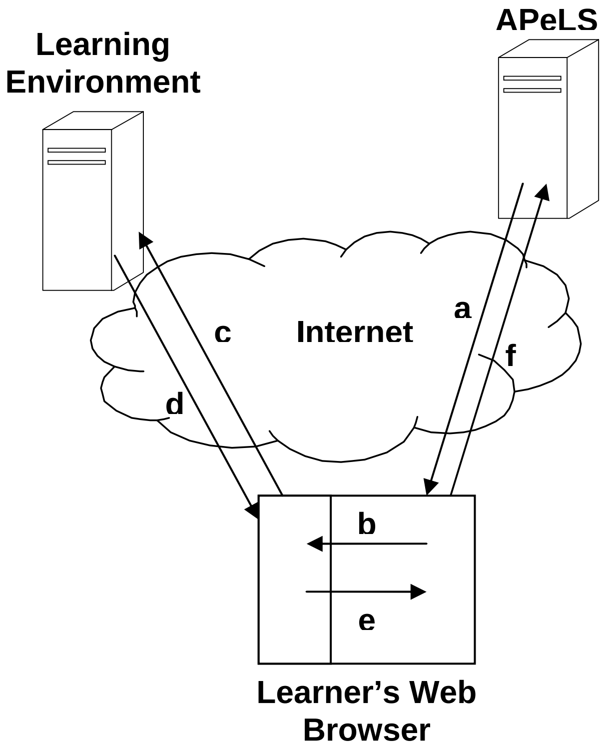


Figure 5.4, APeLS accessing the VLE data model

APeLS calls the Content Interworking API to access the data model on a VLE using the following process –

a) The learning content (right browser frame) and JavaScript API (left browser frame, hidden) are delivered to the learner's browser from APeLS
b) An API function, in the left hand API frame, (e.g. LMSGetValue("cmi.core.lesson_status")) is called from the content frame
c) The API frame communicates the request to the Virtual Learning Environment
d) The Virtual Learning Environment returns the value from the data model (in this case of cmi.core.lesson_status) to the API Frame
e) The function returns the value to the content frame
f) The value may be passed back to APeLS using a JSP method

5.4.5 Common Data Model

The data model required for APeLS/VLE interaction is similar to that proposed in SCORM v1.1. The key requirements of the data model for this interaction are –

- Virtual Learning Environment identifier (to uniquely identify the VLE, possibly it's URL)
- Learner identifier (to uniquely identify a learner from a VLE)
- Section identifier (to determine, using the elements above, whether the learner has attempted this content in APeLS before)
- Section status (completed, incomplete, attempted etc.)
- Score information (across all sections, for the learner and averages for all enrolled learners)
- Score range (the VLE may not use percentile scoring and APeLS may need to normalize the scores)

The number of SCORM data model elements has been significantly reduced between version 1.0 and version 1.1. This reduction was to aid harmonisation between independent developments of the Content Interworking API, i.e. many groups were using the same API, but had subtly different data models. Many of the optional elements defined in SCORM v1.0 had not been implemented by other consortia.

Most of the required data model elements for APeLS/VLE interaction already exist in some guise in the CMI data model used in SCORM v1.1. Learner identifier, section identifier, section status, and score range all have equivalents. The VLE identifier and average score information is not available within the data model and are additions to support the adaptive service architecture.

5.4.6 APeLS Integration with a Commercial VLE

The mechanism for interfacing VLEs and adaptive services described in this section have been implemented using Fretwell-Downing Education's VLE [Fretwell-Downing Education] and APeLS. The data model used is based on SCORM v1.1 [SCORM] with the addition of VLE identifier and average score elements.

Using this model it is possible for APeLS and a VLE to interact, sharing information about the learner including, but not limited to – Learner Identification, Interface Preferences, Pedagogical Preferences and Performance Information.

The first piece of information that is requested by APeLS from the VLE is the VLE identifier. This should be unique, possibly URI based, and allows the service to contextualise all further communications. For example, when it asks for the learner's identification it might assume that this identifier is unique within the VLE, but not across other Virtual Learning Environments.

Learner identification enables APeLS to ascertain if the learner has visited the service before and if they have, determine whether it is necessary to pre-test them. For example, if the learner completed another course within the service already then it may have pedagogical information that is sufficient to adapt to the learner's preferences. If APeLS has no information, or no appropriate information about the learner then it may be necessary to use a pre-test instrument, depending on the nature of the course.

Prior to administering the pre-test the service can query the VLE for assessment scores for any material the learner has attempted in the course (assuming APeLS is used as a component of a larger course) and also request the assessment scores for the learner's peers in the course. These values are returned as normalised values in a range specified by the VLE. Using this information APeLS can determine the difficulty at which to set the pre-test, if it has multiple pre-tests aimed at differing levels of knowledge. Such a knowledge pre-test can be used to determine the prior knowledge about a domain before generating the personalised course.

APeLS can return assessment information to the VLE using the normalised range mentioned above. This allows APeLS to inform the VLE how the learner has performed in any assessments they have completed, but this is only a summary score for all assessments completed in the service. The IMS Question and Test specification was investigated as a candidate for richer communication of learner assessment results, but the uncertainty as to whether any given individual assessment would be present in a personalised course meant the information for different learners may be inconsistent and difficult to draw conclusions about.

5.5 *Instances of APeLS*

To complement the implementation of APeLS two adaptive courses were developed, one on Structured Query Language (SQL) and one on Physics / Mechanics, in accordance with the multi-model, metadata driven approach.

5.5.1 Personalised Course on SQL

The adaptive course on SQL is an adaptable, competencies-based course. It is comprised of a main narrative, thirty sub-narratives and uses learning content from a repository of over 330 fine-grained pagelets. The narrative utilises a tutor-developed competency vocabulary (see *APPENDIX II – Vocabularies - Knowledge Domain Vocabulary for SQL*) of 92 terms.

When a learner enters the adaptive SQL course for the first time they are presented with an instrument for querying prior knowledge/experience in SQL. This simple questionnaire comprises seven increasingly difficult questions asking the learner's experience in SQL. Each question maps to many competencies from the SQL vocabulary. When a learner answers that they know a particular concept this is added to their learner model as a *competencies.learned* adaptivity type.

When a personalised course is being constructed the narrative and sub-narratives look at the learner model and add content for concepts that the learner has not learned. At any stage the learner may revisit the prior knowledge instrument and change the answers, thus initiating the construction of another personalised course offering.

In this instance the scope of adaptivity is only limited by the number of questions used to query the learner. With 92 competencies in the vocabulary and over 330 pagelets there is the possibility to provide very fine-grained adaptive effects based on competencies.

Within the course there also exists a number of candidate content groups representing pagelets of different media types that support different learning preferences. These pagelets are differentiated using VARK. This extension to the original SQL course enables the learner's preference for different forms of content to be catered for [Bruen, 02]. Candidate content groups, the abstraction mechanism and candidate selectors were employed to achieve this additional axis of adaptivity.

As part of another extension to the personalised SQL course, the Kolb/McCarthy Learning Style Inventory was used to differentiate content, in accordance with the inventory's separation of learning style into two continuums [Clarke et al., 03]. Again candidate content groups and the abstraction mechanism were employed to separate narrative from content. Appropriate candidate selectors were developed to select the most effective learning content for the learner based on the Kolb/McCarthy inventory.

The final extension to the personalised SQL course carried out as part of this thesis was to facilitate the selection of Learning Objects appropriate to the device the learner is using. For example, if the learner is learning using a PDA then LOs that may be displayed on a device with low screen real estate would be selected. Unlike the previous two extensions the candidate selector in this case did not use the learner model as a basis for the decision. Instead it used a separate environment model.

5.5.2 Personalised Course on Physics / Mechanics

The Physics / Mechanics course, originally developed by the Department of Psychology at the University of Graz [UoG] and based on the Theory of Knowledge Spaces [Albert and Lukas, 99], was adapted to work with APeLS. The course comprises just under 200 coarse grained (page size) pagelets with associated IMS Learning Resource Metadata and is described by a tutor defined competencies vocabulary of 76 terms (see *APPENDIX II – Vocabularies - Knowledge Domain Vocabulary for Physics / Mechanics*).

The aim in using and adapting this content for APeLS was to determine what a course author had to do to produce an adaptive course based on pre-existing content and metadata. The course developed is a competencies based adaptive course. Learners are given a knowledge pre-test that is used to initially populate their learner model competencies. The relationship between these learned competencies and content is dictated by the theory of knowledge spaces [Albert and Lukas, 1999]. Each piece of content is assigned *competencies.required* and *competencies.taught*, represented using the adaptivity metadata extension. The learner model's learned competencies must satisfy the required competencies of a piece of content for it to be eligible for delivery. Similarly, when a learner has completed a piece of content the competencies it teaches are added to the learner's learned competencies. At any stage the learner may rebuild the table of contents for the course, which will grow as they learn more competencies and more of the course content is made available.

The Physics / Mechanics course was extended with mechanics learning material from the Open University [OU]. The goal of this extension was to investigate the flexibility of the multi-model approach to increase the scope of an existing course. The material from the OU was added by extending the number of concepts in the knowledge space. These additional learning objects were added by extending their metadata to include these additional concepts. The new material shared some prerequisite competencies with those taught by the existing material. When a learner browsed to existing material that completed these prerequisite relationships, the new material became available to them. The navigation structure they were placed in (i.e. the Table of Contents) was extended by using a sub-narrative.

5.6 Summary

This chapter discussed an implementation of the multi-model approach called the Adaptive Personalised eLearning Service (APeLS). Two architectural models of APeLS have been presented, component and technical, for the multi-model approach. These were presented along with the primary constituent models of APeLS (learner, pagelet and narrative) described in depth. The technical approach to the candidacy architecture, rendering and delivery issues were also described in this chapter. The APeLS cycle for personalised course creation was presented, illustrating the operation of APeLS. The implementation of the service architecture used by APeLS was also described. An integration with a commercial learning environment was also detailed. The chapter concluded with a description of two adaptive courses created to run on APeLS covering the topics of SQL and Physics / Mechanics.

6 TRIAL AND EVALUATION

6.1 Introduction

The research question posed in this thesis is whether it is feasible and beneficial to separate the elements of personalised eLearning, such as the content, sequencing logic and the learner model. The feasibility of separating the elements of adaptivity may be evaluated through examining the reusability of these elements. It may also be assessed through evaluating the flexibility and extensibility of the multi-model, metadata driven approach. These are primarily technical concerns that reflect the generic implementation of the Adaptive Engine at the core of APeLS. These technical concerns are what differentiate the multi-model, metadata driven approach, and APeLS, from current Adaptive Hypermedia Systems. However, these differences are of little value if the personalised eLearning courses produced using APeLS do not satisfy the learners who engage in the courses and satisfy the tutors who create the courses.

The benefits of the multi-model, metadata driven approach for the primary actors involved in eLearning may be determined by examining the satisfaction of learners and authors. The evaluation of learner satisfaction asks if personalised courses developed using APeLS are pedagogically appropriate, coherent and effective. This part of the evaluation was carried out by presenting students with a paper-based questionnaire to determine how they used the personalised courses and their level of satisfaction with different features of the courses. The *Student Satisfaction* section of this chapter deals with this part of the trial and evaluation.

The evaluation of author satisfaction examines how the multi-model, metadata driven approach has impacted the authors of eLearning and the process of creating personalised eLearning offerings. The tutors' satisfaction with the multi-model approach was determined through interviewing the tutors who created personalised courses using the approach. The *Tutor Satisfaction* section of this chapter describes the tutor trial and evaluation of their experiences with APeLS.

The feasibility of the multi-model approach may be determined by examining the technical advantages brought about through separating the elements of personalised eLearning. This part of the trial examines the separation and reusability of the elements of adaptivity. The separation and associated capacity for reuse is evaluated in the *Separation and Reusability of Elements of Adaptivity* section. This section examines the separation and reusability issues of each of the three core models – Narrative, Content and Learner.

The evaluation of the technical advantages of the multi-model approach continues with an examination of the flexibility and extensibility of the approach. This is achieved by evaluating the different forms of adaptive effects offered by the different instances of personalized courses offered through APeLS. By evaluating the range of different adaptive effects, and the features of the multi-model approach that facilitate them, the extensibility and flexibility of the approach may be examined. As part of this evaluation the personalised SQL courses with learning style extensions for VARK [Bruen and Conlan, 02] and Kolb/McCarthy [Clarke et al., 03] were examined. The *Flexibility and Extensibility of Adaptive Effects* section describes this part of the trial and evaluation.

Reusability is a key goal of this thesis, and while the *Separation and Reusability of Elements of Adaptivity* section evaluates reusability from the perspective of the models used in the approach, the *Integrating APeLS with a VLE* section examines the potential reusability of the adaptive eLearning service as a whole, through its integration with an existing Virtual Learning Environment. This evaluation is carried out through the examination of the level of integration possible between APeLS and a VLE. This chapter concludes with a summary of the key findings of the evaluation.

6.2 Student Satisfaction

Learners are the most significant stakeholder in eLearning. Therefore, any approach to personalised eLearning, such as the multi-model, metadata driven approach, must produce tailored eLearning offerings that are satisfactory to the learner. This satisfaction may be evaluated by examining how coherent the learners felt the courses were, how appropriate the courses were to their objectives and how pedagogically appropriate they were. This section describes the learner trial and evaluation. It begins with an overview of the trial, continues by examining student responses to various questionnaire questions, provides an examination of student performance and concludes with a summary of the key findings from the evaluation.

6.2.1 Overview

In 1998 and 1999, a non-adaptive SQL (Structured Query Language) online resource [Wade and Power, 98b], also developed by Trinity College, Dublin (TCD), was blended into the teaching of the database courses for the *Computer Engineering*, *Computer Science* and *Computer Science Language and Linguistics* degree programmes. This non-adaptive resource offered students a structured online course in SQL. It also included supplementary features such as a live database that could be queried via a web browser interface, a detailed case study that formed the basis of many of the examples found in the course, and a project reference section. The project reference section detailed the assessment that the students must complete as part of their respective courses. For each course this involved an assignment in which every student created their own database using the SQL they learned through the personalised course. The assignment was the same for all students.

Using the principles described in the multi-model, metadata driven approach this non-adaptive SQL course was chosen as the basis of a personalised eLearning course in SQL. In TCD, SQL is taught as part of seven different degree programmes across ten different courses. Each course differs based on the degree focus and ethos, objectives, student experience and complexity. A goal of the personalised SQL course is to facilitate the teaching of SQL as part of these diverse courses, supporting different objects, pedagogical styles, prior knowledge and academic scope. It is important to note that the curriculum of the course did not change during the evaluation of the APeLS-based personalised eLearning service and that the content used in the personalised service is based directly on that of the original non-adaptive course, although the content from the original course was *chunked* into fine grained pagelets and external references were removed from the content. Evaluations of the personalised SQL course have been carried out over the past four years (2000 – 2003, inclusive).

Junior and Senior Sophister (3rd and 4th year) students in the *Computer Engineering*, *Computer Science*, *Computer Science Language and Linguistics* and *Information and*

Communication Technology degree courses in TCD utilise the personalised SQL (Structured Query Language) course as part of their studies on Database Management and Database Design. Each of these courses have different subject emphasis and different lecturers responsible for teaching the courses. Like many computer science departments, the Department of Computer Science in TCD teaches databases, and SQL in particular, at several levels within different degree programmes. The student evaluation was carried out on over 500 students and this chapter attempts to determine the usability of the personalised SQL course, in particular focusing on learner satisfaction and the effectiveness of the service.

Each of the students was presented with the same paper-based questionnaire (see *APPENDIX VI – Student Evaluation Questionnaire*) giving them an opportunity to evaluate the personalised SQL course they had just completed. The questionnaire comprised thirty one, principally multiple choice, questions and an open comments section. Each question also had space for the student to elaborate on their answers. The questions were divided into a number of different areas, although these sections were not made explicit on the questionnaire they are used for discursive purposes. Through these areas it is hoped to examine student satisfaction with using the personalised eLearning provided by the SQL course. The areas of the questionnaire, and the information they are trying to elicit, are –

- Student's previous experience with online learning resources and SQL; this area examines how experienced students are with online learning and how amenable they are to using online resources for learning.
- Student's usage of the personalisation tool; this area examines how the students used the personalisation instrument offered.
- Student's views on the appropriateness of the personalised courses; examined in this area of the questionnaire is the student's satisfaction with the personalised courses they are presented with.
- Student's usage and views of the personalised course, both in terms of navigation and learning content; this area examines the usability of the personalised courses.

- Student's usage of the additional features provided and overall performance of the personalised courses; examined in this final area of the questionnaire is how students used the additional features offered as part of the online SQL course. Also examined is how the students perceived the technical performance of the personalised courses.

This section looks at the responses of students in these five areas. The final subsection will look at the students' examination performance and use this as an indicator of the educational effectiveness of the personalised SQL course.

6.2.2 Previous Experience of Online Learning and SQL

The goal of capturing this information was to determine the students' level of comfort with using online learning resources as part of a formal curriculum. It was felt that this was an important metric upon which to evaluate the students' satisfaction with the personalised courses. For example, if the students' stated that they were uncomfortable with receiving educational material via the Web then this may be used as a context in which to view their answers to questions about their use of the personalised course.

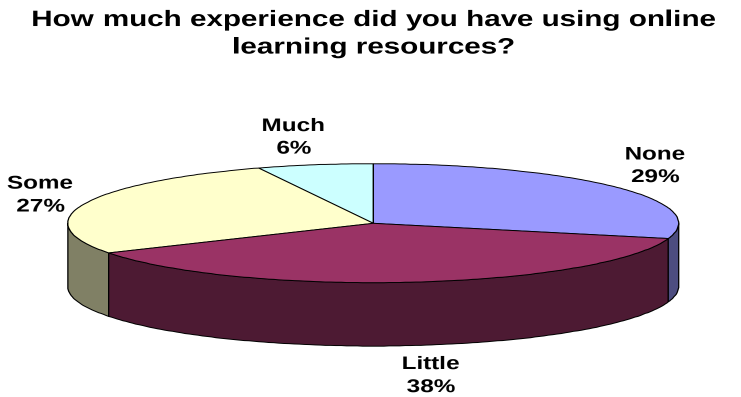


Figure 6.1, Experience using online learning resources

As may be seen in Figure 6.1, 67% of the students considered themselves to have had little or no experience using online learning resources. The students were also asked if they felt they would be comfortable with learning new material from the web (figure not reproduced). Less than 25% replied that the felt they would not be comfortable with learning material this way. This percentage may be higher in classes that are not so inherently IT enabled.

These initial questions indicate that while our students had not received much formal education online they were certainly receptive to do so.

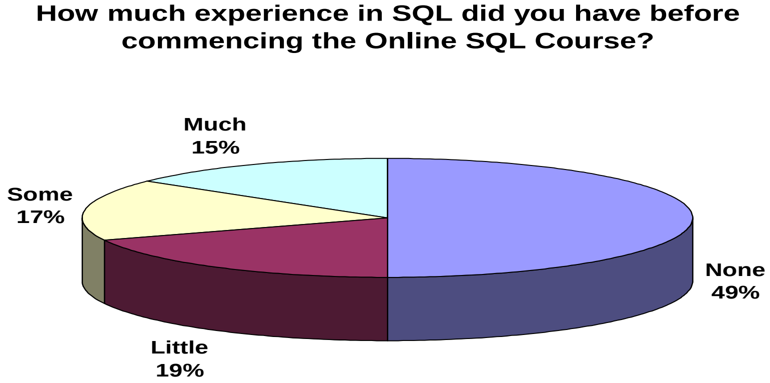


Figure 6.2, Experience of SQL

By the senior years of a degree course in the broad area of Computing there is a high probability that students, despite never having been formally taught it, will have some experience of SQL. For example, this experience may have come from a general interest in the subject matter or from summer work experience. As may be seen in Figure 6.2, just over 30% of the students considered themselves to have some or much experience of SQL, while nearly 50% had none.

Discussion and Findings

These results seem to indicate that the students' were willing to learn new material via the Web, although they had little experience of doing so and generally had little experience of the material. This finding impacts any future findings relating to the students' satisfaction in using the personalised courses. A less IT literate student body may not be as receptive to using online learning resources. Even with this IT literate student body just under 25% said that they would not be comfortable learning material via the web. The main comment of these students was that they preferred reading printed material than reading from a screen. This finding may point to a need for more interactive content, thus alleviating the amount of basic reading the students engage in.

The decision to base the personalised courses in the subject area of SQL seems justified as there are diverse levels of experience among the students. This is important due to the fact the primary adaptive axis for the personalised SQL course is prior knowledge. If the students using the personalised course all had the same previous experience of SQL then the personalisation may have been less effective.

6.2.3 Usage of the Personalisation Tool

The questions asked in this area were to capture the students' experiences using the personalisation instrument to rebuild personalised courses. These questions attempt to capture their satisfaction with using this instrument and the courses produced.

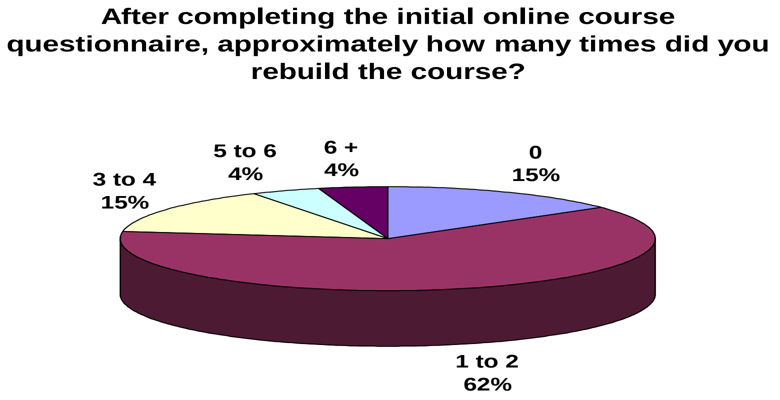


Figure 6.3, Course rebuilds

Upon first entering the personalised SQL course, every student was required to complete the personalisation instrument (referred to as the *Online Course Questionnaire* in Figure 6.3), which was composed of seven questions relating to their prior experience of SQL. The personalisation instrument is the mechanism used by the students to personalise their SQL courses. The seven questions posed in the instrument map into more than ninety concepts taught in the course, so in a sense the instrument provides a very broad and convenient simplification of the underlying concept structure. It was felt that it would be unwieldy to present the students with a large list of concepts, which they may not be familiar with.

Figure 6.3, above, shows that the majority of the students' only used this instrument once or twice after the initial building of their personalised course. Of the 23% who rebuilt the course more than three times, many stated that they did so to build courses that were *bite sized* and could be tackled in a short period of time. For example, one student stated that he liked to use the course during his lunch hour and deliberately altered how he represented his prior knowledge of SQL in the instrument in order to produce short courses. This enabled him to complete the course over about 5 lunch times (as there is approximately 5 hours of content in the course).

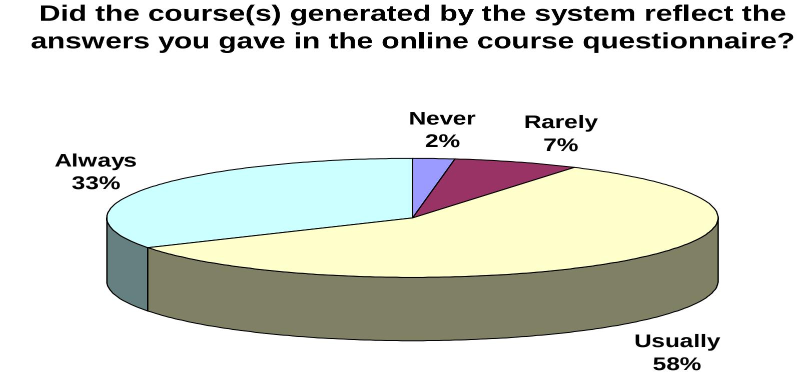


Figure 6.4, Courses reflecting questionnaire answers

This form of *playing the system* was possible as the students had confidence in the instrument. This is also shown in Figure 6.4, where 91% of the students felt that the personalised courses produced after completing the instrument always or usually reflected the answers they had given. The other ten percent may be accounted for in the results represented in Figure 6.5, below.

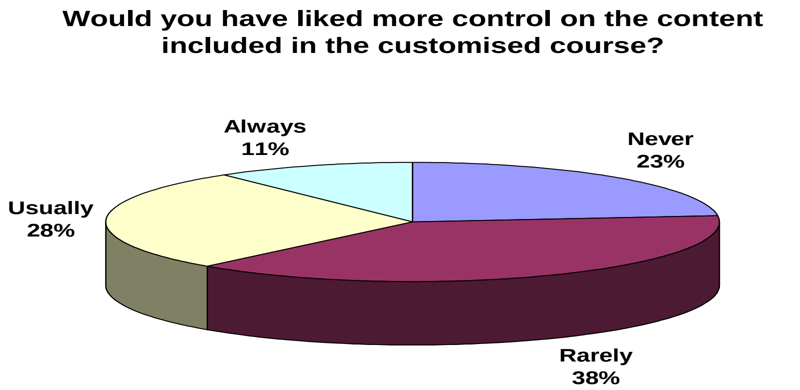


Figure 6.5, Control over customised courses

Here approximately 40% of the students stated that they would like more control over the personalisation process and the content included as a result of this process. This again may be accounted for by the IT literate nature of the students and their desire for more control. On the benefits of rebuilding the course, over 60% of the students found the ability to modify the amount of content in the personalised courses useful and almost 75% found the ability to adjust the scope (number of sections) of their courses useful. 80% of the students also found the maintenance of a familiar link structure across the personalised course important. Many commented that the ability to bookmark this consistent link structure was useful.

Discussion and Findings

The results of these questions indicate that many of the students like to take responsibility for their learning process and become more active *self-learners*, although many of them rebuilt the adaptive course only a few times. This may be attributed to the fact that the students appear to have been satisfied with the personalised courses produced using the instrument. This gave many students a high degree of confidence in the courses produced, and believed the courses were suitable for them.

A very small minority of students (9%) did not feel that the courses generated reflected the answers they provided in the personalisation instrument. These students generally also answered that they would like more control over the content in the personalised courses. A large proportion of these students were also in the 15% of students who had significant prior experience of SQL. These students wished for more control over a subject matter with which they were familiar.

6.2.4 Appropriateness of the Personalised Courses

The questions in this area of the questionnaire were used to determine if the personalised courses produced met with the students' expectations and that the objectives of the courses were clear to them.

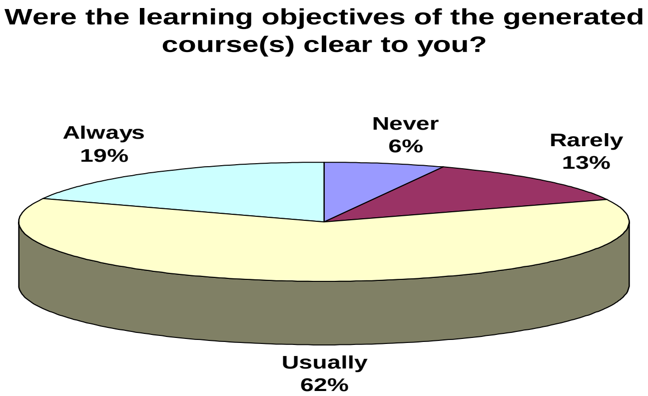


Figure 6.6, Clarity of learning objectives

Figure 6.6 shows that more than 80% of the students felt that the objectives of the personalised courses were clear. This is also supported by Figure 6.7 showing that the majority of the students felt the courses generated reflected their own objectives.

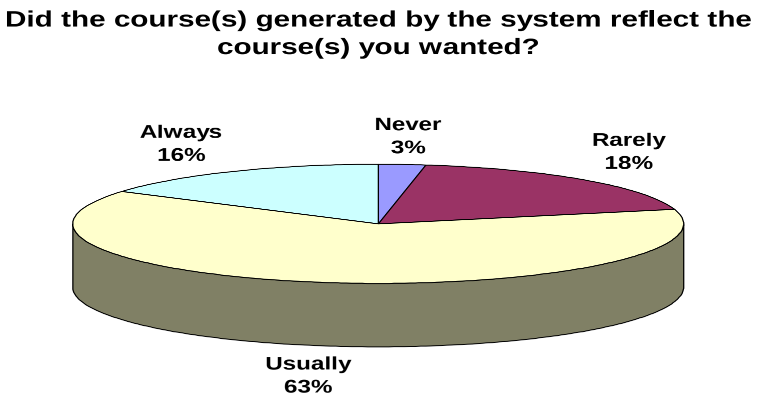


Figure 6.7, Suitability of courses generated

When asked, however, if they felt they had completed these objectives upon completion of the online course, 67% answered yes, they felt they had, and 33% answered no (Figure 6.8). This is clearly not as satisfactory as one would desire. This figure illustrates that a significant percentage of the students either did not have a good sense of the objectives of the course or that the course did not fully satisfy their own objectives.

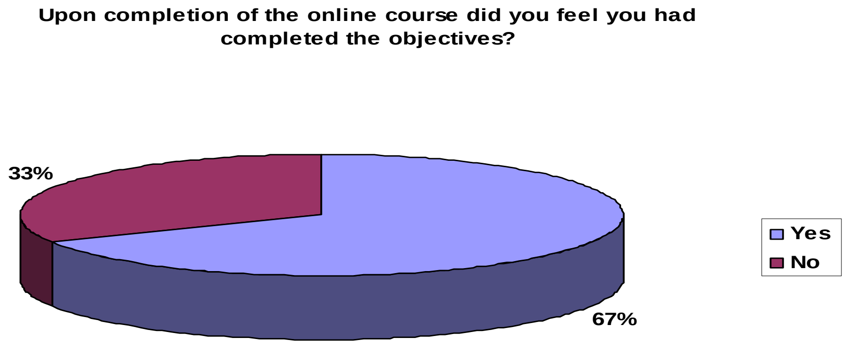


Figure 6.8, Completion of Objectives

Discussion and Findings

The results from these questions seem to indicate that the students found the personalised courses to be suitable for them, but that some did not have a clear sense of achieving all of the objectives of the course. The suitability of the course may be attributed to the fact that each course is aligned with the assignment the students must complete, i.e. the assignment and personalised courses have complementary objectives.

While the majority of students found the courses suitable there is approximately 20% that did not find the objectives of the courses clear, nor felt the personalised courses reflected what they wanted. This would seem to indicate that there is divergence between what these students expected from the course and what was available to them. This may be addressed by a broader body of content, possibly open corpus, and a more detailed, alternative personalisation instrument.

The proportion of students who felt they had not completed all of the course objectives is also a telling result. Despite maintaining consistency of navigation and layout between different personalised courses many students did not get a good spatial orientation of the courses and the concepts they contain. This may be remedied by providing visualisation tools that represent the concept domain ontology of the current personalised course in the context of all possible concepts that may be learned. In this way students would be less likely to feel that they were missing something useful because of the personalisation process.

6.2.5 Satisfaction with Navigation, Learning Content and User Interface

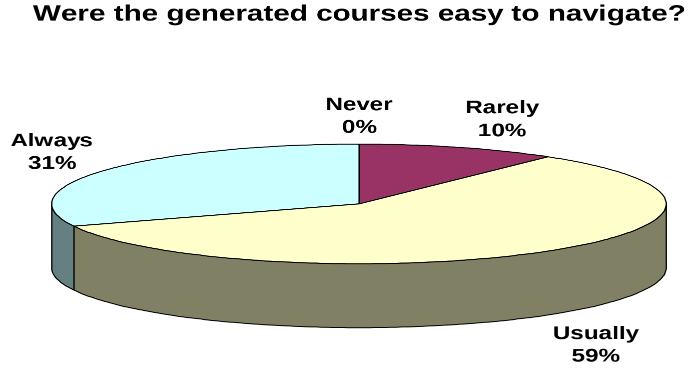


Figure 6.9, Ease of navigation

This area of the questionnaire attempts to determine the students' usage of the navigation features, learning content and the user interface. These usability aspects are important in determining overall student satisfaction with the learning experience presented. Figure 6.9, above, indicates that the vast majority of students found the navigation to be appropriate to their needs.

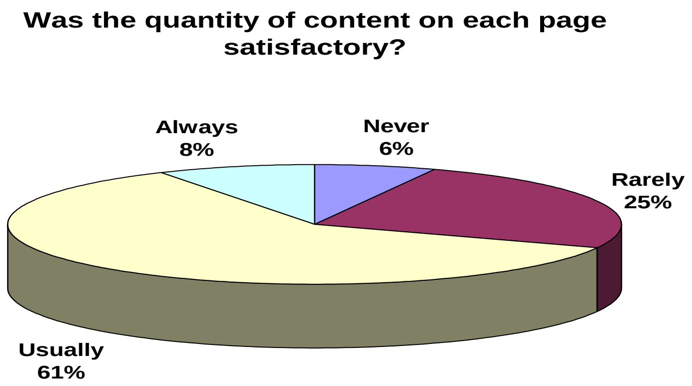


Figure 6.10, Satisfactory level of content

When asked if the content of the personalised courses ever appeared to be disjointed, approximately 80% responded that it rarely or never did (figure not reproduced). The comments supplied by the other 20% of students indicated that the navigation structure was effective, but that sometimes the short content pages made it difficult to read the structure of the textual content fluidly. The alternative was to present longer pages, yet when asked about the quantity of content on each page 69% of students were satisfied, as Figure 6.10 shows.

The quantity of content on each page was not always fully satisfactory for 31% of the students and this should be considered in the future when evaluating the size of the pagelets. With regard to the placement of content in the overall navigation structure, 92% of students said that the content appeared where they expected it to be in the course structure. Similarly, when asked if they would like to be able to reposition content into different sections, the students seemed happy with the structure of the navigation of course, as Figure 6.11 shows.

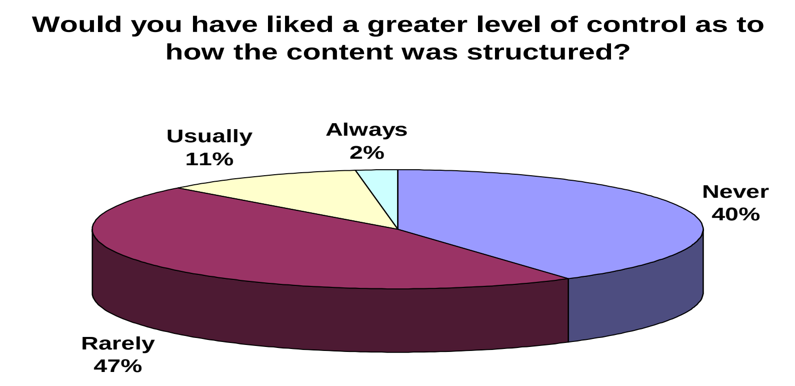


Figure 6.11, Structure of content

Somewhat surprisingly, when asked if they would like more control over the user interface, i.e. placement of buttons, number and type of hyperlinks, the students overwhelmingly replied that they did not want more control, which may be seen below.

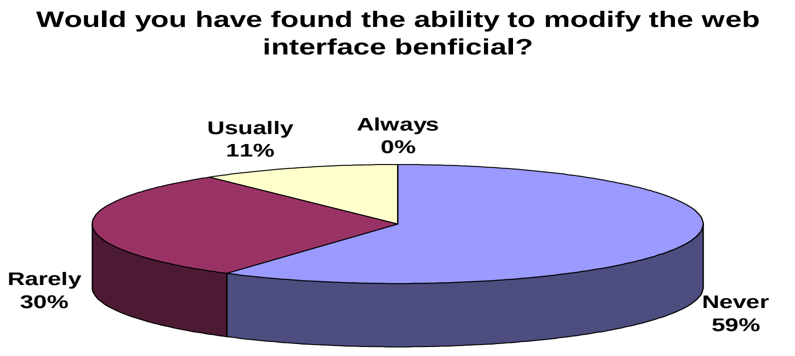


Figure 6.12, Ability to modify web interface

Discussion and Findings

Overall satisfaction with the usability of the courses, namely navigation, learning content and user interface, was high amongst the students. It was expected that more students would like control over the user interface, given the ability to *skin* many desktop applications. Most students, however, liked the simple and effective interface. It provided multiple paradigms for navigation, including both verbose textual links and basic hierarchical navigation buttons. It also maintained the consistent structure embodied by the personalised courses.

As the personalised SQL courses were presented as part of a fixed curriculum the students did not want a greater level of control over the content as it was aligned with the assignments and examinations they had to complete. In a more open learning system the learners may prefer more control over the structuring of the learning content to suit their own objectives.

The quantity of content on each page is an issue with approximately 30% of students. These students felt that there was too little content on some of the pages, i.e. the pagelets were *too* fine grained. This fine granularity also led many of the students (20%) to comment that the content sometimes felt disjoint. This finding should lead to a closer look at the size of pagelets and the conceptual flow between them.

6.2.6 Usage of the Additional Features and Performance of the Personalised Course

The additional features include the ability to query an example database, an example project solution and a case study. In general the students found the case study and example more useful than the live database. All of these features, however, scored highly. This illustrates the desire of these students to have many *streams* of input from which to choose and learn. It also demonstrates how an appropriate assignment for the students to undertake after the course has been completed provides inherent motivation. It should be noted that while the SQL courses were personalised the assignments were not. This motivation is compounded by both the case study and the example project solution as it helps the students to visualise what they are required to do.

With respect to the performance of the personalised SQL course this is an area that approximately 65% of students felt was poor. The rebuild time for producing a new personalised course was in the order of 40 seconds and this was, unsurprisingly, unacceptable to most of the students.

Discussion and Findings

The poor speed performance of the personalisation process is a significant finding of the student evaluation. The students were reluctant to reuse the personalisation instrument because it took too long. This may account for the relatively few times students rebuilt their personalised courses. This performance issue was addressed in later versions of the personalised SQL course. The personalised courses now have an average rebuild time of less than one second.

6.2.7 Student Performance

As part of the evaluation of the personalised SQL course the effectiveness of the course was assessed by examining the students examination performance over the period of the evaluation (2000 – 2003) and for the previous two years when an online non-adaptive (based on the same content) version of the SQL course was used. The effectiveness of the personalised SQL course is examined by comparing it to two different metrics – how students performed using the non-adaptive online course prior to the introduction of the personalised course and how students performed in examination questions related and not related to the material presented in the online SQL course.

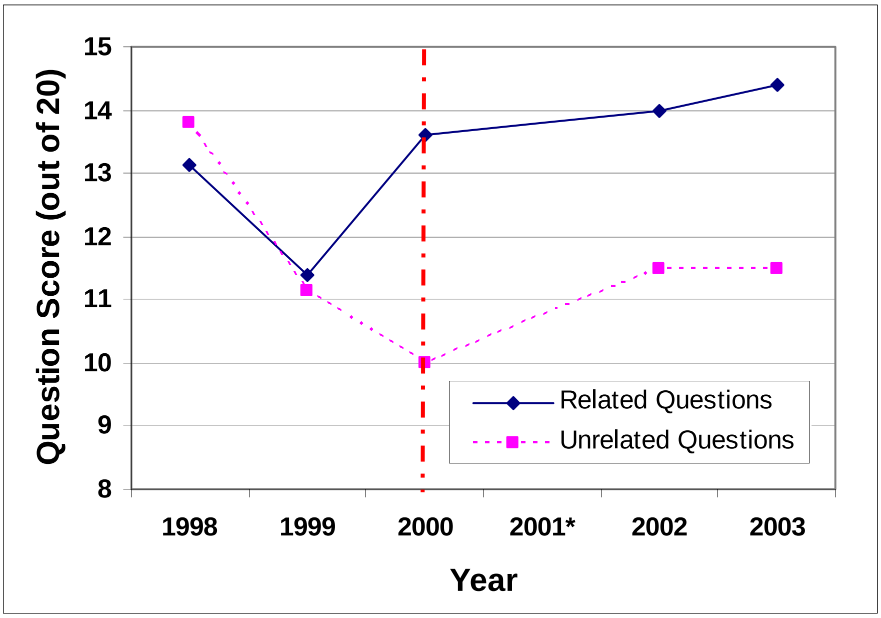


Figure 6.13, Average Question Scores on Database Examinations 1998 – 2003

In Figure 6.13, above, the Question Score axis shows the average question score out of a possible twenty marks. The two lines shown on the graph represent the average question scores in questions related and unrelated to the material taught in the online SQL courses. The personalised SQL course was first introduced in 2000 (signified by the vertical dotted line). In 1998 and 1999 the online tools were non-adaptive. As may be seen from the graph the students in 1998 scored well in both questions related and unrelated to SQL, yet in the following year their scores declined. From 2000, with the introduction of the personalised SQL course, the scores in the questions related to SQL have risen steadily. Indeed, the score in these questions display an increase in marks of 2-3 over the unrelated questions. In a question worth 20 marks this represents approximately a 10-15% increase in performance by the students. (* Note: 2001 statistics were unavailable and have been interpolated.)

Discussion and Findings

The two metrics – how students performed using the non-adaptive online course prior to the introduction of the personalised course and how students performed in examination questions related and not related to the material presented in the online SQL course – gives two different perspectives on the effectiveness of adaptivity. The former metric gives some historical perspective on the impact of the adaptive versus non-adaptive course across different peer groups, while the latter metric shows how, in the same peer groups, the adaptive course affected their performance. This latter metric is probably more significant than the former, as the comparisons across different years have not been normalised, i.e. some years may have had stronger students on average than others.

There are a number of factors that may account for this increase in performance witnessed between the questions related to SQL and those unrelated. While it may simply indicate an increase in the pedagogical effectiveness of the personalised course over the non-adaptive offering, there are other criteria that need to be considered. The students may have found the adaptive course, and in particular the controls available to them, to be more stimulating or more motivating. This increased stimulation/motivation and the subsequent increase in learner engagement may account for the increased performance.

6.2.8 Key Findings

There are a number of key findings that arise from the Evaluation of Student Satisfaction. The foremost is that the personalised courses produced using APeLS, and the multi-model approach, satisfied the vast majority of learners. This is witnessed by a number of the responses given in the questionnaire:

- Despite little or no prior experience of online learning the students felt that the personalised courses generated for them reflected the courses they wanted and contained clear objectives.
- The students were satisfied with the navigation provided in the personalised courses and were also satisfied with the quantity of content provided.
- The students did not want more control over the structure of the content or over the layout of the web interface, indicating that the system interfaces and navigation provided were usable and suitable.

The performance of students has also improved. This may be seen when comparing the scores of examination questions that are related to SQL to those that are not. Students consistently perform better on the questions related to the personalised SQL course. The reason for this is, however, needs to be further investigated. It may be attributed to a number of things, such as the increased pedagogical effectiveness of the course or improved learner engagement brought about through increased stimulation/motivation. From these findings it has been shown that APeLS and the multi-model, metadata driven approach are capable of producing personalised eLearning courses that are appropriate, coherent and effective. Above all, these courses are satisfactory for the learner.

6.3 Tutor Satisfaction

Parallel to the evaluation performed by the students the tutors involved in the teaching of the courses were asked to comment on their experiences. The tutor evaluation centres on the tutor's ability to produce and control a personalised eLearning offering and integrate it as part of their current curriculum. As the personalised SQL course is used across four different degree courses, the tutor evaluation can potentially highlight four different tutor perspectives on integrating personalised eLearning within a curriculum. The objective of the tutor trial was to determine their satisfaction with using APeLS to teach SQL as part of their curriculum. This trial was carried out by interviewing the three tutors responsible for the four degree courses. All of the tutors questioned were pleased not to be teaching SQL in a face to face situation as they felt that it was a subject in which many students had some experience, or if they did not, that it was a subject the students should be capable of learning without direct tutor guidance (especially for Junior and Senior Sophister students). As they felt that the students already had varying degrees of ability in SQL they believed that the personalisation based on prior experience was appropriate.

The main area of disagreement amongst the tutors questioned was in the matter of the course objectives for the personalised SQL courses and on the approach taken to teaching them. While some tutors felt the approach was appropriate and blended well with their current curriculum, others felt that it necessitated change in their curriculum and their approach to teaching SQL. In this sense, while the personalised courses for the students were not one-size-fits-all, the tutors felt the teaching approach was. This disagreement on course objectives and teaching approach highlighted the need for a single course to be able to support different narratives. Different narratives still facilitate content to be reused across all of the courses, but enable tutors to emphasise their personal approach to teaching the material. Initially this was achieved in year two of the APeLS-based SQL course by having a different personalised course builder JSP component for each tutor involved. This temporary solution, however, did not aid the tutors in the actual process of narrative creation. To create a narrative, tutors were required to author the JESS rules by hand. This was somewhat acceptable when the tutor was proficient and comfortable with technology and programming languages, but would clearly not suffice if the Adaptive Engine was to be used as the basis for personalised courses outside of the IT domain. This requirement has spawned separate research on creating an appropriate, pedagogically supported authoring environment [Dagger et al., 04].

The tutors also had different course assignments associated with their courseware. While it was a simple matter to change the example project solution presented to each group of different students, changing the case study material that was intertwined within the adaptive courses was not as straight forward, but could be achieved with candidate content groups. The use of separate examples across different courses is possible without any changes to the primary narrative. As the narrative refers to concepts, and obliquely to candidate content groups, rather than directly to learning content, the same narrative may be used across different courses. By adding the identifiers of different examples to the candidate content groups the appropriate piece of content could be chosen at runtime using a candidate selector. The candidate selector needs some metadata describing each example as being associated with a particular course and can reconcile that with the course the learner is currently enrolled in. In this way the courses may remain conceptually the same (i.e. because of the same narrative), but deliver different content based on the tutor.

Some tutors also wished to use third party material and this proved difficult due to the granularity and scope of that material. Maintaining language and textual flow across learning objects of different granularities can be achieved, but often requires modification of the content to enable it. The conceptual granularity of the concepts in the narrative and the conceptual granularity of the learning content should be aligned as closely as possible. When larger grained learning content is introduced it often spans a number of concepts that were originally taught by many pagelets. This difference in conceptual granularity can negate the benefits of abstraction and candidacy as the candidate content groups can no longer contain objects of the same scope.

In the second year of the evaluation, many tutors expressed the desire to personalise the courses towards other characteristics of the students. For example, depending on how well the student performed in a previous assessment the tutor wished to adjust the level of scaffolding for certain concepts. This was possible due to the flexibility of the candidacy architecture. If the student required further scaffolding then, instead of delivering a piece of candidate content, a candidate sub-narrative that further explored the concept may be chosen. Additional axes of adaptivity, such as adapting to VARK or Kolb/McCarthy preferences, may be added through the introduction of additional candidates into candidate content groups and the enhancement of candidate selectors to reason across them.

6.3.1 Key Findings

The key finding from the tutor evaluation is that while the tutors were satisfied with using the personalised SQL course as part of their teaching they wanted a greater level of control over the personalised courses. As far as the tutors are concerned one size does not fit all when it comes to narrative design, student modelling and content selection. The tutors felt that the reuse of learning resources was important, but that this was secondary to maintaining their approach to teaching. Both the narrative selection and authoring were also highlighted as issues that would need to be addressed.

The tutor experiences were good, but there are four key findings that need to be addressed –

- It is difficult to support multiple tutors' objectives in a single personalised course
- Tutors desired more control over the learner models
- The current process of authoring narratives is too complex
- It is difficult to adapt material from third party suppliers

The findings from the tutor evaluation mostly revolve around the authoring process. Tutors were satisfied with the results of using personalised eLearning courses as part of their curricula. However, appropriate author support is required to ensure that the process of creating and modifying these personalised courses is sufficiently easy in order to ensure tutors will continue to use them.

6.4 Separation and Reusability of Elements of Adaptivity

6.4.1 Overview

The reusability of narratives, content and learner models have been trialled and evaluated in two adaptive personalised courses developed using the multi-model, metadata driven approach. The first, the personalised SQL course, comprises fine-grained learning objects adaptively selected based on the learner's prior knowledge. The second is a personalised course on Physics / Mechanics that performs personalisation based on the theory of knowledge spaces [Albert and Lukas, 99].

This section evaluates how effectively the separation and reuse of narrative, content and learner models are facilitated by the multi-model approach.

6.4.2 Reuse of Narrative

Narratives, and the pedagogical strategies they embody, are not tied directly to the Learning Objects used to fulfil these strategies. This facilitates the reuse of those strategies independently to the Learning Objects. This section evaluates their reusability by examining both the size/scope of the narratives and the metadata used to describe them. It also examines the reusability of candidate selectors, which are a form of narrative, but are designed to be independent of a particular knowledge domain. This sub-section is divided into three aspects relating to narrative – the relationship between the size and scope of narratives pertinent to their reuse, the reuse of Candidate Selectors, a special form of narrative, and the relationship of the metadata that describes a narrative relevant to its potential reuse.

6.4.2.1 Reuse and Size/Scope of Narratives

One of the key factors in determining the reuse of a narrative is its size or scope. The larger and more encompassing a narrative is, the less likely it is to be reused. As narratives may include sub-narratives it is preferable to assemble a narrative describing an adaptive course from several discrete sub-narratives. This is the approach that was taken in the personalised SQL course. By building the initial course in this way the sub-narratives, each of which had responsibility for adaptively teaching a particular aspect of SQL, could be reused. With many different versions of the personalised SQL course, each teaching the material from different perspectives, the sub-narrative could be reused in different sequences. This facilitates their use for sequencing concepts differently, using sub-narratives with different Learning Objects (through the Abstraction Mechanism) or by combining them with new sub-narratives.

In the Physics / Mechanics course the addition of the Open University learning material was achieved through the creation of a new sub-narrative and its addition to the existing narrative in the appropriate place. The Physics / Mechanics course also demonstrates the reusability of the pedagogical strategy from the theory of knowledge spaces. The strategy is embodied as a sub-narrative that generically describes how the theory is employed. This is independent of the domain in which the theory is being applied. The prerequisite conceptual relationships, upon which the theory of knowledge spaces relies, are captured in the metadata describing particular learning objects. The independence of the sub-narrative from a particular domain facilitates the reuse of the pedagogical strategy in other personalised learning courses.

Discussion and Findings

Narratives, sub-narratives and pedagogical strategies may be reused easily. This reuse is facilitated by the abstraction mechanism and the ability of narratives to execute other narratives. The reusability of a narrative is determined by its scope. A large scale, highly specialised narrative is less likely to be reused than one that is smaller and discrete. Pedagogical strategies, if implemented independently from the domain they are being applied to, may also be reused.

The key enabler for narrative reusability is the abstraction architecture provided by the multi-model approach. Through this abstraction narratives are not tied to particular learning resources. They describe how concepts, not content, are sequenced. Sub-narratives, such as the theory of knowledge spaces sub-narrative, do not describe any conceptual relationships, but embody pure instructional or pedagogical approaches. The relationships they rely upon may be captured in the metadata of the content across which they operate.

6.4.2.2 Reuse of Candidate Selectors

Like pedagogical strategies, candidate selectors are highly reusable, providing the metadata they are working with is marked-up appropriately. Candidate selectors were employed in the extensions that utilised VARK [Bruen and Conlan, 02], Kolb/McCarthy [Clarke et al., 03] and different display devices [Brady et al., 04] used in the personalised SQL course to select the appropriate LOs from the candidate content groups. These selectors are highly reusable as they describe a generic approach to selecting the most appropriate candidate when presented with many to choose from. The candidate selectors may, therefore, be used in personalised courses from very different domains.

Discussion and Findings

As candidate selectors are, by their nature, domain independent they are highly reusable. This reusability promotes their reuse across courses in different domains. For example, a candidate selector that selects the most appropriate candidate for a display device does so independently of the subject of the candidate and the course. The proviso in the use and reuse of candidate selectors is that the candidates they reason about must have the appropriate metadata. As currently implemented, candidate selectors are not flexible in the metadata they expect the candidate to be described with.

6.4.2.3 Effectiveness of Metadata

As narratives, sub-narratives and candidate selectors are reusable elements of adaptivity they require appropriate metadata to facilitate their discovery and use. Sub-narratives may also be selected from candidate narrative groups using a candidate selector and will, therefore, require appropriate metadata describing their adaptive features. The content model metadata is also used to represent the narratives, sub-narratives, pedagogical strategies and candidate selectors. As the narratives and sub-narratives embody the approach to teaching a subject matter they lend themselves naturally to description using the content model metadata. The adaptive features of the narrative may be embodied using the adaptivity extension.

However, the candidate selectors do not lend themselves to being easily described using this metadata. This is due to the fact that they are mostly comprised of process and it is this process that the metadata author may wish to describe, i.e. how a candidate selector reasons about candidates is as important to capture as the types of candidates it is capable of reasoning about. The adaptivity extension may be used, but the candidate selectors are still difficult to describe, due to the permutations possible in the processes carried out.

6.4.2.4 Reuse of Narrative in the APeLS Trials

During the trial with the SQL course, four separate root narratives were used to facilitate the different teaching styles of the different tutors. These root narratives used the same set of thirty sub-narratives to produce the personalised courses. The sub-narratives were, however, presented in different orders depending on the individual tutor's wishes. In this way reusability of the sub-narratives was achieved.

The Physics / Mechanics course took a very different approach to the reuse of narrative. The narrative in this course was created to be domain independent, i.e. it implemented the Theory of Knowledge Space approach independently of the subject matter. This narrative was reused across both instances (the original and extended versions) of the Physics / Mechanics demonstrator. In this case, reusability of a generic strategy has been demonstrated.

Metadata is necessary to facilitate the discovery and reuse of narratives, sub-narratives and candidate selectors. The educational outcomes and the concepts taught in narratives and sub-narratives are relatively easy to describe using the content model metadata. The process carried out by candidate selectors, however, is less easy to describe in this way. This leads to some difficulty in identifying the suitability of a candidate selector. For example, its metadata may say that it chooses a content candidate based on its suitability for a given device. It is difficult to describe the thresholds used in choosing one piece of content over another for a PDA and it is this characteristic that the author may wish to know.

6.4.3 Reuse of Content

The reusability of content has been trialled and evaluated in the two adaptive personalised courses developed using the multi-model, metadata driven approach – the personalised SQL course and the personalised Physics / Mechanics course. The first, the personalised SQL course, comprises fine-grained learning objects adaptively selected based on the learner's prior knowledge. The second, the Physics / Mechanics course utilises the Theory of Knowledge Spaces as the basis of adaptation. The knowledge space was not explicitly modelled as a separate model. It was, however, represented through the metadata of the learning objects used. This metadata represents the competencies required (to understand the learning object) and the competencies taught (upon completion of the learning object). The learning objects used in this version of the course are of larger granularity than those used in the SQL course.

This section evaluates the reusability of content by looking at the size, or granularity of that content, and by examining the metadata used to describe it.

6.4.3.1 Reuse and Size of Learning Objects

The size of a learning object has a significant impact on its reuse. Its size is not only a measure of how much screen real estate it occupies, but also of the scope of the learning object. Fine-grained learning objects, or those with discrete scopes are more reusable than larger LOs with broader scopes. This is because authors are more likely to be able to include a discrete and small LO within a new course development and combine it with other LOs.

Both adaptive courses demonstrate the effective reuse of learning objects, but do so in quite different ways with different sized learning content. The personalised SQL course uses fine-grained single image / paragraph sized learning objects with associated descriptive metadata. Any issues or problems with such fine-grained resources may be corrected and those updates would automatically be applied in any adaptive course using the resource. This is due to the candidacy architecture in the multi-model approach. The LOs are not referred to directly by the narrative and may be updated without impacting the course flow. This updating may not be effective if the scope of the LO changes, as it may duplicate (or contradict) material in neighbouring LOs.

Reuse of these fine-grained LOs was demonstrated through evolving different versions of the SQL course. As there are many undergraduate courses in which SQL is taught there was a requirement to have several versions of the personalised SQL course available. The fine granularity of the learning objects facilitated the reuse of this content across the different courses. The variations between the courses were mainly subtle differences in pedagogical approach and preferred approach to teaching SQL, expressed by the tutors on each course. The fine granularity of the learning objects facilitated easy reuse across each of these courses. However, in some instances there existed sequencing issues where narrative flow of the content of the combined LOs was less effective than the original. These sequencing difficulties can arise when trying to place discrete and encapsulated pieces of content together – they may have logical flow from conceptual and pedagogical perspectives, but their language and size make narrative flow difficult. There is no obvious solution to this issue. If the learning objects are larger in size their reusability is diminished. Similarly, if their granularity remains small, but narrative interdependencies are introduced, their potential reusability also suffers.

In the Physics / Mechanics course the size of the Learning Objects were much larger than that of the SQL course. The LOs in the Physics / Mechanics course were approximately a full screen of text with embedded images. Reuse of the individual LOs in this case may not be as easy to achieve due to their larger granularity than that of the SQL content. However, the successful integration of LOs from a third party vendor was demonstrated as part of this course. Learning objects from the Open University [OU] were successfully integrated as part of the personalised course without need to change the LOs or the narrative/pedagogical strategy. The only change necessary was the need to update the metadata of the Open University LOs to include the new theory of knowledge space competencies.

Discussion and Findings

Through the different instances of personalised courses based on APeLS two different models of reuse have been shown. The first model is that of the personalised SQL course, which may be viewed as a suite of courses based on a common fine-grained content base. In this model of reuse the reusability of the LOs is facilitated by the abstraction architecture. The different courses, all teaching SQL, can have different narratives that sequence the concepts in very different ways. When candidates are chosen to realise these concepts, however, they may be chosen from the same candidate repository. A difficulty may arise when threading together fine-grained resources because there is a narrative flow to the learning objects that must be considered. When they are combined in very different ways this flow through the content may become disjoint. The abstraction mechanism has also shown that updates and fixes to such fine-grained resources are automatically affected in any course that uses them.

The second model of reuse showed the reusability of larger scaled LOs. This was demonstrated through the inclusion of third party learning material into an existing adaptive course with minimal changes required; only the metadata representing that material required update. In both models of reusability the scope of the learning object and any external references it may contain had to be considered. LOs with a large scope or those with many external references may be less reusable. If there are many LOs with overlapping scopes, in a course its content may appear repetitious to the learner. External references to other LOs may limit the reusability of an LO because the externally referenced LOs may also need to be included in the personalised learning course. As such, LOs that are encapsulated and have a fine scope are more reusable.

6.4.3.2 Effectiveness of Learning Object Metadata

The metadata used to describe learning resources is evaluated on its flexibility in supporting the selection of appropriate content candidates during the adaptation process. The metadata used to describe the Learning Objects for both the personalised SQL course and the personalised Physics / Mechanics course is based on IMS Learning Resource Metadata. This metadata schema contains elements useful for the discovery and retrieval of Learning Objects, such as title, subject, keywords and usage information. For example, in the Physics / Mechanics course these elements were effectively used to identify suitable LOs from the Open University for inclusion in the existing course. There are, however, some difficulties with vocabulary consistency. Even in a scientific domain such as Physics some inconsistencies between how different metadata authors defined concepts were identified.

The metadata extension used in the adaptive process proved to be highly flexible. It was used to describe the competencies taught and competencies required as part of the theory on knowledge spaces as well as the concepts related to pagelets in the personalised SQL course. In extensions to the personalised SQL course, both the VARK inventory [Bruen and Conlan, 02] and the Kolb/McCarthy Learning Style Inventory [Clarke et al., 03] were catered for at the Learning Object level. To facilitate this, the metadata extension was used to describe these inventories.

While there were no issues with the metadata extension in describing the adaptive use of fine-grained/pagelet Learning Objects, the standard metadata elements of title, subject and keywords were difficult to complete effectively. When LOs are this fine-grained the terms used to describe them are often nearly identical, so although the metadata author may be able to complete the metadata the elements values are also identical. This problem is not easily solved and is particularly apparent in the SQL course where there are just over ninety concepts and over three hundred pagelets. When looking at the metadata of the pagelets many claim to teach the same concept, when in reality, they only teach a part of the concept. Standard domain vocabularies and ontologies do not deal with concepts as fine-grained as those represented by the pagelet Learning Objects. The relation element may be used to describe a pagelet as being part of a larger concept, but this is not strictly correct usage of this element as it is intended to show relationships between Learning Objects and not concepts.

6.4.3.3 Reuse of Content in the APeLS Trials

The four different courses created for the different SQL tutors demonstrated the flexibility of narrative to create diverse adaptive courses from a single body of content. In the SQL trial over 330 pagelets were created and these were reused, in different sequences, in the different SQL courses. The reuse of these fine grained pieces of content was facilitated primarily by their size and scope. The fine granularity facilitated their reuse as the scope of each piece of content was discrete. From a metadata perspective content was differentiated by the concepts taught. However, this lacked precision as the vocabulary used to identify content was coarse. Therefore, a number of pagelets could be tagged as teaching the same concept, even though in actuality they teach different aspects of it. The candidacy architecture facilitated the effective reuse of this content by establishing sequences within concepts.

During the Physics / Mechanics trial larger grained content was used. The primary facilitator for the reuse of this content was the metadata used. This metadata was formed from complete IMS Learning Resource Metadata instances with the addition of `adaptivitytype` elements. Reuse was demonstrated in this APeLS instance by the inclusion of additional, pre-existing content from another content provider. This demonstrated that the content base could be added to, thus facilitating growth within adaptive courses.

Discussion and Findings

For larger LOs, standard metadata was appropriate for their description. This is especially useful when discovery of the LO is the goal. Current vocabularies were generally suited for the completion of the metadata elements. Finer-grained Learning Objects, however, were difficult to describe using existing elements in metadata standards and existing vocabularies. This is a significant finding because, although the fine-grained LOs are more reusable, they are more difficult to discover. Their small size facilitates their reuse, but also prohibits their accurate description.

The metadata extension used to describe the adaptive use of LOs, both coarse and fine-grained, was flexible, appropriate and effective for assisting their use in personalised courses. However, as with all metadata, care needed to be taken to ensure consistency in completing the elements. The issue of vocabularies and ontologies are of key importance in metadata. Metadata schemas provide the syntax of the elements, but do not provide the semantics that may be used to complete them.

6.4.4 Reuse of Learner Models

The learner model utilised in the personalised courses also uses the adaptivity extension to facilitate the description of the various characteristics of the learner that the courses are adapting to. Individual learner models of this form may be reused in part across different personalised courses implemented in APeLS. For example, if a learner undertook a course where their VARK preferences were determined and modelled, this aspect of their model may be reusable in other courses (certainly if they are in a similar knowledge domain). If their competencies in a domain are also modelled these may be less reusable, due to the fact that other personalised courses may deal with an unrelated domain. The more generic elements of a learner's model, such as learning style and learning preferences, tend to be reusable in many APeLS-based personalised courses.

As no standards-based learner models facilitate the level of flexibility required, only the most basic learner attributes, such as name and contact details, are portable across different Virtual Learning Environments. This is an important point for the portability and reusability of learner models. If it is only the basic elements that are truly reusable then should adaptive systems and VLEs be attempting to share them? In the integration of APeLS and a commercial VLE the learner's identity was the most significant piece of data that was passed between them. The specialised adaptivity extension is not portable to such environments.

Discussion and Findings

The learner model is reusable, in part, across different personalised courses based on APeLS. The elements that tend to be most reusable are those that also tend to be difficult to model, such as learning style and learning preferences[3]. Reuse of certain basic elements of the learner model may be achieved across different Virtual Learning Environments. Learner identity tends to be the most valuable piece of data to share between services. The different services may use the identity of the learner to build there own models as in many situations the kind of information that is modelled about the learner is only of value to the service that modelled it.

6.4.5 Key Findings

From the evaluation of the reusability of the models in the multi-model, metadata driven approach there are a number of key findings. The first is that metadata is the primary driver in the reuse of elements of adaptivity. Appropriate metadata facilitates the discovery and reuse of any of these elements. The discovery and reuse of these is related to their size. Larger resources, whether they are learning objects or narratives, tend to be described better than smaller ones. This is generally a vocabulary/ontology shortcoming. Vocabularies do not exist to describe the concepts in fine-grained resources in fine enough detail. Fine-grained resources, while more difficult to discover, are easier to reuse within personalised course offerings. The narrative flow when sequencing such fine-grained resources must be considered. The adaptivity extension for describing how adaptive resources may be used is effective and flexible in describing various forms of adaptation across a broad range of resources – from fine-grained pagelets and sub-narratives to complete personalised course offerings.

The learner model is highly flexible and may be tailored to a broad range of personalised courses. It is, however, the element of adaptivity which presents the most difficulty from a reusability perspective. By its nature the learner model for a given personalised course is focused towards the needs of that application. It is only aspects of this personalised information that may be reused in a wide range of personalised courses. These tend to be independent of the domain or subject matter that is being taught. Indeed, it is often the learner's identity that is of most use.

The elements of adaptivity that are most easily reused are the content and narrative. The main reasons for this ease of reuse are the abstraction mechanism and the

[3] The reusability of learning style and learning preferences may be limited, as learners' learning style and preferences may vary from subject to subject.

candidate content groups. The separation of narratives, sub-narratives, pedagogical strategies and candidate selectors from the content they operate on means that there is no direct interdependency between them. This enables narratives to be reused with different content and content to be reused in more than one narrative. This reuse is demonstrated in the multiple versions of the personalised SQL course that are all based on a common content repository. The ability of narratives to execute other narratives facilitates the reuse of both narrative structures and pedagogical strategies.

6.5 Flexibility and Extensibility of Adaptive Effects

6.5.1 Overview

The multi-model, metadata driven approach and the adaptive engine at its core are designed to be generic, flexible and extensible. This section describes how the different instances of personalised courses, based on APeLS, have demonstrated these features.

6.5.2 Using APeLS to teach SQL

The personalised SQL course, which formed the basis of the student evaluations, has seen a number of extensions. The first extension of the course deals with the inclusion of VARK [VARK] learning material. Within this course there are a number of candidate content groups representing pagelets of different media types that support different learning preferences. These pagelets are differentiated in their metadata by using VARK values in the `adaptivitytype` elements. This extension to the original SQL course enables the learner's preference for different forms of content to be catered for. Candidate content groups, the abstraction mechanism and candidate selectors were employed to achieve this additional axis of adaptivity. Implementing this extension of the course did not require a re-authoring of narrative because conceptually the narrative remained the same. The appropriate identifiers of the additional pieces of content were added to the candidate content groups referenced by the concepts. The only significant authoring that was involved was the creation of a candidate selector that could reason over these VARK values and choose the most appropriate in accordance with the learner model. As the candidate selector is independent of the domain being taught it may be reused in any course where the author wishes to employ VARK.

As part of another extension to the personalised SQL course, the Kolb/McCarthy Learning Style Inventory was used to differentiate content, in accordance with the inventory's separation of learning style into two continuums. Again, candidate content groups and the abstraction mechanism were employed to separate narrative from content. Appropriate candidate selectors were also developed to select the most effective learning content for the learner based on the Kolb/McCarthy inventory. This extension was implemented in the same way as the VARK extension.

Discussion and Findings

The multi-model approach facilitates the extension of existing courses. This extension may be achieved by creating additional content, creating appropriate metadata, adding the content identifiers to candidate content groups and creating (or reusing) a candidate selector. The primary result is that the narrative does not need to be modified to implement these changes. As such, the original instructional strategies described in it are still appropriate.

6.5.3 Using APeLS to teach Physics / Mechanics

The Physics / Mechanics course was extended with mechanics learning material from the Open University [OU] with the goal of investigating the flexibility of the multi-model approach in increasing the scope of an existing course. The material from the OU was added by extending the number of concepts in the knowledge space. These additional learning objects were added by extending their metadata to include these additional concepts. The new material shared some prerequisite competencies with those taught by the existing material. When a learner browsed to existing material that completed these prerequisite relationships, the new material became available to them. The navigation structure they were placed in (i.e. the Table of Contents) was extended by using a sub-narrative.

Discussion and Findings

The multi-model approach is flexible enough to cater for the easy extension of the scope of the courses offered. The only changes necessary to facilitate the extension of the Physics / Mechanics course was the addition of appropriate metadata, i.e. metadata conforming to the theory of knowledge spaces, to the new resources and the addition of a sub-narrative that added the new concepts to the Table of Contents. The author must understand the paradigm for adaptivity and must have knowledge of the current knowledge space for this to be accomplished.

6.5.4 Key Findings

APeLS and the multi-model approach have been shown to be both flexible and extensible. Both additional axes of adaptivity (in the case of the VARK and Kolb/McCarthy extension of the SQL course) and additional learning material have been successfully added to existing courses. Significantly, this was done without a need to re-author the existing narratives. Narratives could be extended by adding sub-narratives. The number of axes of adaptivity could also be increased by adding or extending existing candidate selectors.

The candidacy and abstraction architecture, provided by the multi-model approach, is a key factor in facilitating these extensions. This is due to the fact that additional learning content may be added without changing the concepts in the narrative. These additions occur through the addition of the content identifiers to the relevant candidate content groups.

6.6 Integrating APeLS with a VLE

6.6.1 Overview

The goal of this part of the trial was to evaluate how successfully APeLS could be integrated with a third party Virtual Learning Environment (VLE). In this case integration is defined as the inclusion of a personalised course as part of a standards-based eLearning offering – combining both static learning resources with personalised eLearning provided by APeLS. The personalised course is required to interoperate with the VLE in order to pass learner and assessment information. This integration work was carried out as part of EASEL (Educator Access to Services in the Electronic Landscape) [EASEL], a two year European Commission project funded under the auspices of the IST Programme. Both instances of APeLS, the SQL course and the Physics / Mechanics course, were used as educational services within this project. As part of the EASEL trial, the main aspects with respect to the remote service that were examined were –

- The effectiveness of the metadata describing the service. The effectiveness of the metadata is a key enabler in facilitating the reuse of the service.
- How well the service was integrated with and delivered as part of the Virtual Learning Environment

The examination of these aspects highlights how reusable APeLS is and how easily it may be integrated with a standards-based VLE.

6.6.2 Metadata

The metadata examined as part of this trial was that describing the service. Both of the possible approaches to describing an Adaptive Service described in Section *5.4.1 Metadata Describing the Service* were tested. The first approach was the use of a single metadata record to represent all possible permutations of the adaptive course. The second approach was to use multiple instances of metadata records that represent typical courses that the adaptive service may produce. The role of the metadata describing the service is to facilitate its discovery and integration with eLearning courses offered through a VLE. The metadata should describe the various forms of adaptation supported by the personalised course.

The first approach to describing complete personalised courses, tested as part of the EASEL project, was to use a single metadata record extended using the *adaptivity* extension. This record was used to describe all possible permutations of the personalised course using the *set* and *candidate* elements of *adaptivitytype*. In other words, the metadata attempts to capture the process of adaptivity and the different personalised outcomes that may result. While the range of personalised courses produced by the adaptive service could be described using this approach it presented a number of difficulties –

- The greater the scope of potential adaptivity supported by the service the more unwieldy the metadata and permutations therein became.
- Metadata search tools did not understand the semantics represented by the set structure and often viewed all permutations as representing a single course offering, which resulted in wildcard and inappropriate search results.

The second approach tested was to create a number of separate metadata representations of the adaptive service. Each metadata representation used the *adaptivity* extension, but only attempted to describe a single, typical permutation. In this way the different metadata representations captured a sample of the most popular usage of the adaptive service. The metadata author is responsible for recognising and determining the typical usage and representing them in metadata. This approach resulted in more appropriate and representative search results, although there is the possibility of less frequently used permutations of the adaptive service not being discovered.

Discussion and Findings

The metadata representation of adaptive services as a whole is not a trivial problem. The potentially wide range of courses that may be produced by an adaptive service means that no single standards-based metadata can fully represent it. The approach of representing the adaptive service and all of its permutations using a single all encompassing metadata record proved to be a complex procedure. Since the metadata search tools used were unable to decipher the semantic relationships within the *adaptivity* extension, searching across the single record was ineffective.

The second approach to representing typical personalised courses that the adaptive service could produce proved more effective, although it had the potential to overlook some of the less frequently used aspects of the adaptive service. Ultimately, this approach was adopted within the EASEL project. Ideally, the metadata for describing an adaptive service should be automatically generated by examining how it performs personalisation, thus alleviating the metadata author from what can be a very complex task.

6.6.3 Service Integration and Delivery

Section *5.4 APeLS as a Service* describes the implementation of the integration between APeLS and the commercial Virtual Learning Environment produced by Fretwell-Downing Education. The three issues for integrating an adaptive service and a VLE are –

- Launching the remote adaptive service from the VLE

- Establishing communications between the service and the VLE
- Passing data and models between the service and the VLE

The location of the adaptive service was imported into the VLE using the metadata stored in the IMS Content Packaging manifest. As this was the mechanism used to import regular, static content, into the VLE it required no changes to its current content import mechanisms. Launching the adaptive service is carried out in the same way as launching a remote learning object. The only significant difference is that most learning objects do not have in-built navigation and rely on the VLE to provide navigation to the learner. The personalised courses produced by the adaptive service generally have a personalised navigation structure and do not rely on the VLE for navigation. Indeed, any additional navigation provided by the VLE may hinder the learner's use of the personalised course. To this end the personalised course is launched in a separate browser window with no additional navigation structure provided by the VLE.

Establishing and maintaining communications between the adaptive service and the VLE is achieved using the process described in Section *5.4.4 Communication between the AHS and* the VLE, and involves a JavaScript implementation of the Content Interworking API. This communication facilitated by this API utilises the SCORM Runtime Communication API for data models and uses a hidden JavaScript enabled frame to provide the HTTP-based communication infrastructure. Using this infrastructure and the data models provided by SCORM, the adaptive service and the VLE could communicate learner and assessment information.

The main challenge faced in the trial was not establishing communication, rather it was determining the form of the data that should be passed between the adaptive service and the VLE. With regard to learner information, the information stored by the VLE about a learner is either inappropriate or insufficient for facilitating the personalisation of courses. This is similar to the issue described in Section *6.4.4 Reuse of Learner Models*, whereby the nature of the information stored by the VLE and adaptive service was for different purposes. The VLE stored basic contact information about the learner, as well as information pertinent to their interactions with the VLE. For the adaptive system the most important information was that which it modelled to carry out the personalisation process. These two sets of information both refer to the same learner, but other than the identity of the learner have little, or nothing, in common.

A different issue affects the communication of assessment results. Assessment information is returned to the VLE using a normalised range. This allows the adaptive service to inform the VLE how the learner has performed in any assessments they have completed, but this is only a summary score for all assessments completed in the adaptive service. As there is no guarantee that an individual assessment will be delivered as part of a personalised course, it is not possible to provide more detailed assessment information back to the VLE. Even if this were not the case there is still a vocabulary issue, i.e. both the VLE and adaptive service would be required to understand a common vocabulary in which to communicate fine-detailed concept-related results.

Discussion and Findings

Standards-based communication was achieved between the adaptive service and the VLE with relative ease as the Content Interworking API is the standard mechanism used by the VLE to communicate with static content. The more complex issues, however, arose when trying to identify information that the VLE and adaptive service could share successfully. As much of the learner modelling information in the adaptive service is quite specific to how it implements adaptivity internally there is rarely a requirement to share this information with a VLE as the VLE has no use for such information.

What proved more useful was the sharing of assessment information. Assessment information about the learner prior to their entering the adaptive service could be used to gauge the level of complexity at which to start the personalised course. Similarly, assessment information returned to the VLE upon completion of the personalised course could contribute towards an overall course score. However, as it may not be possible to guarantee the delivery of any individual assessment as part of a personalised course the information returned to the VLE is usually non-specific using a normalised score. This may lead to difficulties if the VLE requires detailed information about the assessments carried out by the learner. Such information may be required for accreditation purposes. If this is the case, it may no be possible to rely on adaptive services for such accreditation processes. By their nature the assessments tend to be personalised and do not fit with a standardised assessment point of view.

6.6.4 Key Findings

From the integration evaluation it has been shown that APeLS may be successfully employed as a remote service and integrated as the personalised part of an eLearning offering. The technical aspects of this integration, namely the import, launch and communication with the VLE were relatively straightforward. However, the semantic aspects of this integration proved to be more problematic.

The first semantic issue was the appropriate representation of the adaptive service to facilitate search, discovery and use of the personalised courses offered. Current metadata standards do not facilitate the effective description of adaptive services and their features. Two approaches to representing the adaptive service using metadata were described. The first attempted to use a single *intelligent* piece of metadata to represent the adaptive service and all of its permutations. This proved difficult to author and was interpreted poorly by metadata search tools that did not understand the complex semantic relationships expressed. The second, more successful approach was to describe a number of discrete courses that the adaptive service may deliver as separate metadata instances. This approach is easier to author and provides more satisfactory searching results, but may not describe all of the permutations possible in the adaptive service.

The second semantic issue that arose in the integration trial was that of the passing of information between the VLE and adaptive service. While they had the infrastructure for communication, the question arose as to what information was appropriate to communicate. The VLE and adaptive service do not share many learner model elements in common and those that they do share are rarely pertinent to the personalisation process, e.g. the learner's forename and surname. Similarly, the passing of assessment information between the VLE and the adaptive service was simplified to summary scores as they shared no common semantic understanding upon which to share more detailed information. This summary information could still be effectively used, but a richer semantic exchange may alleviate some learner modelling requirements of the adaptive service.

6.7 Summary

This chapter has discussed the trial and evaluation of the multi-model, metadata driven approach and APeLS across a number of areas. The first two sections of this chapter discussed the evaluation of student and tutor satisfaction with personalised courses produced and delivered by APeLS. From a pedagogical perspective the evaluation has shown that students were satisfied with the personalised learning experiences delivered through APeLS and that they achieved genuine learning outcomes. Students were successfully offered a range of adaptivity in their personalised courses, demonstrating the extensibility of the multi-model, metadata driven approach. Through the discrete separation of models, and the learner model in particular, the students were given a high level of control over their learning experience. This level of control and the ability to extend the adaptive functionality of personalised eLearning courses was achieved in the multi-model, metadata driven approach through the discrete and separate modelling of the elements of adaptivity and their reconciliation at runtime. Through the separation of content and narrative, tutors were given control over the scope and adaptive features of their courses. By modifying the narrative, while basing their course on the same body of content, these changes could be made without authoring or modifying content.

These outcomes from evaluating student and tutor satisfaction correspond to the goal of APeLS that through using courses developed in APeLS students should receive offerings that are pedagogically appropriate, coherent and effective. The research question of this thesis is whether it is feasible and beneficial to separate the elements of personalised eLearning, such as the content, sequencing logic and the learner model. Satisfaction of the actors involved is a key outcome. This chapter has also evaluated the success of separating the elements of adaptivity. The *Separation and Reusability of Elements of Adaptivity* section examined the reusability of the narrative, content and learner models facilitated by their separation.

The reusability aspect of the trial demonstrated that the discrete elements of adaptivity could be reused across many adaptive eLearning applications. Moreover, it demonstrated that adaptive solutions could be developed using the content from multiple providers. It also showed that a range of different types of content, pedagogical strategies and learner modelling techniques may be successfully employed using the multi-model, metadata driven approach. The findings in this section identified a number of features of each model that facilitate their reuse. For example, in both narrative and content the scope of the model was an important characteristic in determining its reusability. Appropriate metadata is also a key feature in enabling the reuse of each of the elements of adaptivity.

The following section described the flexibility and extensibility of the multi-model approach and looked at the instances of APeLS and the features of the approach that facilitated extensibility. Through the use of pedagogical strategies from external sources, namely the theory of knowledge spaces, Kolb/McCarthy and VARK, it was demonstrated that adaptive applications developed using the multi-model approach could be easily extended.

The final section described the integration trial and validated the service approach to producing adaptive systems. It showed that an adaptive service may be successfully integrated with an existing Virtual Learning Environment to provide learners with a richer learning experience through the combination of personalised eLearning and the advanced course management features of a VLE.

7 CONCLUSION

7.1 Introduction

This thesis has presented the multi-model, metadata driven approach to producing adaptive systems for personalised eLearning. This innovative approach proposed the runtime reconciliation of discrete elements of adaptivity to produce extensible personalised eLearning courses. This reconciliation is facilitated by a generic adaptive engine that may be repurposed in many adaptive applications.

This chapter discusses the objectives of this thesis and how they were achieved. It also identifies the contribution this work has made to the state of the art of adaptive systems and in particular Adaptive Hypermedia Systems. Finally, it concludes with a discussion of the future work that describes research areas in which this work may be carried forward.

7.2 Objectives and Achievements

The research question driving this thesis was whether it is feasible and beneficial to separate the elements of personalised eLearning, such as the content, adaptive logic and the learner model, while still being able to produce tailored courses that are pedagogically appropriate, coherent, satisfactory and effective for the learner.

The primary objective of this thesis was to propose and evaluate an architectural approach for implementing dynamic adaptive courseware with sufficient flexibility to support multiple instructional models and multiple axes of adaptivity, whose selection is independent of the architectural implementation. It was proposed that this objective could be achieved through the development of a new approach for implementing adaptive systems that facilitates the reuse of the main elements of adaptivity and promotes a generic and extensible adaptive engine that can be repurposed across many different personalised eLearning applications. It was also proposed that through the separation and discrete modelling of the different elements of eLearning and adaptivity this generic and extensible architecture for Adaptive Hypermedia Systems would be possible. Furthermore, this separation of the elements of adaptivity, both physically and logically, enables their reuse across multiple personalised eLearning solutions.

The multi-model, metadata driven approach to the development of adaptive systems was proposed at the core of this thesis. This approach advocates the separation of the elements of adaptivity into discrete models that are reconciled at runtime through a generic and extensible adaptive engine. Much of the design and development effort of this thesis focused on the realisation of this generic adaptive engine, while ensuring that the models it utilised could be employed successfully as part of a personalised eLearning solution. The models and elements of eLearning and adaptivity that this thesis focused on were Digital Learning Objects (LOs), Narratives (in the form of Course Structures and Instructional Strategies) and Learner Models, although additional models such as Environment Models and Device Models are also possible.

Potentially the most significant model in the multi-model approach is the Narrative Model as it is this model that embodies strategies, adaptive course structures and the principles of candidate selection. Through the state of the art survey of Adaptive Hypermedia Systems it was determined that many AHSs had their adaptive logic either embedded in the core of the system or in the content model. The narrative model is an extensible model that enables the authors of personalised eLearning solutions to easily update and expand the adaptive logic of their courses. This logic embodied in the narrative does not, however, exist at the core of the generic adaptive engine. It remains as a separate and discrete model.

As such the narrative, along with the other models used in the multi-model approach, is reusable across different personalised eLearning applications. This reuse is facilitated through the abstraction mechanism. Using the abstraction mechanism the narrative is not constrained to referring to Learning Objects directly, but uses conceptual abstraction to refer to knowledge domain concepts rather than LOs. The ability of the narrative to execute sub-narratives coupled with the abstraction mechanism has enabled the development of a candidacy architecture for facilitating multiple axes of adaptivity. Through candidacy, the most appropriate element of adaptivity (either a Learning Object or Narrative) may be selected to fulfil the personalisation criteria. For example, if a narrative determines that a learning concept should be added to a personalised course a candidate selector may be invoked to choose the most appropriate sub-narrative or learning object to teach that concept from a group of potential candidates. This selection is usually based on the learner's needs and preferences, but may be based on any model in the multi-model approach.

The generic adaptive engine at the core of the Adaptive Personalised eLearning Service (APeLS), based on the multi-model, metadata driven approach, provides the functionality to execute narratives and provide them with access to the various models required. As instructional strategies, adaptive course structures and candidate selectors are all forms of the narrative model they can make use of any functions provided to narratives by the adaptive engine. The abilities to access, compare and reason about the various models available in a given personalised eLearning application are central to the ability of the narrative to adapt and it is this type of functionality that the engine provides. The information that is reasoned over is encapsulated as metadata. Every model in the multi-model approach has some metadata associated with it. Indeed, some models such as the Candidate Content Groups and the Learner Model are purely comprised of metadata. It is through this metadata that models may be compared and reasoned about by narratives.

The main goals of APeLS, based on the architectural approach, were that it –

- Delivers a satisfactory and pedagogically sound eLearning experience to the learner
- Provides different adaptive effects based on different sets of models
- Separates the different concerns and elements of learning
- Is flexible and extensible in the types of adaptivity offered
- Is easily integrated with existing Virtual Learning Environments

This thesis has achieved the initial objective of developing an approach to producing personalised eLearning courses that are pedagogically effective, enable learners to achieve genuine learning outcomes and, above all, provide a satisfactory learning experience to the learner. These achievements were demonstrated through the successful evaluations described in the *Trial and Evaluation* chapter. This chapter demonstrated how the multi-model approach of utilising separate and discrete models, described using metadata, facilitated the reuse of all of the elements of adaptivity in an adaptive system.

This separation also showed how tutors could be empowered to create different narratives across a common body of content, thus reusing that content. Through trialling and multiple instances of APeLS the adaptive engine itself has also been shown to be reusable and flexible through its generic implementation. APeLS is also capable of producing different adaptive effects based on different sets of models. Through the multiple instances of courses based on APeLS it has been demonstrated that the adaptive engine is extensible with the use of additional models. Finally, it has been shown that APeLS may be integrated successfully with third party products, such as Virtual Learning Environments, in order to facilitate the production of composite online learning experiences.

7.3 Contribution to State of the Art

The multi-model, metadata driven approach, as an approach for the development of adaptive systems, is the primary contribution to the State of the Art made by this thesis and work described in it. This approach is significantly different to that used in the development of current Adaptive Hypermedia Systems. Firstly, it utilises a generic and extensible adaptive engine that is application independent. This engine provides functionality that allows adaptive logic from external sources to utilise and manipulate multiple models to produce an adapted effect. This approach is different from current adaptive systems in that both the adaptive engine and the adaptive logic remain separated and are reusable independently of each other. The models used in the adaptation process are also kept separate and discrete. This separation of elements of adaptivity, that is fundamental to the multi-model, metadata driven approach, benefits the State of the Art by providing an approach that enables authors to reuse adaptive logic and learning resources, which are expensive to develop.

The influence of this approach on the State of the Art has been witnessed by its direct contribution to over twenty publications in international conferences and journals such as - International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems; Ed-Media - World Conference on Educational Multimedia, Hypermedia and Telecommunications; International Conference on Hypertext and Hypermedia; E-

Learn, World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education; the International Journal on E-Learning and the Journal of Universal Computer Science and the Journal of the Japanese Society for Information and Systems in Education. The most significant of these publications have been –

- “Multi-Model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning” [Conlan et al., 02a].
- “Towards a Standards-based Approach to e-Learning Personalization using Reusable Learning Objects” [Conlan et al., 02c].
- “An Architecture for Candidacy in Adaptive eLearning Systems to Facilitate the Reuse of Learning Resources” [Dagger et al., 03].
- “Extending educational metadata schemas to describe adaptive learning resources” [Conlan et al., 01].
- “An Architecture for integrating Adaptive Hypermedia Service with Open Learning Environments” [Conlan et al., 02b].

This multi-model, metadata driven approach has been shown through this thesis to be capable of producing pedagogically effective adaptive eLearning courses. It has also shown that these courses may be improved without having to modify either the generic adaptive engine or the existing pedagogical strategies employed. This coupled with the separation of the elements of adaptivity enables authors to improve and update adaptive eLearning courses iteratively. This is a significant contribution to the State of the Art where many existing Adaptive Hypermedia Systems require a complete reworking of the core of the application to implement changes.

The metadata driven aspect of the approach is used to describe the elements of adaptivity and how they may be used adaptively. This form of descriptive metadata has not previously been required in many AHS as they do not generally facilitate the reuse of elements of adaptivity. The use of metadata, and the adaptivity extension in particular, to describe elements of adaptivity beyond just Learning Objects is an innovative feature of this work. The metadata extension proposed enables metadata creators to describe the adaptive features of adaptable content as well as enabling them to describe how a resource (Learning Objects, Narrative, Pedagogical Strategy, Adaptive Course Structure, etc.) may be used adaptively.

The abstraction mechanism utilised in APeLS is an innovative solution that has not been used in an AHS before to maintain abstraction between adaptive logic and the learning material it is acting upon. Furthermore, this abstraction layer facilitates the use of additional axes of adaptation through candidacy.

Adaptive Hypermedia is still a relatively young research field and it is hoped that the work described in this thesis will have a beneficial and significant impact upon it.

7.4 Future Work

There are many areas in which the work described in this thesis may be taken forward. In the domain of personalised eLearning there are many different forms of personalisation and characteristics of the learner that may be adapted to, only a subset of which was examined as part of this thesis. The multi-model, metadata driven approach provides an extensible platform for exploring further strategies for personalising eLearning material. With the proliferation of devices that a learner may use to learn (PDAs, Mobile Telephones, Laptop Computers, etc.), there is also a requirement to support the adaptive layout of learning material to tailor the presentation of personalised eLearning.

7.4.1 Narratives

There is also much scope for future work in the exploration of narratives and their potential. As part of this thesis, the basic technology of narratives has been employed to create instructional strategies, adaptive course structures and candidate selectors. Narratives may be employed to provide additional intelligent adaptation effects such as peer-based adaptation, collaborative personalised eLearning and the personalisation of open hypermedia.

The nature of adaptive narratives for eLearning, a definition of narrative and a formulation of its capabilities is necessary future work. At the moment the capabilities and limitations of narrative are ill-defined. Through the formal evaluation of narrative types, their suitability in different situations may be determined.

Such research may yield a deeper understanding into how narratives may be separated into constituent parts. At the moment the narratives in APeLS, despite the use of sub-narratives, primarily perform a simple set of adaptive functions. Namely, they add appropriate concepts to a course navigation structure and select the most appropriate candidate to fulfil that concept. It may be appropriate to separate these adaptive responsibilities across different narratives.

7.4.2 Authoring

In the area of authoring for adaptive systems there is much work to be carried out. There are several tools to aid authors in the creation of Learning Objects, but there are few to aid them in the creation of personalised eLearning courses. The multi-model approach enables authors to compile new adaptive courses from pre-existing elements of adaptivity. The metadata associated with these elements facilitates their discovery. However, authors require a tool to facilitate the process of discovering, creating and combining elements of adaptivity into a new personalised eLearning course.

Moreover, such a tool should actively support the author in creating pedagogically sound courses. This process is related to the future work required on narrative, mentioned above, as a clearer understanding of narrative and its capabilities will lead towards identifying which type of narrative is most appropriate to support which pedagogy. The identification and representation of abstract ‘types’ of narratives is also important. This abstraction stage will facilitate authors in choosing, or being supported in the choice of, the most appropriate narrative for a given course. The abstracted narrative may then be populated through an appropriate authoring tool.

7.4.3 eLearning Services

From a more technical perspective the adaptive engine may be extended to support web service technologies. This support could be twofold with the adaptive engine given the ability to access models and elements of adaptivity from remote services as well as exposing the adaptive engine as a web service in its own right. This exposure would enable an adaptive service architecture to be created where adaptive services could be chained and linked together to form multi-adaptive outputs. Each adaptive service in the framework may have discrete adaptation responsibilities. For example, one adaptive service may be responsible for selecting learning material that is most suited to the learner's objective, whilst another is responsible for rendering it on the learner's terminal device.

This technical aspect of future work may be used to facilitate the separation of narratives into discrete responsibilities mentioned above. The ability to access and process information from other web services also expands the number of potential model sources available. Since narrative decisions and adaptivity are based on input and composite models the availability of more models may facilitate more precise levels of adaptivity.

7.4.4 Ontologies and Repositories

Another area for future work is in exploring the potential relationship between adaptive knowledge-based systems and emerging vocabulary and ontology definition languages. Vocabularies and ontologies may be seen as models for use as part of the multi-model approach. However, they have the potential to provide a wealth of information that may assist in the adaptation process and, therefore, have a larger scope, and potential impact, than any of the current information models in the multi-model approach. Ontologies can supply important information that helps in the syntactic and semantic reconciliation of other models.

Metadata enabled content repositories can provide an additional source of composite models to the multi-model approach. By looking beyond local content repositories the scope of adaptive courses may be increased. There are, however, challenges in combining such content together towards an adaptive outcome. The first challenge lies at the metadata level – for the adaptation to be effective the metadata describing the content resource must be appropriate. There is potential for ontologies, and specifically ontology mapping approaches, to facilitate the reconciliation of diverse metadata specifications. The second challenge lies in how the content itself is authored. The granularity and scope of the content, as discussed earlier, is an issue. So too, is the format and styling of the content. These issues need to be addressed if content repositories are to be successfully leveraged towards personalised eLearning.

7.4.5 Other Areas of Future Work

Finally, there are many areas outside of personalised eLearning where the multi-model, metadata driven approach may be employed. Other areas of adaptive system research include –

- Ubiquitous Computing – the multi-model approach may be employed to create systems that manage how the physical and computing environment adapts to a user's needs and preferences. The use of Context Information as a potential input model may be used to yield fine-level adaptive effects
- Service Composition – with the emerging web service technologies it may be possible to adaptively compile new conglomerate services from pre-existing ones. Narrative may be used to encapsulate service composition techniques and strategies.
- Policy Refinement – policies may be used to control the behaviour of many computer systems such as Networks. Adaptive policy refinement may enable the semantically driven dynamic adaptive development of policies to manage aspects of large systems.

BIBLIOGRAPHY

[ADL] Advanced Distributed Learning (ADL) Sharable Content Object Reference Model (SCORM) (2004). Available online at http://www.adlnet.org/.

[AICC] Aviation Industry CBT Committee (AICC), Available online at http://www.aicc.org.

[Albert, 94] Albert, D. (1994). *Knowledge Structures*. Heidelberg, Germany: Springer.

[Albert and Lukas, 99] Albert, D. and Lukas, J. (1999*). Knowledge Spaces: Theories, empirical Research, and Applications.* Mahwah, NJ: Lawrence Erlbaum.

[APA, 97] APA (1997). *Learner-Centered Psychological Principles: A Framework for School Reform and Redesign*. Online available at http://www.apa.org/ed/lcp2/homepage.html.

[Anderson et al., 98] Anderson, John R., Reder, L. M., and Simon, H. A. (1998). *Applications and Misapplications of Cognitive Psychology to Mathematics Education.* Online available at http://act.psy.cmu.edu/personal/ja/misapplied.html.

[Apache] Apache HTTP Server Project. Available online at http://httpd.apache.org/.

[Atherton, 03a] Atherton, J. (2003) *Learning and Teaching: Learning index*. Available online at http://www.dmu.ac.uk/~jamesa/learning/.

[Atherton, 03b] Atherton, J. (2003) *Learning and Teaching: Knowles' Andragogy*. Available online at http://www.dmu.ac.uk/~jamesa/learning/knowlesa.htm.

[AVANTI] AVANTI Project. Available online at http://www.ifac.cnr.it/avanti/contents/over.htm

[Bancroft, 01] Bancroft, D., Healy, Y. (2001). *Caution on E-learning*. Irish Times – Education and Living, (Tuesday 17/04/01).

[Bechtel et al., 98] Bechtel, W., Abrahamsen, A., and Graham, G. (1998). *The Life of Cognitive Science*. In W. Bechtel and G. Graham (eds).

[Ben-Ari, 1998] Ben-Ari, M. (1998). *Constructivism in computer science education*. In Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education, Atlanta, Georgia, United States.

[Blackboard] Blackboard Inc.: Blackboard Course Management System (2002). Blackboard Inc. Available online at http://www.blackboard.com/.

[Brady et al., 04] Brady, A., Conlan, O., Wade, V. (2004*) Dynamic Composition and Personalization of PDA-based eLearning – Personalized mLearning*. In Proceedings E-Learn 2004, World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education, Washington, November 2004.

[Bruen, 02] Bruen, C. (2002). *Development Framework for Re-useable Learning Resources for Different Learning Styles and Requirements*. M.Sc. Thesis submitted to Trinity College, Dublin 2002.

[Bruen and Conlan, 02] Bruen, C.; Conlan, O. (2002). *Dynamic Adaptive ICT Support for learning Styles - A Development Framework for re-useable learning resources for different learning styles & requirements. ITTE 2002, Annual Conference of the Association of Information Technology for Teacher Education.*

[Bruner, 60] Bruner, J. (1960). *The Process of Education*. Cambridge, MA: Harvard University Press.

[Bruner, 66] Bruner, J. (1966). *Toward a Theory of Instruction*. Cambridge, MA: Harvard University Press.

[Bruner, 73] Bruner, J. (1973). *Going Beyond the Information Given*. New York: Norton.

[Bruner, 83] Bruner, J. (1983). *Child's Talk: Learning to Use Language*. New York: Norton.

[Bruner, 86] Bruner, J. (1986). *Actual Minds, Possible Worlds*. Cambridge, MA: Harvard University Press.

[Bruner, 90] Bruner, J. (1990). *Acts of Meaning*. Cambridge, MA: Harvard University Press.

[Bruner et al., 56] Bruner, J., Goodnow, J., and Austin, A. (1956). *A Study of Thinking*. New York: Wiley

[Brusilovsky, 93] Brusilovsky P., Pesin L., and Zyryanov M. (1993). *Towards an adaptive hypermedia component for an intelligent learning environment*. Human-Computer Interaction, Springer-Verlag, Berlin, 1993, pp. 348-358

[Brusilovsky, 94] Brusilovsky P. (1994). *Adaptive hypermedia: the state of the art*. In Proceedings of MHVR'94 Multimedia, Hypermedia and Virtual Reality (Moscow, Sep. 14-16, 1994), ICSTI, pp.24-29.

[Brusilovsky and Pesin, 94] Brusilovsky P. and Pesin L. (1994). *ISIS-Tutor: An adaptive hypertext learning environment*. In Proceedings of JCKBSE'94, Japanese-CIS Symposium on knowledge-based software engineering. (Pereslavl-Zalesski, May 10-13, 1994). Tokyo, 1994, pp 83-87.

[Brusilovsky et al., 96] Brusilovsky, P., Schwarz, E., Weber, G., (1996). *ELM-ART: An Intelligent Tutoring System on the World Wide Web*. In Proceedings of the Third International Conference, ITS, 1996.

[Brusilovsky, 96b] Brusilovsky, P. (1996). *Methods and techniques of adaptive hypermedia*. In P. Brusilovsky and J. Vassileva (eds.), Special Issue on Adaptive

Hypertext and Hypermedia, User Modeling and User-Adapted Interaction 6 (2-3), 87-129.

[Brusilovsky and Schwarz, 97] Brusilovsky, P. and Schwarz, E. (1997). *User as student: Towards an adaptive interface for advanced Web-based applications*. In: A. Jameson, C. Paris and C. Tasso (eds.) Proceedings of 6th International Conference on User Modeling, Chia Laguna, Sardinia, Italy, June 2-5, 1997 Wien: Springer-Verlag, pp. 177-188.

[Brusilovsky and De Bra, 98a] Brusilovsky, Peter and De Bra, Paul (1998*). Second Workshop on Adaptive Hypertext and Hypermedia.* Eindhoven University of Technology, NL. Online available at http://wwwis.win.tue.nl/ah98/Proceedings.html.

[Brusilovsky et al., 98b] Brusilovsky, P., Kobsa, A., and Vassileva, J. (1998*). Adaptive Hypertext and Hypermedia.* Dordrecht, NL: Kluwer Academic Publishers.

[Brusilovsky, 98c] P. Brusilovsky. (1998). *Adaptive educational systems on the world-wide web: A review of available technologies*. In Proceedings of Workshop "WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, TX, 1998.

[Brusilovsky et al., 98d] Brusilovsky, P., Eklund, J., and Schwarz, E. (1998). *Web-based education for all: A tool for developing adaptive courseware*. Computer Networks and ISDN Systems (Proceedings of Seventh International World Wide Web Conference, 14-18 April 1998) 30 (1-7), 291-300.

[Brusilovsky and Pesin, 98] Brusilovsky, P., Pesin, L. (1998). *Adaptive Navigation Support in Educational Hypermedia: An Evaluation of the ISIS-tutor*. In Journal of Computing and Information Technology, 6(1), 27-38, 1998.

[Brusilovsky and De Bra, 99] Brusilovsky, P., De Bra, P. (1999). *Second Workshop on Adaptive Systems and User Modeling on the World Wide Web.* Eindhoven University of Technology, NL. http://wwwis.win.tue.nl/asum99/contents.html.

[Brusilovsky, 99b] Brusilovsky, P. (1999). *Adaptive and Intelligent Technologies for Web-based Education*. Künstliche Intelligenz , 4 (1999) 19-25, available online at http://www2.sis.pitt.edu/~peterb/papers/KI-review.html.

[Brusilovsky, 01] Brusilovsky, P. (2001). *Adaptive hypermedia*. User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 87-110.

[Brusilovsky, 04] Brusilovsky, P. (2004). *KnowledgeTree: A Distributed Architecture for Adaptive E-Learning*. In Proceedings of the thirteenth International World Wide Web Conference, 2004.

[Brusilovsky et al., 05] Brusilovsky, P., Wade, V., Conlan, O. (2005) *From Learning Objects to Adaptive Content Services for E-Learning*. In International Journal on E-Learning (2005) (In press).

[Caprio, 94] Caprio, M.W. (1994). *Easing into constructivism, connecting meaningful learning with student experience*. Journal of College Science Teaching, 23 (4), pp210-212, 1994.

[Cheek, 92] Cheek, D.W. (1992) *Thinking Constructively About Science*. In Technology and Society Education. Albany, NY: State University of New York Press.

[Cheniti-Belcadhi et al. 04] Cheniti-Belcadhi, L., Henze, N., Braham, R. (2004) *An Assessment Framework for eLearning in the Semantic Web*. Proceedings of the Twelfth GI- Workshop on Adaptation and User Modeling in interactive Systems (ABIS 04), October 2004, Berlin, Germany.

[Clarke et al., 03] Clarke, L.; Wade, V.; Conlan, O.; Dagger, D. (2003). *Personalisation for Adult eLearning - an AHS Approach*. E-Learn 2003, World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education, Phoenix, November 2003.

[CLIPS] C Language Integrated Production System (CLIPS), http://www.ghg.net/clips/CLIPS.html.

[Collins et al., 96] Collins, A., Beranek, B., Newman Inc. (1996). *Design Issues for Learning Environments*. In International Perspectives on the Design of Technology-Supported Learning Environments, Lawrence Erlbaum Associates, 1996.

[Conklin, 87] Conklin, Jeff (1987). *Hypertext: An Introduction and Survey*. IEEE Computer, 20(9), 17-41.

[Conlan, 00] Conlan, O. (2000). *Novel Components for Supporting Adaptivity in Education Systems - Model-based Integration Approach*. In Proceedings of International Conference on Multimedia 2000, Los Angeles.

[Conlan et al., 01] Conlan, O., Hockemeyer, C., Lefrere, P., Wade, V., and Albert, D. (2001). *Extending educational metadata schemas to describe adaptive learning resources*. In Hugh Davies, Yellowlees Douglas, & David G. Durand (eds.), Hypertext 01: Proceedings of the twelfth ACM Conference on Hypertext and Hypermedia, pp. 161-162, New York: Association of Computing Machinery (ACM).

[Conlan et al., 02a] Conlan, O.; Wade, V.; Bruen, C.; Gargan, M. (2002). *Multi-Model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning*. In De Bra, P., Brusilovsky, P. and Conejo, R. (eds.) Proc. of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002) Proceedings, Málaga, Spain (2002) 100-111.

[Conlan et al., 02b] Conlan, O.; Hockemeyer, C.; Wade, V.; Albert, D.; Gargan, M. (2002) *An Architecture for integrating Adaptive Hypermedia Service with Open Learning Environments*. In Proceedings of ED-MEDIA 2002, World Conference on Educational Multimedia, Hypermedia and Telecommunications, Denver, Colorado, June 2002.

[Conlan et al., 02c] Conlan, O.; Dagger, D.; Wade, V. (2002) *Towards a Standards-based Approach to e-Learning Personalization using Reusable Learning Objects*. E-

Learn 2002, World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education, Montreal, September 2002.

[Conlan et al., 03a] Conlan, O.; Hockemeyer, C.; Wade, V.; Albert, D. (2003). *Metadata Driven Approaches to Facilitate Adaptivity in Personalized eLearning Systems*. In the Journal of the Japanese Society for Information and Systems in Education, 2003, http://www.jsise.org.

[Conway, 97] Conway, J. (1997). *Educational technology's effect on models of instruction* [online]. www: http://copland.udel.edu/~jconway/EDST666.htm.

[Cowie, 98] Cowie, F. 1998. *What's Within: Nativism Reconsidered*. Oxford: Oxford.

[Cristea and de Mooij, 03] Cristea, A.I., de Mooij, A. (2003) *Designer Adaptation in Adaptive Hypermedia Authoring* In the proceedings of the International Conference on Information Technology: Computers and Communications, ITCC'03, Las Vegas, USA, IEEE Computer Science.

[CSS] Cascading Style Sheets (CSS). Available online at http://www.w3.org/Style/CSS/.

[Cumming et al., 99] Cumming, G., Okamoto, T., and Gomez, L. (1999*). Advanced Research in Computers and Communications in Education.* Amsterdam: IOS Press.

[Czarkowski and Kay, 03] Czarkowski, M. and Kay, J., (2003). *Challenges of Scrutable Adaptivity*. In Proceedings of AIED Conference, 11th International Conference on Artificial Intelligence in Education, IOS Press 404 - 407.

[Dagger et al., 03] Dagger, D.; Conlan, O.; Wade, V. (2003). *An Architecture for Candidacy in Adaptive eLearning Systems to Facilitate the Reuse of Learning Resources*. In Proceedings of E-Learn 2003, World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education, Phoenix, November 2003.

[Dagger et al., 04] Dagger, D., Wade, V., Conlan, O. (2004). *Developing Adaptive Pedagogy with the Adaptive Course Construction Toolkit (ACCT)*. In Proceedings of the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004), Eindhoven, Netherlands (2004), 55-64.

[Dao and Parent, 98] Dao, Dr. K., Parent L. (1998) *Adaptive Learning Model for Workplace Environmental Learning*. In Proceedings of Edmedia, 1998.

[DC] Dublin Core Metadata Initiative (DC or DCMI). Available online at http://dublincore.org/.

[DCEd] DCMI Education Working Group (DCEd). Available online at http://dublincore.org/groups/education/.

[DDC] Dewey Decimal Classification System (13th Abridged) (DDC). Available online at http://www-lib.nearnorth.edu.on.ca/dewey/ddc.htm.

[De Bra, 96] De Bra, P. (1996). *Teaching Hypertext and Hypermedia through the Web*. In the Journal of Universal Computer Science 2, 12 (1996) 797-804.

[De Bra and Calvi, 97] De Bra, P., Calvi, L. (1997) *Creating Adaptive Hyperdocuments for and on the Web*. In Proceedings of the AACE WebNet'97 Conference, pp. 149-154, Toronto, 1997.

[De Bra and Calvi, 98] De Bra, P., Calvi, L. (1998). *AHA: a Generic Adaptive Hypermedia System*. In Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia, pp. 5-12, Pittsburgh, 1998. (Proceedings available as CSN 98-12, TUE, or online at http://wwwis.win.tue.nl/ah98/)

[De Bra and Ruiter, 01] De Bra, P., Ruiter, J.P. (2001) *AHA! Adaptive Hypermedia for All*. In Proceedings of the WebNet Conference, pp. 262-268, October 2001.

[De Bra et al., 02a] De Bra, P., Aerts, A., Smits, D., and Stash, N. (2002) *AHA! Version 2.0: More Adaptation Flexibility for Authors*. In: M. Driscoll and T. C. Reeves (eds.) Proceedings of World Conference on E-Learning, E-Learn 2002, Montreal, Canada, October 15-19, 2002, AACE, pp. 240-246.

[De Bra et al., 02b] De Bra, P., Aerts, A. T. M., Smits, D., Stash , N. (2002) *AHA! Meets AHAM*. In Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems(AH2002), 388-391.

[De Bra et al., 03] De Bra, P., Santic, T., Brusilovsky, P. (2003) *AHA! meets Interbook, and more...* In Proceedings of the AACE ELearn 2003 Conference, Phoenix, Arizona, November 2003, pp. 57-64.

[De Corte and Weinert, 96] De Corte, E. and Weinert, F. (1996*) International Encyclopedia of Developmental and Instructional Psychology.* Oxford, UK: Elsevier.

[De La Passardiere and Dufresne, 92] De La Passardiere, B., Dufresne, D. (1992) *Adaptive Navigational Tools for Educational Hypermedia*. ICCAL 1992: 555-567

[DeVries and Kohlberg, 87] DeVries, R.; Kohlberg, L. (1987) *Programs of Early Education: The Constructivist View*. Longman, New York, 1987.

[Dewey, 66] Dewey, J. (1966) *Democracy and Education*. Free Press, New York, http://www.ilt.columbia.edu/ academic/texts/dewey/d_e/contents.html,1966.

[Doignon and Falmagne, 99] Doignon, J.and Falmagne, J. (1999). *Knowledge Spaces.* Berlin: Springer Verlag.

[Dolog, 04] Dolog, P. (2004) *Model-Driven Navigation Design for Semantic Web Applications with the UML-Guide*. In Maristella Matera and Sara Comai (eds.), Engineering Advanced Web Applications. To appear.

[Dovey, 00] Dovey, M (2000).*Stuff about Stuff – the different meanings of 'metadata'*. VINE 116, 2000 pp 6-13.

[Driscoll, 94] Driscoll, M. P., (1994). *Psychology of Learning for Instruction*. Needham Heights, MA: Allyn & Bacon.

[DTD] Document Type Definition (DTD).

[Duffy and Jonassen, 92] Duffy, T., Jonassen, D. (1992). *Constructivism and the Technology of Instruction: A Conversation*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1992.

[EASEL] Educator Access to Services in the Electronic Landscape (EASEL). Education and training - Open platforms and tools for personalised learning. Funded under 5th FWP (Fifth Framework Programme) (IST-1999-10051).

[EC, 98] European Commission – *Fifth Framework Programme, Information SocietyProgramme Technologies for Knowledge and Skills Acquisition,* 1998 http://www2.echo.lu/telematics/education/en/interact/bul_5th2.html

[Eklund, 95] Eklund, J. (1995). *Cognitive models for structuring Hypermedia and implications for learning from the world wide web.* In Proceedings of AusWeb, 1995.

[Eklund et al., 97] Eklund, J., Brusilovsky, P., Schwarz, E. (1997). *A Study of Adaptive Link Annotation in Educational Hypermedia*. Available online at http://wwwis.win.tue.nl/ah/.

[Eklund and Brusilovsky, 98] Eklund, J., Brusilovsky, P. (1998). *The Value of Adaptivity in Hypermedia Learning Environments: A Short Review of Empirical Evidence*. http://wwwis.win.tue.nl/ah98/Eklund.html, 1998

[Eklund and Sawers, 96] Eklund J. and Sawers J. (1996). *Customising Web-Based Course Delivery in WEST with Navigation Support*. Paper presented at WebNet96, San Francisco, October 1996. p. 534-535.

[Elliott et al., 96] Elliott, S., Kratochwill,, T.., Littlefield, J., and T.(1996). *Educational Psychology: Effective Teaching - Effective Learning.* Madison: Brown & Benchmark.

[Espinoza and Hook 95] Espinoza F., Hook K. (1995). *An Interactive interface to an Adaptive Information System*. In Proceedings of the User Modelling for Information on the World Wide Web, a mini-workshop at the Fifth International Conference on User Modelling, 1995.

[Fink et al., 96] [Fink et al 96] Fink, J., Kobsa, A., Nill, A. (1996). *User-Oriented Adaptivity and Adaptability in the AVANTI project*. http://wwwis.win.tue.nl/ah/.

[Fleming, 87] Fleming, N. (1987), *Teaching and Learning Styles, VARK Strategies*. Available online at http://www.vark-learn.com.

[Fleming and Mills, 92] Fleming, N., Mills, C. (1992). *Not another inventory, rather a catalyst for reflection*. To improve the academy, 1992, (11), 137-149.

[Frankola, 01] Frankola, K. (2001). *Why online learners dropout*. Workforce, 10, 53-63.

[Fretwell-Downing Education] Fretwell-Downing Group (formally Fretwell-Downing Education). Available online at http://www.fdgroup.com/.

[Funk and Conlan, 03] Funk, P., Conlan, O. (2003) *Case-Based Reasoning to Improve Adaptability of Intelligent Tutoring Systems*. Adaptive Hypermedia Workshop at the HyperText 2003, The Fourteenth Conference on Hypertext and Hypermedia, Nottingham, UK.

[Garcia, 95] Garcia, T. (1995). *The role of motivational strategies in self-regulated learning*. In R. J. Menges & M. D. Svinicki (Eds.) Understanding self-regulated learning, New directions for teaching, 1995.

[GESTALT, 99] "Getting Educational Systems Talking Across Leading Edge Technologies", Work Package 3, D0301 Design and Specification of the RDS (Resource Discovery Service), 1999.

[Gilbert and Han, 99] Gilbert, J. E. and Han C. Y. (1999). *Arthur: Adapting Instruction to Accommodate Learning Style*. Proceedings of WebNet'99, World Conference of the WWW and Internet, Honolulu, HI, pp. 433-438.

[Gilliland-Swetland, 99] Gilliland-Swetland, A. (1999). *Defining Metadata*. Department of Library and Information Science, University of California, Los Angeles

[Gredler, 01] Gredler, M. E. (2001). *Learning and Instruction: Theory into Practice*. (4th ed.). New Jersey: Prentice Hall, Inc.

[Grunst, 93] Grunst, G. (1993) *Adaptive hypermedia for support systems*. In: M. Schneider-Hufschmidt, T. Kühme and U. Malinowski (eds.): Adaptive user interfaces: Principles and practice. Amsterdam: North-Holland, pp. 269-283.

[Halaz and Schwartz, 90] Halasz, F.and Schwartz, M. (1990) *The Dexter Reference Model*. In Proc. NIST Hypertext Standardization Workshop, pp. 95-133,1990.

[Harel, 91] Harel, I. (1991). *Children Designers. Interdisciplinary Constructions for Learning and Knowing Mathematics in a Computer-Rich School*, Ablex Publishing, Norwood, NJ, 1991.

[Harel and Papert, 91] Harel, I., Papert, S. (1991). *Constructionism*. Ablex Publishing, Norwood, NJ, 1991.

[Helander et al., 97] Helander, M., Landauer, T. and Prabhu, P. (1997). *Handbook of Human-Computer Interaction.* 2nd edition. Amsterdam, NL: Elsevier.

[Henze et al., 99a] Henze, N., Naceur, K., Nejdl, W. and Wolpers, M. (1999). *Adaptive hyperbooks for constructivist teaching*. Kunstliche Intelligenz, 26-31.

[Henze et al., 99b] Henze, N., Nejdl, W. and Wolpers, M. (1999) *Modeling Constructivist Teaching Functionality and Structure in the KBS Hyperbook System*. AIED99 Workshop on Ontologies for Intelligent Educational Systems, Le Mans, France, July 18-19, 1999

[Henze et al., 04] Henze, N., Dolog, P., Nejdl, W. (2004) *Reasoning and Ontologies for Personalized E-Learning*. Educational Technology & Society, 2004, Vol. 7, Issue 4.

[Henze and Nejdl, 04] Henze, N. and Nejdl, W. (2004)*A Logical Characterization of Adaptive Educational Hypermedia.* To appear in New Revies on Hypertext and Hypermedia (NRHM), 2004.

[Hockemeyer et al., 98] Hockemeyer, C., Held T., and Albert, D. (1998). *RATH A relational adaptive tutoring hypertext WWW-environment based on knowledge space theory*. Proceedings of CALISCE'98, 4th International conference on Computer Aided Learning and Instruction in Science and Engineering, Goteborg, Sweden, pp. 417-423.

[Holcombe et al., 98] Holcomb, L., Young, I., Kuhataparuks, S. (1998) *Computer Social Interaction and Individual Cognitive Growth: Vygotsky's Role of Language – Both Social and Private Speech*. In Proceedings of Edmedia 1998 pg 167.

[Honey, 86] Honey, P. and Mumford, A. (1986). *The Manual of Learning Styles*. Maidenhead.

[Hook, 97] Hook, C. (1997). *Evaluating the Utility and Usability of an Adaptive Hypermedia System*. In Proceedings of IUI Conference.

[Houben et al., 05] Houben, G.J., Aroyo, L., Dicheva, D., De Bra, P. (2005) *Adaptation Engineering in Adaptive Concept-based Systems*. Book chapter in: Adaptable and Adaptive Hypermedia Systems, edited by S.Y. Chen, G. Magoulas, p. 36-58, 2005, Idea Group.

[IEEE SUO] Standard Upper Ontology Working Group (SUO WG), IEEE P1600.1. Available online at http://suo.ieee.org/.

[IIS] Microsoft Internet Information Services (IIS). Available online at http://www.microsoft.com/windowsserver2003/iis/default.mspx.

[IMS Content Packaging] IMS Content Packaging v1.1.3. Available online at http://www.imsglobal.org/content/packaging/index.cfm.

[IMS LIP] IMS Learner Information Package (LIP) v1.0. Available online at http://www.imsglobal.org/profiles/index.cfm.

[IMS Metadata] IMS Meta-Data v1.3. Available online at http://www.imsglobal.org/metadata/index.cfm.

[IMS SS] IMS Simple Sequencing (SS) v1.0. Available online at http://www.imsglobal.org/simplesequencing/index.cfm.

[JDOM] Java Document Object Reference Model (JDOM) v1.0. Available online at http://www.jdom.org/.

[JSP] JavaServer Pages Technology (JSP). Available online at http://java.sun.com/products/jsp/.

[JESS] Java Expert System Shell (JESS) v7.0. Available online at http://herzberg.ca.sandia.gov/jess/.

[Kaplan et al., 93] Kaplan, C., Fenwick, J., and Chen, J. (1993). *Adaptive hypertext navigation based on user goals and context*. User Modeling and User-Adapted Interaction 3(3), 193-220.

[Kass and Stadnyk, 92] Kass, R. and Stadnyk, I. (1992). *Using User Models to Improve Organizational Information*. In Proceedings of the 3rd International Workshop on User Modeling. 1992, Dagstuhl, Germany.

[Katz, 96] Katz., A. (1996*). The Relationship Between Learner's Goal Orientation and Their Cognitive Tool Use and Achievement in an Interactive Hypermedia Environment.* In Proceedings of Edmedia 1996 pg 1697.

[Kayama and Okamoto, 98] Kayama, M. and Okamoto, T. (1998). *A mechanism for knowledge-navigation in hyperspace with neural networks to support exploring activities*. Proceedings of Workshop 'Current Trends and Applications of Artificial Intelligence in Education' at the 4th World Congress on Expert Systems, Mexico City, Mexico, pp. 41-48.

[Kay and Kummerfeld, 95] Kay, J. and Kummerfeld, R. (1995). *Intelligent agents derived from user models*. ED-MEDIA 95, June 18-21 Graz, Austria, 766 - 767.

[Kearsley, 99] Kearsley, G. (1999*). Explorations in Learning & Instruction: The Theory Into Practice Database.* Online available at http://www.gwu.edu/~tip/.

[Kirkwood, 98] Kirkwood, A. (1998). *New Media Mania: Can Information and Communication technologies enhance the quality of open and distance learning?*. In Distance Education – An International Journal, Vol.19, No.2, 1998, p.229.

[Knowles, 80] Knowles, M.S. (1980). *The modern practice of adult education*. New York: Cambridge, The Adult Education Company.

[Knowles, 84] Knowles, M.S. (1984). *Andragogy in Action*. San Francisco: Jossey-Bass.

[Knowles, 98] Knowles, M. (1998). *The adult learner. The definitive classic in adult education and human resource development*. Houston, TX: Gulf Publishing.

[Kobsa, 93] Kobsa, A. (1993). *User modeling: Recent Work, Prospects and Hazards*. In Adaptive User Interfaces: Principles and Practice, M. Schneider-Hufschmidt, T. Kühme, and U. Malinowski, (eds.). 1993, North-Holland: Amsterdam.

[Kobsa, 94] Kobsa, A. (1994) *KN-AHS: An Adaptive Hypertext Klient of the User Modelling System*. BGP-MS, Proceedings of 4th Conference on User Modeling.

[Kolb, 79] Kolb. D. A. (1979). *Organizational psychology: A book of readings*. Englewood Cliffs, N.J.: Prentice-Hall.

[Laurillard, 93] Laurillard, D. (1993). *Rethinking University Teaching: A Framework for the Effective Use of Educational Technology*, Routledge & Kegen, P., 1993.

[Laurillard, 02] Laurillard, D. (2002). *Rethinking University Teaching*, London: Routledge Falmer, 2002 2nd ed.

[LTSC] IEEE LTCS WG12: LOM: Working Draft Document v6.1, Learning Object Metadata Working Group of the IEEE Learning Technology Standards Committee (2001) available online at http://ltsc.ieee.org/wg12/doc.html

[MacLean, 90] MacLean (1990). *The Triune Brain in Evolution*. Plenum Press: New York

[Marchionini, 95] Marchionini, G. (1995). *The costs of educational technology: A framework for assessing change*. In H. Maurer (Ed.), Proceedings of Ed-Media 95, World conference of educational multimedia and hypermedia, Graz, Austria.

[Marco, 00] Marco, D. (2000). *Building and Managing the Metadata Repository*. In John Wiley Sons ISBN 0-471-35523-2.

[McCarthy, 87] McCarthy, B. (1987). *The 4mat system: Teaching to learning styles with right/left mode techniques*. Barrington, IL:Excel, Inc.

[McCarthy, 96] McCarthy, B. (1996). *The 4mat system research: reviews of the literature on the differences and hemispheric specialization and their influence on learning*. Barrington, IL: Excel, Inc.

[Meister, 02] Meister, J. (2002). *Pillars of e-learning success*. New York: Corporate University Exchange.

[Mesick, 76] Mesick, S. (1976). *Individuality in learning*. San Francisco: Jossey-Bass.

[Microsoft LRN] Microsoft LRN v3.0. Available online at http://www.microsoft.com/Education/LRN2Tutorial.aspx.

[Miller, 99] Miller, E. (1999). *Guidance on Expressing the Dublin Core within the Resource Description Framework (RDF)) Section 3. Enriching the Dublin Core* July 1999.

[Milosavljevic, 97] Milosavljevic, M. (1997). *Augmenting the user's knowledge via comparison*. In: A. Jameson, C. Paris and C. Tasso (eds.): Proceedings of 6th International Conference on User Modeling, UM97. Wien: SpringerWienNewYork, pp. 119-130.

[Milosavljevic et al., 97] Milosavljevic, M., Tulloch, A., Dale, R. (1997). *Text Generation in a Dynamic Hypertext Environment*. Available online at http://wwwis.win.tue.nl/ah/.

[Milosavljevic and Oberlander, 98] Milosavljevic, M., Oberlander, J. (1998). *Dynamic Hypertext Catalogues: Helping Users to Help Themselves*. In Proceedings of the 9th ACM Conference on Hypertext and Hypermedia. June 1998

[Myers and McCaulley, 85] Myers, I., McCaulley, M. (1985). *Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator*. Palo Alto, CA: Consulting Psychologists Press, Inc., 1985.

[Newby, 96] Newby, T. (1996). *Instructional technology for teaching and learning : designing instruction, integrating computers and using media*. Chapters 1 – 3. Prentice-Hall International (UK), 1996. ISBN 0023866950. 1996

[OPEN, 96] Open Learning Technology (1996). *Learning with Software: Pedagogies and Practice.* Online available at http://www.educationau.edu.au/archives/cp/.

[Oppenheimer, 97] Oppenheimer, T. (1997) *The Computer Delusion*. The Atlantic Monthly. July 1997.

[Ottmann, et al., 98] Ottmann, T. and Tomek, I. (1998). *Proceedings of ED-MEDIA/ED-TELECOM 98 - World conference on Educational Multimedia and Hypermedia & World Conference on Educational Telecommunications.*

Charlottesville, VA: Association for the Advancement of Computers in Education (AACE).

[OU] The Open University, http://www.open.ac.uk/.

[OWL] Web Ontology Language (OWL). Available online at http://www.w3.org/2004/OWL/.

[Paolucci, 98] Paolucci, R. (1998). *Hypermedia and Learning: The Relationship of Cognitive Style and Knowledge Structure.* In Proceedings of EdMedia, 1998.

[Papert, 80] Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas.* Basic Books, Inc., New York, 1980.

[Papert, 93] Papert, S. (1993) *The Children's Machine: Rethinking School in the Age of the Computer*. pp82-105, Basic Books Inc., New York, 1993.

[PAPI] Farance, F., Schoening, J. (2002). *Public and Private Information (PAPI) Specification*, Version 8.00, 2002.

[Parson, 98] Parson, R. (1998). *An investigation into instruction*. Available online at http://www.oise.utoronto.ca/~rparson/abstract.html.

[Pérez et al., 95] Pérez, T., Lopistéguy, P., Gutiérrez, J., and Usandizaga, I. (1995). *HyperTutor: From hypermedia to intelligent adaptive hypermedia*. ED-MEDIA'95, World conference on educational multimedia and hypermedia, Graz, Austria, pp. 529-534.

[Philips and Soltis, 03] Phillips, D., Soltis, J. (2003). *Thinking About Education--Perspectives on Learning.* 4th Edition, Chapter 3. Teachers College Press.

[Piaget, 70] Piaget, J. (1970). *The Science of Education and the Psychology of the Child*. Grossman, New York, 1970.

[Piaget, 73] Piaget, J. (1973). *To Understand is to Invent: The Future of Education*. Grossman, New York,1973.

[Pintrich and Schunk, 96] Pintrich, P. R. and Schunk, D. H. (1996). *Motivation in education: Theory, research, and application*. Englewood Cliffs, NJ: Prentice-Hall Inc.

[Postlethwaite, 93] Postlethwaite, K. (1993). *Differentiated Science Teaching*. Philadelphia: Open University Press.

[QTI] Question and Test Interoperability Specification V0.1 – Draft, Tan Boon Tee, IMS Asia Centre, 8/03/99;

[Reeves, 98] Reeves, T. (1998). *Answering critics of Media and Technology in Education*. In Proceedings of Edmedia, 1998.

[Rich, 89] Rich, E. (1989). *Stereotypes and User Modeling*. In: A. Kobsa and W. Wahlster: 1989, User Models in Dialog Systems, Heidelberg: Springer Verlag.

[Russell, 98] Russell, T. L. (1998) *The "no significant difference" phenomenon.* 4th ed., 1998. http://tenb.mta.ca/phenom/phenom.html.

[Sadler-Smith, 96] Sadler-Smith, E. (1996). *Learning Styles and Instructional Design*. Innovations in Education and Training International, Vol 33, number 4, November 1996, pp 185 – 193.

[Saettler, 90] Saettler, P. (1990). *The Evolution of American Educational Technology*. Englewood, CO: Libraries Unlimited, 1990.

[SCORM] Advanced Distributed Learning - Sharable Content Object Reference Model (SCORM) http://www.adlnet.org.

[Seefelder and Schwabe, 04] Seefelder de Assis, P. and Schwabe, D. *A General Meta-model for Adaptive Hypermedia Systems*. In Proceedings of the Third International

Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004), Eindhoven, Netherlands (2004), 433-436.

[Semantic Web, 01] *W3C Semantic Web Framework* (2001). http://www.w3.org/2001/sw/.

[Skinner, 76] Skinner, B. F. (1976). *About Behaviorism*. New York: Vintage.

[Skinner, 77] Skinner, B. F. (1977). *Why I am not a Cognitive Psychologist*. Behaviorism, 5, 1-10. (This journal is now known as ‘Behavior and Philosophy’.)

[Solomon, 92] Solomon, B. A., (1992). *Inventory of Learning Styles*. North Carolina State University.

[Solomon and Felder] Solomon, B. A., Felder, R. M., *Learning Styles and Strategies*. Available online at http://www.ncsu.edu/felder-public/ILSdir/styles.htm, North Carolina State University.

[Specht, 98a] Specht, M. (1998). *Empirical Evaluation of Adaptive Annotation in Hypermedia*. In Proceedings of Ed-Media, 1998.

[Specht, 98b] Specht, M. (1998). *Adaptive Methoden in computerbasierten Lehr-/Lernsystemen*. Ph.D. thesis, Universität Trier, Germany.

[Specht and Oppermann, 98] Specht, M. and Oppermann, R. (1998). *ACE - Adaptive Courseware Environment*. The New Review of Hypermedia and Multimedia 4, 141-161.

[Sperry, 97] Sperry R. W. (1997). *The impact and promise of the cognitive revolution*. In Yamaguchi, K., (Ed.), Sustainable global communities in the information age. Twickingham, UK: Adamantine Press Ltd.

[Stauffer, 96] Stauffer, K. (1996). *Student Modelling & Web-Based Learning Systems*. 1996.

[Tochtermann et al., 99] Tochtermann, K., Westbomke, J., Wiil, U. and Legget, J. (1999*). Hypertext'99: Returning to Our Diverse Roots.* New York: ACM.

[Tomcat] Apache Jakarta Tomcat. Available online at http://jakarta.apache.org/tomcat/.

[Tuckey, 92] Tuckey, C. (1992). *Uses of New Technology in Higher Education - Guiding Principles*. ICBL Reports, 001/92 Institute for Computer Based Learning, Heriot-Watt University, Edinburgh. 1992

[UDC] Universal Decimal Classification (UDC). Available online at http://www.udcc.org/

[UFI] University for Industry (UFI). Available online at http://www.ufi.com/.

[UoG] Karl Franzens Universität, Graz, Austria. Available online at http://www.kfunigraz.ac.at/homepage.html.

[Urban-Lurain, 02] Urban-Lurain, M. (2002). *Intelligent Tutoring Systems: An Historic Review*. In the Context of the Development of Artificial Intelligence and Educational Psychology. Available online at http://www.cse.msu.edu/rgroups/cse101/ITS/its.htm. Accessed June 2004.

[VARK] Visual, Auditory, Read/Write, Kinaesthetic (VARK) – A guide to Learning Styles. Available online at http://www.vark-learn.com/english/index.asp.

[Vassileva, 96] Vassileva, J. (1996). *A task-centered approach for user modeling in a hypermedia office documentation system*. In P. Brusilovsky and J. Vassileva (eds.), Spec. Iss. on Adaptive Hypertext and Hypermedia, User Modeling and User-Adapted Interaction 6 (2-3).

[Vdovjak et al., 03] Vdovjak, R., Frasincar, F., Houben, G.J., Barna, P. (2003) *Engineering Semantic Web Information Systems in Hera*. In: Journal of Web Engineering, Vol. 2, No. 1&2, p. 3-26, 2003, Rinton Press.

[Wade and Doherty, 97] Wade, V. P. and Doherty, P. (1997). *A Meta-data Driven Approach to Searching for Educational Resources in a Global Context*. In: Davies, G. and Owen, C. (eds.) Proc. of WebNet'2000, World Conference of the WWW and Internet, San Antonio, TX, AACE (2000) 136-145.

[Wade and Power, 98a] Wade, V.P. and Power, C. (1998). *Evaluating the Design and Delivery of WWW Based Educational Environments and Courseware*. In Proceedings of the 6th. Annual Conference on the Teaching of Computing, August 1998, Ireland, pp, 243-248.

[Wade and Power, 98b] Wade, V.P. and Power, C. (1998). *Network Based Delivery and Automated Management of Virtual University Courses*. pp 1421-1427 in Proceedings of the 10th World Conference on Educational Multimedia and Hypermedia & World Conference on Educational Telecommunications (ED-MEDIA & ED-TELCOM '98).

[WebCT] WebCT: WebCT Course Management System, Lynnfield, MA, WebCT, Inc. (2002) available online at http://www.webct.com

[Wedekind at al., 98] Wedekind, J., Lechner, M., Tergan, S. (1998). *Teaching and Learning with Flexible Hypermedia Learning Environments*. In Proceedings of Edmedia. 1998

[Xalan] Xalan-Java v2.6.0. Available online at http://xml.apache.org/xalan-j/.

[XDR] XML-Data Reduced (XDR) schema. Available online at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/xmlsdk/html/xmconxdr_whatis.asp.

[Xerces] Xerces Java Parser v1.4.4. Available online at http://xml.apache.org/xerces-j/.

[XHTML] The Extensible HyperText Markup Language (XHTML) (Second Edition). A Reformulation of HTML 4 in XML 1.0. Available online at http://www.w3.org/TR/xhtml1/.

[Xindice] Apache Xindice v1.1. Available online at http://xml.apache.org/xindice/.

[XML] Extensible Markup Language (XML). Available online at http://www.w3.org/XML/.

[XPath] XML Path Language (XPath) Version 1.0. Available online at http://www.w3.org/TR/xpath.

[XSD] XML Schema Definition (XSD). Available online at http://www.w3.org/XML/Schema.

[XSL] The Extensible Stylesheet Language Family (XSL). Available online at http://www.w3.org/Style/XSL/.

GLOSSARY

Abstraction When applied to narratives abstraction is the mechanism that keeps the concepts used in the narrative separate from the identifiers for individual learning resources. This enables narratives to be repurposed across different content bases.

Adaptive Adaptive programs adapt to a particular model or set of models, e.g. the skill level or preferences of the learner.

Adaptive Hypermedia Adaptive Hypermedia systems tailor information to the user and may guide the user in the information space to present the most relevant material. Abbrev. *AH*.

Adult Learning Theory Principles and practices of providing instruction to the adult learner. Primarily concerned with an adult's well-defined learning goals, wealth of experience and ability/desire to direct his or her own learning.

Andragogy The opposite of pedagogy. A term coined by Malcolm Knowles as the art and science of helping adults learn. A prime contributor to most theories of adult learning, andragogy as set out by Knowles emphasises adults' capabilities to direct and motivate themselves, utilise past knowledge to assist learning and evaluate the contents of training for relevance and quality.

APeLS Adaptive Personalised eLearning Service. An implementation of a personalised eLearning system based on the multi-model, metadata driven approach.

Candidate On of a number of models (content, narrative, sub-narrative, etc.) that fulfils a particular (usually educational or technical) function. Candidates are considered a model and have associated metadata.

Candidate Group A group of candidates. Groups are formed by grouping candidates that fulfil the same function, but perform it in different ways. Candidate Groups are metadata-only models. Their metadata contains the identifiers of their constituent candidates.

Candidate Selector A form of narrative that is capable of reasoning across the metadata of candidate models in a candidate group. This reasoning may utilise any model in the multi-model approach to select the most appropriate candidate(s) for a given function.

Collaborative Learning Learning through the exchange and sharing of information and opinions among a peer group.

Competencies	A structured list of knowledge, skills and attitudes that are possessed or required by a learner. Competencies are used as the foundation to guide needs analyses and evaluations.
Content Model	One of the three core models of the multi-model approach. The content model is the metadata describing a learning resource.
Curriculum	A group of related courses, often in a special field of study.
eLearning	Broad definition of the field of using technology to deliver learning and training programmes. Typically used to describe media such as CD-ROM, Internet, Intranet, wireless and mobile learning. Also known as *Web-based Training*, *Internet-based Training* and *Online Learning*.
Hypermedia	The combination of text, video, graphic images, sound, hyperlinks, and other elements in the form typical of Web documents. Hypermedia is the modern extension of hypertext, the hyperlinked, text–based documents of the original Internet.
Knowledge Domain	The content of a particular domain or field of knowledge.
Learner Model	One of the three core models of the multi-model approach. The learner model is the metadata describing an individual learner including their competencies, characteristics and preferences. It is an input model in the multi-model approach.
Learning Management System	A program that manages the administration of training and eLearning. Typically includes functionality for course catalogues, launching courses, registering students, tracking student progress and assessments. Abbrev. *LMS*.
Learning Object	Any digital resource that can be reused to support learning. Abbrev. *LO*.
Learning Style	An individual's unique approach to learning based on strengths, weaknesses, and preferences.
Metadata	Information that provides macro-level details about an object, such as author, title, subject, date created, etc. Usually used to describe models.
Model	A representation of an object, process, behaviour or attitude.
Multi-Model, Metadata	An approach for producing adaptive systems. This

driven approach	approach separates elements of adaptivity into discrete models. Utilises metadata to reason about models. Uses a generic engine to execute process models (narratives) and access other model metadata.
Narrative (Model)	One of the three core models of the multi-model approach. A process model that in personalised eLearning systems, such as APeLS, reconciles input models to form an output. May be considered the embodiment of adaptive logic and reasoning.
Pagelet	A discrete and fine grained Learning Object and associated metadata.
Pedagogy	Opposite of andragogy. The art and science of how children learn.
Personalisation	The process of being adaptive towards a user's characteristics and preferences. Personalised eLearning generally refers to personalisation towards a learner's objectives, prior knowledge and learning preferences.
Sub-narrative	A narrative that may be executed from within another narrative. May be seen as a mechanism for separating out reusable process to facilitate its reuse.
Virtual Learning Environment	A virtual learning environment is a set of teaching and learning tools designed to enhance a student's learning experience by including computers and the Internet in the learning process. Abbrev. *VLE*. See *Learning Management System*, also.

APPENDICES

APPENDIX I – Metadata

Metadata for describing Adaptivity Information Model

Nr	Name	Explanation	Multiplicity	Type	Note
1	**adaptivity**	Root element of metadata schema	no	-	
1.1	**adaptivitytype**	Names the adaptivity type and provides a reference to it through its attributes.	yes	-	Two attributes – name and ref
1.1.1	**set**	Type of set determined by attribute. May have sets or candidates as children.	yes	-	One attribute – type
1.1.2	**candidate**	Gives a value to the adaptivity type.	yes	LangString Type(100)	

Schema for Content Model

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to XML Data subset for IE 5-->
<Schema name = "adapt_ims_metadatav1p1_oc1.xdr"
         xmlns = "urn:schemas-microsoft-com:xml-data"
         xmlns:dt = "urn:schemas-microsoft-com:datatypes">
        <description><p> adapt_ims_metadatav1p1.xdr </p></description>
        <description><p> DTD supporting the EASEL XML binding of the EASEL Version of
the IEEE LTSC LOM Version 3.5 </p></description>
        <description><p> 2001-02-23 Owen Conlan: Modified for EASEL Project
</p></description>
        <description><p> Nothing is mandatory. </p></description>
        <description><p> Simple elements  </p></description>
        <description><p> extension: The extension element is used to create extensions
at anyplace in the XML instantiation.  If the extension contains only elements from
this DTD, maintaining those content models, then additional elements do not need to be
declared.  It is encouraged that extensions be created from the exisiting library of
elements whenever possible.  </p></description>
        <ElementType name = "extension" content = "mixed" model = "open"/>
        <ElementType name = "language" content = "mixed" order = "many">
                <description><p> language: A Human Language </p></description>
                <element type = "extension"/>
        </ElementType>
        <ElementType name = "adaptivitytype" content = "eltOnly" order = "seq">
                <description><p> Added by Owen Conlan, 2001-02-23 </p></description>
                <description><p> adaptivitytype: Type of Adaptivity, e.g. Navigation,
Structural, Presentation etc. </p></description>
                <AttributeType name = "ref" dt:type = "string"/>
                <AttributeType name = "name" dt:type = "string"/>
                <attribute type = "ref"/>
                <attribute type = "name"/>
                <group order = "one">
                        <element type = "set" minOccurs = "0" maxOccurs = "*"/>
                        <element type = "candidate" minOccurs = "0" maxOccurs = "*"/>
                        <element type = "langstring" minOccurs = "0" maxOccurs = "*"/>
```

```
		</group>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "aggregationlevel" content = "mixed" order = "many">
		<description><p> aggregationlevel: 0..3, 0 = raw media</p></description>
		<element type = "extension"/>
	</ElementType>
	<ElementType name = "langstring" content = "mixed" order = "many">
		<description><p> langstring: A string in a particular language
</p></description>
		<AttributeType name = "lang" dt:type = "string"/>
		<attribute type = "lang"/>
		<element type = "extension"/>
	</ElementType>
	<ElementType name = "STRING" content = "mixed" order = "many">
		<description><p> String with no defined language.  </p></description>
		<element type = "extension"/>
	</ElementType>
	<ElementType name = "catalogue" content = "mixed" order = "many">
		<description><p> catalogue: a reference index or listing
</p></description>
		<element type = "extension"/>
	</ElementType>
	<ElementType name = "datetime" content = "mixed" order = "many">
		<description><p> datetime: Date or time Per  W3C, e.g., 1999-08-07
</p></description>
		<element type = "extension"/>
	</ElementType>
	<ElementType name = "difficulty" content = "mixed" order = "many">
		<description><p> difficulty: A scale of 0..4, 0 = Very Easy
</p></description>
		<element type = "extension"/>
	</ElementType>
	<ElementType name = "metadatascheme" content = "mixed" order = "many">
		<description><p> The metadatascheme value should include version.
</p></description>
		<element type = "extension"/>
	</ElementType>
	<ElementType name = "size" content = "mixed" order = "many">
		<description><p> size: Integer, in bytes </p></description>
		<element type = "extension"/>
	</ElementType>
	<ElementType name = "location" content = "mixed" order = "many">
		<description><p> location: an Ordered list of places </p></description>
		<AttributeType name = "type" dt:type = "enumeration" dt:values = "URI
TEXT" default = "URI"/>
		<attribute type = "type"/>
		<element type = "extension"/>
	</ElementType>
	<ElementType name = "minimumversion" content = "mixed" order = "many">
		<description><p> minimumversion: the lowest or earliest
</p></description>
		<element type = "extension"/>
	</ElementType>
	<ElementType name = "maximumversion" content = "mixed" order = "many">
		<description><p> maximumversion: the highest or latest
</p></description>
		<element type = "extension"/>
	</ElementType>
	<ElementType name = "interactivitylevel" content = "mixed" order = "many">
		<description><p> interactivitylevel: a scale of 0..4, 0 = Very Low
</p></description>
		<element type = "extension"/>
	</ElementType>
	<ElementType name = "semanticdensity" content = "mixed" order = "many">
		<description><p> semanticdensity: number of concepts as a fucntion of
size of the resource. Scale of 0..4, 0 = Low </p></description>
		<element type = "extension"/>
	</ElementType>
	<ElementType name = "source" content = "eltOnly">
		<description><p> source: authoratative generator </p></description>
		<group order = "seq" minOccurs = "0" maxOccurs = "*">
			<element type = "extension"/>
		</group>
	</ElementType>
	<ElementType name = "id" content = "mixed" order = "many">
		<description><p> id: An alphanumeric identifier </p></description>
```

```
                <element type = "extension"/>
        </ElementType>
        <ElementType name = "vcard" content = "mixed" order = "many">
                <description><p> vcard: a STRING per vCard specification
</p></description>
                <element type = "extension"/>
        </ElementType>
        <ElementType name = "cost" content = "eltOnly" order = "seq">
                <description><p> HHHHHHHHHHHHHH Elements with content models of elements
HHHHHHHHHHH </p></description>
                <description><p> cost: is there?  A Boolean, yes|no </p></description>
                <element type = "langstring" minOccurs = "0" maxOccurs = "*"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "copyrightandotherrestrictions" content = "eltOnly" order =
"seq">
                <description><p> copyrightandotherrestrictions: are there any use
restrictions? A Boolean, yes|no </p></description>
                <element type = "langstring" minOccurs = "0" maxOccurs = "*"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "intendedenduserrole" content = "eltOnly" order = "seq">
                <description><p> intendedenduserrole: Teacher, Author, Learner, Manager
</p></description>
                <description><p> Ordered list </p></description>
                <element type = "langstring" minOccurs = "0" maxOccurs = "*"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "interactivitytype" content = "eltOnly" order = "seq">
                <description><p> interactivitytype: Direction of information flow
between user and resource </p></description>
                <element type = "langstring" minOccurs = "0" maxOccurs = "*"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "format" content = "eltOnly" order = "seq">
                <description><p> format, technical:MIME or non-digital
</p></description>
                <element type = "langstring" minOccurs = "0" maxOccurs = "*"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "status" content = "eltOnly" order = "seq">
                <description><p> status: state, e.g., preliminary, draft, final,
terminated </p></description>
                <element type = "langstring" minOccurs = "0" maxOccurs = "*"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "structure" content = "eltOnly" order = "seq">
                <description><p> structure: Logical structure, e.g., atomic,
hierachical, linear, parceled, network </p></description>
                <element type = "langstring" minOccurs = "0" maxOccurs = "*"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "centity" content = "eltOnly" order = "seq">
                <description><p> centity: The is the ENTITY element.  The word ENTITY is
reserved within XML, so the name has been changed to centity to stand for "Contributing
Entity".   </p></description>
                <description><p> Ordered list </p></description>
                <element type = "vcard" minOccurs = "0" maxOccurs = "1"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "entry" content = "eltOnly" order = "seq">
                <description><p> entry: an alphanumeric string, normally human readable
</p></description>
                <element type = "langstring" minOccurs = "0" maxOccurs = "*"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "purpose" content = "eltOnly" order = "seq">
                <description><p> purpose: reason or use </p></description>
                <element type = "langstring" minOccurs = "0" maxOccurs = "1"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "title" content = "eltOnly" order = "seq">
                <description><p> title: brief descriptive string </p></description>
                <element type = "langstring" minOccurs = "0" maxOccurs = "*"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "description" content = "eltOnly" order = "seq">
```

```
		<description><p> description: A textual summary. </p></description>
		<element type = "langstring" minOccurs = "0" maxOccurs = "*"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "keywords" content = "eltOnly" order = "seq">
		<description><p> keywords: Decriptive words and/or phrases.  One keyword
or keyphrase per langstring </p></description>
		<element type = "langstring" minOccurs = "0" maxOccurs = "*"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "coverage" content = "eltOnly" order = "seq">
		<description><p> coverage: Span, e.g., Spatial, temporal, cultural
</p></description>
		<element type = "langstring" minOccurs = "0" maxOccurs = "*"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "version" content = "eltOnly" order = "seq">
		<description><p> version: edition </p></description>
		<element type = "langstring" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "type" content = "eltOnly" order = "seq">
		<description><p> type: specifies kind </p></description>
		<element type = "langstring" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "name" content = "eltOnly" order = "seq">
		<description><p> name: Textual designation </p></description>
		<element type = "langstring" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "installationremarks" content = "eltOnly" order = "seq">
		<description><p> installationremarks: technical notes how to import or
set up </p></description>
		<element type = "langstring" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "otherplatformrequirements" content = "eltOnly" order =
"seq">
		<description><p> otherplatformrequirements: technical notes
</p></description>
		<element type = "langstring" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "role" content = "eltOnly" order = "seq">
		<description><p> role: Kind of participation or contribution
</p></description>
		<element type = "langstring" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "adaptivity" content = "eltOnly" order = "seq">
		<description><p> adaptivity: Information about adaptivity employed
</p></description>
		<element type = "adaptivitytype" minOccurs = "0" maxOccurs = "*"/>
		<element type = "description" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "learningresourcetype" content = "eltOnly" order = "seq">
		<description><p> learningresourcetype: Educationally specific resource
type.  Ordered list </p></description>
		<element type = "langstring" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "learningcontext" content = "eltOnly" order = "seq">
		<description><p> learningcontext: Primary Education, Secondary
Education, Higher Education, University First Cycle, University Second Cycle,
University Postgrade, Technical School First Cycle, Technical School Second Cycle,
Professional Formation, Continuous Formation, Vocational Training, Other
</p></description>
		<element type = "langstring" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "typicalagerange" content = "eltOnly" order = "seq">
		<description><p> typicalagerange: typical ages of intended users
</p></description>
		<element type = "langstring" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
```

```
        </ElementType>
        <ElementType name = "kind" content = "eltOnly" order = "seq">
                <description><p> kind: type or role </p></description>
                <element type = "langstring" minOccurs = "0" maxOccurs = "1"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "taxon" content = "eltOnly" order = "seq">
                <description><p> taxon: a specific node ata specific level in a
taxonomy. An ordered list of taxons comprise a taxonpath</p></description>
                <element type = "id" minOccurs = "0" maxOccurs = "1"/>
                <element type = "entry" minOccurs = "0" maxOccurs = "1"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "date" content = "eltOnly" order = "seq">
                <description><p> date: calendar and/or clock instant.  date may be
repeated when both DateTime and Description are used. </p></description>
                <element type = "datetime" minOccurs = "0" maxOccurs = "1"/>
                <element type = "description" minOccurs = "0" maxOccurs = "1"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "duration" content = "eltOnly" order = "seq">
                <description><p> duration: temporal span </p></description>
                <element type = "datetime" minOccurs = "0" maxOccurs = "1"/>
                <element type = "description" minOccurs = "0" maxOccurs = "1"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "typicallearningtime" content = "eltOnly" order = "seq">
                <element type = "datetime" minOccurs = "0" maxOccurs = "1"/>
                <element type = "description" minOccurs = "0" maxOccurs = "1"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "taxonpath" content = "eltOnly" order = "seq">
                <description><p> taxonpath: A taxonomic path of taxon nodes in a
specific classification or txonomy.  Taxonomies are typically hierachies of tems or
phases with increasinly finer resolutions deeper in the taxonpath.  There may be
different paths, in the same or different classifications, that describe the same
characteristic. </p></description>
                <element type = "source" minOccurs = "0" maxOccurs = "1"/>
                <element type = "taxon" minOccurs = "0" maxOccurs = "*"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "catalogentry" content = "eltOnly" order = "seq">
                <description><p> catalogentry: a specific entry from a specific
catalogue or index.  Ordered list </p></description>
                <element type = "catalogue" minOccurs = "0" maxOccurs = "1"/>
                <element type = "entry" minOccurs = "0" maxOccurs = "1"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "contribute" content = "eltOnly" order = "seq">
                <description><p> contribute: Type of contribution or participation as
defined by Role </p></description>
                <element type = "role" minOccurs = "0" maxOccurs = "1"/>
                <element type = "centity" minOccurs = "0" maxOccurs = "*"/>
                <element type = "date" minOccurs = "0" maxOccurs = "1"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "resource" content = "eltOnly" order = "seq">
                <description><p> resource: a specific target resource in a relation
</p></description>
                <element type = "description" minOccurs = "0" maxOccurs = "1"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "requirements" content = "eltOnly" order = "seq">
                <description><p> requirements: specific needs for use.  Usually
technical </p></description>
                <element type = "type" minOccurs = "0" maxOccurs = "1"/>
                <element type = "name" minOccurs = "0" maxOccurs = "1"/>
                <element type = "minimumversion" minOccurs = "0" maxOccurs = "1"/>
                <element type = "maximumversion" minOccurs = "0" maxOccurs = "1"/>
                <element type = "extension" minOccurs = "0" maxOccurs = "1"/>
        </ElementType>
        <ElementType name = "annotation" content = "eltOnly" order = "seq">
                <description><p> HHHHHHHHHHHHHHH  Category Elements HHHHHHHHHHHHHHH
</p></description>
                <description><p> annotation is a category element </p></description>
                <description><p> Evaluations or comments. </p></description>
                <element type = "centity" minOccurs = "0" maxOccurs = "1"/>
```

```
		<element type = "date" minOccurs = "0" maxOccurs = "1"/>
		<element type = "description" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "classification" content = "eltOnly" order = "seq">
		<description><p> classification is a category element </p></description>
		<description><p> Description or catalogueing of a characteristic.
</p></description>
		<element type = "purpose" minOccurs = "0" maxOccurs = "1"/>
		<element type = "taxonpath" minOccurs = "0" maxOccurs = "*"/>
		<element type = "description" minOccurs = "0" maxOccurs = "1"/>
		<element type = "keywords" minOccurs = "0" maxOccurs = "*"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "educational" content = "eltOnly" order = "seq">
		<description><p> educational is a Category element. </p></description>
		<description><p> Educational or pedagogic features of the resource.
</p></description>
		<description><p> 2000-04-17. language was ? is now * </p></description>
		<element type = "adaptivity" minOccurs = "0" maxOccurs = "1"/>
		<element type = "interactivitytype" minOccurs = "0" maxOccurs = "1"/>
		<element type = "interactivitylevel" minOccurs = "0" maxOccurs = "1"/>
		<element type = "learningresourcetype" minOccurs = "0" maxOccurs = "*"/>
		<element type = "semanticdensity" minOccurs = "0" maxOccurs = "1"/>
		<element type = "intendedenduserrole" minOccurs = "0" maxOccurs = "*"/>
		<element type = "learningcontext" minOccurs = "0" maxOccurs = "*"/>
		<element type = "typicalagerange" minOccurs = "0" maxOccurs = "*"/>
		<element type = "difficulty" minOccurs = "0" maxOccurs = "1"/>
		<element type = "typicallearningtime" minOccurs = "0" maxOccurs = "1"/>
		<element type = "description" minOccurs = "0" maxOccurs = "1"/>
		<element type = "language" minOccurs = "0" maxOccurs = "*"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "general" content = "eltOnly" order = "seq">
		<description><p> Category element general </p></description>
		<description><p> Context-independent features of the resource.
</p></description>
		<element type = "title" minOccurs = "0" maxOccurs = "1"/>
		<element type = "catalogentry" minOccurs = "0" maxOccurs = "*"/>
		<element type = "language" minOccurs = "0" maxOccurs = "*"/>
		<element type = "description" minOccurs = "0" maxOccurs = "*"/>
		<element type = "keywords" minOccurs = "0" maxOccurs = "*"/>
		<element type = "coverage" minOccurs = "0" maxOccurs = "*"/>
		<element type = "structure" minOccurs = "0" maxOccurs = "1"/>
		<element type = "aggregationlevel" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "lifecycle" content = "eltOnly" order = "seq">
		<description><p> Category Element lifecycle </p></description>
		<description><p> Features related to the life cycle of the resource.
</p></description>
		<element type = "version" minOccurs = "0" maxOccurs = "1"/>
		<element type = "status" minOccurs = "0" maxOccurs = "1"/>
		<element type = "contribute" minOccurs = "0" maxOccurs = "*"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "metametadata" content = "eltOnly" order = "seq">
		<description><p> metametadata is a category element </p></description>
		<description><p> Features of the description rather than the resource.
</p></description>
		<element type = "catalogentry" minOccurs = "0" maxOccurs = "*"/>
		<element type = "contribute" minOccurs = "0" maxOccurs = "*"/>
		<element type = "metadatascheme" minOccurs = "0" maxOccurs = "*"/>
		<element type = "language" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "relation" content = "eltOnly" order = "seq">
		<description><p> relation is a category element </p></description>
		<description><p> Features of the resource in relationship to other
resources. </p></description>
		<element type = "kind" minOccurs = "0" maxOccurs = "1"/>
		<element type = "resource" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "rights" content = "eltOnly" order = "seq">
		<description><p> rights is a category element </p></description>
		<description><p> Conditions of use of the resource. </p></description>
```

```
		<element type = "cost" minOccurs = "0" maxOccurs = "1"/>
		<element type = "copyrightandotherrestrictions" minOccurs = "0" 
maxOccurs = "1"/>
		<element type = "description" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "technical" content = "eltOnly" order = "seq">
		<description><p> technical is a category element </p></description>
		<description><p> Technical features of the resource. </p></description>
		<element type = "format" minOccurs = "0" maxOccurs = "1"/>
		<element type = "size" minOccurs = "0" maxOccurs = "1"/>
		<element type = "location" minOccurs = "0" maxOccurs = "*"/>
		<element type = "requirements" minOccurs = "0" maxOccurs = "*"/>
		<element type = "installationremarks" minOccurs = "0" maxOccurs = "1"/>
		<element type = "otherplatformrequirements" minOccurs = "0" maxOccurs = 
"1"/>
		<element type = "duration" minOccurs = "0" maxOccurs = "1"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "record" content = "eltOnly" order = "seq">
		<description><p> record is the Root element </p></description>
		<AttributeType name = "easelns" dt:type = "string" default = "http://" 
required = "yes"/>
		<attribute type = "easelns"/>
		<element type = "general" minOccurs = "0" maxOccurs = "1"/>
		<element type = "lifecycle" minOccurs = "0" maxOccurs = "1"/>
		<element type = "metametadata" minOccurs = "0" maxOccurs = "1"/>
		<element type = "technical" minOccurs = "0" maxOccurs = "1"/>
		<element type = "educational" minOccurs = "0" maxOccurs = "1"/>
		<element type = "rights" minOccurs = "0" maxOccurs = "1"/>
		<element type = "relation" minOccurs = "0" maxOccurs = "*"/>
		<element type = "annotation" minOccurs = "0" maxOccurs = "*"/>
		<element type = "classification" minOccurs = "0" maxOccurs = "*"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
	<ElementType name = "set" content = "eltOnly" order = "one">
		<AttributeType name = "type" dt:type = "enumeration" dt:values = "ALL 
ONE-OR-MORE NONE" default = "ALL"/>
		<attribute type = "type"/>
		<element type = "set" minOccurs = "0" maxOccurs = "*"/>
		<element type = "candidate" minOccurs = "0" maxOccurs = "*"/>
	</ElementType>
	<ElementType name = "candidate" content = "eltOnly" order = "seq">
		<element type = "langstring" minOccurs = "0" maxOccurs = "*"/>
		<element type = "extension" minOccurs = "0" maxOccurs = "1"/>
	</ElementType>
</Schema>
```

APPENDIX II – Vocabularies

Knowledge Domain Vocabulary for SQL

```
db.concepts.introduction.objectives
db.concepts.introduction.introduction
db.concepts.introduction.classification
db.concepts.relational_model.objectives
db.concepts.relational_model.sets
db.concepts.relational_model.further_reading
db.concepts.rdbms.objectives
db.concepts.rdbms.architecture
db.concepts.sql.syntax
db.tables.data_types.types
db.tables.data_types.string
db.tables.data_types.bit
db.tables.create.introduction
db.tables.create.example
db.tables.constraints.objectives
db.tables.constraints.example
db.tables.constraints.types.uniqueness
db.tables.constraints.types.example
```

```
db.tables.modify.introduction
db.views.objectives
db.views.introduction
db.views.advantages
db.views.disadvantages
db.views.how
db.createview.objectives
db.createview.introduction
db.createview.syntax
db.createview.example
db.tables.populate.dml.objectives
db.tables.populate.dml.introduction
db.tables.populate.dml.statements
db.tables.populate.dml.statements
db.tables.populate.insert.objectives
db.tables.populate.insert.introduction
db.tables.populate.insert.syntax
db.tables.populate.insert.example
db.tables.populate.update.objectives
db.tables.populate.update.introduction
db.tables.populate.update.syntax
db.tables.populate.update.example
db.tables.populate.delete.objectives
db.tables.populate.delete.introduction
db.tables.populate.delete.syntax
db.tables.populate.delete.example
db.retrieval.simple.select.objectives
db.retrieval.simple.select.introduction
db.retrieval.simple.select.syntax
db.retrieval.simple.select.example
db.retrieval.select.objectives
db.retrieval.select.introduction
db.retrieval.select.syntax
db.retrieval.select.clauses
db.retrieval.select.operators.objectives
db.retrieval.select.operators.introduction
db.retrieval.select.operators.operators
db.retrieval.select.operators.date
db.retrieval.select.operators.string
db.retrieval.select.operators.example
db.retrieval.clauses.where.objectives
db.retrieval.clauses.where.introduction
db.retrieval.clauses.where.syntax
db.retrieval.clauses.where.example
db.retrieval.clauses.where.search_conditions.objectives
db.retrieval.clauses.where.search_conditions.introduction
db.retrieval.clauses.where.search_conditions.predicates
db.retrieval.clauses.where.search_conditions.considerations
db.retrieval.clauses.where.search_conditions.examples
db.retrieval.joins.objectives
db.retrieval.joins.introduction
db.retrieval.joins.syntax
db.retrieval.joins.general
db.retrieval.joins.how
db.retrieval.joins.example
db.retrieval.simple.joins.objectives
db.retrieval.simple.joins.introduction
db.retrieval.simple.joins.syntax
db.retrieval.simple.joins.considerations
db.retrieval.simple.joins.example
db.retrieval.aggregate_functions.objectives
db.retrieval.aggregate_functions.introduction
db.retrieval.aggregate_functions.details
db.retrieval.aggregate_functions.standards
db.retrieval.aggregate_functions.types
db.retrieval.aggregate_functions.example
db.applications.embedded_sql.objectives
db.applications.embedded_sql.introduction
```

```
db.applications.embedded_sql.static_dynamic
db.applications.embedded_sql.embedding
db.applications.embedded_sql.c
db.applications.embedded_sql.structure.objectives
db.applications.embedded_sql.structure.structure
db.applications.embedded_sql.structure.constructs
```

Knowledge Domain Vocabulary for Physics / Mechanics

```
ElementaryNumberTheory.Concept
Functions.Terms
Functions.Concept
Functions.TableOfValues
Functions.2dGraph
Functions.RateOfChange.Concept
Function.RateOfChange.Linear
Function.RateOfChange.NonLinear
Analysis.Derivatives.Concept
Analysis.Derivatives.Calculation
Analysis.ImportantDerivatives
ElementaryAlgebra.Operations
TrigonometricFunctions.Terms
VectorArithmetic.Concept
VectorArithmetic.AlgebraicAddition
VectorArithmetic.GeometricAddition
VectorArithmetic.MultiplicationScalar
VectorArithmetic.VectorComponents
VectorArithmetic.UnitVector
VectorArithmetic.GeometricSubstraction
VectorArithmetic.DotProduct.Concept
VectorArithmetic.DotProduct.Calculation
VectorArithmetic.CrossProduct.Concept
VectorArithmetic.CrossProduct.Calculation
Units.Terms
Units.Concept
Particle.Concept
ReferenceOfFrame.Concept
Particle.Position1D
ReferenceOfFrame.CoordinateSystem2D
ReferenceOfFrame.CoordinateSystem3D
Displacement.Concept
Displacement.TimeCurve
AverageVelocity.Concept
AverageVelocity.Formula
Velocity.TimeCurve
InstantaneousVelocity.Concept
AverageAcceleration.Concept
AverageAcceleration.Formula
InstantaneousAcceleration.Concept
ConstantAcceleration.Formula
Displacement.Formula
Displacement.Curves
Displacement.VectorConcept2D
AverageVelocity.Vector
InstantaneousVelocity.Vector
ParametricRepresentation.Concept
AverageAcceleration.Vector
InstantaneousAcceleration.Vector
ProblemSolving.ReferenceOfFrame
ProblemSolving.Independencies
Dynamics.Concept
Newtons1Law.Concept
Newtons2Law.Concept
Newtons2Law.Formula
Newtons3Law.Concept
Problemsolving.PerpendicularForces
Problemsolving.FreeBodyDiagramm
```

```
Newtons3Law.Collision
Dynamics.NormalForce
Friction.Concept
Friction.Formula
CentripetalAcceleration.Concept
CentripetalMotion.Formula
CurvedMotion.Concept
Work.Concept
Work.Formula
KineticEnergy.Concept
KineticEnergy.Formula
ProblemSolving.Interpretation
Work.SpringBlockSystem
PotentialEnergy.Concept
ConservativeSystem.Concept
PotentialFields.Concept
Gravity.Concept
ProblemSolving.NewtonsLawsVsEnergyApproach
```

APPENDIX III – Narratives and Style sheets

APeLS Narrative for SQL Course

```
;; Function to add a new section to the DOM and populate
;; it with candidate group (cg) elements.
;;
;; parameters:  ?name -- The name of the section
;;             ?list -- A multifield containing an entry
;;                      for every cg in the section
;;

(deffunction add-subsection ( ?sectionname ?require ?list )
        ;; Declare variable to count page numbers in section
        ;; variable is set to zero initially
        (bind ?pageCounter 0)

        ;; if learner contains a match to the contents of ?require
        ;;      then add section to DOM
        ;;      else do nothing
        (if (eq (search-learner ?require "competencies.required") "TRUE" )
                then
                        (dom "add-to-parent" "subsection")
                        (dom "add-to-parent" "name" ?sectionname)
                        (dom "add-to-parent" "id" ?sectionname)


                        (foreach ?var ?list
                                (dom "add-to-parent" "card")
                                (dom "add-to-parent" "cg" ?var)
                                (dom "set-parent" "..")
                        )

                        (dom "set-parent" "..")
                else
                        (return 1)
        )
)

(dom "set-root" "course")
(dom "add-to-parent" "name" "SQL Course")

;; Add first sections to DOM
;; N.B. the function create$ creates a multifield
(dom "add-to-parent" "section")
(dom "add-to-parent" "name" "Database Concepts")

(add-subsection "Introduction" "db.concepts.introduction" (create$ cg011-000 cg012-000
cg013-000 cg014-000))
```

```
(add-subsection "The Relational Model" "db.concepts.relational_model" (create$ cg021-
000 cg023-000 cg023-001 cg023-002 cg023-003 cg025-000))

(add-subsection "Relational DBMS Architecture" "db.concepts.rdbms" (create$ cg031b-000
cg033-000))

(add-subsection "Structured Query Language" "db.concepts.sql.syntax" (create$ cg035b-
000 cg035b-001 cg035b-002))

;; Add second section to DOM
(dom "set-parent" "..")
(dom "add-to-parent" "section")
(dom "add-to-parent" "name" "Creating a Database")

(add-subsection "Data Types" "db.tables.data_types" (create$ cg051-000 cg052-000 cg052-
001 cg053-000 cg053-001 cg053-002 cg054-000 cg054-001 cg054-002 cg054-003 cg054-004
cg055-000 cg055-001))

(add-subsection "DDL : Create Statement" "db.tables.create" (create$ cg061-000 cg062-
000 cg062-001 cg063-000 cg063-001 cg064-000 cg064-001))

(add-subsection "Table Constraints" "db.tables.constraints" (create$ cg071-000 cg073-
000 cg073-001 cg073-002 cg074-000 cg074-006 cg074-007))

(add-subsection "DDL : Alter and Drop Statements" "db.tables.modify" (create$ cg075-000
cg076-000 cg076-001 cg077-000))

(add-subsection "Introduction to Views" "db.views" (create$ cg081-000 cg082-000 cg082-
001 cg082-002 cg083-000 cg083-001 cg084-000))

(add-subsection "Create View Statement" "db.createview" (create$ cg091-000 cg092-000
cg092-001 cg093-000 cg093-001 cg093-002 cg093-003 cg094-000))

;; Add third section to DOM
(dom "set-parent" "..")
(dom "add-to-parent" "section")
(dom "add-to-parent" "name" "Populating a Database")

(add-subsection "Introduction to Data Manipulation Language" "db.tables.populate"
(create$ cg101-000 cg102-000 cg102-001 cg103-000 cg103-001 cg103-002))

(add-subsection "Insert Statement" "db.tables.populate.insert" (create$ cg111-000
cg112-000 cg112-001 cg113-000 cg113-001 cg113-002 cg113-003 cg113-004 cg113-005 cg113-
006 cg113-007 cg113-008 cg115-000 cg115-001))

(add-subsection "Update Statement" "db.tables.populate.update" (create$ cg121-000
cg122-000 cg122-001 cg122-002 cg123-000 cg123-001 cg123-002 cg123-003 cg123-004 cg125-
000 cg125-001))

(add-subsection "Delete Statement" "db.tables.populate.delete" (create$ cg131-000
cg132-001 cg133-000 cg133-001 cg133-002 cg133-003 cg136-000 cg136-001))

;; add forth section to DOM
(dom "set-parent" "..")
(dom "add-to-parent" "section")
(dom "add-to-parent" "name" "Database Retrieval")

(add-subsection "The Simple Select Statement" "db.retrieval.simple.select" (create$
cg141-000 cg142-000 cg142-001 cg143-000 cg143-001 cg143-002 cg143-003 cg143-004 cg143-
005 cg143-006 cg144-000 cg144-001 cg144-002 cg144-003))

(add-subsection "The Select Statement" "db.retrieval.select" (create$ cg151-000 cg152-
000 cg152-001 cg153-000 cg153-001 cg153-002 cg153-003 cg154-000 cg154-001 cg154-002))

(add-subsection "Expression Operators" "db.retrieval.select.operators" (create$ cg161-
000 cg162-000 cg163-000 cg163-001 cg163-002 cg163-003 cg163-004 cg163-005 cg164-000))

(add-subsection "The Where Clause" "db.retrieval.clauses.where" (create$ cg171-000
cg172-000 cg172-001 cg172-002 cg172-003 cg173-000 cg173-001 cg174-000 cg174-001))

(add-subsection "Search Conditions" "db.retrieval.clauses.where.search_conditions"
(create$ cg181-000 cg182-000 cg183-000 cg183-001 cg183-002 cg183-003 cg183-004 cg183-
005 cg183-006 cg183-007 cg183-008 cg183-009 cg183-010 cg183-011 cg183-012 cg183-013
cg183-014 cg183-015 cg183-016 cg184-000 cg184-001 cg184-002 cg184-003 cg184-004 cg185-
000 cg185-001 cg185-002 cg185-003 cg185-004 cg185-005 cg185-006))
```

```
(add-subsection "An Introduction to Joins" "db.retrieval.joins" (create$ cg191-000
cg192-001 cg193-000 cg193-001 cg193-002 cg193-003 cg194-000 cg194-001 cg194-002 cg194-
003 cg194-004 cg195-000 cg195-001 cg195-002 cg195-003 cg197-000))

(add-subsection "An Introduction to Simple Joins" "db.retrieval.simple.joins" (create$
cg201-000 cg202-000 cg203-000 cg203-001 cg203-002 cg203-003 cg203-004 cg203-005 cg203-
006 cg204-000 cg204-001 cg204-002 cg204-003 cg204-004 cg204-005 cg205-001 cg205-002))

(add-subsection "Aggregate Functions" "db.retrieval.aggregate_functions" (create$
cg211-000 cg212-000 cg213-000 cg213-001 cg213-002 cg213-003 cg213-004 cg214-000 cg214-
001 cg214-002 cg214-003 cg214-004 cg215-000 cg215-001 cg215-002 cg215-003 cg216-000
cg216-001 cg216-002))

;; add fifth section to DOM
(dom "set-parent" "..")
(dom "add-to-parent" "section")
(dom "add-to-parent" "name" "Database Applications")

(add-subsection "Writing Database Applications using Embedded SQL"
"db.applications.embedded_sql" (create$ cg221-000 cg222-001 cg222-002 cg223-000 cg223-
001 cg224-000 cg224-001 cg225-001))

(add-subsection "Structure of Embedded SQL Applications"
"db.applications.embedded_sql.structure" (create$ cg231-000 cg232-000 cg233-000 cg233-
001 cg233-002))
```

APeLS Narrative for Physics / Mechanics Course

```
(deffunction add-unit (?name ?link)
   (if (eq (compare-attrib "competencies.learned" "competencies.learned" ?name) "TRUE")
       then
               (dom "add-to-parent" "unit")
               (dom "add-to-parent" "identifier" ?name)
               (dom "add-to-parent" "name" (get-content "subcollection2" ?name
"//*[name()='title']"))
               (dom "add-to-parent" "link" ?link)
               (if (eq (compare-attrib "competencies.taught" "competencies.learned"
?name) "TRUE")
                       then
                               (dom "add-to-parent" "status" "visited")
                       else
                               (dom "add-to-parent" "status" "unvisited")
               )

               (dom "add-to-parent" "competencies" (get-content "subcollection2" ?name
"//*[name()='adaptivitytype'][@name='competencies.taught']"))
               (dom "add-to-parent" "location" (get-content "subcollection2" ?name
"//*[name()='location']"))
               (dom "set-parent" "..")
       else
               (+ 1 1)
   )
)

(dom "set-root" "course")
(dom "add-to-parent" "name" "Mechanics Course")

(dom "add-to-parent" "section")
(dom "add-to-parent" "name" "Numbers, Functions and Graphs")
(add-unit "unit_nfg1" "nfg1.html")
(add-unit "unit_nfg2" "nfg2.html")
(add-unit "unit_nfg3" "nfg3.html")

;; The original Narrative has 12 sections (10 have been omitted for space reasons).

(dom "set-parent" "..")

(dom "add-to-parent" "section")
(dom "add-to-parent" "name" "Potential Energy and Fields")
(add-unit "unit_pef1" "p1pef1.html")
(add-unit "unit_pef2" "p1pef2.html")
(add-unit "unit_pef3" "p1pef3.html")
(add-unit "unit_pef4" "p1pef4.html")
```

APeLS Style sheet for SQL course

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    version="1.0"
    xmlns:redirect="org.apache.xalan.xslt.extensions.Redirect"
    extension-element-prefixes="redirect">

<xsl:variable name="parent">../ae-app/learner_<xsl:value-of select =
"//learnerid"/>_index.jsp</xsl:variable>

  <xsl:template match="/">
    <standard-out>
      Standard output:
      <xsl:apply-templates/>
    </standard-out>
  </xsl:template>

  <xsl:template match="course">
    <redirect:write file="{$parent}">
               <xsl:text disable-output-escaping="yes"><![CDATA[<%
session.setAttribute("coursetitle",new String("]]></xsl:text><xsl:value-of select =
"/course/*/title"/><xsl:text disable-output-
escaping="yes"><![CDATA["));%>]]></xsl:text>
               <xsl:apply-templates select = "/course/learnerid"/>
               <xsl:text disable-output-escaping="yes"><![CDATA[<%@ include
file="includes/index_header.jsp"%>]]></xsl:text>
               <xsl:text disable-output-escaping="yes"><![CDATA[<%@ include
file="includes/navigation.jsp"%>]]></xsl:text>
               <xsl:apply-templates select = "/course/ladder | /course/directedweb"/>
               <xsl:text disable-output-escaping="yes"><![CDATA[<%@ include
file="includes/navigation.jsp"%>]]></xsl:text>
               <xsl:text disable-output-escaping="yes"><![CDATA[<%@ include
file="includes/footer.jsp"%>]]></xsl:text>
    </redirect:write>
  </xsl:template>

  <xsl:template match="/course/ladder | /course/directedweb">
       <div class="Heading1"><xsl:value-of select = "title"/></div>
       <div class="Content">
       <ul type="square">
       <xsl:apply-templates select = "child::*"/>
       </ul>
       </div>
</xsl:template>

<xsl:template match="ladder | directedweb">
       <xsl:variable name="filename">learner_<xsl:value-of select =
"//learnerid"/>_<xsl:value-of select="identifier"/>.jsp</xsl:variable>
       <xsl:variable name="file">../ae-app/<xsl:value-of
select="$filename"/></xsl:variable>
       <redirect:write select="$file"><xsl:text disable-output-
escaping="yes"><![CDATA[<% session.setAttribute("title",new
String("]]></xsl:text><xsl:value-of select = "title"/><xsl:text disable-output-
escaping="yes"><![CDATA["));%>]]></xsl:text>
               <xsl:text disable-output-escaping="yes"><![CDATA[<%@ include
file="includes/header.jsp"%>]]></xsl:text>
               <xsl:text disable-output-escaping="yes"><![CDATA[<%@ include
file="includes/navigation.jsp"%>]]></xsl:text>
               <div class="Heading2"><xsl:value-of select="title"/></div>
               <div class="Content">
               <ul type="square">
                      <xsl:apply-templates select = "child::*"/>
               </ul>
               </div>
               <xsl:text disable-output-escaping="yes"><![CDATA[<%@ include
file="includes/navigation.jsp"%>]]></xsl:text>
               <xsl:text disable-output-escaping="yes"><![CDATA[<%@ include
file="includes/footer.jsp"%>]]></xsl:text>
       </redirect:write>
       <li><div class="Bullet"><a><xsl:attribute name = "href"><xsl:value-of
select="$filename"/></xsl:attribute><xsl:value-of select = "title"/></a></div></li>
</xsl:template>


<xsl:template match="location">
```

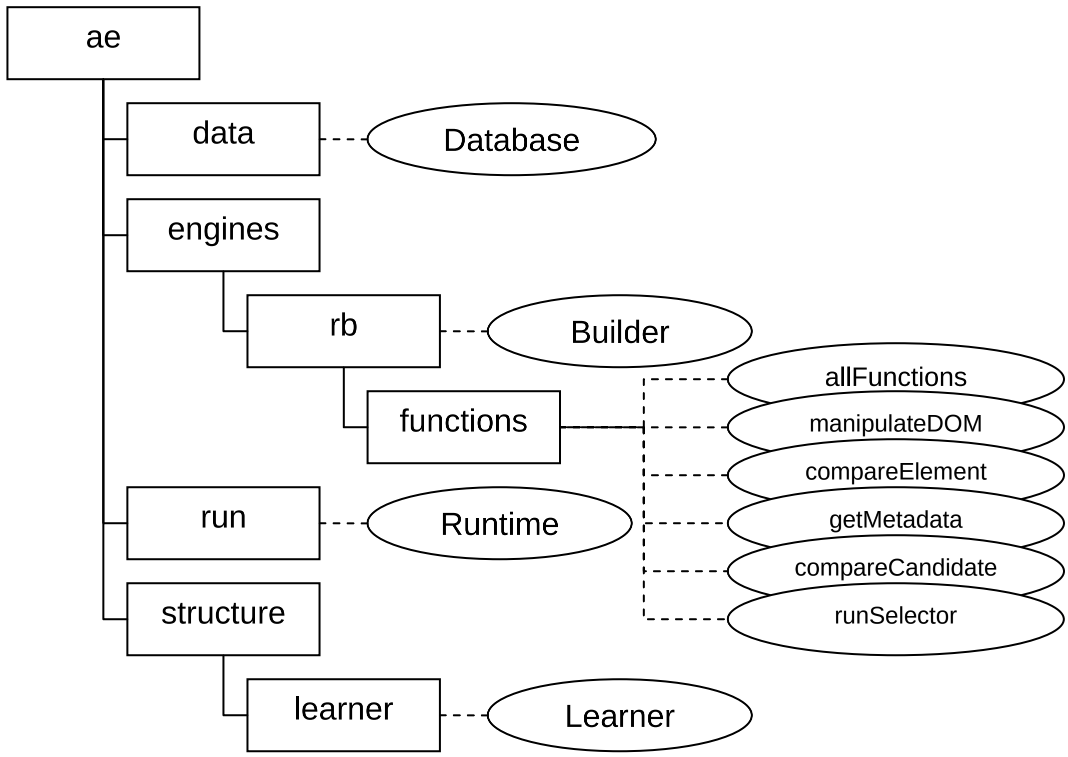
```
                <a><xsl:attribute name = "href">content/modules/<xsl:value-of
select="."/></xsl:attribute><xsl:value-of select = "."/></a>
</xsl:template>

<xsl:template match="learnerid">
                <xsl:text disable-output-escaping="yes"><![CDATA[<%
session.setAttribute("learnerid",new String("]]></xsl:text>
                <xsl:value-of select="."/>
                <xsl:text disable-output-escaping="yes"><![CDATA[")); %>]]></xsl:text>
</xsl:template>

<xsl:template match="pagelet | title | identifier">

</xsl:template>

<xsl:template match="childrengroup">
                <xsl:apply-templates select = "selectedchild"/>
</xsl:template>

<xsl:template match="selectedchild">
                <xsl:comment> START OF <xsl:value-of
select="child::*/identifier"/></xsl:comment>
                <div class="identifier"><xsl:value-of select =
"child::*/location"/></div>
                <xsl:text disable-output-escaping="yes"><![CDATA[<%@ include
file="]]></xsl:text>content/modules/<xsl:value-of select =
"child::*/location"/><xsl:text disable-output-escaping="yes"><![CDATA["%>]]></xsl:text>
                <xsl:comment> END OF <xsl:value-of
select="child::*/identifier"/></xsl:comment>
</xsl:template>

</xsl:stylesheet>
```

APeLS Style sheet for Physics / Mechanics course

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/course">
        <div class="course">
                <xsl:apply-templates select="section"/>
        </div>
</xsl:template>

<xsl:template match="section">
        <xsl:choose>
                <xsl:when test="count(unit) > 0">
                        <div class="section"><xsl:value-of select="name"/></div>
                        <xsl:apply-templates select="unit"/>
                </xsl:when>
                <xsl:otherwise/>
        </xsl:choose>
</xsl:template>

<xsl:template match="unit">
        <div class="unit">
                <a><xsl:attribute name="href">index.jsp?competencies=<xsl:value-of
select="competencies"/>&amp;link=<xsl:value-of
select="link"/>&amp;rebuild=false</xsl:attribute><xsl:attribute
name="class"><xsl:value-of select="status"/></xsl:attribute><xsl:value-of
select="name"/></a>
        </div>
</xsl:template>

</xsl:stylesheet>
```

APPENDIX IV – Implementation

Package and Class Hierarchy of APeLS



Resource Bundles

APeLS.properties File

```
maincollection=/apels
subcollection1=/learners
subcollection2=/pagelet_metadata
subcollection3=/narrative_metadata
subcollection4=/pagelet_groups
xsl_file=http://127.0.0.1:8080/aeapp/test.xsl
rule_file=/usr/java/jakarta-tomcat-4.0.3/webapps/aeapp/base.clp
candidate_selector1=/usr/java/Jakarta-tomcat-4.0.3/webapps/aeapp/vark.clp
```

APeLS_comparisons.properties File

```
competencies.learned=multiple subcollection2
//*[name()='adaptivitytype'][@name='competencies.learned']
//*[name()='adaptivitytype'][@name='competencies.required']

competencies.taught=multiple subcollection2
//*[name()='adaptivitytype'][@name='competencies.learned']
//*[name()='adaptivitytype'][@name='competencies.taught']

learningstyle.sequence=multiple subcollection3
//*[name()='adaptivitytype'][@name='learningstyle.sequence']
//*[name()='adaptivitytype'][@name='learningstyle.sequence’]
```

JSP Interfaces

SQL Course

```
<%@ page import="ae.structure.learner.Learner" %>

<%@ page import="ae.engines.rb.Builder" %>
<%@ page import="ae.data.Database" %>
<%@ page import="ae.run.Run" %>
<%@ page import="java.util.StringTokenizer" %>
<%@ page import="java.util.Vector" %>
<%@ page import="java.io.StringReader" %>
<%@ page import="org.jdom.Document" %>
<%@ page import="org.jdom.Element" %>
<%@ page import="org.jdom.output.XMLOutputter" %>
<%@ page import="org.jdom.transform.JDOMResult" %>
<%@ page import="org.jdom.transform.JDOMSource" %>
<%@ page import="org.jdom.input.SAXBuilder" %>
<%@ page import="org.w3c.dom.Node" %>
<%@ page import="org.xmldb.api.modules.XMLResource" %>
<%@ page import="javax.xml.transform.Transformer" %>
<%@ page import="javax.xml.transform.TransformerFactory" %>
<%@ page import="javax.xml.transform.stream.StreamSource" %>
<%@ page import="org.xmldb.api.base.XMLDBException" %>

<%@ page errorPage="error.jsp" %>

<%
        String content = "";
        Learner l = null;
        String learner = null;

        /* Get parameters */
        learner = request.getParameter("learner");

        Database dBase = new Database(  Run.resources.getString("maincollection"),
                                      Run.resources.getString("subcollection4") );


        try{
                /* Query Database */
                 String xmlContent = dBase.executeXPathQuery( learner, "/" )

        .getMembersAsResource().getContent().toString();
                dBase.deregister();
                StringReader xmlStream = new StringReader( xmlContent );

                 SAXBuilder builder = new SAXBuilder();
                Document contentDoc = builder.build( xmlStream );

                 /* Carry out transformation */

                Transformer transformer = null;
                try
                {
                        transformer   =   TransformerFactory.newInstance().newTransformer(
new StreamSource( Run.resources.getString( "xsl_file" ) ) );
                }
                catch( Exception e )
                {
                        out.print( "An error occured! " +e.toString() );
                }

                JDOMResult outputter = new JDOMResult();

                 /* pass parameter to transform */
                transformer.setParameter( "learner", learner );

                 transformer.transform( new JDOMSource( contentDoc ), outputter );

                 Builder b = new Builder();
                 content = b.XMLtoString( outputter.getDocument() );
        }
        catch(Exception e)
```

```
        {
                e.printStackTrace();
        }
%>

<html>
        <head>
                <title>
                        SQL course for <%= learner %>
                </title>
                <link rel=stylesheet type="text/css" href="./stylesheets/default.css">
                <META HTTP-EQUIV="PRAGMA" CONTENT="NO-CACHE">
        </head>
        <body>
                <%= content %>
        </body>
</html>
```

Physics / Mechanics Course

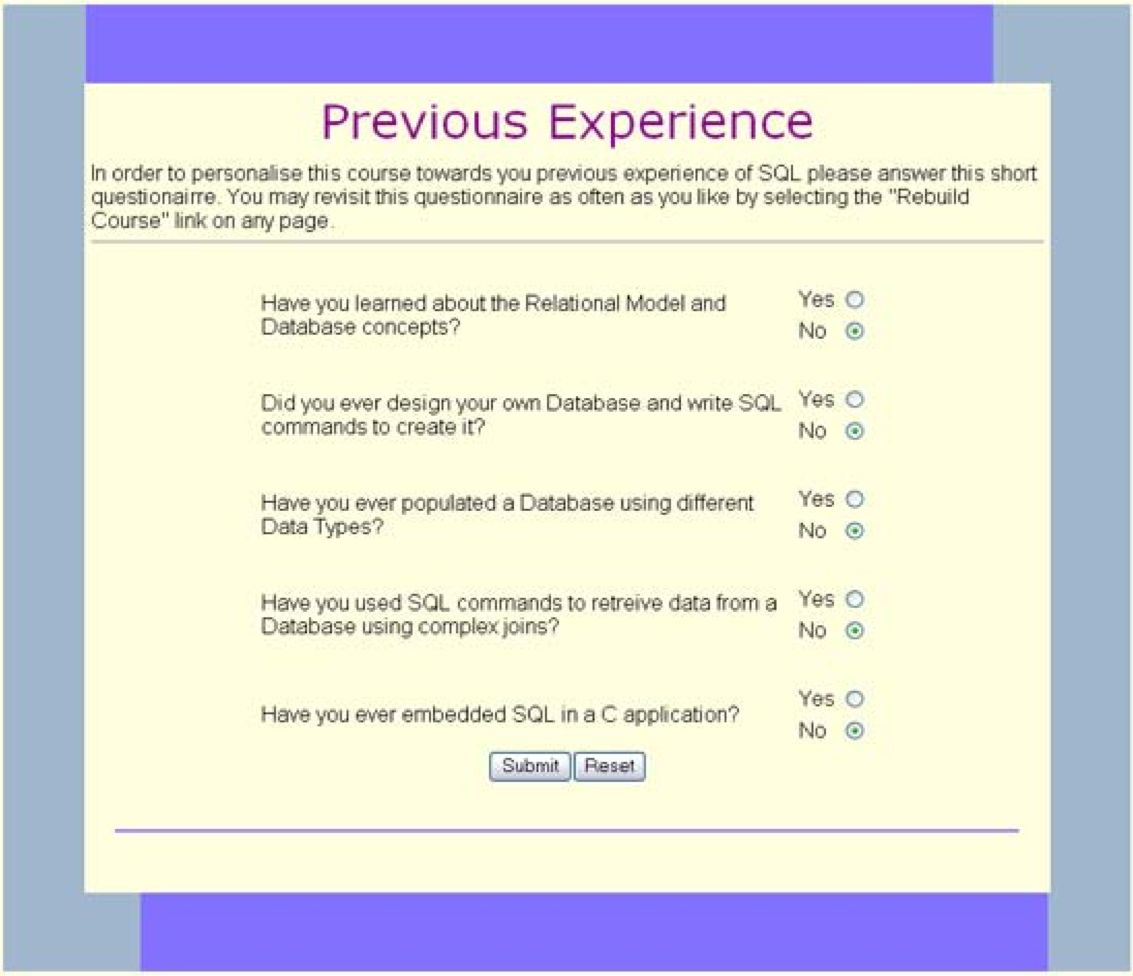
```
<%@ page import="ae.structure.learner.Learner" %>
<%@ page import="ae.engines.rb.Builder" %>
<%@ page import="ae.run.Runtime" %>

<%@ page import="java.util.StringTokenizer" %>
<%@ page import="org.xmldb.api.base.XMLDBException" %>

<%
        boolean newlearner = false;
        boolean cleansession = false;

        String currentlink = request.getParameter("link");
        if(currentlink == null) currentlink = "index.html";

        String learnerid = request.getParameter("learner");
        Learner l = null;
        if(learnerid == null){
                try{
                        l = (Learner) session.getAttribute("learner");
                        learnerid = l.getIdentifier();
                }
                catch(Exception e){
                    e.printStackTrace();
                }
        }
        else{
                try{
                        l = Runtime.getLearner(learnerid);
                        cleansession = true;
                }
                catch(XMLDBException ex){
                        newlearner = true;
                        l = new Learner();
                        l.setIdentifier(learnerid);
                        l.addAdaptivityType("competencies.learned", "None");
                        l.addAdaptivityType("competencies.learned",
"TrigonometricFunctions.Terms");
                        l.addAdaptivityType("competencies.learned",
"ElementaryAlgebra.Operations");
                        //ex.printStackTrace();
                        cleansession = true;
                }
                catch(Exception e){
                        newlearner = true;
                        l = new Learner();
                        l.setIdentifier(learnerid);
                        l.addAdaptivityType("competencies.learned", "None");
                        l.addAdaptivityType("competencies.learned",
"TrigonometricFunctions.Terms");
                        l.addAdaptivityType("competencies.learned",
"ElementaryAlgebra.Operations");
                        //e.printStackTrace();
                        cleansession=true;
                }
        }
```

```
        String competencies = request.getParameter("competencies");
        if(competencies != null){
                StringTokenizer st = new StringTokenizer(competencies, ",");
                while(st.hasMoreTokens()){
                        l.addAdaptivityType("competencies.learned",
st.nextToken().trim());
                }
        }

        String rebuild = request.getParameter("rebuild");
        if(rebuild == null) rebuild = "false";


         String links = "";
        try{
                links = (String) session.getAttribute("links");
        }
        catch(Exception e){
                        e.printStackTrace();
        }

        if(links == null || cleansession) rebuild = "true";

        if(rebuild.equals("true")){
                try{
                Builder b=new Builder(Runtime.resources.getString("rule_file"), l);
                b.build(Run.verbose);
                links=
                b.XMLtoString(b.transform(Runtime.resources.getString("xsl_file")));
                }
                catch(Exception e){
                        e.printStackTrace();
                }
        }


%>
<html>
<head>
        <title>Personalized Mechanics Course for <%= l.getIdentifier()%></title>
        <link rel="STYLESHEET" type="text/css" href="third.css" />
</head>
<body bgcolor=white>

<table border="0" border="100%" cellspacing="7">
<tr>
  <td width="280" height="20" bgcolor="silver" align="center"><div
class="section">Table of Contents</div></td>
  <td bgcolor="silver" align="Center"><div class="section">Content</div></td>
</tr>
<tr>
  <td valign="top" width="300" bgcolor="silver"><div><a
href="index.jsp?rebuild=true&link=<%= currentlink%>" class="unvisited">Rebuild
TOC</a></div><%= links %></td>
  <td valign="top"><jsp:include page="<%= currentlink %>" flush="true"/></td>
</tr>
</table>

<%session.setAttribute("learner", l);
  session.setAttribute("links", links);
  try{
        if(newlearner)
                Runtime.newLearner(l);
        else
                Runtime.updateLearner(l);
  }
  catch(XMLDBException ex){
                ex.printStackTrace();
  }
%>
</body>
</html>
```

APPENDIX V – Screenshots

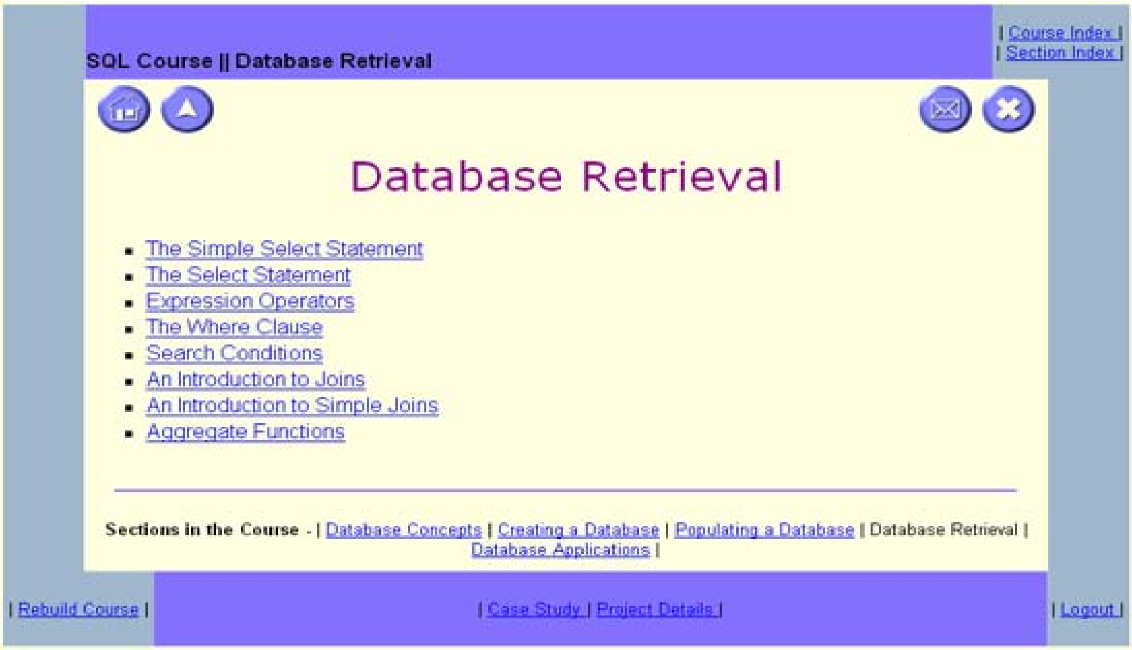
Personalised SQL Course

Screenshot: SQL Login Page

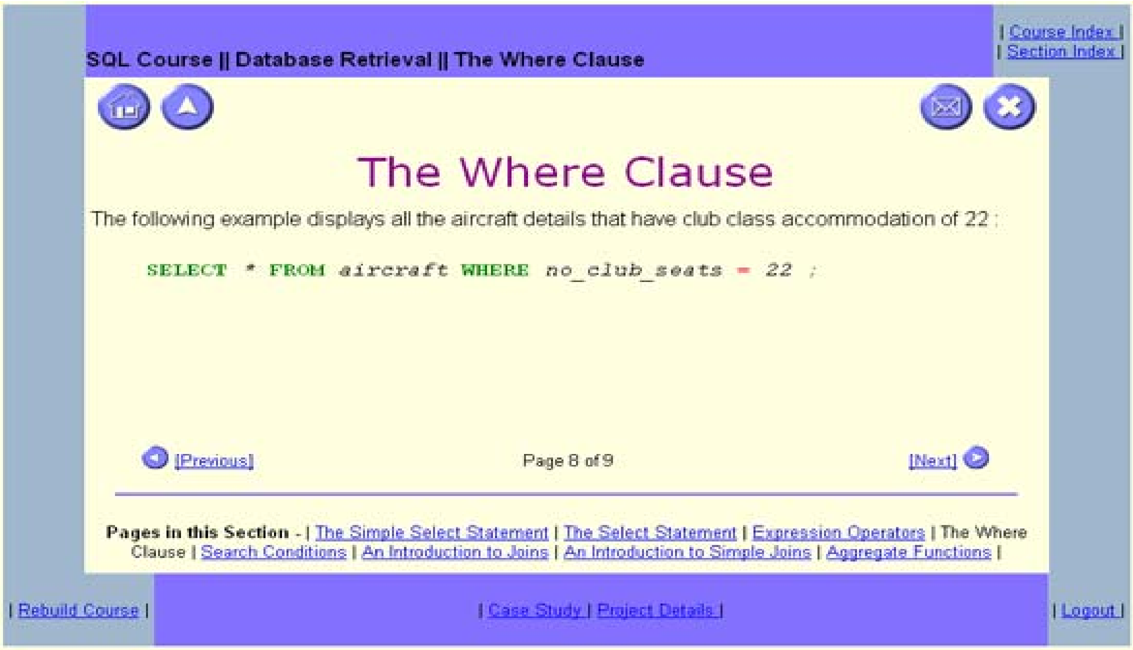


Screenshot: SQL Prior Knowledge Test

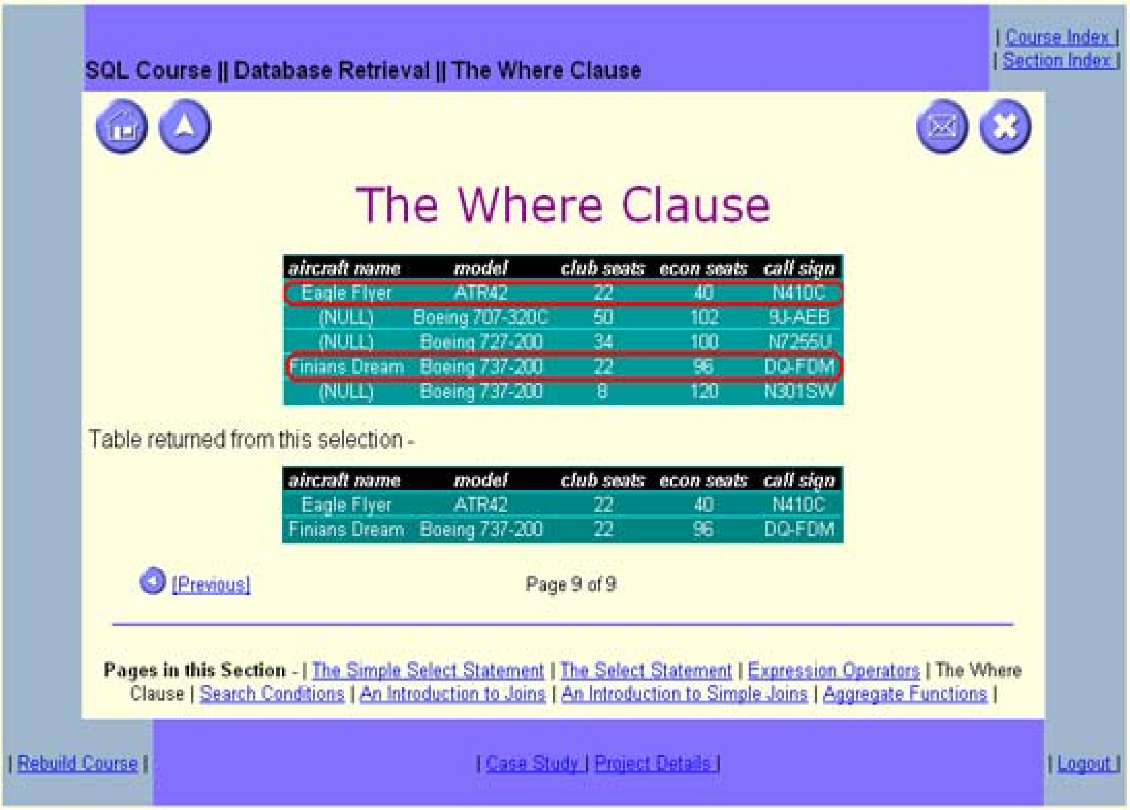
Screenshot: Top Level Personalised Table of Contents



Screenshot: Content List for *Database Retrieval* Section

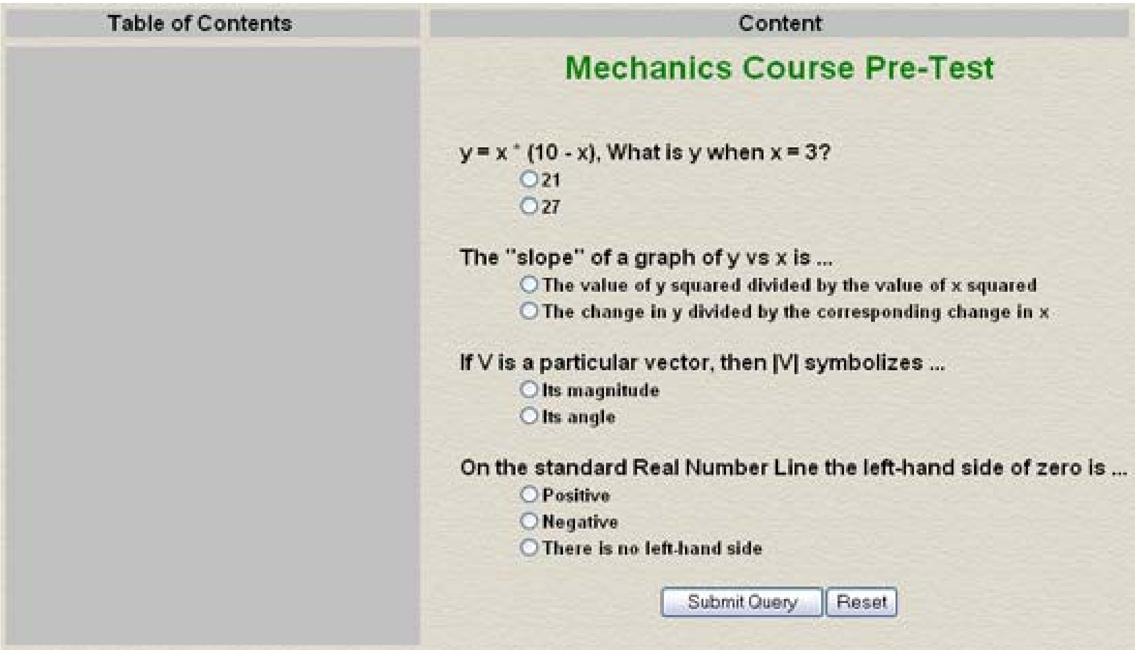


Screenshot: A pagelet from *The Where Clause* showing an SQL example

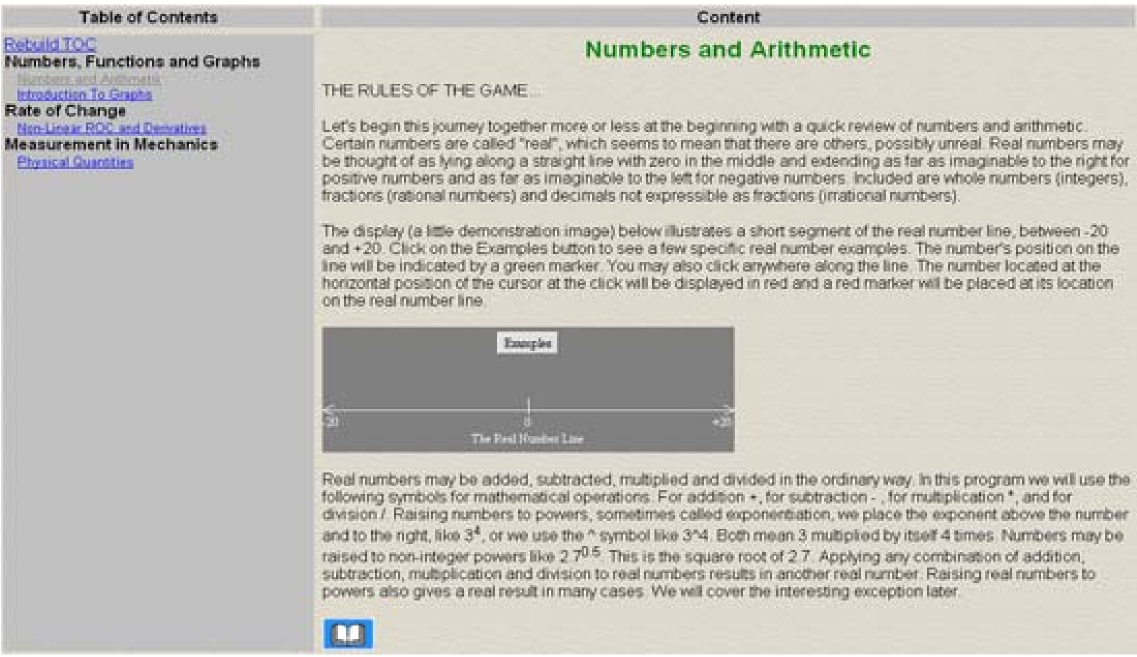


Screenshot: A pagelet from *The Where Clause* showing tables

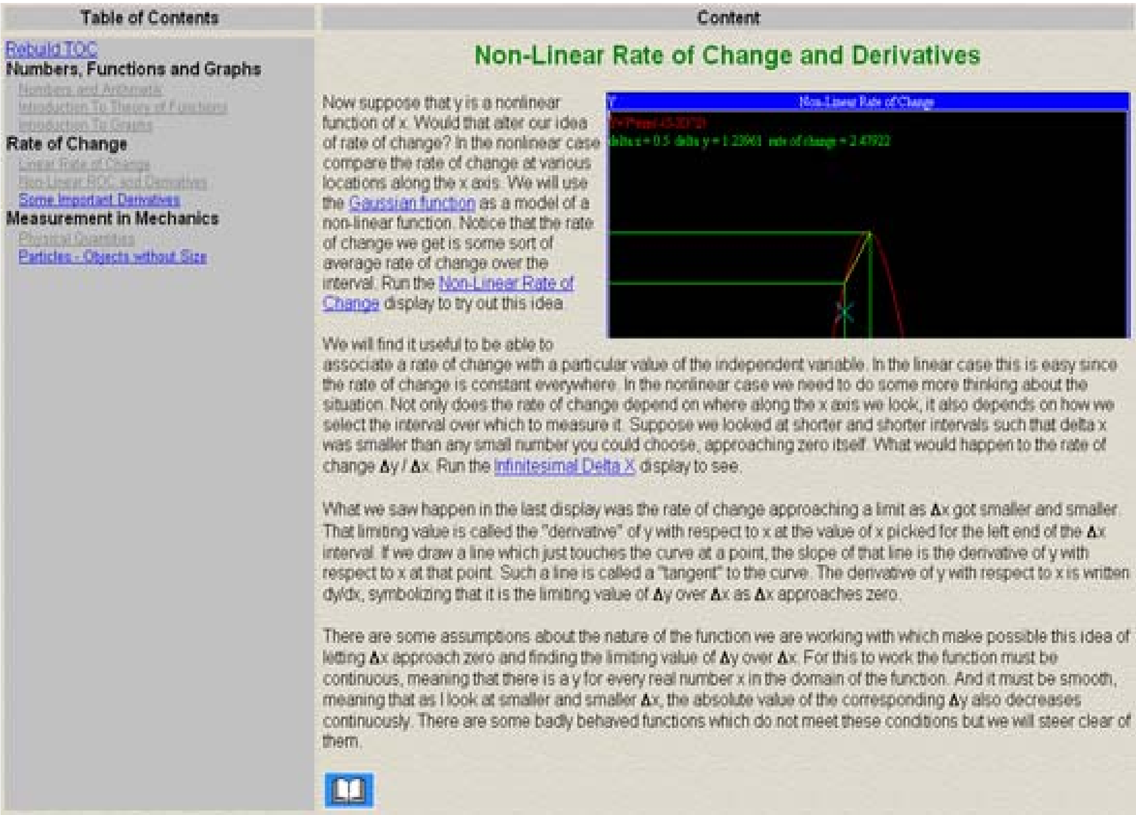
Personalised Physics / Mechanics Course



Screenshot: Physics / Mechanics Prior Knowledge Test



Screenshot: Physics / Mechanics Content showing Adaptive TOC



Screenshot: Physics / Mechanics Content showing Adaptive TOC growth

APPENDIX VI – Student Evaluation Questionnaire

Q1 How much experience do you have using online learning resources?

☐ None ☐ Little ☐ Some ☐ Much

Comments ____

Q2 Are you comfortable learning new course material via the Web?

☐ Not at all ☐ Not very ☐ Quite ☐ Very

Comments ____

Q3 How much experience in SQL did you have before commencing the Online SQL Course?

☐ None ☐ Little ☐ Some ☐ Much

Comments ____

Q4 After completing the initial online questionnaire, approximately how many times did you rebuild the course?

☐ 1-2 ☐ 3-4 ☐ 5-6 ☐ 6+

Comments ____

Q5 Were the objectives of the generated course(s) clear to you?

☐ Never ☐ Rarely ☐ Usually ☐ Always

Comments ____

Q6 Did the course(s) generated by the system reflect the answers you gave in the online course questionnaire?

☐ Never ☐ Rarely ☐ Usually ☐ Always

Comments ____

Q7 Did the course(s) generated by the system reflect the course(s) you wanted?

☐ Never ☐ Rarely ☐ Usually ☐ Always

Comments ____

Q8 Upon completion of the online course did you feel you had completed the objectives?

☐ Yes ☐ No

Comments ______________________________

Q9 Please rate the following aspects of rebuilding the course for usefulness (1-5; 1 = not useful, 5 = very useful).

_ Ability to modify the amount of content in the course

_ Ability to adjust the scope (number of sections) of the course

_ Availability of the Case Study in all courses

_ Maintenance of familiar link structure and layout

_ Availability of the Example Database for querying in all courses

Comments ______________________________

Q10 Were the courses generated easy to navigate?

☐ Never ☐ Rarely ☐ Usually ☐ Always

Comments ______________________________

Q11 Did the course content of the generated course(s) appear disjoint?

☐ Never ☐ Rarely ☐ Usually ☐ Always

Comments ______________________________

Q12 Would you have liked more control on the content included in the customised courses?

☐ Never ☐ Rarely ☐ Usually ☐ Always

Comments ______________________________

Q13 Did the course sections contain the content you expected?

☐ Never ☐ Rarely ☐ Usually ☐ Always

Comments ______________________________

Q14 Was the quantity of content on each page satisfactory?

☐ Never ☐ Rarely ☐ Usually ☐ Always

Comments ______________________________

Q15 Would you have liked a greater level of control as to how the content was structured? (i.e. the ability to place content in difference sections)

☐ Never ☐ Rarely ☐ Usually ☐ Always

Comments ______________________________

Q16 Would you have found the ability to modify the web interface beneficial? (i.e. placement of buttons, number and type of hyperlinks)

☐ Never ☐ Rarely ☐ Usually ☐ Always

Comments ______________________________

Q17 Please rate the course sections on how effectively you felt they represented the subject matter (1-5; 1 = not at all, 5 = completely).

_ Database Concepts
_ Creating a Database
_ Populating a Database
_ Database Retrieval
_ Database Applications

Comments ______________________________

Q18 Please rate the course features on how useful you found them (1-5; 1 = not useful, 5 = very useful).

_ Query Example Database
_ Example Project Solution
_ Case Study

Comments ______________________________

Q19 Are there any additional features you would like to see in the course?

☐ Yes ☐ No

Comments ______________________________

Q20 Did you experience any technical difficulties with the course?

☐ Yes ☐ No

Comments ____

Q21 Please use the space below to add any additional comments.

Thank you for answering the questionnaire.