Adaptive Composition of Personalised Learning Activities

A thesis submitted to the

University of Dublin, Trinity College

in fulfillment of the requirements for the degree of

Doctor of Philosophy

Ian O'Keeffe
Knowledge and Data Engineering Group,
School of Computer Science and Statistics,
Trinity College Dublin,
Ireland.

September 2012

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Acknowledgements

I would like to thank my supervisor, Prof. Vincent Wade, whose support and guidance has been invaluable from the very start. His experience and insight have been instrumental in ensuring that I continued along the right path.

I would also like to recognise the contributions that my collegues in the Knowledge and Data Engineering Group at TCD have made. Both directly through collaboration and indirectly through the sharing of their own research experience. They have provided me with a wealth of new ideas and new ways to look at problems. As James Watson, best known as a co-discoverer of the structure of DNA, once said "if you're the brightest person in the room, you're in trouble".

Finally I would like to pay thanks to my family for their love, support and understanding.

Ian O'Keeffe
University of Dublin, Trinity College
September 2012

Abstract

eLearning practitioners are increasingly adopting more activity based approaches to online learning, as they move away from more traditional content centric approaches in an effort to provide their learners with more engaging and effective learning. This trend can also be seen as part of the wider evolution of the internet from a content delivery platform to a more collaborative and creative environment.

However, although existing eLearning platforms provide some access to services they do not fully meet the needs of this new generation of eLearning. Platforms such as Moodle or Blackboard do not provide support for the sequencing of services as part of an activity, a requirement of pedagogically sound eLearning activities. Additionally, they represent a walled garden with a limited range of services available with which to implement activities. Existing approaches to the delivery of eLearning activities also lack support for personalisation. Technologies such as Adaptive Hypermedia have successfully demonstrated the benefits that personalisation of a content composition can provide to the learner. However, such technologies do not currently extend to the personalisation of service compositions.

The key motivation of this research is to address these limitations through an innovative approach to the adaptive composition of both multimedia content and services in a unified manner. This thesis describes the requirements for a system that can adaptively select and sequence multimedia content and services as part of a pedagogically driven eLearning activity. Based on these requirements this thesis presents the design and implementation of such a system.

To evaluate and validate this research a series of experiments were carried out and are described in this thesis. The evaluation experiments provide a means of analysing four key aspects of the system described; (i) the ability to support a range of eLearning activities;

(ii) support for the adaptive sequencing and selection of content; (iii) the support for the adaptive sequencing and selection of services; and (iv) the performance and scalability of the system.

An analysis of the results from this evaluation validates the approach taken in this research as it demonstrates the ability of the system to generate a range of personalised eLearning activities. Furthermore the evaluation shows that the system can deliver personalised eLearning in an effective manner without significant adverse impact on the educational or usability aspects of the composition.

Contents

List of Tables

List of Figures

Abbreviations

AE	Adaptive Engine
AH	Adaptive Hypermedia
AI	Artificial Intelligence
AJAX	Asynchronous JavaScript and XML
AHS	Adaptive Hypermedia System
AM	Adaptation Model
APeLS	Adaptive Personalised eLearning Service
API	Application Programming Interface
ASP	Active Server Pages
CAS	Computer Algebra System
CAWE	Context Aware Workflow Execution Framework
CPU	Central Processing Unit
DAML-S	DARPA Agent Markup Language for Services
DM	Domain Model
DOM	Document Object Model
ELF	E-Learning Framework
EMML	Enterprise Mashup Markup Language
ESB	Enterprise Service Bus
HCI	Human Computer Interaction
HTML	Hypertext Markup Language
HTN	Hierarchical Task Network
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
J2EE	Java Platform Enterprise Edition
JCP	Java Community Process
JDBC	Java Database Connectivity
JDOM	Java Document Object Model
JISC	Joint Information Systems Committee
JSON	JavaScript Object Notation
JSP	Java Server Pages
JSR	Java Specification Request
JVM	Java Virtual Machine
LADiE	Learning Activity Design in Education
LAN	Local Area Network
LIP	Learner Information Profile
LMS	Learning Management System
LOM	Learning Object Metadata

OMDoc	Open Mathematical Document
OWL-S	Web Ontology Language for Services
PC	Personal Computer
PDD	Process Definition Document
PDDL	Planning Domain Definition Language
PWE	Personalised Web Experience
RDF	Resource Description Framework
RPC	Remote Procedure Call
RSS	Really Simple Syndication
SaaS	Software as a Service
SCORM	Sharable Content Object Reference Model
SCP	Symbolic Computational Program
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
STRIPS	Stanford Research Institute Problem Solver
SUS	System Usability Scale
TWT	Tolerable Wait Time
UI	User Interface
UM	User Model
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WS	Web Service
WSBPEL	Web Service Business Process Execution Language
WSDL	Web Service Definition Language
WSFL	Web Service Flow Language
WSRP	Web Services for Remote Portlets
WSRP4J	Web Services for Remote Portlets for Java
XHTML	eXtensible Hypertext Markup Language
XML	eXtensible Markup Language
XMLDB	eXtensible Markup Language Database
XSL	eXtensible Stylesheet Language
XSLT	eXtensible Stylesheet Language Transformations
YAWL	Yet Another Workflow Language

Chapter 1

Introduction

1.1 Motivation

Users on the internet have become accustomed to a web that is more than an interactive hypermedia but is a integrated mix of rich multimedia services and hypermedia content. Users are now contributors and active participants on the web, communicating and interacting with each other using a range of services such as Gmail, MSN Messenger. They are no longer simply consumers of content but are also content creators and publishers, using content sharing and collaborative tools like Flickr, YouTube, wikis and blogs. This evolution is moving us towards the original vision for the internet as "a collaborative medium, a place where we [could] all meet and read and write" [Berners-Lee 05]. In this environment, users seamlessly move between interacting with services, for example voting, rating, annotating and communicating, to content interaction, viewing, navigating, downloading, etc. The internet is increasingly being used as a platform to carry out complex tasks that require users to combine services and related content in order to achieve their goals.

In the educational domain, the application of eLearning technologies has become increasingly more advanced. It has become inherently web based with a growing emphasis on activity based learning. An example of such an activity, within an educational context, is a peer review activity, which not only requires access to appropriate content but also requires services to support authoring, submission, annotation and discussion. Not only do these services need to be made available but they must be presented in a specific order according to the educational requirements of the activity.

There is an ongoing move from the simple use of such technology for the delivery of content via the internet using bespoke websites towards the use of more integrated environments such as Blackboard [Blackboard Inc. 10], Sakai [Sakai Foundation 10] and Moodle [Moodle Trust 10]. These systems provide environments that support the application of more complex pedagogical strategies by providing services that the educator can combine with their own content to produce learning experiences that engage the learner. In this way the educator can apply strategies such as activity based learning, which they commonly apply in the classroom, in their eLearning based teaching. However, such environments do not go far enough in addressing the needs of eLearning practitioners. Despite their support for services, they do not provide any means by which the author of a learning experience can control the sequencing of the services included in their activity. As such, these environments do not fully support activities such as peer review because they cannot control exactly when a specific service is provided to a Learner. Furthermore, they still suffer from the same problems as traditional content centric eLearning. They can only offer a one size fits all approach which does not take into account the needs of the user as an individual. The closed nature of such systems presents another limitation. They limit the educator to using the services that are provided by the system, they cannot take advantage of services that are not available outside of the environment provided by the system.

Personalisation technologies, such as Adaptive Hypermedia [Knutov 09, Brusilovsky 01], have been successfully shown to provide real benefits to the user by adaptively selecting and sequencing content to meet the needs of the user. However, such systems have so far focused almost exclusively on the adaptive delivery of interactive content, e.g. GALE [Smits 11], KnowledgeTree [Brusilovsky 05b] and PersonalReader [Henze 05]. As we move towards next generation web technologies, there is a need to provide a combination of adaptive selection and sequencing of multimedia content with adaptive selection and sequencing of user centric services. We define the notion of a **Personalised Web Experience** (PWE) as an experience that involves the integration of the personalised selection and presentation of content, personalised service adaptation and personalised service composition. Thus the personalised web experience provides a significant engagement of the user in carrying out activities on the web.

This research aims to provide a radical rethink of Adaptive Engines (AE) [Brusilovsky 01], where the AE supports adaptive composition of web services as well as multimedia web content. Such a next generation AE will effectively generate adaptive service workflows and adaptively compose content, seamlessly integrating the adaptive selection, composition and presentation of content and services. This work builds upon existing AE technology and integrates portal and semantic web business process and planning techniques to support the unified AE.

1.2 Research Question

The research question posed in this thesis asks "what are the appropriate techniques and technologies required to support the delivery of personalised web based experiences that combine adaptively sequenced and selected multimedia content with adaptively composed interactive services in a unified manner".

The focus of this research is on the design and implementation of a framework capable of generating and delivering personalised web experiences. The aim of this framework is to support the strategically driven, adaptive selection and sequencing of multimedia content and interactive, user centric services in order to deliver a web based experience that is tailored to the needs of the user.

1.3 Objectives

In answering the research question posed in this thesis, three objectives will be addressed. The first objective is to carry out a literature review of the state of the art in those research areas that are of significance to this work. Specifically, this review will investigate the technologies and techniques that can be used for the adaptive selection and composition of content and services for eLearning. The state of the art review will also investigate systems and techniques for the composition of services and the management of control flow between services.

The second objective is to research, iteratively develop and test an integrated adaptive system suitable for the pedagogically driven composition of multimedia content and interactive services. The developed system will also be capable of delivering the generated educational compositions to the user.

The third objective is to carry out a detailed evaluation of the implemented system. This evaluation will look at the complexity and performance of the system as well as the usability of the system from the perspective of the relevant stakeholders.

1.4 Contribution to State of the Art

This thesis makes two notable contributions to the state of the art in Adaptive Educational Hypermedia Systems (AEHS). The major contribution is the development of an innovative framework, which enables adaptive content presentation and navigation as well as flexible and extensible service composition in a unified environment. This framework facilitates the delivery of activity based eLearning through a web based environment.

The second minor contribution is the development of design concepts for the integration of adaptive behaviours in the composition of services. These design concepts, as illustrated by the developed framework, can be utilised in order to support both adaptive sequencing and adaptive selection of services.

1.5 Technical Approach

To achieve the specified research goals, an experimental approach was employed as illustrated in figure 1.1. The first step in the employed methodology was to carry out a survey of existing systems and techniques with respect to content centric approaches to the delivery of adaptive eLearning. This was then combined with requirements for learning activities. A survey of the techniques currently available for the composition of services was also carried out. This aspect of the survey investigated both the area of dynamic service composition and the orchestration of services.

Based on this survey a set of learning activities, designed by an expert community of practice in eLearning, were identified. These activities were analysed in order to identify requirements in terms of workflow support. Based on this analysis, a test implementation for an integrated adaptive service and content composition system was developed. This test implementation was then subject to an iterative development and testing process to reach a stable system.

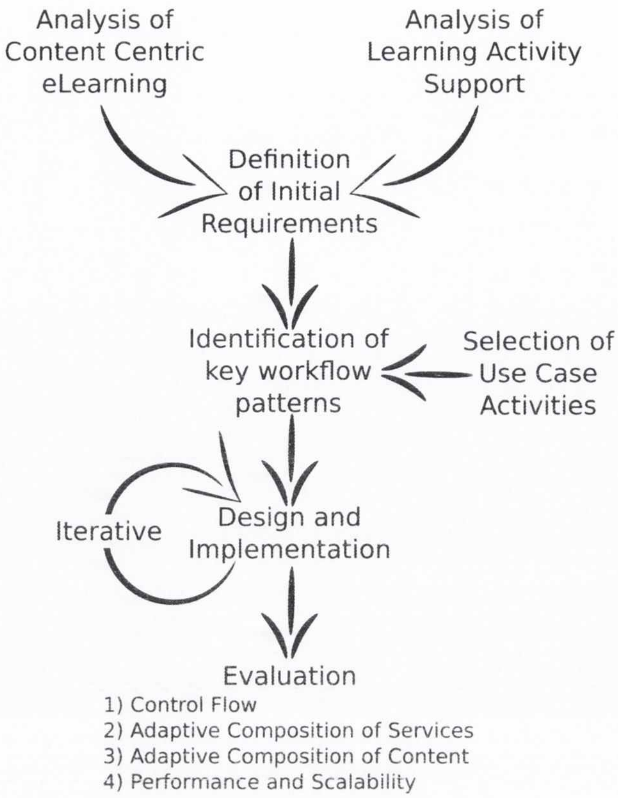

Analysis of Content Centric eLearning
Analysis of Learning Activity Support
Definition of Initial Requirements
Identification of key workflow patterns
Selection of Use Case Activities
Iterative
Design and Implementation
Evaluation
1) Control Flow
2) Adaptive Composition of Services
3) Adaptive Composition of Content
4) Performance and Scalability

Figure 1.1: Research Methodology

Following the development cycle a set of four experiments were devised and carried out in order to evaluate the developed system with respect to the research question.

The first experiment was an analysis of the service sequencing capabilities of the system based on the learning activities identified as part of the state of the art survey. The second experiment was based on the implementation of a set of learning activities as the basis for an analysis of the adaptive service selection and sequencing functionality of the system. The third experiment was a user study based evaluation of a system capable of adaptively selecting and sequencing content. The fourth experiment carried out as part of the evaluation consisted of a set of benchmarks designed to evaluate the performance and scalability of the system.

1.6 Thesis Outline

The rest of this thesis is structured as follows, the state of the art review of Adaptive Learning is provided in Chapter 2. This is followed by the review of the state of the art in Adaptive Service Composition in Chapter 3. The design of the prototype system is discussed in Chapter 4 with the details of its implementation provided in Chapter 5. Chapter 6 describes the evaluation of the research. The thesis concludes with a discussion of the conclusions and future work in Chapter 7.

Chapter 2

State of the Art - Adaptive Learning

2.1 Introduction

The use of the internet as a medium for accessing information has exploded in recent years providing more and more of the world's population with access to readily available information that they would not otherwise have access to. There are now 2 billion internet users worldwide, double the number from 5 years ago [Telecommunication Development Bureau 11]. This growth has also seen the internet become a popular tool for the delivery of both formal and informal learning. However, the very nature of the internet, access to massive volumes of content, can also present users with significant problems as they attempt to access information. Not only must they find appropriate content that covers the concepts that they are interested in but that content must also be in a form that is accessible to them.

Even in the context of formal learning environments on the internet, where the content is generally restricted to a closed corpus of high quality content, the learner can still have difficulty in making use of the information due, for example, to a lack of prior knowledge in the necessary prerequisite concepts. Many factors can result in a learner not being able to take full advantage of the information contained in the content, hardware restrictions such as network bandwidth or screen size can affect their ability to access some forms of content while disabilities or a lack of literacy can make some forms of content more appropriate relative to others.

Adaptive Hypermedia (AH) is an area of research that attempts to address these issues by adapting or personalising multimedia content, such as that found on the internet, based on the needs of the individual. This is achieved by capturing information that can be used to inform the process of personalising the content.

This chapter discusses the techniques and mechanisms used in AH to adapt content to the needs of the user. As part of this discussion, four mature Adaptive Hypermedia Systems are discussed in detail and their design and functionalities compared.

As a significant application domain for personalised web technologies, such as AH, a discussion of the eLearning domain is also presented. The discussion provides a summary of the main theoretical basis for learning theory and discusses their relevance with respect to the personalised web and activity based learning.

2.2 Learning to Support Personalisation

2.2.1 Learning Theory Perspectives

Learning Theory is the study of the psychological theory that underpins educational design. As with many areas of psychology, there is no single accepted view of how the human mind learns. Instead there are many different 'perspectives' on the nature of learning. These perspectives can be grouped together into clusters based on the fundamental assumptions that underpin each perspective. This section follows the approach of Greeno, Collins and Resnick [Greeno 96], who identified three clusters or broad perspectives: the Associationist/Empiricist perspective, the Cognitive perspective and the Situative perspective.

2.2.1.1 Associationist/Empiricist Perspective

The Associationist perspective views learning as the creation, reorganising and re-enforcement, through repetition, of associations between units of knowledge. In associationism, learning is considered to be the process of building associations between elementary units, which is achieved through sequences of activity. In associationism, learning tasks are constructed hierarchically with simpler tasks as pre-requisites of more complex tasks. This sequencing is achieved through 'task analysis', a process developed by Gagné [Gagné 85]. In this approach the assumption is that simpler tasks must be mastered before more complex ones.

One area of learning theory that can be aligned with the associationist perspective is that of Behaviourism [Watson 25], which views the mind as a 'black box' and defines learning as the change in behaviour resulting from the experiences of the learner. This perspective emphasises learning as an active process in which 'learning by doing' [Dewey 38][1] or activity based learning is the mechanism through which these changes in behaviour are made. Another important aspect of the Behaviourist approach is the provision of immediate feedback based on the activities of the learner. This emphasis on feedback to the user implies that the composition/sequencing of tasks for an individual learner should be personalised so that the learner's performance in prior tasks can be taken into account.

2.2.1.2 Cognitive Perspective

In the Cognitive perspective learning is seen as the interaction between concepts and the learner's mental model of the world. As such, it is how the learner processes the information that is important rather than the information itself, as is the case in associationism. As well as the learner's mental model, the cognitive perspective places importance on meta-cognition or 'thinking about thinking'. This is because, from a cognitive perspective, the challenge for the learner is to be build a framework for understanding the domain in question.

As each individual learner develops their own mental model and so makes sense of the world in their own way, the cognitive perspective can be seen to promote the idea of treating each learner as an individual and providing them with personalised activities that take into account their needs [Meyes 04].

Constructivism is an increasingly important aspect of the cognitive perspective, which places importance on activity as a means of building new forms of understanding. As the cognitive perspective in general sees learning as the construction of a broad mental model of a domain, it is important that the activities provided to a learner are authentic to the

[1] Although 'learning by doing' is not exclusively associated with Behaviourism, it is one of the theoretical perspectives that have arisen from it.

domain in question. Activities that are out of context do not allow the learner to develop the skills necessary to apply the knowledge in the actual domain. This view can be seen to draw from the situative perspective on learning theory and has lead to the development of the social constructivist paradigm, which is influenced by the work of Vygotsky. In the social constructivist paradigm, collaboration among learners as they take part in an activity is emphasised as social interaction is considered to be an important factor in constructing knowledge [Vygotsky 78].

Cognitive Constructivism is based on the work of Piaget [Piaget 70] and theorises that humans must construct their own knowledge rather than being able to immediately understand and use information that is given to them. This construction of new knowledge is achieved though experiences that allow the learner to create mental models, which can then be modified through processes referred to as assimilation and accommodation [Piaget 70]. Assimilation is the process of incorporating what is perceived in the outside world (concepts and experiences) into the learner's internal model without changing that model. As such, the learner can 'pigeon hole' what they perceive in order to make it fit into their mental model. Accommodation is the process of changing the internal model to fit the evidence that is presented to the learner.

In cognitive constructivism the teacher's role in a classroom is to provide a rich environment in which the learner is free to explore. This encourages the learner to become an active constructor of their own knowledge.

2.2.1.3 Situative Perspective

The situative perspective is based on the principle that all learning occurs in a social context and that the learner will be influenced by the social and cultural setting in which learning occurs. As such, the focus is on the way in which knowledge is distributed socially. Learning then becomes about the ability of an individual to successfully participate in the practices of a community. From this perspective learning must also be personally meaningful as the learner must have some reason for being part of the community in the first place.

Situated learning can be viewed as coming in one of two 'flavours' [Barah 00]. The first is a socio-psychological view in which learning is seen as activity based. The emphasis however is on the authenticity of the learning activities with respect to the social context in which the skills or knowledge are normally embedded. Problem based learning [Barrows 80] and anchored instruction [Crews 97] are examples of such activities.

The second is an anthropological view of situated learning, which places more emphasis on the relationships between the learner and the other members of a community of practice rather than on the activities themselves. The emphasis is then on how the learner's identity derives from being part of that community [Lave 91]. In order for this to happen, the learner must be provided with opportunities for legitimate participation in the community that allow them to move from simply observing the practices of the community to being active participants. Examples of this approach include apprenticeships, role play and debates.

2.2.2 Learning Activities

2.2.2.1 DialogPlus Project

The DialogPlus project [The Dialog Plus Team 04] took an activity based eLearning approach. The aim of the project was to develop a framework that would support the authoring of pedagogically informed learning activities. As part of this framework a model for the authoring of learning activities [Bailey 06] and a set of tools that apply that model to the authoring process [Conole 05] were developed. The project involved researchers from the Geography, Education and Computer Science domains from UK and US universities although the focus of the project was on

In the DialogPlus model, learning activities are referred to as 'Learning Nuggets' [Bailey 06] and consist of a sequence of tasks with each task providing access to specific resources and tools [Bailey 06]. Tasks are contextually delivered based on various properties including level of difficulty and prerequisite skills and are composed of a traditional media types such as text, images, audio and video as well as other digital media types such as interactive maps, Flash objects, databases and modelling applications. Learning Nuggets can also include quizzes, exercises, submission of written answers and communication tools such as discussion boards and email [Bailey 06].

Over the course of the DialogPlus project several learning activities were developed and used in the teaching of Geography to third level students. The topics covered by these activities included Academic Integrity and image processing for Global Information Systems analysis.

In addition to the technological contributions of the DialogPlus project, it also produced a taxonomy of learning activities. This taxonomy was designed to provide a basis for describing learning activities in a manner that would support reuse through a shared vocabulary for the description of activities. The taxonomy includes both pedagogical and technological taxa[2] including, Context (Aims, Environment, Difficulty, etc.), Learning Outcomes, Pedagogical Approaches, Type (what), Technique (how), interaction (who), Roles (which), Tools and Assessment.

The taxonomy provides a very detailed basis for the description of activities with each taxa have many sub types. An example of this detail is the Type taxon, which is broken down into 6 different sub types; Assimilative, Information Handling, Adaptive[3], Communicative, Productive and Experiential. Each of these has further subtypes providing a total of 35 different categories of task types.

The complexity of the DialogPlus taxonomy is in contrast to the 8LEM [D 05] taxonomy, which consists of only 8 types of task (Imitation, Reception, Creation, Exercising, Exploration, Debate, Experimentation, Self-reflection).

2.2.2.2 LADiE Project

The Learning Activity Design in Education (LADiE) project [Jeffery 06] was set up as part of the E-Learning Framework (ELF) [Elf 10], which was funded by the UK's Joint Information Systems Committee (JISC) and Australia's Department of Education, Science and Training (DEST). The aim of ELF was to "produce an evolving and sustainable, open standards based, service oriented technical framework to support the education and research communities". As part of ELF, the LADiE project aimed to support the authoring and realisation of Learning Activities. The authoring aspect of the project aimed to develop a Learning Activity Reference Model [Jeffery 06], which covered every aspect of the authoring process from design to packaging of the learning activity, including the discovery of resources and the specification of activity sequencing. To support the realisation of these learning activities, the project developed a specification for an execution

[2] Taxa is the plural of taxon, which is taxonomic category or group.

[3] 'Adaptive' in the context of the DialogPlus taxonomy means tasks that involve simulation or modelling.

environment in which such activities could be run. This environment was designed based on a component/service orientated approach.

To support the development of this reference model, the LADiE project ran a series of workshops with the aim of eliciting activity based eLearning case studies from learning practitioners. From these case studies, 16 use cases were developed, again in consultation with the learning practitioners. The definitions of these use cases are provided in appendix A.1.

Each of the LADiE case studies describe an authentic learning activity designed to be applied in an eLearning context. They describe the steps involved in the activity from the perspective of all of the stakeholders, students, teachers and, where necessary, technical support staff. The use cases also specify the teaching approaches on which the use case is based. In addition to the 16 main use cases, an additional set of use cases describe the steps involved in an online quiz and discussion activity, which are frequently reused in the 16 main use cases. Table 2.1 provides a list of the steps that make up LADiE use case 4. This is a discussion based activity in which the student(s) discuss a set of resources and write a report based on that discussion.

1	Teacher designs a scenario, collects appropriate resources, and saves them to the system
2	Teacher defines student groups and permissions for discussion (see usecase 'discussion')
3	Teacher briefs students on the activity and refers them to the resources and discussion forum on the system
4	Students log into system and access the resources
5	Students discuss the problem (see usecase 'discussion')
6	Teacher sends questions intended to stimulate discussion, and guidance on writing a report, to the forum
7	Students write report and save it to the system
8	System notifies teacher that report has been submitted and teacher retrieves it
9	System saves records of activity, discussion and student work for future access for quality assurance and benchmarking

Table 2.1: LADiE Use Case 4 Definition

An interesting aspect of many of the use cases is that they provide a set of alternative tasks within the activity that could be selected based on the learner's prior knowledge, abilities, the time available for the activity or the progress within the activity. For example in use case 4 it is suggested that the asynchronous discussion in step 5 could be replaced with a synchronous discussion (this could be realised using, for example, a chat tool or instant messaging). Similarly, it is suggested that students with reading/writing disabilities could use a VOIP tool such as Skype to carry out the discussion. Another example of possible adaptation to the context in which the activity is running is the suggestion that step 7, where the students save their work to the system could be replaced with the students emailing the report to the teacher.

The alternative paths and tasks specified in the use cases are interesting as the use cases were not written with the explicit intention to personalise the activities or to deliver them adaptively in any other way. It is apparent that the practitioners from whom the use cases were derived clearly saw the need to tailor the activities to the needs and context of the student.

As part of the LADiE project, a gap analysis was carried out on the set of use cases developed by the project. This was achieved through the use of the DialogPlus taxonomy to identify any areas that might not be well represented by the use cases [Falconer 06]. This analysis showed that the use cases were heavily discussion focused although it was not clear from the projects findings whether this was representative of the activities that learning practitioners apply in general or whether it was due to the methodology used to capture the use cases.

A significant part of the gap analysis was focused on analysing the types of task that were used in the LADiE use cases. The results of this analysis are reproduced in table 2.2 and represented visually in figure 2.1[4]. The graph shows the number of occurrences of each DialogPlus task type in the LADiE use cases. As shown the LADiE use cases show a strong representation of the Assimilative, Information Handling and Communicative task types and to a lesser degree the Productive task type. However, Experiential and Adaptive tasks are not well represented at all with only three experiential tasks and no adaptive tasks.

2.2.3 Analysis

The three perspectives on learning theory discussed associationist, cognitive and situative can be characterised as follows [Meyes 04]:

[4] Use Case 10 was not included in the original analysis carried out by the LADiE project due to its similarity to Use Case 16.

Use Case	Assimilative	Info. Handling	Adaptive	Communicative	Productive	Experiential
1	YES	YES		YES		
2		YES				
3	YES					
4	YES			YES	YES	
5		YES				
6	YES	YES		YES		
7	YES	YES		YES		
8	YES	YES		YES	YES	
9	YES			YES		
10	-	-	-	-	-	-
11				YES	YES	YES
12	YES			YES	YES	YES
13	YES			YES		
14	YES			YES	YES	
15	YES			YES		
16	YES	YES		YES	YES	

Table 2.2: LADiE Use Case Gap Analysis - DialogPlus Tasks

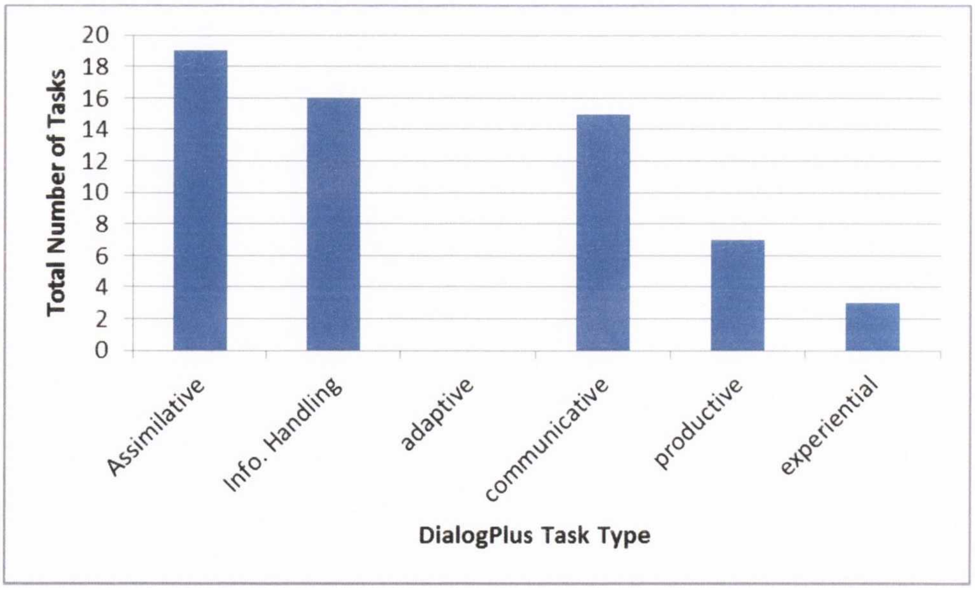


Figure 2.1: LADiE Use Case Gap Analysis - Breakdown of DialogPlus Task Usage

- Associationist - learning as activity
- Cognitive - learning as achieving understanding
- Situative - learning as social practice

Despite the different background and motivations of each of these perspectives one aspect that they have in common is the use of activity as a means of learning. For the Associationist/Empiricist perspective 'learning by doing' is the key to the process of learning as it is the mechanism through which learners build connections between units of knowledge. The Cognitive perspective encourages the use of activities that are as authentic to the real world as possible as a means to allow the learner to construct mental models that can be applied to real world problems. The authenticity of the activity is also emphasised by the Situative perspective where the focus is on the realistic interactions of learners with the other members of the community as the learner participates in the activity.

Personalisation is another common theme across the different perspectives. Associationism places an emphasis on personalising the composition of tasks to the individual learner based on their performance in prior tasks. Similarly, the Cognitive perspective suggests that learners should be treated as individuals and activities personalised to their needs.

Clearly, the motivation for learning activities that are personalised to the needs of the individual learner is supported by learning theory. It is also clear that eLearning practitioners are applying this theory in the real world. This is shown by the emphasis that has been placed on research in this area. Projects such as LADiE and DialogPlus have illustrated how learning activities are applied by learning practitioners.

2.3 Adaptation Techniques for Learning Content

Adaptive Hypermedia (AH) [Brusilovsky 01] is an area of research that attempts to address some of the issues that have been identified in traditional hypermedia based systems. Examples of such issues that are commonly attributed to hypermedia include the 'Lost in Space' phenomenon [Conklin 87], the non-linear paths through the hypermedia and the one-size-fits-all nature of traditional hypermedia. The 'Lost in Space' or 'Lost in Hyperspace' phenomenon refers to when a user has difficulty knowing where in a hypermedia they currently are and how they can get to where they want to go. Traditional one-size-fits all hypermedia present all users with the same hypermedia document irrespective of their information needs. These issues can have the effect of increasing the cognitive load on a learner as they must not only attempt to learn the subject matter but also successfully navigate the hypermedia [Dias 99, Conklin 87]. AH attempts to address these issues by adapting the hypermedia to the individual user based on various properties of the user, for example, the user's goals, prior knowledge or preferences.

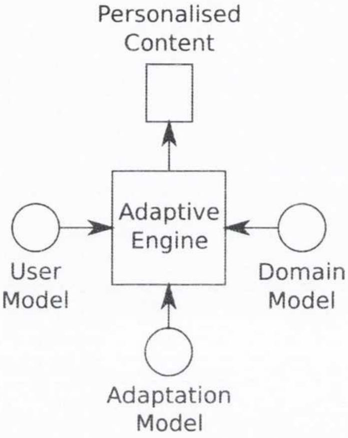


Figure 2.2: Abstract Model of an Adaptive Hypermedia System

Adaptive Hypermedia Systems (AHS) are generally characterised as consisting of an Adaptation Engine and three metadata models, the User Model, Domain Model and Adaptation Model [Koch 02, De Bra 99]. Figure 2.2 provides an illustration of a generic model for an AHS. The User Model provides the system with a model of the user that describes, for example, the goals, prior knowledge and interests of the user. The Domain Model describes the conceptual structure of the application domain while the Adaptation Model consists of the adaptation rules that determine how the hypermedia can be adapted to the user. The three models described represent the Storage Layer of a hypermedia system that is based on the Dexter Model [Halasz 90]. The Adaptation Model rules are executed by the Adaptive Engine in order to generate a personalised hypermedia. This is achieved by reconciling information from the Domain Model with information about the user from the User Model. For example, only present concept B to the user if their User Model says that they have completed the prerequisite concept A.

The techniques used in AH to personalise the hypermedia, as categorised by Brusilovsky in his updated taxonomy of AH techniques [Brusilovsky 01], are considered to fall into

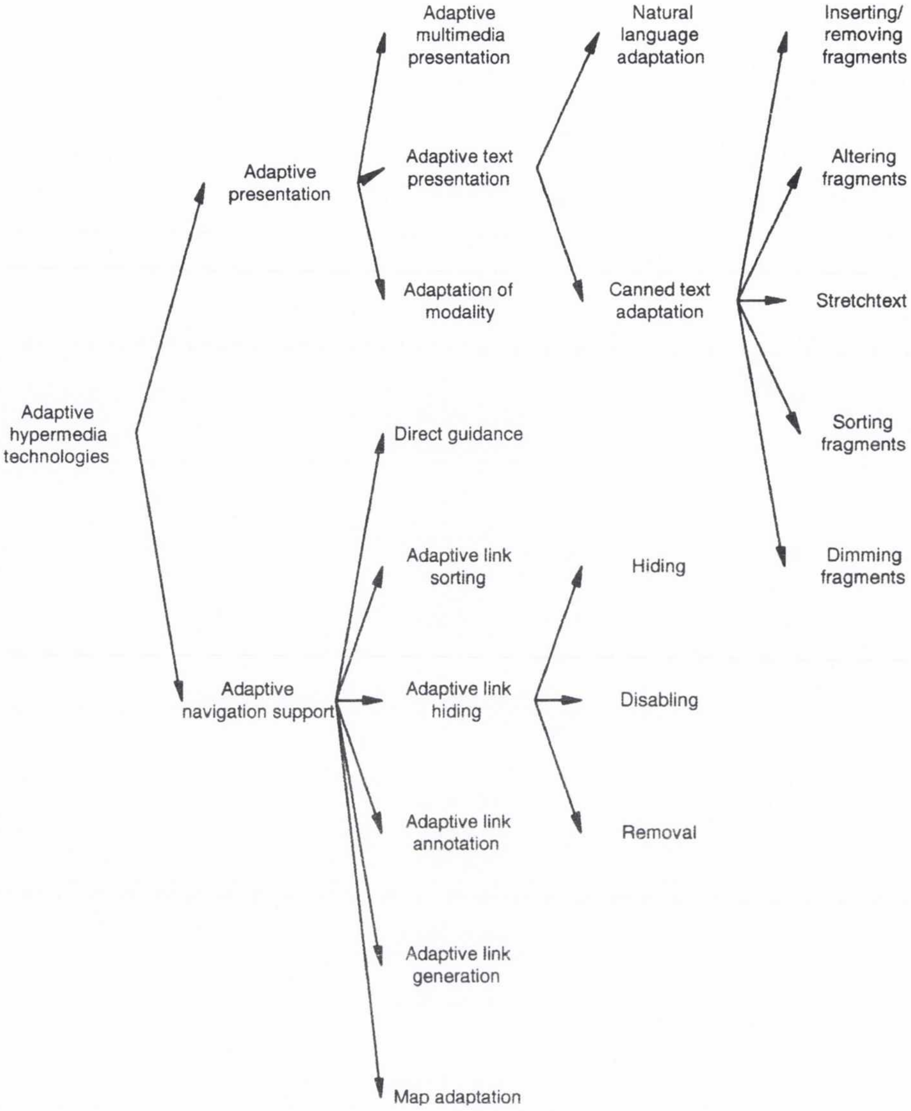


Figure 2.3: Brusilovsky's Taxonomy of Adaptive Hypermedia Techniques [Brusilovsky 01]

two different categories, Adaptive Presentation and Adaptive Navigation, as is shown in figure 2.3. Adaptive Presentation techniques adapt the content of a page that is presented to the user in order to adapt to the users needs. A significant portion of the research in Adaptive Presentation is in 'canned text adaptation' where content is adaptively included or excluded from the page using techniques such as stretchtext, inserting/removing fragments or the sorting of fragments [Brusilovsky 96a]. Adaptation of the modality of the content, for example based on user preferences, is also included in this category.

Adaptive Navigation techniques attempt to provide a personalised navigation structure across the hypermedia in order to help the user to find the best path through the hyperspace [Brusilovsky 96a]. This is achieved through a range of techniques that range from Direct Guidance, where the user is provided with the next link that they should follow to Adaptive Link Annotation where the user is given guidance as to which links might be appropriate for them.

2.3.1 Adaptive Presentation Techniques

Adaptive Multimedia Presentation The adaptation of the multimedia resources used in a hypermedia is referred to as Adaptive Multimedia Presentation. Examples of how this technique can be applied include changing the quality of an image or video based on the available bandwidth or device characteristics.

Adaptive Text Presentation Adaptive Text Presentation generally refers to a category of adaptive presentation techniques that attempt to personalise the content of a hypermedia document through the conditional inclusion of fragments of 'canned text' within a page of text. Such techniques are often applied in order to adapt the text to the user's prior knowledge, for example to provide an explanation of a term to a novice user.

One technique that falls into this category is Stretchtext [Brusilovsky 96a], which adapts the text of a hypermedia by adaptively expanding so called 'hotwords' to provide more detailed information on a concept depending on the needs of the user. Early examples of such systems include MetaDoc [Boyle 94] and KN-AHS [Kobsa 94].

Other techniques that can be classified as Adaptive Text Presentation are the Dimming of Fragments [Brusilovsky 96a] and the adaptive insertion and removal of text fragments, as exemplified by the AHA! system [De Bra 06].

Adaptation of Modality Adaptation of the modality of a hypermedia generally refers to adaptation of the underlying format in which the hypermedia is delivered, for example delivering audio instead of video based on available bandwidth or synthesising text to speech when the user is in an 'eyes busy' environment.

2.3.2 Adaptive Navigation Techniques

Direct Guidance Direct Guidance is considered to be the simplest of the adaptive navigation techniques to implement. In this technique, the AHS adaptively selects the next node in the hypermedia that should be visited by the user based on, for example, the user's goals or prior knowledge. A link to that node is then added to the current page. Examples of systems that implement this technique include Interbook [Brusilovsky 98] and ELM-ART [Brusilovsky 96b].

Adaptive Link Sorting Adaptive Link Sorting provides the user with an ordered list of links from the current page in the hypermedia based on the relevance of the linked nodes to the user. The relevance is determined by the system based on the User Model, for example based on the goals of the user or their prior knowledge. The user can then choose whether or not to follow the links that they system has identified as most relevant or to follow alternate links.

Adaptive Link Hiding Adaptive Link Hiding is a mechanism through which the possible paths through the hypertext can be restricted by 'hiding' or turning off the links from one page to another. Links between nodes can be disabled in order to hide content from the user that they do not yet have the prerequisite knowledge to understand. This approach can also be used to constrain the hyperspace that the user sees, based on their current goals, so that irrelevant or unnecessary content is hidden from them. The AHA! system [De Bra 06] is a popular example of a AHS that implements this technique.

Adaptive Link Annotation Adaptive Link Annotation is a similar technique to Link Hiding as it attempts to guide the user through the hypertext by annotating the links in a page in order to provide visual cues to the user as to the appropriateness of the link based on the users prior knowledge, goals, interests, etc. The default behaviour of a Web Browser implements a simple form of this technique. A Web Browser provides the user with a visual cue, through the colour coding of the hyperlink text, that tells them whether or not they have previously visited a link thus allowing the user to decide whether or not they want to follow the link. ELM-ART [Brusilovsky 96b] was one of the first AHS to implement this technique using it's 'traffic light' annotations.

Adaptive Link Generation Unlike the Adaptive Navigation techniques discussed previously, which annotate or re-order existing links in a hypertext. Adaptive Link Generation techniques attempt to generate new links between resources. These new links can be generated based on the discovery of relationships between concepts, or by recommending links to topics that are related in some way to the topic currently being viewed by the user. Another approach to Link Generation is the generation of a navigation structure, such as a 'table of contents' that spans the content/concept space taking into account the user's prior knowledge, learning goals, preferences, etc. The APeLS system is an example of how this form of Adaptive Link Generation can be applied [Conlan 03a].

Map Adaptation Map Adaptation refers to the adaptation of a local or global site map that provides the user with a graphical representation of the hyperspace. Examples systems that do apply this technique include HYPERCASE [Micarelli 96] and ExploraGraph [Dufresne 00].

2.4 Survey of Adaptive Hypermedia Systems

In the previous section, the techniques used in AH to provide both Adaptive Navigation and Adaptive Presentation were summarised and examples of systems that apply those techniques were provided. This section presents a survey of highly cited adaptive eLearning systems, providing a summary of the features of each system. It then provides a thorough comparative analysis of the systems against important AH capabilities and affordances.

2.4.1 AHA!

AHA! [De Bra 06] is an open source AHS that is developed and maintained by the Technical University of Eindhoven. It follows closely the classical model of an AHS, as described by Brusilovsky [Brusilovsky 01]. As such, its architecture consists of an Adaptive Engine (AE), which is used to execute a set of rules, referred to as the Adaptation Model (AM), in order to select appropriate concepts from a Domain Model (DM) based on the systems knowledge of the user, as captured in the User Model (UM).

The DM consists of a directed graph of concepts in which the edges represent relationships between the concepts, for example the concept *HTML tag* is a prerequisite of the concept

Javascript [De Bra 02a]. Content, in the form of XHTML pages, can be associated with a concept in the DM.

The AM consists of event-condition-action rules that determine any actions that the AE should take. Each rule is associated with a specific concept from the DM so that for example, when a user reads a page associated with a specific concept the UM is updated to reflect an increase in the user's knowledge of that concept. The rules in the AM also allow such increases in knowledge to propagate so that an increase in a user's knowledge of a subconcept can also trigger their knowledge of a superconcept to be updated as well. As well as knowledge of a concept, adaptation rules in the AM can also be used to update other properties that are associated with a concept in the UM.

AHA! supports both adaptive navigation, through the use of adaptive link annotation, and adaptive presentation, through the use of adaptive multimedia presentation and canned text adaptation. Link annotation is based on the traffic light metaphor with the links to appropriate concepts annotated with a green light, links to optional concepts annotated with an amber light and links to concepts that are considered to be inappropriate annotated with a red light. Annotations are added to any hyperlink in a document that is marked as conditional based on the adaptation rules associated with the linked concept. These rules determine the appropriate annotation based on the users knowledge of the concept.

The content that AHA! uses to deliver adaptive hypertexts consists of XHTML compliant documents in which the anchor tag, for any hyperlinks that are required to have adaptive behaviours, are annotated by adding a 'class' attribute with the value 'conditional'. Such hyperlinks can be used to point to a concept in the DM rather than a specific page or resource. This allows the AE to be used to select an appropriate page to present to the user from the set of pages that are associated with that concept [De Bra 06]. In AHA! this feature is referred to as 'adaptive link destinations'. This is a new feature of AHA! 3.0, in previous versions pages of content were explicitly referenced in the DM [De Bra 02b]. AHA! can also support the adaptive insertion of fragments of text within a page. This can be achieved using one of two supported mechanisms, the first is through the use of a custom <if> tag that is embedded in the content [De Bra 02b]. The use of this tag allows adaptation rules to be embedded in the content and used to select an appropriate block of text to show to the user. This mechanism has been available in previous versions of

AHA! whereas the second mechanism, which makes use of the XHTML <object> tag, is only available in the more recent version 3.0 of AHA! [De Bra 03]. This new approach removes the need to use the non standard tag <if> tag in the content markup as well as removing the need to embed the rules in the content. Instead the rules are associated with a concept in the AM. Using the object based mechanism, <object> elements are inserted into the content at the point where a conditional fragment should be added. This element has two attributes, the type attribute with value 'aha/text' is used to identify the object to the AHA! engine as it processes the XHTML document and the name attribute is used to specify the concept that the conditional fragment corresponds to.

2.4.2 ADAPT^2

ADAPT^2 [Brusilovsky 05b] is a distributed AHS, based on the earlier KnowledgeTree [Brusilovsky 04a] system, in which the various functionalities of the system are separated out into individual services. The architecture consists of a user portal, through which users can interact with the system, a user modelling service [Brusilovsky 05a], a set of services that provide support for a range of different adaptive functionalities and value-adding services [Brusilovsky 04a]. These adaptation services, referred to as 'activity servers' [Brusilovsky 04a], can then be combined as required in order to deliver personalised content. Examples of adaptive services available as part of ADAPT^2 include QuizGuide [Sosnovsky 04], an adaptive quiz service, KnowledgeSea [Brusilovsky 04b], a personalised content recommender and WebEx [Brusilovsky 08], which delivers interactive examples.

As part of the ADAPT^2 architecture, adaptive behaviours are handled by the value adding services. On such service is Nav-Ex [Brusilovsky 08, Brusilovsky 05c], which uses link annotation to provide adaptive navigation support to courses delivered using ADAPT^2. As part of the ADAPT^2 architecture, services such as Nav-Ex can be used to add adaptive behaviours to content that was not originally designed to be used adaptively.

Nav-Ex provides link annotations to the user based on the user's prior knowledge of the prerequisite concepts for a given link. Red bullet annotations are used to indicate that a user does not have sufficient prior knowledge while a green bullet annotation indicates that a link is appropriate for the user. Links can also be annotated with a green check mark to indicate that the user has already covered the concepts covered by that link. Nav-Ex identifies links as appropriate or not using a domain model, which is auto-generated from the content. The user's knowledge level for the prerequisite concepts of a given link are retrieved from a binary overlay user model so that a user is considered to have learned a concept as soon as they have read the corresponding content.

An interesting feature of ADAPT^2 is its support for what is referred to as 'intelligent content' [Rey-lópez 08]. This consists of pieces of dynamic content, which can be implemented using various technologies including ASP and Java Applets, that provide the user with interactive content that provides features such as quizzes and interactive programming problems [Brusilovsky 05c]. As mentioned previously, dynamic content in the ADAPT^2 architecture is delivered using 'activity servers'.

2.4.3 APeLS

The Adaptive Personalised eLearning Service (APeLS) [Conlan 03a] is a web based AHS that was developed in Trinity College Dublin (TCD). The APeLS system is specifically targeted at eLearning applications and as such has been used to deliver several adaptive courses covering topics ranging from Mechanics (physics) [Conlan 03a] to the SQL database query language [Conlan 02]. The adaptive SQL course has been used to teach undergraduate students in TCD for several years, in that time, it has been shown to have provided real benefits to the students that have used it in comparison to non adaptive online courses [Conlan 04].

The APeLS architecture is similar to that of other AHS, as shown in figure 2.4 it consists of an Adaptive Engine, which is used to reconcile the Learner Model with the Content Model through the execution of the Narrative Model [Conlan 02]. The Narrative Model is a key architectural difference of APeLS in comparison with other AHS, such as AHA!

The Learner Model used by APeLS is an XML based description of the user which is primarily designed to capture information about the competencies that the user has learned as well as the competencies that they wish to acquire. The schema of this model is fixed and cannot be modified although it does provide enough flexibility to store the competencies relating to an arbitrary number of concepts. It is also possible to capture properties of the learner other than competencies required/learned as long as the information to be captured conforms to the concept overlay approach to user modelling that is applied in APeLS.

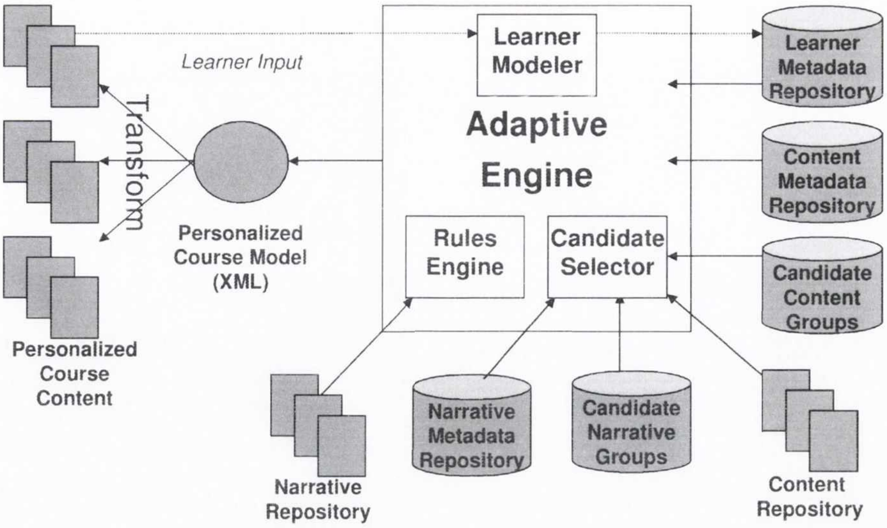

Learner Input
Learner Modeler
Adaptive Engine
Rules Engine
Candidate Selector
Learner Metadata Repository
Content Metadata Repository
Candidate Content Groups
Transform
Personalized Course Model (XML)
Personalized Course Content
Narrative Repository
Narrative Metadata Repository
Candidate Narrative Groups
Content Repository

Figure 2.4: APeLS Architecture

APeLS does not define what information about the user is captured in the Learner Model or how that information about the user is captured. This is up to the developer of the specific instance of an APeLS course to specify based on the needs of the adaptive course they are designing. For example, in the Physics course mentioned previously, once a user has viewed a page it is assumed that their knowledge of the corresponding concept increased and so their user model is updated. In contrast, the SQL Course instance of APeLS only updates the user model based on an explicit action of the learner.

The Content Model in APeLS is a collection or XML based metadata descriptions of each of the content resources that are available to the system for inclusion in a personalised course. Each metadata description conforms to the same schema although this schema can be defined by the designer or an adaptive course. This allows APeLS to search across all of the content metadata as one model in order to identify appropriate content.

The Narrative Model in APeLS is used to encapsulate the necessary domain model information as well as the adaptation rules that allow the AE to adaptively select appropriate concepts as part of a user's personalised course. In these respects, the APeLS Narrative Model is equivalent to the Adaptation Model in AHA! However, unlike AHA!, which only considers the sequencing of concepts as defined in the domain model, APeLS allows concepts to be sequenced based on strategies that are defined by the author of the adaptive course, while still taking into account the relationships between the concepts that are defined in the domain model. This allows adaptive courses that are based on educationally sound strategies to be generated. These strategies are captured in the Narrative Model.

In addition to the adaptive sequencing of concepts, APeLS also supports the adaptive selection of appropriate content through a technique referred to as Candidacy [Dagger 03]. Candidacy is based on the grouping of content resources that cover the same concept but which differ in how that concept is presented. For example, different content resources might be available to explain a concept to a learner with limited knowledge of a domain and a learner with a more detailed knowledge. A specialised set of rules, referred to as a Candidate Selector [Dagger 03], is then used to choose the most appropriate content for an individual learner. This selection process can also be passed on to an external system, which has specialised knowledge of how to make the selection.

In both adaptive sequencing and selection, APeLS can make use of any property of the learner or content that is available to the Engine in order to influence the execution of the narrative rules. These rules are executed using the JESS [Friedman-Hill 08] rule engine, which has been extended to provide a set of Custom Functions [Conlan 02] that allow the rules to access and manipulate the metadata models. The combination of the flexible rule language provided by JESS and its support for applying sequencing strategies, APeLS affords the narrative author a lot of flexibility in how they express the adaptive behaviours necessary for their course.

2.4.4 ActiveMath

ActiveMath [Ullrich 08, Melis 01] is a web based system for teaching mathematics, which supports the personalised composition of learning objects. These learning objects are based on the Open Mathematical Document (OMDoc) specification [Kohlhase 06] for mathematical documents, which ActiveMath interprets and transforms into a format that can be delivered to the user, e.g. HTML. In addition to learning objects consisting of static content, ActiveMath also supports the delivery of interactive exercises [Goguadze 05] that allow the user to engage with the system by completing exercises such as answering mathematical problems, either by calculating the answer or submitting a formula, which is then evaluated by a Computer Algebra System (CAS). As with static content objects, the exercises are encoded using an extension of the OMDoc specification, which is transformed into a HTML form. The system can then evaluate how well a user does in an exercise and provide personalised feedback based on that evaluation. ActiveMath supports Adaptive Navigation through the integration of a course generator component, PAIGOS [Ullrich 09], which builds personalised courses based on the user's goals.

Unlike the AH based personalised eLearning systems discussed previously, which use an application specific set of rules to make decisions about concepts in the domain model, PAIGOS uses a Hierarchical Task Network (HTN) based AI planning technique to compose personalised courses (see section 3.3 for more details on HTN and other AI Planning techniques). This approach allows PAIGOS to decompose the goal of the user into a set of subtasks that can be satisfied by the resources available to the system. To achieve this, PAIGOS has a set of rules that describe how a specific educational goal should be decomposed in order to produce an educationally sound course. As such, these rules represent a pedagogical model that describes how courses can be generated in accordance with pedagogically sound strategies. PAIGOS supports six different strategies or 'scenarios' [Ullrich 08], which have been developed based on a constructivist perspective on learning theory. The six scenarios supported are: discover new content, rehearse weak points, establish connections between concepts, train intensively, train competencies and exam simulation.

The PAIGOS course generator affords ActiveMath with a lot of flexibility in terms of how a course can be delivered as it supports both a priori composition of a course and just in time selection of appropriate resources. Sections of a course structure can be statically defined by a course author while still allowing other sections of the course to be adaptively generated. The content to teach these concepts can then be selected on a just in time basis, allowing the system to select appropriate content based on the current state of the User Model. This allows the system to take into account the user's performance in exercises that they have recently completed.

2.4.5 IMS Learning Design

Learning Design [IMS 03] is an IMS Specification for the description of pedagogically driven learning activities using a platform independent language. The specification is broken into three levels, level A provides support for describing the basic structure of a static Learning Design that can be executed by a Learning Design player such as CopperCore [Martens 05] or learning management systems such as .LRN [Del Cid 07] and

LAMS [Dalziel 03]. Level B builds upon the basic structure of a Learning design specified by level A by adding support for adaptivity within the Learning Design while level C adds support for notifications, which can be used to send messages to a user or to enable activities.

The basic components of a Learning Design are **Roles**, **Activities** and **Environments**. Roles allow the author to define different roles that users can carry out within the system, these roles can be defined as either **learner** or **staff** roles although there can be many different roles defined within these types. Activities are the basic components of a Learning Design representing individual parts of the course, which are realised through the delivery of one or more Environments. Activities can be nested to create a tree structure of activities. Environments are containers for content and services that are necessary to deliver the Learning Design. By separating the definition of Environments from Activities it is possible to reuse the Environments as part of multiple Activities.

The sequencing of the delivery of Activities in a Learning Design is controlled by the **Play**, which contains one or more **Acts**. In turn, an Act contains one or more **Role-parts**, which link a Role with that Activity that the role should take part in during the Act.

As mentioned previously, adaptive behaviours can be implemented in a Learning Design using the level B features, of specific interest are **Properties** and **Conditions** that this level introduces. Properties are variables that can be scoped with respect to the 'run' (specific execution of the Learning Design) or can be global (exist across all runs of the Learning Design). Properties can also be personal, reflecting the properties of individual users. In order to implement adaptive behaviours these Properties can be accessed and manipulated using Conditions, which are essentially if-then-else statements that can be triggered by changes to Property values. The most common use of conditions is to set the **isvisible** attribute that is part of the definition of all Activities, Environments and Acts. This allows parts of the Learning Design to be enabled or disabled based on the value of a Property, which could be set as the result of a previous interaction with the system, or in the case of a fully integrated LMS such as .LRN, the properties from a Learner Model. Conditions can also be used to conditionally show/hide sections of a HTML document being presented to the user. In this case the result of the condition would be to show or hide a <div> element within the HTML document based on the value of the **class** attribute.

Learning Design also supports the use of services are part of an activity. The Specification itself supports three types of services an email service, **send-mail**; a discussion service, **conference**; and a search service, **index-search**. The implementation of these services is left up to the platform so that the Learning Design can remain independent of the platform so that it can be reused. Additional services can be added to Learning Design by extending the specification with additional service types, which would be specific to a given platform as other Learning Design platforms would not be able to handle the extensions correctly. Various different approaches have been taken to address this issue such as that taken by LAMS as well as others [Del Cid 07].

2.4.6 Comparison of Content Adaptation Systems

To compare the adaptive eLearning systems presented in this chapter, a set of criteria have been identified. These criteria cover five areas of interest with respect to the research question, namely:

- Adaptation Techniques
- Adaptation Mechanisms
- Adaptation Scaffolding
- Content Flexibility
- Content Interaction with System

The adaptation techniques criteria allow the systems to be compared based on the adaptive behaviours that they support. These behaviours have been taken from the main stream classification of adaptive behaviours as discussed in section 2.3. By comparing the adaptive behaviours supported by the systems discussed it will be possible to identify any trends that exist in the state of the art in adaptive eLearning systems.

The adaptation mechanism criteria cover the techniques used by adaptive systems to realise the adaptive behaviours that they exhibit. As such, they cover both architectural considerations such as the types of content supported and how they make use of the metadata models available to the system.

The underlying scaffolding upon which the adaptation rules are built can have a significant impact on how the adaptation rules are authored. For this reason the comparison looks at how the adaptation rules of the different systems are structured upon the underlying models of the system.

The content flexibility is an important aspect to investigate as it provides an insight into the overall flexibility of the adaptive system and how easily it can be applied across different domains. For example, does a system only support content in a specific format or does the system require content to be tailored or customised specifically to the system.

Related to the flexible support for content is whether or not the system supports or requires the interaction of the content directly with the system in order to provide some or all of it's adaptive behaviours.

These five criteria can be considered to fall into two categories, those that are used to evaluate the systems based on the Adaptive behaviours supported and those that are used to evaluate the system based on it's content requirements. These categorisations are shown below. The following sections provide an analysis of the systems surveyed based on these criteria, which is summarised in tables 2.3 and 2.4.

- Adaptation Criteria
  - Adaptation Techniques
  - Adaptation Mechanisms
  - Adaptation Scaffolding
- Content Criteria
  - Content Flexibility
  - Content Interaction with System

2.4.6.1 Adaptation Comparison

Adaptation Techniques The traditional categorisation of adaptation techniques focuses on Adaptive Navigation (AN) and Adaptive Presentation (AP). With respect to AN all of the systems discussed support the adaptation of the navigation structure. In the case of AHA! and ADAPT^2 this support is in the form of Link Annotation where as APeLS and ActiveMath can both be considered to apply Link Hiding and Link Generation techniques. As discussed by Berlanga et al. [Berlanga 08] and Hendrix et al [Hendrix 09], Learning Design can support AN using several different mechanisms including Direct Guidance, Link Hiding and Link Annotation.

	AHA! 3.0	ADAPT^2	APeLS	ActiveMath	Learning Design
Adaptation Techniques					
Adaptive Navigation	Yes Link annotation using traffic light metaphor	Yes Link Annotation	Yes Link Hiding/Generation	Yes Link Generation	Yes Direct Guidance, Link Hiding, Link Generation
Adaptive Presentation	Yes Canned/Conditional text and 'adaptive link destinations'	No	Yes Candidacy mechanism	No	Yes Conditional inclusion of fragments of canned text
Adaptive Retrieval	No	Yes Supported using KnowledgeSea service	No	No Static search only	No Static search only
Adaptation Mechanisms					
Rules embedded in content	Yes (for conditional/canned text)	No	No	No	No
Separate Rule/Adaptation Model	Yes	Yes	Yes	Yes	Yes
Run time use of Domain Model	No	Yes	No	Yes	No
Adaptation Scaffolding Adaptive Links overlaid on:					
Domain Model	Yes	Yes	Yes	No	No
Pedagogy/Strategy	No	No	Yes	Yes	Yes
Service Workflow	No	No	No	No	No

Table 2.3: Comparison of Supported Adaptation Techniques and Model Usage of Adaptive eLearning Systems

ADAPT^2 and ActiveMath do not support any AP mechanisms, however the other three systems in the survey do implement various Adaptive Presentation mechanisms. Both APeLS and AHA! can adaptively select appropriate content resources using their respective Candidacy and 'adaptive link destinations' techniques. AHA! also supports the conditional inclusion of fragments of text in an appropriately marked up piece of content, a feature that is also supported by Learning Design. The implementation of this feature in AHA! however is a lot more flexible as the individual fragments are separate to the document in which they are included where as Learning Design requires every possible version of the text fragment to be included in the document. This is similar to the technique that was used in older versions of AHA!.

In order to fully cover the full range of adaptive behaviours, Adaptive Retrieval has also been included as a criteria for comparison. In their general form, none of the systems covered in this chapter support Adaptive Retrieval. However, ADAPT^2 can support Adaptive Retrieval through the use of the KnowledgeSea [Brusilovsky 02] service. KnowledgeSea can be integrated into an adaptive eLearning course in order to support the searching of open corpus content. In the case of KnowledgeSea II [Brusilovsky 04b], the basic search functionality is extended through an adaptive annotation mechanism, similar to the traditional Link Annotation techniques, that suggests to the user content that others have browsed.

Courses delivered using ActiveMath can take advantage of a Search Tool, which allows the user to search all of the content available in ActiveMath by entering mathematical formulae as well as text. Similarly, a search service can be added to a Learning Design. However, in both cases the search functionality is not adaptive.

Adaptation Mechanisms The first aspect of AHS design that was investigated was the mechanism used to encapsulate the adaptation rules. Early AHS embedded the adaptation rules in the content that they delivered, while others embedded the rules in the engine itself. Embedding the adaptation rules in the content has several adverse effects, it increased the complexity of the content authoring, restricted the availability of content and greatly reduced the ability to reuse or re-purpose the adaptation strategy.

Several subsequent models for the design of AHS, such as AHAM [De Bra 99] and the Multi Model, Metadata Driven Approach [Conlan 02, Conlan 05], identified this issue and recommended the separation of the adaptation rules into a separate model, referred to as the Adaptation Model or Narrative Model respectively. This approach would allow the rules to be modified independently of the engine and/or content and opened up the possibility of reusing the strategy embodied by the Adaptation Model. Similarly, embedding the rules in the engine itself reduces the applicability of the engine to different applications. If the rules are specific to the application domain, e.g. Database Theory then the engine software is tied to that application domain. Even if the rules are independent of the domain, they are still likely to be designed based on a specific pedagogical strategy or teaching approach. To change the strategy used would require the engine itself to be modified, increasing the cost and complexity of applying the technology.

In all of the systems discussed in this chapter, the rules used to implement Adaptive Navigation are indeed contained in a separate Adaptation Model. This can also be considered to be the case for Learning Design as the description of a Unit of Learning in Learning Design is only focused on the description of the activity sequencing. As such, it is independent of the engine itself. Based on this agreement in the design of the different AHS, a separate Adaptation Model can be seen as a characteristic of a modern AHS. When looking at how Adaptive Presentation is implemented by the systems in this study that support it, namely Learning Design, APeLS and AHA!, there is not such agreement. To support conditional/canned text functionality as described by [Gutierrez-Santos 08], Learning Design requires the content to be written especially, with all possible versions of the fragment contained in the same document. However, the rules that actually control which fragment is displayed are part of the Learning Design specification for the Unit of Learning. AHA! also supports conditional text fragments but it has moved from an approach in which both the conditional fragments and adaptation rules were encapsulated in the content resources to a more flexible approach in which the rules are now part of the Adaptation Model and the content fragments are stored in individual files. APeLS on the other hand does not require rules or content fragments to be embedded in the content resources it delivers in order to provide Adaptive Presentation although its implementation of this behaviour, through Candidacy, does not provide the same level of adaptation granularity as AHA! or Learning Design.

The way in which an AHS utilises its Adaptation Model is also an interesting aspect of their design. The question here is when should the adaptive selection and sequencing be carried out, a priori or just in time. In the a priori approach, the AHS would generate a complete personalised course or presentation before the user is allowed to access the content. The just in time approach differs in that is makes adaptive decisions about which concept or specific piece of content to present to the user only at the point when the user makes a request to the AHS for the next piece of content. For example, when the user clicks a hyperlink in a web based interface. The a priori approach has the benefit of allowing the system to generate a complete course in which the user is free to navigate without the structure of the course or the content presented to the user changing. This helps to address the issues raised by De Bra [De Bra 00], who identified how an 'unstable' hypertext that changed between user visits could be confusing to the user. The problem with this approach is that the course or presentation generated by the AHS can become out of date as the user's knowledge in the domain changes. The just in time approach results in a course that is always appropriate given the system's current model of the user but can cause the user to become disorientated within the hypertext as a result of the change structure of the document.

The ADAPT^2 system takes a just in time approach to Adaptive Navigation, however the possible adverse affects of this approach on the user are not a significant issue for this system as it does not directly manipulates the link structure of the personalised course. If an ADAPT^2 user revisits a page the only change they are likely to notice is that the annotations on the adaptive links might have changed. AHA! also employs just in time Adaptive Navigation but unlike ADAPT^2 it does manipulate the link structure through the use of its 'adaptive link destinations' technique as well as adapting the content itself dynamically using conditional fragments. As a means of controlling the potential for confusing the user as they navigate, the designers of AHA! recommend that authors of adaptive courses design their courses so that they contain 'explicit static link structures' and that adaptive links should be used for additional information [De Bra 98].

In contrast, the APeLS system applies an a priori based approach in which the user is provided with instruments that allow them to control when the system rebuilds their personalised course. If the user feels that the course generated by the AHS is no longer appropriate for them then it is up to the user to prompt the system to build a new course for them, taking into account the systems updated view of the user. Any changes to the structure of the course are directly related to the user's own actions. The ActiveMath system takes a similar approach to APeLS in that the course generation process is carried out a priori by the PAIGOS component. The actual content delivered to the user is selected just in time. This allows ActiveMath to select appropriate content for the user based on how well the user did in previous interactions with the system's interactive components.

When looking at Learning Design with respect to just in time adaptation versus a priori adaptation it is a little more difficult to analyse as there is nothing specific in the specification that defines when the adaptation occurs, instead it is dependent on the player or LMS that is executing the Learning Design. In the case of a player such as Coppercore, it is obvious that any adaptation can only be at run time as Coppercore only has access to user properties that it has acquired during the run of a Unit of Learner. An LMS such as LAMS or .LRN has access to a user model in order to populate properties in the Learning Design, which could be used to adapt the Unit of Learning to the user's needs prior to their interaction with the system. However, there is nothing in the specification to restrict the Learning Design from further adapting the Unit of Learning as the User Model changes over time. This results from the way in which, according to the specification, a Learning Design player should re-evaluate any conditions as soon as the vales of the properties that the conditions operate on change. This behaviour would result in the Learning Design changing as the user navigates through it.

Adaptation Model Scaffolding When looking at how different AHS encode their supported adaptive behaviours in rules, it is clear that there are several different approaches that can be taken and that the choices made directly affect the type of course or presentation that the system can generate.

In order to be able to write rules that describe the adaptation mechanisms in an AHS, it is necessary to first identify a scaffolding upon which the rules can be layered. Such a scaffolding would need to be an existing structure that is either available to the AHS at run time or is considered at design time by the author of an adaptive course or presentation. Examples of such scaffolds include the Domain Model, the pedagogy or other strategy and, in the case of activities or services, the workflow the describes the interaction between the services.

The use of the Domain Model as a scaffolding for overlaying the adaptation rules implies that the system makes decisions about concepts to be included in a personalised course or presentation based on relationships between the concepts in the Domain Model. AHA!, ADAPT^2 and APeLS all support the execution of rules based on the Domain Model. In the cases of AHA! and ADAPT^2, this is the only type or rule scaffolding that they support for the implementation of Adaptive Navigation. This is because the rules are designed to be executed in a just in time manner that reconciles the relationships between concepts in the Domain Model with the system's knowledge of the user based on an overlay style User Model. APeLS, in addition to supporting Domain Model based scaffolding of the rules, can support the use of other scaffoldings at the same time. ActiveMath, unlike the other three systems, does not layer its rules on the Domain Model. Although ActiveMath does generate courses based on a set of concepts to be covered, the rules that determine how this is achieved are independent of the concepts themselves.

An alternative to using the Domain Model as a scaffolding for the Adaptation Model rules is to use pedagogy or some other strategy that should influence the structure of the generated course or presentation. This approach allows the AHS to take into account the expertise of the course author as well as accepted design principles from the domain in question, for example pedagogical strategies in eLearning. Both ActiveMath and APeLS adopt this approach to rule authoring to varying degrees. In ActiveMath, the rules are entirely based on pedagogical strategies and are independent of the domain over which the rules are executed. This allows PAIGOS, the course generator component of ActiveMath, to be used in different application domains without the need to author any rules. The limitation of this approach is that the author of the course is restricted to using pedagogies supported by PAIGOS.

APeLS takes a more pragmatic approach to the use of strategy as a scaffolding for the overlay of rules. It does not prescribe any pedagogical strategies to the author but instead allows them to make decisions about how the course should be structured for themselves. As mentioned previously, these strategy based rules can be combined with Domain Model based rules so that the system can make decisions about both educational strategy and the relationships between concepts. The complexity of such rules mean that the author is faced with a significant task, especially if they are not experts in authoring such rules. This complexity is handled through the use of authoring tools such as the ACCT [Dagger 05] that allow pedagogical strategies and Domain Model information to be combined graphically.

The main focus of the Learning Design specification is to support the application of a wide range of pedagogical strategies in eLearning. A such, the basis of a Learning Design is the sequencing of acts and activities in order to implement the strategy that the designer wishes to employ. There are limitations in the approach taken by Learning Design such as the inability to allow the learner to review activities that they have already completed, as discussed by [Gutierrez-Santos 08]. Learnin Design also requires that the strategy being employed must be fitted within the hierarchical structure of acts and activities that Learning Design is based on. Irrespective of these limitations, the adaptation rules that can be incorporated into a Learning Design are primarily based on the structure of activities in the Learning Design. Furthermore, Learning Design authoring tools such as ASK-LDT [Sampson 05], Reload [Reload Project 10] and the LAMS authoring tool are generally focused on facilitating the designer in structuring the activities. This is in contrast to authoring tools such as those for APeLS or AHA!, which base the authoring methodology on the Domain Model.

A third possible approach to the scaffolding of adaptation rules is to overlay them on the control flow (workflow) between the activities or services in a composition. The integration of services into an AH course is something that has seen limited application. As is the case with pedagogy in eLearning, there are a complex set of rules that govern the interaction of services in a composition. In order to adapt the composition it is not only necessary to take into account the services that are available but also the control flow between the services, which describes the business logic of the composition. None of the four systems looked at in this chapter provide any explicit support for this approach. Although ADAPT^2 supports the delivery of interactive or 'intelligent' content these are treated in the same way as static content with no support for control flow between the 'intelligent content' resources. Similarly, Learning Design supports the inclusion of services such as communication and email in a Unit of Learning but these are also treated in the same way as any other content resource specified in the Learning Design. In contrast, the flexible nature of APeLS could provide some degree of support for this form of rule scaffolding as speculated about in [Conlan 03b] although this has not been realised.

2.4.6.2 Content Support Comparison

Flexible use of Content The content that an AHS uses to generate a personalised course or presentation is an important aspect of the design of such systems with a direct impact on the functionality that the system provides as well as several non functional properties of the system. Content is also expensive to produce and as such can represent a significant barrier to the deployment of AHS. As part of this review, two areas of interest have been identified with respect to the relationship between an AHS and the content that it utilises, the flexibility of an AHS to utilise content and the need/ability for content to directly interact with the AHS.

The ability of an AHS to use content that consists of content types, such as HTML, images and video files, that are commonly used in non-adaptive, web based courses affords such a system with significant advantages. It allows existing content to be reused by the AHS without the need for extensive refactoring of the content. Similarly, content that is developed specifically for delivery as part of an adaptive course can be used by other systems, both adaptive and non-adaptive. AHA!, ADAPT^2 and APeLS are all designed to deliver XHTML based content, however ActiveMath requires 'learning objects' to be developed using the OMDoc XML specification.

A drawback that comes from supporting generic content types is that it prevents the AHS from providing some Adaptive Presentation behaviours. For example, of the four AHS looked at in this chapter only AHA! supports the runtime interpretation of adaptive links. However, in order to achieve this it is necessary for the links in the XHTML content to be annotated in the appropriate manner. In contrast both ActiveMath and APeLS provide the structure of the course to the user and do not make decisions about Adaptive Navigation at runtime. Similarly, ADAPT^2 only provides link annotation.

	AHA! 3.0	ADAPT^2	APeLS	Active Math	Learning Design
Flexible Use of Content					
Heterogeneous Content	No HTML Content containing special links	Yes	Yes	No Content must be in OMDoc format	Yes
Interpretation of Adaptive Links	Yes	No	No	No	No
Content Inspection	No	Yes	No	No	No
Content Interaction with Adaptive System					
Content directly accesses User Model	No	Yes	No	Yes	No
Adaptively Parameterised Content	No	No Content from 'activity server' is statically parameterised but not adaptively configured	No	Yes	No
Support for Services	No	Yes Limited support for services retrieved from 'activity server' such as quizes	No	Supports 'interactive exercises'	Yes Support for services defined by LD specification or which are specific to the LD implementation
User Inspection of User Model	No	No	Yes	No	No Not explicitly part of LD but could be provided by the implementation
Implicit User Model update	Yes	Yes	No User model only updated on explicit action of user	Yes	Yes

Table 2.4: Comparison of Support for Content and Services in Adaptive eLearning Systems

In order to implement canned text adaptation, both AHA! and any system that supports the Learning Design specification must pre-process any HTML content on the server before passing it to the client to be rendered in the browser. This is necessary in order to identify fragments of the HTML that should be conditionally shown or hidden. In the case of AHA! this is either the special <if> tags or HTML <object> tags with the appropriate class type. Similarly a Learning Design system must look for any <div> element with a class type that is specified in a condition rule. In the case of content used in a Learning Design, this means that the content must be designed with this feature in mind, with all possible alternative content fragments included in the document. This does not prevent the content from being reused in other systems although it could be confusing for the user to see all of the possible alternative fragments. The use of the <object> tag in AHA! removes the need to include all possible alternative fragments in a page but it still requires the content to be written with the application in mind. Presenting such AHA! content in a different system would produce a page that potentially had pieces of content missing. Further more, the use of a mime type that the client browser is not aware of would have undesirable effects.

Content Interaction with AHS In order to achieve some adaptive behaviours it is necessary for the content to directly interact with the AHS itself. Examples of such direct interactions include content that retrieves information about the user from the User Model in order to adapt itself to the user or similarly content that directly updates the User Model based on the interactions of the user with the content. In the delivery of static content, none of the four AHS discussed exhibit this behaviour. When considering the delivery of interactive content, which ADAPT^2 and ActiveMath both support, we see that in order to provide this functionality the content must interact directly with the AHS. In the case of ADAPT^2 the content can directly query the Cumulate user modelling service for information that it then uses to adapt itself to the user. Subsequently, the content can then update the system's model of the user based on their interactions with the content. Similarly, ActiveMath's activities provide information to the system that is stored in the systems global User Model. The ability of content to directly interact with the AHS can provide benefits to the user however it does mean that the content must be specifically tailored to the architecture of the system and makes it difficult to reuse the content on any other system.

Another mechanism through which content and the AHS can interact with each other is in the adaptive parametrisation of content. This mechanism would allow an AHS to directly influence the behaviour of a content resource by provide it with additional information, for example user preferences. None of the systems discussed provide such a mechanism for passing parameters to content however, Learning Design does allow parameters to be passed to services. As Learning Design only allows services to be specified abstractly the parameters passed to the services can only be general properties that the author of the Unit of Learning thinks the service might be interested in but there is no guarantee that the service will actually make use of the information.

2.5 Summary

In this chapter, a state of the art review of the adaptive web has been presented. As part of this review, the mechanisms and techniques used in Adaptive Hypermedia to provide personalised web content to the user were described. To capture a clear view of the current state of the art, a set of long standing Adaptive Hypermedia Systems were discussed. Based on the analysis of these systems, a set of properties that can be used to characterise the architecture and functionality of an Adaptive Hypermedia System were developed. These properties were then used to compare and contrast the systems discussed.

Based on the analysis carried out, it is clear that, as a whole, modern Adaptive Hypermedia Systems have all adopted the model driven principles put forward by approaches such as AHAM and the Multi-Model, Metadata Driven Approach. However, the adoption of a pedagogical model, which can be abstractly considered as a strategy model, has not, so far, been unanimously adopted.

It is interesting to note how ADAPT^2 and ActiveMath both support what could be referred to as 'intelligent content', that is content that adapts to the user independently of the Adaptive Hypermedia System itself. Although this can be seen to have benefits from the perspective of the end user, it does present significant problems in relation to interoperability and content reuse. An important outcome of the analysis is the limited support provided by any of the Adaptive Hypermedia Systems surveyed for services. Although ActiveMath and ADAPT^2 both support content that the user can interact with, this 'interactive content' is, from an implementation perspective, tightly bound to the functionality and infrastructure of the respective systems. Furthermore, this interactive content is treated in the same manner as non interactive content with no provision made for the support of control flow or data flow between the interactive components.

Chapter 3

State of the Art - Adaptive Service Composition

3.1 Introduction

In the previous chapter the different techniques that can be applied in order to personalise multimedia content based on the needs of the individual were discussed. However, to be able to provide the Learner with adaptively composed activities, it is necessary to be able to not only adaptively select and sequence content but also to do the same for the interactive services that need to be combined with the content to create an activity.

The composition of services can be categorised as either static or dynamic [Rao 04, Dustdar 05]. Static composition techniques are generally those in which the composition of services is generated at design time with the desired services selected by the designer and the necessary control flow put in place. Research in the area of workflow can be placed into this category. Dynamic composition techniques are those in which the composition is generated at runtime with the necessary services selected and sequenced as part of the orchestration. This category of composition covers a wide range of techniques ranging from rule based to AI planning.

This chapter presents a state of the art review of both static and dynamic service composition domains, introducing the various techniques that can be applied as well as examples of systems that apply those techniques. The chapter is structured as follows, a survey of service orchestration systems and technology are presented in section 3.2, this is followed by a survey of service composition techniques based on AI planning in section 3.3. Some background on techniques and technologies related to the composition of services is also provided in section 3.4. An analysis of the systems surveyed in this chapter is then presented in section 3.5. This analysis is based on systems and technologies from both the service orchestration and AI planning surveys and focuses on the applicability to the techniques and technology in addressing the research question.

3.2 Service Orchestration

This section discusses various techniques for the orchestration of Web Services. Such techniques are traditionally seen as static composition techniques. This is essentially the case with technologies such as Web Service Business Process Execution Language (WSBPEL 2.0) [Oasis Consortium 07], which is the de facto industry standard for Web Service Orchestration. However, the state of the art clearly indicates a trend towards orchestrations that can adapt towards the needs of the user or the context in which the workflow is being executed. This is illustrated by the varying degrees of dynamism supported by systems discussed in this section. A comparative analysis of the systems discussed in this section along with the AI planning based systems discussed in the next section is provided in section 3.5.

3.2.1 Web Service Business Process Execution Language

Web Service Business Process Execution Language (WSBPEL) is an Oasis specification for the description of executable Web Service orchestrations using an XML grammar. WSBPEL is based on the work carried out by Microsoft and IBM on their earlier workflow languages, XLANG and WSFL [Leymann 01]. As WSBPEL is designed to orchestrate Web Services it supports communication between the process and the services or 'partners' that are used in the workflow using SOAP as the underlying transport mechanism.

The basic component of a WSBPEL process is the activity. There are activities to support the basic functionality of invoking partner services:

- <receive>
- <reply>

- <invoke>

Control flow in a WSBPEL process can either be handled using hierarchical structured programming constructs or, alternatively, graph based techniques. The structured approach is supported by a set of control flow activities:

- <sequence>
- <if>
- <while>
- <repeatUntil>
- <forEach>
- <pick>

The execution of activities in parallel is supported through the use of the <flow> activity, which acts as a container for activities that can be executed in parallel. Dependencies between parallel activities can be defined through the use of a <link>. Links can also be used to link execution of any activity in a WSBPEL process, even outside of a <flow>.

Messages passed to and from Web Services can be stored in local variables using the <assign> activity. During the execution of an assign activity the information in the message can be manipulated so that, for example, only the necessary information is stored to the variable. This is achieved through the use of XPath to select the appropriate parts of the message. Similarly, the data can be manipulated using basic operations to perform tasks such as basic arithmetic, boolean logic and string manipulation. The assign activity also supports the transformation of messages using XSLT transforms.

In WSBPEL, correlation, the matching of messages to the appropriate running instance of a workflow, is carried out based on declarative properties in the messages. All messages sent to a running WSBPEL workflow must contain appropriate fields that can be used to uniquely identify a specific instance of a workflow. For example, in a workflow designed to support a B2B transaction, the customerID property of a message might be used to uniquely identify the appropriate instance. If a corresponding instance of a workflow is not currently running on the workflow engine then it can be started, if appropriate. The properties used for correlation must be defined in the workflow at design time.

WSBPEL has seen widespread adoption with many commercial and open source workflow engines supporting the language. These include IBM Websphere [IBM 08], Microsoft Biztalk [Microsoft 09], Oracle BPEL Process Manager [Oracle 09], Apache ODE [Apache Foundation 10b] and ActiveVOS[1] [Active Endpoints 10]. WSBPEL is also an important component of many Enterprise Service Bus implementations, for example JBoss Riftsaw [JBOSS Community 11], which uses Apache ODE, and Glassfish ESB [Glassfish Community 11]. There is also a wide range of commercial and open source authoring tools for WSBPEL including Netbeans SOA Project [Netbeans Community 09], Eclipse BPEL Project [W3C 11], ActiveVOS [Active Endpoints 10] as well as tools from Oracle and other commercial vendors.

WSBPEL supports two methods of binding concrete Web Services with partnerlinks. The first is at design time when the endpoint for the Web Service must be explicitly specified in the process definition. In this case, the existence of the endpoint is often verified by the workflow engine as part of the deployment of the process definition. Alternatively, the partner can be dynamically assigned at run time by setting the value of a variable in the workflow to the desired endpoint prior to the invocation of the activity in question. The URI for the Web Service can be obtained by the workflow, for example, as a result of the invocation of a Web Service earlier in the process flow.

3.2.2 Yet Another Workflow Language

Yet Another Workflow Language (YAWL) [van der Aalst 04] is a workflow language developed by Technical University of Eindhoven based on the work of van der Aalst on workflow patterns [van der Aalst 98]. YAWL is a graphical language that is influenced by earlier work on the use of petri-nets to describe workflow [van Der Aalst 03]. As such it is based on a 'net' or graph consisting of tasks to be completed.

The basic language constructs in YAWL are Tasks, Composite Tasks, Conditions and Flow Relations. Figure 3.1 provides an example of a simple YAWL process. As can be seen from the diagram, a YAWL net bears a strong resemblance to a petri net with Tasks corresponding to 'places' and Conditions to 'transitions'. Unlike in petri-nets, Tasks can be linked directly to other Tasks without the need for a Condition in between. In this case the Condition can be considered to have been added implicitly. Composite Tasks are used to link an abstract task within a workflow to an additional 'net' that defines the implementation of the task. Using this approach it is possible to break a complex workflow

[1] ActiveVOS was formerly ActiveBPEL.

into a hierarchical structure which is easier to author and manage.

Tasks in YAWL are linked to 'YAWL services', which are part of the extension architecture of YAWL that allows the workflow engine to be used to orchestrate different types of tasks, the supported services include the orchestration of Web Services using the 'YAWL Web Service Invoker' as well as SMS, Twitter and Email messages. Additionally, YAWL can be used to orchestrate human tasks using the 'Default Engine Worklist' service.

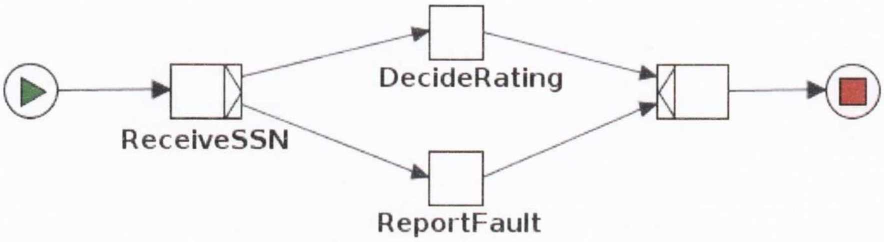


Figure 3.1: Simple YAWL process definition.

By linking together Tasks using the graphical editor it is possible to generate workflows. Tasks can be 'decorated' with 'splits' and 'joins', as illustrated in figure 3.1, in order to add more complex control flow to a composition. Three different types of split/join are supported, AND, OR and XOR. In the case of OR and XOR, conditions can be added to the task in order to control which path should be followed. These conditions are based on the value of variables defined within the 'net'. Conditions are used in the same way as transitions in petri-nets, they act as place holders to temporarily store tokens, for example to facilitate merges.

The YAWL platform consists of the YAWL Editor and the YAWL workflow engine. The engine itself is a web based platform that supports the management of users and processes. Unlike WSBPEL, processes can only be started through the web based user interface, which allows tasks to be assigned to users who can then accept or reject tasks as well as indicate that the task has been completed. As such a workflow running on the YAWL workflow engine cannot be directly invoked by a client application.

YAWL supports the dynamic selection of 'worklets' at runtime based on a set of rules that are written by the workflow designer [Adams 06]. This allows tasks within a parent net to be implemented by child nets that are dynamically selected at runtime from a pool of available nets.

This feature is based on YAWL's support for Composite Tasks except that worklets are selected dynamically at runtime based on the execution of a set of rules that are associated with the Composite Task. These rules can be modified independently of the process net itself and so allow for new candidate worklets to be deployed even after the original net. The rules dynamically select from the available worklets based on 'conditional' data about the running case that is stored in the case's process variables.

3.2.3 CAWE Framework

The Context Aware Workflow Execution (CAWE) Framework [Ardissono 10] is an adaptive workflow system based on the JBoss jBPM workflow engine [JBoss 11]. The aim of this system is to support user centric workflows that adapt based on the individual user as well as contextual properties such as the device they are using. The CAWE Framework supports the runtime adaptation of the workflow through the use of 'abstract activities', a construct similar to the Complex Tasks used in YAWL, that are adaptively bound to a concrete workflow implementation at runtime. To support this adaptive selection mechanism, the CAWE Framework utilises a set of models that represent the User, their Role and their Context. These models are used to influence the execution of 'adaptation policies', which are used to select an appropriate workflow at runtime.

The CAWE Framework provides the user with a graphical interface to the services that the workflow is composed of by generating HTML pages consisting of the necessary user interface components to represent the inputs and outputs of the service. This interface can be adapted to the needs of the user or to their context by adaptively selecting an appropriate stylesheet.

One of the advantages of the approach taken in the CAWE Framework is that it decomposes the workflow into a hierarchy of steps that can be adaptively selected. This reduces the complexity of the authoring process as well as the task of maintaining or modifying the workflow as the workflow does not become overly complex as is the case in systems that attempt to incorporate the possible adaptive branches into a single workflow process definition. The model driven approach also offers a significant advantage over the approach taken in YAWL to the dynamic selection of 'worklets', which is restricted to operating on variables contained in the workflow itself.

However, to compose an adaptive workflow for the CAWE Framework still requires the author to generate jBPM workflow definitions for every possible path in the workflow as well as the adaptation rules to carry out the adaptive selection. Even if the only difference between two possible paths is a single service instance.

3.2.4 C-BPEL

C-BPEL [Ghedira 06] is a personalised workflow system that supports the adaptive selection of services to instantiate the activities in a workflow at runtime. C-BPEL is an extension of the WSBPEL workflow language that allows for the dynamic selection of services by comparing models of the available services with a User Context Model and a Web Service Context Model. The author of such an adaptive workflow specifies the necessary control flow as is the case with a normal WSBPEL workflow but instead of explicitly referring to the services that should be executed, abstract activities are specified in the workflow. When, as part of the execution of a workflow, the C-BPEL engine is required to invoke an abstract service it instead makes a request to the 'Context Matching Module'. This component of the C-BPEL system is responsible for dynamically selecting an appropriate Web Service to fulfil the requirements of the abstract service specified in the workflow.

In order to allow the Context Matching Module to make a selection, the User Context Model and Web Service Context Model are passed as part of the request along with parameter values that are considered necessary for the execution of such a service. The UM captures information about the user including the device they are using, their preferences and their goals while the Web Service Context Model contains information about both the static and dynamic properties of the service that is required. The static properties include cost, access privileges, quality, etc. while the dynamic properties describe features such as availability and the response time of the service.

The selection mechanism used to identify appropriate services involves the comparison of the models passed to the Context Matching Module with the metadata model associated with all of the services known to the Context Matching Module. The matching algorithm looks for properties in the available service models that match the properties in the user model in order to identify the service that is most appropriate in order to implement that abstract service specified in the workflow.

3.3 Service Composition as Planning

Planning as a means of dynamically generating Web Service Compositions can be achieved through the application of many different techniques. Examples of such techniques include rule based planning, AI planning graphs, situation calculus and hierarchical task networks. This section provides an overview of some of the techniques commonly used in dynamic Web Service composition as well as some related technologies. Following this, systems that apply these techniques specifically to the problem of Web Service composition are discussed. A comparative analysis of the systems discussed in this section along with the service orchestration based systems discussed in the previous section is provided in section 3.5.

3.3.1 Planning Techniques

The automated generation of strategies consisting of sequences of actions is an area of Artificial Intelligence (AI) research that has seen significant advances in the last 20 years. Many different techniques have been developed to solve planning problems, from treating the problem as a search problem to advanced algorithms such as Hierarchical Task Networks (HTN) [Erol 94] and Graphplan [Blum 97]. This section introduces some of the basic AI Planning techniques that will be discussed later in this chapter in the context of Web Service composition systems. Also discussed in this section is the Planning Domain Definition Language (PDDL) as it is closely related to the topic of AI Planning.

3.3.1.1 Planning as Search

One approach that can be applied to the automated composition of services is to treat the problem as a search problem. Tree search algorithms such as forward chaining, e.g. breadth first search and depth first search, and backward chaining can be used to generate a solution to a planning problem by iteratively searching through the available services, the search space, in order to find services that are composable with the current service. That is, services whose input parameters match the output parameters of the current service. More advanced search algorithms, such as A*[Hart 68], which use heuristics to improve the performance of the search can also be applied.

3.3.1.2 Rule Based Composition

Rule based planning [Rao 04] is based on the selection and sequencing of services using a rule base consisting of rules that determine the appropriateness of a given service. These rules primarily look at three main areas message composability, operation composability and qualitative composability. Message composability deals with whether or not the output of a given service is compatible with the input of another. Two services are deemed to be composable if this is the case. Examples of systems that apply this technique to the composition of Web Services include [Arpinar 05] and [Chan 08]. Operation composability looks at whether or not the domain, categories, etc. of services are composable. For example a service in a financial services domain is unlikely to be appropriate if you are looking for a hotel booking service. Qualitative composability looks at selecting services that meet the requirements of the end user of the composition based on non-functional properties of the services such as quality of service, cost, etc.

3.3.1.3 Hierarchical Task Networks

Hierarchical Task Networks (HTN) [Erol 94] take a different approach to planning compared to classical AI techniques such as tree search algorithms or STRIPS style approaches such as Graphplan. A HTN attempts to solve a planning problem by decomposing a high level task, i.e. the goal of the planner, into a set of sub tasks. This decomposition process is carried out recursively until the high level task has been decomposed into a set of primitive tasks that can be executed. A HTN planning domain consists of tasks, operators and methods. Tasks are the set of 'things that need to be done', each of which consists of a task name and a list of arguments. Operators define the effects of each primitive task. Methods describe how non-primitive tasks can be performed by describing the possible decompositions for a given task.

HTN based planning is suitable for us in domains for which the methods for decomposing the domain based on a hierarchical structure exists. However, a limitation of HTN based planners is that the quality of the plans produced is limited to the quality of the human authored methods that describe how to decompose the tasks [Klusch 05].

3.3.1.4 Graphplan

Graphplan [Blum 97] is an example of a graph based AI planner and was the first to make use of this technique. Graph based planners construct a 'planning graph' in order to reduce the amount of search that has to be carried out to find a solution. As such, Graphplan's algorithm involves two steps, the generation of the graph and the search step.

The planning graph is generated by first taking the initial conditions (facts) that are known to the planner, this is the initial step or 'level'. The second level consists of all the actions that can be performed based on the initial facts. The third level is made up of all of the facts that might be true based on the effects of the actions from the previous step. This process is repeated iteratively.

The process of generating the planning graph reduces the overall graph size by removing incompatible propositions and actions for example an action that negates the effects of another action is removed. Another example of an action that can be excluded from the planning graph would be an action that deletes the preconditions of another action.

When a planning graph has been generated a solution to the planning problem can then be found using a backward chaining search.

3.3.1.5 PDDL

The Planning Domain Definition Language (PDDL) [Fox 03] is a language originally developed for use in the 1998/2000 International Planning Competition. The aim of the language was to provide a standardised means of describing AI planning domains and problems. PDDL is a STRIPS [Fikes 71] style language consisting of Types, Predicates, Actions, Initial State and Goal State.

```
(define (domain gripper-strips) (:predicates (room ?r) (ball ?b) (gripper ?g) (at-robby ?r) (at ?b ?r) (free ?g)
(carry ?o ?g))
(:action move :parameters (?from ?to) :precondition (and (room ?from) (room ?to) (at-robby ?from)) :effect (and
(at-robby ?to) (not (at-robby ?from))))
(:action pick :parameters (?obj ?room ?gripper) :precondition (and (ball ?obj) (room ?room) (gripper ?gripper) (at
?obj ?room) (at-robby ?room) (free ?gripper)) :effect (and (carry ?obj ?gripper) (not (at ?obj ?room)) (not (free
?gripper))))
(:action drop :parameters (?obj ?room ?gripper) :precondition (and (ball ?obj) (room ?room) (gripper ?gripper)
(carry ?obj ?gripper) (at-robby ?room)) :effect (and (at ?obj ?room) (free ?gripper) (not (carry ?obj ?gripper)))))
```

Figure 3.2: Gripper planning domain described in PDDL

Types are objects that exist in the domain, for example a ball or a room in the case of the classic 'gripper' planning problem defined in figure 3.2. Predicates specify the state of an object in the domain based on boolean logic.

Actions define the actions/operations that can be carried out. An action in PDDL is defined in terms of a unique name, a set of input parameters, a set of preconditions and a set of effects. Preconditions are set of 'goal descriptions' that must be satisfied before an action can be applied and are defined in the form of first order logic predicates. Similarly, effects are defined using predicates that specify changes to the domain that occur as a result of the action.

```
(define (problem gripperProblem) (:domain gripper-strips) (:objects rooma roomb ball4 ball3 ball2 ball1 left right)
(:init (room rooma) (room roomb) (ball ball4) (ball ball3) (ball ball2) (ball ball1) (at ball4 rooma) (at ball3
rooma) (at ball2 rooma) (at ball1 rooma) (at-robby rooma) (free left) (free right) (gripper left) (gripper right))
(:goal (and (at ball4 roomb) (at ball3 roomb) (at ball2 roomb) (at ball1 roomb))))
```

Figure 3.3: Gripper planning problem described in PDDL

A planning problem declared in PDDL consists of four parts, as defined in figure 3.3, the objects that exist in the domain, the initial state of the planning domain and the goal state of the domain. The initial state of the domain declares the predicates that hold true at the start of the planning problem. For example in the example planning problem the location of each ball in the gripper problem is specified with respect to a specific room. Similarly, the desired solution to the planning problem or goal state of the planning problem is defined in terms of the predicates that should hold true following the execution of the solution that the planner generates. It is also necessary to specify the planning domain to which the problem applies. This allows for the separation of the domain and problem definitions.

3.3.2 Web Service Composition Systems

The AI planning techniques discussed in the previous section are all designed to work on closed domains that are defined using special languages. In order for these techniques to be useful in the composition of Web Services, the gap between the technologies used in AI Planning, for example PDDL, and the technologies used in the Web Services domain must be bridged. This section describes a set of systems that attempt to do just that.

Before discussing the systems surveyed, a brief summary of OWL-S [Martin 04] is provided in order to provide some background to the discussion. OWL-S is a semantic technology commonly used by Web Service composition systems such as those discussed in this section as it is specifically designed to provide semantic descriptions of services.

3.3.2.1 OWL-S

The Semantic Markup for Web Services (OWL-S), based on the earlier DAML-S [DAML-S Coalition 02] specification, is a W3C specification for the semantic description of Web Services using the OWL ontology language. The aim of the OWL-S ontology is to support the discovery, invocation and composition of services not only by users but by software agents. The ontology consists of three three main parts, as illustrated in figure 3.4, the service profile, the process model and the grounding.

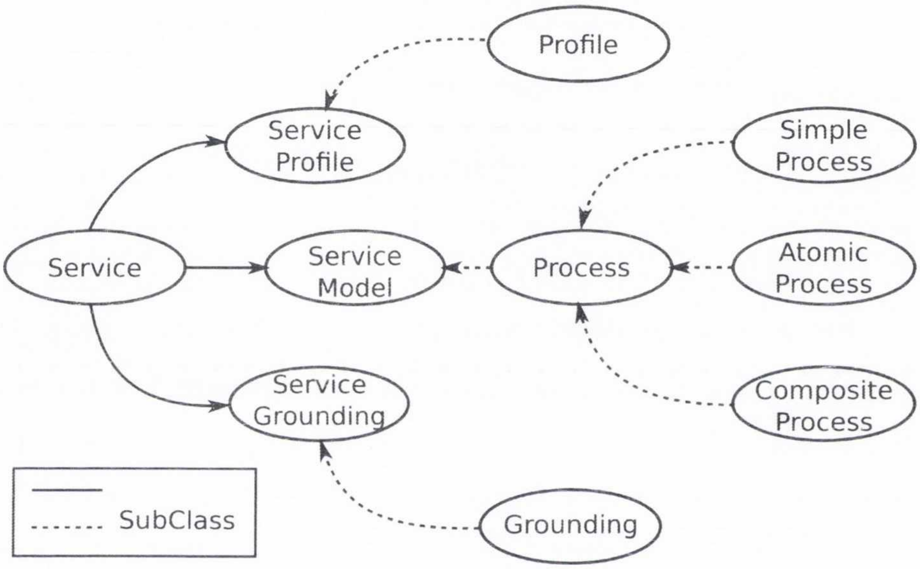


Figure 3.4: OWL-S Ontology Structure

Service Profile This ontology describes "what the service does", although it does not mandate a specific representation. However one possible representation is provided in the form of the Profile ontology which provides basic information about the service such as its name and a textual description. The Profile is primarily used to provide a functional description of the service with the inputs, outputs, preconditions and effects of the service defined as properties. The Profile also allows for the service to be categorised by means of the 'serviceCategory' property, which can be used to relate the service to a separate ontology representing a structured categorisation of service types.

Service Model This ontology describes the service as a process, allowing a client to understand how to interact with the service. The Service Model is represented by the Process ontology, which has three subclasses Atomic Process Simple Process and Composite Process. An Atomic Process represents a single service that has a concrete implementation, i.e. a service that can be directly invoked in a single step by a client. Similar to an Atomic Process, a Simple Process represents a service that can be completed in a single step. However, a Simple Process does not have an associated service implementation and cannot be executed by a client. Instead it is intended to be used to provide a level of abstraction.

Composite Processes are processes that can be decomposed into other processes, either Atomic, Simple or Composite. The composite process can be described in terms of set of control flow constructs, sequence, split, split-join, any order, choice, if-then-else, iterate, repeat-while and repeat-until. Although a composite process is described using these control flow constructs it is not intended to act as a workflow to be executed but rather describes how a client can interact with the service in order to achieve the desired outcome.

The Service Model also provides mechanisms through which data flow can be defined between, for example to output of one service and the input of another.

A lot of work has been carried out on the mapping of WSBPEL to OWL-S such as [Shen 05] while others have focused on the inverse mapping from OWL-S to WSBPEL [Fuentes 06]. It is not clear, however, that this work aligns with the original purpose of Composite Processes in the OWL-S ontolgoy.

Service Grounding This ontology describes how a client can access a service by providing a mechanism through which the service can be grounded to some implementation specific details such as addressing, protocols, etc. One such grounding is from the OWL-S ontology to WSDL.

3.3.2.2 SHOP2

SHOP2 [Nau 03] is a HTN based AI planner, which was built upon by Sirin et al. [Sirin 04] in order to support the composition of Semantic Web Services. In this system the Web Services are described using an OWL-S ontology, facilitating their discovery and execution by machines.

In SHOP2, operators are not only described in terms of parameters but also have associated preconditions and effects.

In order to be able to plan over a domain described using OWL-S, the authors developed a translator that mapped a set of OWL-S process ontologies to a SHOP2 planning domain that could then be reasoned over by the planner. The translation involves the mapping of OWL-S atomic processes to SHOP2 operators while simple and composite OWL-S processes are mapped to SHOP2 methods. SHOP2 was then used to generate a plan based on the resulting domain with the resulting solution to the planning problem transformed back into OWL-S.

The solution/plan generated by SHOP2 is limited to a sequence of services as SHOP2 does not support concurrency [Sirin 04]. Another limitation is that the solutions do not support data flow between the services. As the solutions generated by the SHOP2 based system are transformed back into OWL-S, they are intended to describe how a client can invoke the services in the appropriate order to achieve the desired outcome, as such the plan itself is not intended to be executable.

3.3.2.3 OWLS-XPlan

Another system that applies AI planning techniques to dynamically compose semantically described Web Services is OWLS-XPlan [Klusch 05]. This system is based on a hybrid AI planner, XPlan, that combines a graph-plan based planner with a HTN component. As was the case for the SHOP2 based planner discussed previously, OWLS-XPlan transforms a set of OWL-S ontologies into a planning domain and problem before attempting to solve that problem using the AI planner.

The mapping of OWL-S in this case is to PDDL as XPlan's graph-plan based planner operates over more conventional STRIPS style planning domains. As such, the goal of the converter component is to map OWL-S service descriptions to PDDL actions. OWL-S service inputs, preconditions and effects are relatively easy to map to their equivalents in PDDL. However, OWL-S output parameters do not have any corresponding equivalent in PDDL, OWLS-XPlan addresses this limitation by treating them as a special type of PDDL effect.

The service compositions generated by OWLS-XPlan are in the form of a sequence of actions that should be executed in order to have the desired outcome. A such, they do not support any of the control flow constructs supported by OWL-S. To address this issue, the designers of OWLS-XPlan propose that the solutions generated by OWLS-XPlan can be analysed in order to identify parts of the solution that do not contain dependencies between services and insert appropriate split and join control flow constructs into the solution [Klusch 05]. However, this approach is still limited as control flow constructs such as 'choice' and 'unordered sequence' cannot be identified in this way.

3.3.2.4 PORSCE II

PORSCE II [Hatzi 09] is another Semantic Web Service composition system. As was the case with SHOP2 and OWLS-XPlan, this system is designed to compose services based on OWL-S ontologies. PORSCE II only considers the inputs and outputs defined in the OWL-S service description when generating a planning problem. For each service description, PORSCE II generates a PDDL action with the OWL-S hasInput mapped to preconditions of the action and the OWL-S hasOutput mapped to the effects of the action. As such, PORSCE II does not take into account the OWL-S preconditions or effects when mapping the service descriptions of available services to a PDDL planning domain.

An interesting feature of PORSCE II is its ability to generate 'semantically relaxed' service compositions in which equivalent or semantically close services might be selected by the planner when an exact semantic match is not available. This is achieved by analysing the outputs of the services and comparing them to the concepts in a domain ontology used by the system. If a semantically similar concept is found in the domain ontology for an output of a service (represented as an effect in the generated PDDL) then that concept is added as an additional output of the service.

The use of PDDL allows PORSCE II to make use of any third party AI Planner that supports the language although the implemented system only supports JPlan [EL-Manzalawy 04] and LPG-td [Gerevini 04]. The planning problems that are to be solved by PORSCE II are provided by an end user through a graphical interface that allows that user to select concepts from the domain ontology used by the system. The plans produced by the respective AI planners can differ depending on how the planners work, for example the plans produced by JPlan are purely sequential where as LPG-td breaks the plans down into steps in which services can be run in parallel if there are no dependencies between them.

3.4 Related Technologies

3.4.1 Mashups

An emerging trend in Service Orientated Architecture (SOA) and service composition is to have light weight compositions that can be generated by users without the need for a high degree of technical knowledge using simple graphical user interfaces. These compositions or 'Mashups' were originally aimed at use cases that required the combination of data from multiple sources, for example merging data from a RSS feed with data scraped from another website to meed the needs of a user. Platforms such as Yahoo! Pipes [Yahoo! 11] provided such functionality, allowing users to link together various data sources, including Web Services described using WSDL. However mashups, by virtue of their lightweight nature, were never intended to provide the sort of functionality that workflow systems such as BPEL and YAWL provide. As such, they generally do not provide any fault handling capabilities and have limited support for control flow. Surprisingly, mashup platforms such as Yahoo! Pipes or WSO2 Mashup Server [WSO2 11] do not provide any mechanisms for generating user interfaces. This is left to the user to develop an interface based on traditional web technologies such as JSON, AJAX, etc.

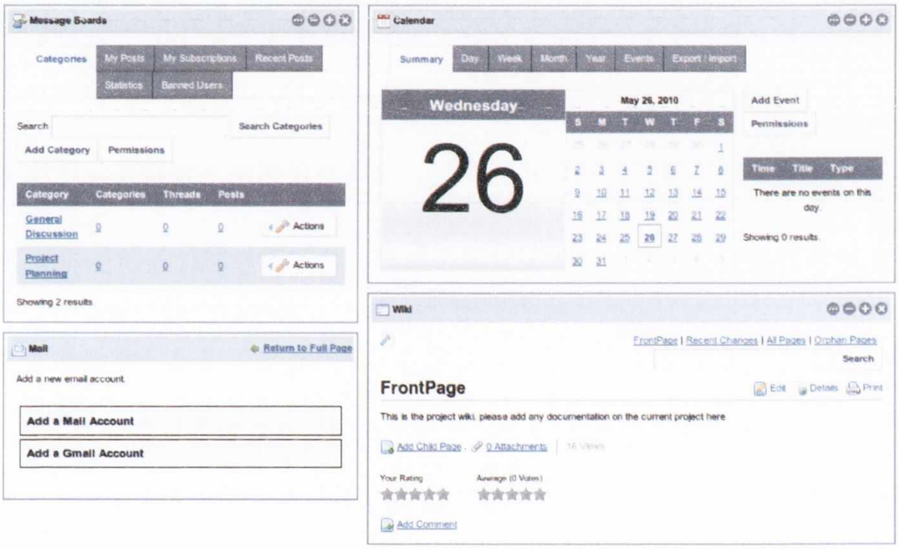
Due to the popularity of user based mashups, they have seen significant interest from more traditional business users, this has resulted in the emergence of two additional classes of mashup, Business Mashups and Enterprise Mashups. Business Mashups are seen as mashups that combine internal business data sources as well as web based data sources. The aim of such business mashup platforms is to provide businesses with the same degree of flexibility as earlier mashup platforms but with the security of hosting their own service compositions. With the corporate adoption of mashups has come a proliferation of platforms including IBM Mashup Center [IBM 10] and JackBe Presto [JackBe 10]. The wide range of platforms available has also seen efforts to develop standards for the definition of mashups. The Open Mashup Alliance[2] have developed the Enterprise Mashup Markup Language (EMML) [Open Mashup Alliance 09], which pushes mashups further

[2] The OMA consists of a wide range of companies including Adobe, Intel, Capgemini and HP.

into the traditional business process domain with support for process flow constructs such as **if**, **for**, **foreach**, **while** and **parallel**. EMML also supports the use of scripting languages such as Javascript and Ruby to manipulate the data as well as XSLT.

3.4.2 Portlets and WSRP

Portal servers are a popular technology for the aggregation of web based content that use the familiar computer desktop paradigm in which individual content sources or widgets are represented as windows or 'portlets' in a shared workspace. Figure 3.5 provides an illustration of such a portal server interface[3]. One of the driving motivations behind the paradigm is that it allows for a highly configurable user interface that can be customised to meet the needs of a specific class of users or even the individual user.



Category	Categories	Threads	Posts	
General Discussion	0	0	0	Actions
Project Planning	0	0	0	Actions

Showing 2 results

Add a new email account

There are no events on this day.

Showing 0 results.

This is the project wiki, please add any documentation on the current project here

Figure 3.5: Example portal user interface consisting of four individual portlets

A significant limitation of such portal servers is the need to install the necessary portlets locally on the portal server. This means that it is not possible to take advantage of a more service orientated approach to the provisioning of the functionality that a portlet provides. This limitation can also be seen as a flaw in the attempts of portal servers to act as aggregators. From a technical perspective, this means that a portlet might have to be rewritten in order for it to be used on a different platform. For example, a JSR-168[4] compatible Java based portlet would need to be rewritten completely in order to run on

[3] Portal interface shown is from the Liferay portal server

[4] JSR-168 is a Java Community Process specification for a Portlet API.

Microsoft's Sharepoint portal server.

One approach that can be used to address this issue is to make use of the Web Service for Remote Portlets (WSRP) [Oasis Consortium 08] specification. This is an OASIS specification that defines a Web Service based protocol for the delivery of portlets, allowing portlet providers and consumers to be decoupled.

The WSRP specification defines four services that a portlet producer, a server that provides portlets, must implement in order to take part in the WSRP protocol. These include a service description interface, a registration interface, a markup interface and a portlet management interface.

The service description interface is used by a portlet consumer to query a provider for information about the portlets that it provides, as such it facilitates the discovery of portlets given a known provider and allows a consumer to get information necessary to add a portlet. To support this functionality, the service description interface provides a single operation, **getServiceDescription**.

The registration interface allows consumers to register with a producer. This is not always a required step but if used can allow the producer to customise portlets for a given consumer based on the capabilities of that consumer. Registration can also be used by a producer to restrict the portlets or capabilities that it reports to the consumer via the service description interface. To support this functionality, the registration interface provides three operations: **register**, **modifyRegistration** and **deregister**.

The markup interface allows a consumer to request the information necessary to render a portlet from the producer. It is also used by the consumer to notify the producer of any interactions that a user has with the portlet. The markup that the producer passes to the consumer is in the form of a fragment of HTML. The markup interface provides the following operations:

- getMarkup
- performBlockingInteraction
- initCookie
- releaseSession

The portlet management interface allows a Consumer to mange the persistent state of a portlet. Portlets can have persistent properties that affect their behaviour, such a portlet is referred to as a 'Consumer Configured Portlet' and can be created by a Consumer using the **clonePortlet** operation. Subsequently, the properties of a cloned portlet can be modified using the **setPortletProperties** and **getPortletProperties** operations. A cloned portlet can be destroyed using the **destroyPortlets** operation.

Assuming that a consumer has successfully discovered the desired portlet and registered with the producer, the consumer can then make use of the getMarkup and performBlocking interaction operations of the markup service to allow it to provide the portlet to the user. Figure 3.6 provides an activity diagram illustrating how these operations are used as part of a two step protocol. The first step is to request the markup fragment for the portlet from the provider using the getMarkup operation (step 1 in the diagram). The fragment can then be used to render the UI of portlet as part of the webpage seen by the user (step 2). This process can involve rewriting certain URLs in the portlet interface so that the consumer can intercept requests made as a result of a user's interaction with the portlet.

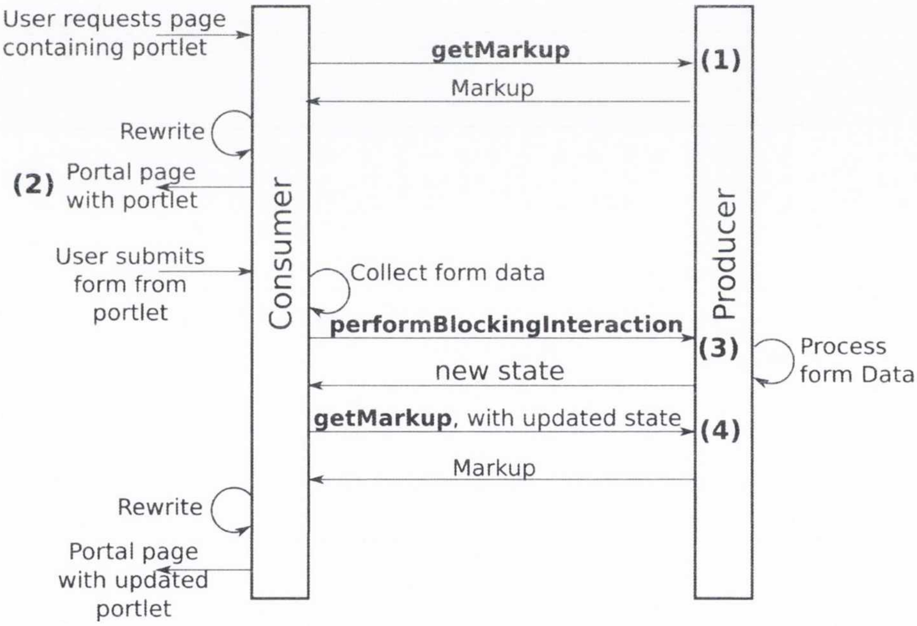

User requests page containing portlet
getMarkup
(1)
Markup
Rewrite
(2)
Portal page with portlet
User submits form from portlet
Consumer
Collect form data
Producer
performBlockingInteraction
(3)
Process form Data
new state
getMarkup, with updated state
(4)
Markup
Rewrite
Portal page with updated portlet

Figure 3.6: Activity Diagram of the WSRP two step protocol.

If the user does carry out an action that requires the portlet to process so data, for example if the user clicks a button in the portlet or submits a form full of information, then the consumer can notify the producer using the performBlockingInteraction operation (step 3). To invoke this operation, the consumer sends to producer a set of key-value pairs corresponding to the fields in the user interface that were updated. To provide the user with an updated interface for the portlet based on the performed action the consumer must again request the markup fragment for the portlet (step 4).

3.4.3 WSDL

The Web Service Description Language (WSDL) [Curbera 01] is a W3C specification for the description of services using a XML grammar. The aim of this specification is to describe a service in terms of the location of the service and the messages that need to be passed to the service in order to invoke it. As such, it aims to allow different Web Service implementations, for example JAX-WS [JAX 10] and Microsoft .NET [Microsoft 10c], to interoperate with each other. The WSDL specification supports the description of services that utilise either SOAP or HTTP as their underlying communication mechanism. In fact, a WSDL document can provide descriptions of both SOAP and HTTP service interfaces in the case where the service implementations supports both protocols.

```
<definitions name="HelloService"
    targetNamespace="http://www.examples.com/wsdl/HelloService.wsdl"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://www.examples.com/wsdl/HelloService.wsdl"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <message name="SayHelloRequest">
        <part name="firstName" type="xsd:string"/>
    </message>
    <message name="SayHelloResponse">
        <part name="greeting" type="xsd:string"/>
    </message>
    <portType name="Hello_PortType">
        <operation name="sayHello">
            <input message="tns:SayHelloRequest"/>
            <output message="tns:SayHelloResponse"/>
        </operation>
    </portType>
    <binding name="Hello_Binding" type="tns:Hello_PortType">
    <soap:binding style="rpc"
        transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="sayHello">
        <soap:operation soapAction="sayHello"/>
        <input>
            <soap:body
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:examples:helloservice"
                use="encoded"/>
        </input>
        <output>
            <soap:body
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:examples:helloservice"
                use="encoded"/>
        </output>
    </operation>
    </binding>
    <service name="Hello_Service">
        <documentation>WSDL File for HelloService</documentation>
        <port binding="tns:Hello_Binding" name="Hello_Port">
            <soap:address
                location="http://www.examples.com/SayHello/">
        </port>
    </service>
</definitions>
```

Figure 3.7: Example WSDL document describing a simple 'Hello World' service

Figure 3.7 provides an example of a WSDL document that describes a simple 'Hello World' web service. The starting point for a WSDL document is the **service** element.

This provides the name of the service and contains one or more **endpoints/ports**. Each endpoint that is defined in a WSDL document corresponds to a specific communication protocol that the service supports. As such, each endpoint has an address in the form of a URL that represents the point to which messages should be sent. Each endpoint refers to a **binding**, which describes the specific details of the service interface in accordance with the communication mechanism used by the endpoint. The binding provides technical information about the operations that the service provides, for example in the case of a HTTP based binding, the HTTP method used. The actual specification of the message type that should be passed to the service during the invocation of a specific operation is defined as part of the **interface**. The interface is an abstract definition of the operations that the service provides and the message types that are used as inputs and outputs of the operation. The input and output messages for each operation are defined in the **type** element of the WSDL document, which can either embed a XML schema describing the message types or refer to an external schema document.

3.5 Comparing Composition Techniques

This chapter has presented a review of the different techniques, both static and dynamic, for the composition of Web Services. As discussed, these techniques each have their own advantages and limitations. To compare and contrast these techniques, serveral properties have been identified that are relevent when discussing service composition as it might be applied in the generation of personalised web experiences.

- **Static/Dynamic Composition** Whether the composition is generated by hand by a domain expert or by a machine.
- **Workflow** Can control flow constucts be used to control the order of execution of services in the compostion.
- **Adaptivity** Can the composition be adapted to better suit the needs of the user or the context in which it is being executed.

The findings of the comparison based in these properties are summarised in table 3.1 and subsequently discussed in detail.

Static/Dynamic As mentioned previously, service composition techniques can be broadly categorised as either static or dynamic. Static compositions are those that are

	WS-BPEL	**YAWL**	**CAWE**	**C-BPEL**	**Planning**
Static/Dynamic	Static	Static	Static	Static	Dynamic
Workflow	YES	YES	YES	YES	NO
Adaptivity	NO	Limited	Limited	Limited	Limited

Table 3.1: Summary of comparison of service composition techniques.

authored by a domain expert while dynamic compositions are generated automatically by an intelligent agent. Obviously work in the area of workflow composition such as eFlow and C-BPEL illustrate that the line between these two categories have been blurred, however, they are still useful categorisations as a means of comparing systems based on the degree of dynamism that exists within the composition and how much manual effort is involved in generating the composition.

Clearly workflow techniques such as WSBPEL are for the most part static with the need for a domain expert to explicitly define the entire composition, both selecting the appropriate services and specifying the control flow between those services. Systems such as YAWL and the CAWE Framework allow for a degree of runtime dynamism in the workflow through the dynamic selection of sub processes but this still requires the domain expert to author all of the possible branches in the subprocesses as well as the rules that govern the selection process. As such, these approaches can be seen as simply a mechanism through which the authoring and maintenance of complex workflows can be simplified. Similarly, C-BPEL only supports the dynamic selection of services based on the needs of the user, requiring the control flow and indeed the data flow to be authored at design time.

The dynamic composition of services as illustrated in SHOP2, OWLS-XPlan and PORSCE II address this authoring complexity through the automated selection and sequencing of services, assuming that the planning system has access to sufficiently detailed service descriptions. In this case, the end user/designer is only required to specify the desired outcome of the composition.

Workflow The ability to explicitly specify the sequencing of services in a composition can be of critical importance in application domains in which the business logic plays an important part in the execution of a service composition and the end result is not the only concern. In eLearning, for example, the sequencing of task in an activity, such as those the proposed system aims to deliver, is critically important. In this case dynamic composition techniques such as AI planning can be problematic as the planner is only concerned with achieving the goal state. Furthermore, as discussed previously in this chapter, dynamic service composition techniques are often limited to producing compositions that are purely sequential and do not support complex control flow constructs such as branching and merging.

In cases where control flow is of such importance, static workflow techniques have an obvious advantage as they are designed specifically to meet this need. The need for human authoring allows for the domain expert's business logic to be embedded directly into the composition. While the advanced support for complex control flow available in modern workflow systems such as WSBPEL and YAWL allow complex business process to be represented.

Adaptability The ability to dynamically adapt a service composition to meet the needs of the user or to the context in which the composition is to be executed can result in significant improvement in the overall composition relative to a 'one size fits all' static compostion.

The workflow based systems that have been discussed in this chapter support varying degrees of adaptivity. The WSBPEL language supports the dynamic resolution of business partners allowing the actual Web Service used to implement an activity in the workflow to be specified at runtime, for example based on a parameter passed to the process as part of the execution of a previous activity in the workflow. This feature can be seen as facilitating the adaptive selection of Web Services although the mechanism used to carry out the selection is not part of the WSBPEL specification and would need to be implemented separately.

YAWL goes a step further by allowing the dynamic selection of 'worklets' based on the execution of rules at runtime. This technique can be used to implement both adaptive selection and adaptive sequencing although the implementation of the selection could be seen to be overly complex as it would require a degree of workflow to be included in the worklet. The limitation of this approach is that the rules that govern the dynamic selection are limited to operating on 'contextual' data available as process variables.

Despite their obvious advantages, the techniques applied in YAWL and WSBPEL are still limited in that the workflow and the services used to implement the process must all be explicitly defined by the author at design time.

The CAWE Framework takes the sub process technique used in YAWL to a higher degree of flexibility by basing the selection of sub processes on a model driven approach in which both the adaptation rules and the information they operate on are separated from the process itself. Of the systems discussed, both static and dynamic, the CAWE Framework and C-BPEL are the only systems that explicitly take into account the needs of the user. However, C-BPEL is limited in that it only supports the adaptive selection of services and does not provide any mechanisms for adaptive sequencing.

Although the focus in Web Service composition systems such as those discussed in this chapter is on the dynamic selection and sequencing of services based on the description of the service from a functional perspective, it is clear that the techniques can be easily applied to support the adaptive selection of services based on, for example, the needs of the user. The AI planning techniques already require that initial properties of the problem are provided to the system. If these were used to provide additional contextual or user info, the planner would automatically take this information into account. Obviously this would also require the services to be described in terms of the non-functional properties that we are interested in adapting on. This could be achieved though the use of the preconditions and effects that are already used to describe the services in an AI planning domain.

3.6 Summary

In this chapter, various different techniques for the composition of Web Services have been discussed. Based on the state of the art systems that were described, covering both static and dynamic composition techniques, an analysis was carried out with the aim of comparing these systems and the techniques they apply. From this analysis it is clear that the state of the art in workflow based composition provides strong support for the specification of control flow, though at the cost of effort at design time and limited support for change in the workflow except through further engineering effort. Planning techniques address the authoring complexity of workflow as well as facilitating the need for dynamic change in the composition. This is achieved at the expense of control flow which is very limited in planning based systems.

With respect to adaptivity and specifically personalisation, none of the systems presented provide strong support. Although planning based systems are well positioned to take into account the needs of the user, they are primarily focused on dealing with change in the environment, such as the availability of Web Services and non functional properties such as quality of service and cost. Of the systems discussed, only the CAWE Framework and C-BPEL explicitly support adaptation based on the needs of the user, with the CAWE Framework supporting adaptive control flow and C-BPEL supporting adaptive selection of Web Services. Nether supports the full range of adaptive behaviours, i.e. adaptive control flow and adaptive selection.

Chapter 4

Design

4.1 Introduction

This chapter presents the architecture for a system that supports the adaptive selection and sequencing of both content and services in order to deliver an educationally sound learning activity to the learner. The influence of the state of the art, discussed in chapters 2 and 3, on this architecture is also discussed.

This chapter continues by providing an overview of the architecture of the system, clearly defining the fundamental objectives of the system and describing how each of these objectives are met by the architecture. This overview is followed by a more detailed discussion of how each of the individual objectives of is met.

The approach taken in the architecture to model the different concerns is discussed first. This is followed by details of how the system can utilise these models in conjunction with an adaptation strategy in order to compose an educationally meaningful activity. The chapter continues by providing further detail on how services are modelled in order to support their composition within the context of an adaptive strategy while taking advantage of existing work on service composition.

Next we discuss how a set of appropriately sequenced services can be delivered to the user while maintaining the sequencing that is explicitly defined in the adaptation strategy as well as the sequencing and selection that has been carried out by the system itself. As part of this discussion of sequencing and delivery of services, we identify a set of workflow patterns that the system will support.

Finally, this chapter provides a more detailed discussion of how the proposed system will support the execution of the adaptation strategy and provide access to the required models.

4.2 Requirements

In Chapter One, the following question was posed as the subject of this research:

"What are the appropriate techniques and technologies required to support the delivery of personalised web based experiences."

In addressing this question, and based on the state of the art reviews that were carried out in the domains of personalised content and service composition, a series of requirements were identified. These requirements fall into one of two categories, Educational Motivations and Technical Requirements.

4.2.1 Educational Motivations

From the analysis of learning theory presented in Chapter Two we can see a clear motivation for the use of personalisation as a means to tailor a learning activity to the needs of the individual learner. This applies to both the composition of content and activities. The motivation for the focus on eLearning activities themselves is also drawn directly from learning theory where the engagement of the learner in more active forms of learning is promoted by the three different learning theory perspectives discussed.

From the state of the art review of personalised eLearning systems it is clear that although such systems have addressed the need for personalisation, they have not yet applied this to the personalisation of activities, which not only require the personalised delivery of content but also of services.

The state of the art review also provided an insight into the types of activities that eLearning practitioners design and apply in their teaching. Activities such as those developed by the LADiE and DialogPlus projects illustrate the requirement for control flow that is not currently supported by existing personalised eLearning systems. The LADiE activities, for example, require services to be provided in parallel. Such activities also illustrate the need for any system that supports the personalised composition of activities to support both multimedia content and services. This is necessary because, as shown by the types of tasks that make up the LADiE use cases, they are a mix of both content and services involving a range of different tasks including discussion, creation and reception of information.

4.2.2 Technical Requirements

In addition to the educational motivations that were identified, a series of technical requirements were also identified based on the state of the art review of adaptive content and service composition.

A common theme across the personalised eLearning systems surveyed was the adoption of a multi model approach that separates the metadata models and adaptation rules from the Adaptation Engine itself. For example the Multi-Model Metadata Driven Approach taken by APeLS and that AHAM based approach in AHA!. This approach ensures that the Adaptation Engine can be applied across a wide range of applications while also facilitating the reuse of both the adaptation rules and the metadata models.

An additional requirement that stems from the application of a multi-model based approach is that the system should also be flexible in the models that it can interpret. The system should not, for example, be constrained to a fixed information model for any of it's models but rather be capable of supporting many different standards and schemas, for example IMS Learning Design. This approach would address some of the limitations that, for example, APeLS had in terms of it's fixed Learner Model structure.

One of the limitations of systems based on Learning Design such as Coppercore is that it is difficult for the developer of an activity to make use of services that are external to the system. The system developed in addressing the research question should take a more open approach to the use of both content and services. To achieve this, techniques from the service composition domain should be used to inform the design of the system.

As discussed in the analysis of the state of the art in Adaptive learning (Chapter 2), there is limited support for the adaptive selection and sequencing of services in the systems surveyed. The design of the system implemented as part of this research should address this limitation and in so doing draw from the state of the art in Adaptive Service Composition

(Chapter 3).

One of the technical requirements for a system that is capable of delivering personalised learning activities is that it can not only adapt the control flow between services but also enforce that sequencing as users interact with their personalised activities. From the state of the art review it is clear that orchestration technologies such as WSBPEL are appropriate solutions to this requirement as they are primarily focused on maintaining such control flow although generally as it applies to business processes rather than eLearning.

The next requirement is the need to adaptively select services in order to instantiate the learning activity. This would seem to be a similar problem to that of content selection. In that case, techniques such as Candidacy and metadata driven search would seem appropriate. However, this does not account for some of the issues that are specific to services such as parameterisation and information flow between services. Both of these issues are taken into account by dynamic service composition technologies such as AI Planning, as shown in the state of the art review.

From the review of AI Planning techniques, it can be seen such approaches may not be appropriate for the composition of an entire learning activity due to the requirement that they are pedagogically sequenced. However, their ability to dynamically compose services to solve a problem can be used to generate small compositions to meet the requirements of a single task. Such a feature would allow the system to adaptively generate compositions where no single service exists to meet the requirements of a given task.

4.3 Architecture Overview

The proposed architecture is influenced heavily by the principles that underpinned APeLS, discussed previously in the state of the art chapter (section 2.4.3). Of most interest are the concepts or 'narrative' and the multimodel, metadata driven approach. Both of these principles have been adopted for this system. By adopting the use of narrative, the sequencing of both content and services can be influenced by an expert activity designer allowing the system to produce activities that are educationally sound. The application of the multimodel, metadata driven approach means that the 'intelligence' of the system is moved from the core of the system towards the edges. As such both the modelling of the application and the application logic itself are separate from the system enabling it to be flexible with respect to the application domains that it can operate in. By applying both of these techniques, the proposed architecture is designed to deliver a wide range of different adaptive activities.

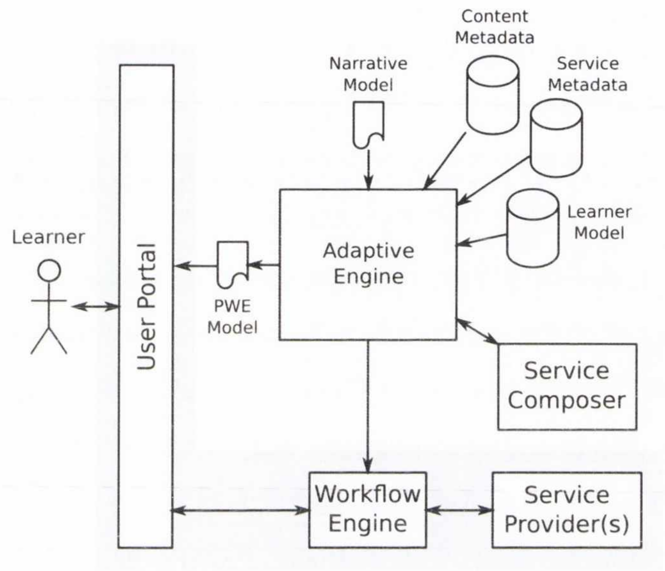


Figure 4.1: A high level overview of the system architecture

As illustrated in figure 4.1, the core component of the system is the Adaptive Engine (AE). Although based on the same underlying principles as the AE developed for the APeLS system, this component represents a complete redevelopment, addressing some of the limitations of the earlier design as outlined in the discussion of APeLS in the state of the art chapter. The AE is responsible for the execution of the educational strategy, which is referred to as the 'narrative'. As well as embodying the educational strategy, the narrative also describes the adaptive behaviours that the adaptive activity supports. As the AE executes the narrative, it reconciles the metadata models that are made available to it in order to adapt the activity to the needs of the learner.

To adaptively compose an activity, a set of three metadata models need to be made available to the AE. This is in addition to the narrative, which can also be considered as a model. These three model types are the Content Model, Service Model and Learner Model. The Content Model describes all of the content that is available to the AE for selection, similarly the Service Model describes all of the services that the AE can select in order to satisfy the requirements of the narrative. The Learner Model describes the attributes of the learner that are important to the system. The ability of the AE to reconcile models is not limited to these three model types but these three models represent a minimal set required for the system to adaptively compose an activity.

As part of the selection and sequencing of services, the AE makes use of an external component, the Service Composer, which is capable of dynamically selecting appropriate services from the collection of available services, which the AE is made aware of through the Service Model. This component will not only support the selection of existing services to meet a specific set of requirements but will also enable the dynamic generation of service compositions from the available services in order to satisfy requirements that are not be met by the existing services.

The delivery of the activity to the user requires that the system is able to make the services available in a manner that allows the user to interact with the appropriate service while still maintaining the sequencing of the service composition. To support this functionality, the system deploys the personalised service composition (generated by the AE) to a workflow engine. This acts as a proxy between the user and the services that they are interacting with and allows services to be delivered to the user in accordance to the design of the activity. The services that are orchestrated by the workflow engine are delivered to the user as part of the activity in an integrated environment along with the personalised content allowing the user to access the content and interact with the appropriate services in a unified manner.

4.4 Modelling

As mentioned in the previous section, the process of adaptively composing an activity is driven by the use of metadata models. These models provide the adaptive system with important information about the various aspects of the system that can have an influence on the adaptation process.

The modelling of information in this system is based on the multi-model approach taken in APeLS [Conlan 05, Conlan 02]. As such, the different concerns of the system are modelled as separate entities from each other. This allows the system to be flexible as different model types can be added or removed from the system without affecting the other models that are in use by the system.

In order to be able to apply this multi-model based approach it is necessary for all of the models used by the system to share a common vocabulary. This means that the terms used to describe learners, concepts, content, etc. in the different models should be consistent in order to allow the system to reconcile the models with each other as part of the adaptation process. The vocabulary used in the models is not controlled by the system at runtime although the use of inconsistent vocabulary in the metadata models will result in unpredictable behaviour.

As the metadata models used by the system underpin the adaptation process, it is important that they are able to provide the system with the necessary information in a way that is flexible enough to allow the metadata to be semantically rich while not imposing restrictions on the type of information that it is possible to capture. To achieve this, the system uses XML [W3C 99a] to encode the metadata models. The use of XML, as a meta language for describing markup languages means that there are very few limitations on the information that can be captured in the models. This flexibility comes at the cost of having to deal with the loosely typed nature of XML, however, this system is designed to be used in domains in which the sources of metadata will have their own controls. Based on this, it is not necessary to require the explicit definition of metadata schemas for the metadata models using technologies such as XML Schema [W3C 10]. Such an approach would require the schemas to be registered with the Adaptive Engine and for every model loaded to be validated against the schema. Such an overhead would provide very little benefit and could be redundant as the metadata models are already validated when they are stored in their respective repositories. Furthermore, the schema for the metadata models used by the system remain static over the life of the application (the instance of the system delivering a specific PWE). As such, technologies such as RDF [W3C 04], which are designed to handle dynamic information models, are not necessary.

Another advantage that is afforded to the system by the use of XML is that it allows the system to support the use of existing markup standards. XML bindings exist for many specifications that are used by systems in the eLearning domain. For example, the IMS Learner Information Profile (LIP) [IMS 05] specification that can be used to model learners and the ADL SCORM [ADL Initiative 09] specification for describing content.

The use of the multi-model, metadata driven approach [Conlan 02], when applied to narrative, means that the composition strategy and the adaptive engine are independent of each other. This means that the system can be used to compose different activities without the need to modify the system architecture.

As proposed in this thesis, the adaptive composition of an activity involves the reconciliation of four model types, the Narrative, Learner, Content and Service models. The Narrative Model contains the adaptation rules used to compose the activity. The Learner Model provides the system with information about the learner necessary for adaptation. The Content Model describes all of the content that is available to the system for use in an activity. The Service Model describes the services that are available to the system for selection. In addition to these models, the system also produces a model of the activity, the PWE Model, as the output of the composition process. This is a representation of an activity that has been personalised to an individual learner.

4.4.1 Narrative Model

The narrative model is the embodiment of the strategy that guides the adaptation process. As such, it provides a framework around which the adaptation is hung. In this role, the narrative serves two functions. First it provides the system with an outline for activities that are described in abstract terms. This outline is the basic structure that all of the activities composed by the system should have, irrespective of any adaptive behaviours. The second function of the narrative is to describe these adaptive behaviours.

In order to adaptively compose an activity, the narrative supports two fundamental behaviours, adaptive sequencing and adaptive selection. These behaviours can be applied to both content and services. The support for adaptive behaviours in the narrative is discussed further in section 4.6.

The encapsulation of the adaptation strategy in a model provides two significant benefits, first is the separation of the strategy from the Adaptive Engine, facilitating the reuse of the engine to execute other strategies. The second related benefit is that it allows the strategy to be separated from the content and services that are being composed, allowing the strategy to be applied to different collections of content and services.

4.4.1.1 Sequencing Constructs

To be able to compose educationally meaningful activities, it is necessary to be able to sequence the services that make up the activity. Unlike the composition of the content parts of the activity, the sequencing of services needs to be enforced both during the adaptive selection of tasks by the Adaptive Engine and while the user interacts with the services. This requires that constructs used by the narrative author to define the sequencing of services are well defined and that they can be interpreted by the system so that the delivery of the activity to the user is consistent with the activity as defined by the activity designer in the narrative.

When selecting a set of constructs to use as the building blocks for the sequencing of tasks it is important to ensure that they represent a sufficiently rich set of behaviours that will allow the activity designer to create educationally meaningful activities.

The work of van der Aalst [van Der Aalst 03] on workflow patterns was used to identify a set of constructs that would be used in this system. As part of this work, van der Aalst identified a set of control flow patterns that can be used to sequence services in a workflow. From this set of patterns, the five 'basic' control flow patterns have been identified as providing the necessary functionality for this system. Table 4.1 provides an overview of the 5 basic control flow patterns that will be supported by this system, namely: Sequence, Parallel Split, Synchronisation, Exclusive Choice and Simple Merge.

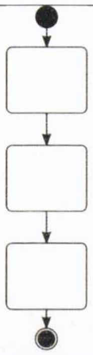
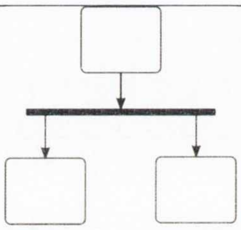
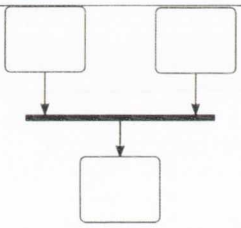
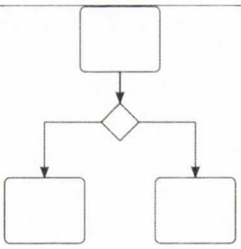
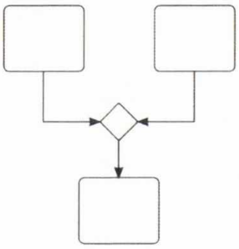
Pattern Description	**Pattern Representation**
The Sequence Pattern consists of a consecutive series of tasks with a task only being executed after the execution of the preceding task has been completed. As such it forms the basic structure of an activity composition.	
The Parallel Split Pattern allows a thread of execution to branch into two or more parallel threads. This is, it allows two or more tasks to be run concurrently. This pattern is also referred to as an 'AND split'.	
The Synchronisation Pattern allows two or more threads of execution to converge at a point into a single thread of execution. This pattern is essentially an 'OR join', therefore only one of the incoming threads needs to reach the synchronisation point in order for execution to proceed on the outgoing thread.	
The Exclusive Choice Pattern allows the execution flow to pass from one thread to a subsequent thread based on the outcome of the preceding task. As can be seen in the diagram, a single thread of execution reaches a decision point where a choice is made as to which outgoing path to take. This pattern is also known as an 'XOR split'.	
The Simple Merge Pattern, or 'XOR' join, is similar to the Synchronisation Pattern as it facilitates the merging of threads of execution. The difference is that the Simple Merge does not require synchronisation. When an incoming thread of execution reaches the merge point, the flow of control passes to the outgoing thread even if another incoming thread has previously reached that point. Essentially, this pattern allows parts that are common to multiple parallel threads of execution to be defined efficiently without the need for repetition.	

Table 4.1: Control Flow Constructs

4.4.2 Learner Model

The learner model provides the system with information about the learner so that the system can personalise the activity to the needs of the individual learner. When adaptively composing activities, there are many different aspects of the learner to which the system can adapt. The information needed is dependent on the specific requirements of the adaptive composition carried out by the system. This is one of the reasons why it is important for the system to be flexible in its support for model structures. Some examples of the types of information that the learner model can be required to contain include prior knowledge about concepts within a domain, information about the role of the learner or the educational goal of the learner.

Within the set of four basic models used to adaptively compose an activity, the Learner Model is the primary source of information upon which the narrative will operate.

4.4.3 Content Model

The Content Model describes all of the content that is known to the system and so available for selection in order to produce a personalised activity. The Content Model facilitates the selection process by providing the system with information about the content that can be used to identify appropriate content and to differentiate between similar pieces of content. The primary use of this model in the adaptation process is to facilitate the selection of appropriate content to satisfy the requirements of the narrative for a specific concept.

As with the Learner Model, the information that the Content Model is required to represent about the available content is influenced by the specific adaptive application. This is because some of the information is based on the actual features of the content. Some basic types of information are commonly used by this system to compose activities, This includes information about the 'physical' attributes of the content (file name, size, location, media type, etc). Information about the purpose of the content is commonly used as a means of selecting appropriate content based on the concepts covered by the content or the educational usage of the content, for example, whether the content is appropriate for a novice or an advanced learner in the subject domain. Irrespective of the information captured in the Content Model, it should be described using a vocabulary that it shares with the other metadata models.

Unlike the learner model, the content model is a composite model in which each individual piece of content has its own corresponding piece of metadata. As part of the adaptation process, the sum of these metadata models can be considered as one model since they share a common information model. As such, they are treated as a single information source by the adaptive system when searching the content model for information. This approach allows content to be added or removed from the system by adding or removing the appropriate metadata instance. Similarly, the metadata for an individual content resource can be updated without affecting the rest of the Content Model. The system will automatically be aware of the changes and make use of the updated content model for subsequent activity compositions.

4.4.4 Service Model

As is the case for content, the system is designed to operate in a 'closed world' with respect to services. This means that the system must know about the services that are available to it before they can be selected for inclusion in a PWE. The Service Model provides metadata descriptions of the available services, which provide information to the system that facilitates the system in identifying appropriate services based on the functionality that they provide. There are two aspects to the problem of describing the services to the system, first the system must be able to identify the core functionality of the service, for example, in the case of a 'chat' service that the service allows the user to communicate with others. In addition to this, the system must be able to differentiate between different services with the same core functionality. For example, both a chat service and a forum service provide the same core functionality since they are both communication services, but they have additional non-functional properties that differentiate them. A chat service can be considered as a synchronous communication service that is appropriate when all the users are online together where as a forum service is asynchronous, suitable for long term communication or where all of the users are not online at the same time.

Furthermore, the service model should also allow the system to execute the services at run time by providing the information necessary to invoke the service

4.4.5 PWE Model

This model represents a PWE that has been adapted to meet the needs of the user. PWE Models are generated by the AE as the result of the adaptation process. As such, it must be capable of describing the sequencing of the services and content that have been selected. Unlike the narrative model, which describes the PWE using abstract terms such as 'concepts' and 'tasks', the activity model references the actual content resources and services that will be delivered to the user.

As outlined in section 4.3 of this chapter, the generation of this model by the AE allows for the personalisation phase of the systems execution to be decoupled from the delivery of the PWE through the User Portal. The PWE Model allows an individual Learner's PWE to remain consistent across multiple sessions of interaction with the User Portal, only changing in response to the explicit actions of the Learner.

4.5 Use Case Scenario

This section presents a use case scenario in order to illustrate how the architecture presented in this chapter can be used to generate a PWE and how the metadata models described in the previous section are utilised in that process. This scenario is based on the generation of a 'peer review' PWE in which the learner is presented with an activity that requires them to write a report based on a subject for which appropriate content is provided. Upon completion, this report must then be submitted for review. The activity proceeds with a review task in which the learner reviews another learners report. This activity, illustrated in figure 4.2, is designed to be applied in two different educational domains, one of which requires the learners to discuss their reports following the review task.

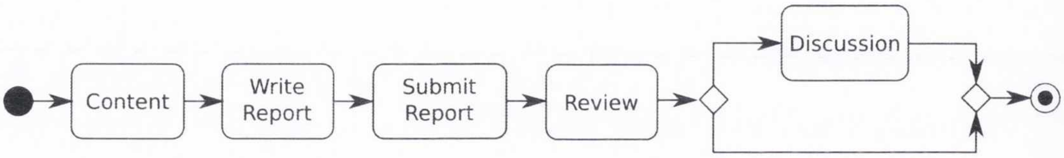


Figure 4.2: Sequencing of tasks in a Peer Review Activity

The structure of this activity, as illustrated in the diagram, is described in the Narrative Model along with the adaptation rules that describe how the concepts and tasks can be selected and sequenced for an individual learner within the constraints of the design of the activity. In this scenario, the adaptation rules are designed to allow the sequencing of the tasks that make up the peer review activity to be adapted while maintaining the essential structure of the activity.

The Narrative Model is executed by the Adaptive Engine in order to generate a PWE. As the narrative is executed, the Adaptive Engine uses information retrieved from the Learner Model to adapt the sequencing of the concepts in the PWE, for example, the concepts provided to the user in the Peer Review activity could be selected based on the learner's interests or prior knowledge. To instantiate these concepts, appropriate content needs to be selected, this can be done using various different adaptive strategies, for example based on the learner's prior knowledge in the concept.

As with the content sequencing, the task sequencing is achieved by reconciling the properties in the Learner Model with the rules defined by the narrative author. To instantiate the tasks with executable services that the learner can interact with, the Adaptive Engine makes a request to the Service Composer. This request provides the Service Composer with the requirements for the selection, what the selected service should be able to do, and information about the services that are available for selection, which the Adaptive Engine retrieves from the Service Model. In the case of the authoring service required to instantiate the first task in the peer review activity, the desired outcome could be to select a service that allows a report to be authored using a rich interface (high bandwidth). This set of requirements provides the system with a primary functional requirement, to produce a report, and a non-functional requirement that the system can use to differentiate between available services.

4.6 Supporting Adaptive Activities

4.6.1 Adaptive Behaviour

As mentioned previously in section 4.3, the Narrative Model is the embodiment of the strategy that underpins the activity. It allows the activity designer to have control over the sequencing of content and services ensuring that the activity remains educationally sound. Through the narrative, the activity designer also has influence over the adaptive behaviour of the system.

When authoring an adaptive activity, the designer can choose which parts of the activity can be adaptively selected as well as which metrics influence the selection process. Similarly, the designer can identify, as part of the narrative, the metrics that influence the adaptive selection of content and services.

Adaptive Sequencing

The narrative, as the embodiment of strategy, describes the activity in terms of concepts that the activity covers and the tasks that should be completed by the learner. It is with these constructs (concepts and tasks) that the activity designer defines the sequencing of the activity to meet the educational requirements.

Adaptive behaviour can be interwoven into the structure of the activity by wrapping specific constructs or groups of constructs with conditional rules. These rules allow specific parts of the activity to be adaptively turned on or off depending on the criteria defined by the narrative author. In order to satisfy the conditions specified in an adaptation rule, the engine makes use of the available metadata models in order to reconcile the rule conditions with the information available to the engine about the learner and the available content and services.

The purpose of an adaptation rule is to reconcile specific properties of the available metadata models with each other or alternatively with an expected value. If the condition is met then the related parts of the activity will be included.

For example, a narrative could consist of a sequence of concepts, each of which is associated with a rule that determines if the learner has the required prerequisite knowledge in order to be presented with the concept. In this case it would be necessary to compare the set of all of the concepts that the learner has covered with the known requirements for the concept in question. To do this, the engine would retrieve the appropriate values from the learner model and compare them against the known values.

Adaptive Selection

As discussed previously, an adaptive activity is defined in terms of the concepts that should be covered and the tasks that are to be made available. Both of these basic constructs are defined in the narrative using abstract terms based on a vocabulary that is shared with the Content and Service Models. This means that the content that covers a specific topic and the services that are used to carry out a task need to be selected in order to create an activity that can be delivered to a user. As with the adaptive sequencing of the activity, the narrative guides the reconciliation of models in order to select the most appropriate content and services.

4.6.2 Service Selection and Dynamic Composition

A simple approach to the adaptive selection of services would be to apply the same techniques as those used for the adaptive selection of content. Each service would be described using a metadata model that the AE, through the execution of the narrative, would use to identify appropriate services for inclusion in an activity. Candidacy could also be used as part of the selection process.

However, there would be several shortcomings with this approach. First of all, it would rely on an overly simplified view of a service. It does not take into account the possibility that a service could require parameterisation in order to be executed. Even if the simple approach could provide the service with the appropriate parameters, these parameters would need to be defined statically in the metadata and could not be changed by the system. Without the ability to modify the parameters passed to a service, the system would not be able to influence the behaviour of the selected service. The second limitation of this simple approach is that it assumes that an appropriate service will always exist to meet the selection requirements for a given task. If a task defines a set of requirements that are not satisfied by an existing service or if they can only be satisfied by more than one service then the composition of the activity could not be completed.

A better solution to the problem of adaptively selecting services would address both of these issues. It would be capable not only of dynamically selecting a service but also of selecting the appropriate parameters to provide to the service. In this way the behaviour of the service could be influenced, essentially providing the system with support for an additional adaptive behaviour, Adaptive Parameterisation. In order to allow the system to deal with scenarios where a service does not exist to meet the composition requirements, the system should be able to dynamically compose services from the existing services. The ability to dynamically compose new services to satisfy the requirements of a specific task with respect to an individual user also allows the narrative author a level of freedom when designing the narrative. If a task requires several services to be made available to the user but the order in which they are provided is not important then the narrative author can define the task as a 'black box' without the need to explicitly define the sequencing of the services within the task.

To support this more advanced approach to the adaptive selection of services, the selection of services is offloaded from the AE to a dedicated service selection component. The Service Composer provides a range of capabilities, in simple scenarios, it can select a single appropriate service from the available services. It can also be used to dynamically generate a simple composition of services to carry out a functionality that is required by the Learner to complete a task but which is not provided by any of the available services.

The Service Composer is initialised by the AE by providing it with information about the services that are available. This information is obtained from the Service Model, discussed in section 4.4. The Service Composer uses this information to build an internal model of the service domain. Once the Service Composer has been initialised it can be used by the AE as a Selector to instantiate tasks with an appropriate service or with a new service composition. The AE requests a selection from the planning component by invoking it with the rules/conditions that the narrative defines for the task. In this way, the AE can influence the selection process not only through the specification of the requirements for the selection/composition but also by controlling the set of services made available to the Service Composer. For example, in scenarios where there is a large number of services available for selection/composition, it could be beneficial to cluster the services based on functionality in order to reduce the complexity of the selection/composition process. This could be considered as a candidacy group similar to that used for content in the APeLS Adaptive Hypermedia System.

4.7 Interactive Service Control Flow

The delivery of personalised activities not only requires that the system is capable of sequencing tasks during the composition of an activity but that it must also be able to maintain the integrity of the task sequencing when delivering the activity to the user. In order to do so, the system must be able to identify the specific task that the user is taking part in at any given time so that only the appropriate services are provided to the learner, as required by the activity sequencing. In addition to this, the system must be able to support the control flow patterns, used in the narrative to sequence the tasks, during this delivery stage.

Section 3.2 discussed the state of the art in the composition of services using workflow techniques. An appropriate workflow engine and corresponding workflow language will be used to support the required control flow between interactive services.

4.8 Adaptive Engine Design

The Adaptive Engine (AE) is the main component of the system architecture as its role is to control the adaptive composition of activities. To support this process, the AE is required to provide three main functionalities; the execution of the narrative, to provide access to the necessary metadata models and the retrieval of metadata models from persistent storage.

As shown in figure 4.3, the AE architecture consists of three sub systems. Each of these sub systems corresponds to one of the functional requirements of the AE. The execution of the narrative is controlled by the EngineManager with access to metadata models provided by the ModelManager. The DataManager allows the system to retrieve models from data repositories.

In addition to the three main sub systems, the AE also provides an interface through which the system can be configured and controlled. This interface is used to simplify the use of the engine by abstracting some of the interactions between the main sub systems into operations that are commonly used to build an adaptive system on top of the AE. The AE also provides a collection of utilities that can be used during the execution of the narrative to provide non standard functionalities that are useful when building adaptive systems using the AE.

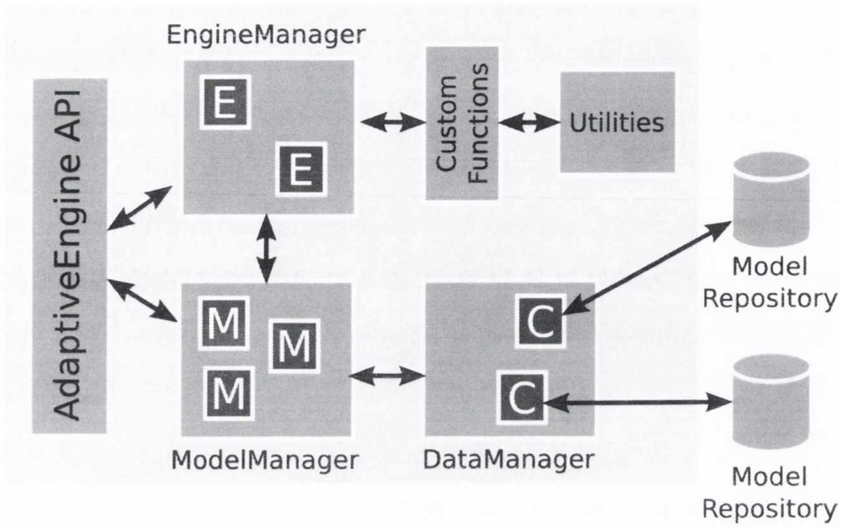


Figure 4.3: Logical Architecture diagram of the Adaptive Engine component

4.8.1 Engine Management

The EngineManager acts as a wrapper for the rule/scripting engine that executes the narrative. It provides a common interface through which other AE components can interact with the rule/scripting engine. As such, it supports tasks such as the initialisation of new scripting engine instances and the control of their execution, for example to start or stop the execution of an engine. In addition to this, the EngineManager provides extensions that are 'hooked' into the scripting/rule engine. These extensions add support to the scripting/rule engine language for functionalities that are specific to the execution of a narrative. Such functionalities include providing access to models by enabling the engine to search the models that are available to the AE. The extensions also allow the scripting/rule engine to directly manipulate the structure of a model.

The ability to search models is key to the adaptation process as it allows the adaptive behaviour described by the narrative to be informed by the metadata models, discussed previously in section 4.6. The ability to manipulate the structure of metadata models allows the engine to dynamically generate new models. This is used, for example, when dynamically generating a personalised course model.

4.8.2 Model Management

At any one time the execution of a narrative can require that AE to have access to many different models both for the purpose of acquiring information and to facilitate the generation of an output to the adaptation process. This means that the ModelManager, as the subsystem responsible for providing access to models, must be capable of accessing many models at the same time.

As discussed in section 4.4, the adaptive composition of an activity is informed by at least three different types of model, each of which will have a different structure. Therefore, the ModelManager must be capable of storing and providing access to models in an open manner that does not restrict the complexity of the models. In addition, the number of models used in the adaptation process is not limited to the three model types discussed previously. This means that the ModelManager must also be able to provide access to an arbitrary number of models.

The ModelManager serves two purposes. First it acts as a store for all of the models that the AE loads into memory, providing mechanisms to support the addition and removal of models from the system as well as supporting the creation of new models. The second function of the ModelManager is to act as an interface to the models that it stores. This allows other AE sub systems, such as the scripting/rule engine, to interact with the models stored in the ModelManager.

As mentioned previously, it is necessary for the ModelManager to be able to store an arbitrary number of models in order to provide flexibility to the designer of the adaptive system. The ModelManager does this by storing each model in memory and proving access to that model using a unique identifier. The unique identifier is assigned to a specific model by the author of the adaptive system built on the AE. This means that the identifier can be meaningful to the designer and simplifies the authoring of narratives as models can be accessed easily by referring to them by their unique identifier.

The ModelManager must also deal with different types of models that have different structures. It would be very inflexible for the ModelManager to restrict the models that it can handle to a specific structure or set of structures. Instead the ModelManager supports the use of models with arbitrary structures. As each model is stored individually, models with different structures can be handled by the ModelManager at the same time.

4.8.3 Data Management

The DataManager provides the Adaptive Engine with the ability to access models that are stored in some form of permanent storage. This allows the AE to load the models required to execute a narrative from their persistent storage location. The two common types of storage that the AE needs to have access to are the local file system and remote databases.

The DataManager can maintain multiple connections to different storage repositories at the same time. This is desirable since different types of models can be stored in different locations. For example the content metadata could be stored in a content repository while the narrative and learner models are stored in a separate repository.

The DataManager provides an interface through which the other AE sub systems can avail of the connections that it maintains. This allows the underlying storage systems to be abstracted so as to provide a common interface that supports simple actions such as loading a model from a repository and storing a model to a repository.

4.8.4 Additional Functionality

In addition to the core functionalities required for the Narrative to interact with the available models, it is also necessary to make additional functionalities available to the Narrative. For example, in order to the AE to function as a component in the architecture described in this chapter, it is necessary for the AE to be able to invoke other components in order to carry out its role. It is necessary to make such functionalities available to the Narrative as their use is dependent on information needs and functional requirements that arise during the execution of the narrative.

4.8.5 Adaptive Engine Execution

When executing a narrative, the different AE subsystems need to interact with each other. This process begins with a new instance of the AE being created and initialised. The initialisation of the AE requires that each of the three main subsystems are configured and primed so that the AE is ready to execute the narrative.

The first step in the initialisation is to set up the data repository connections that will be used to load the necessary metadata models. Once the connections have been created the models can be loaded from the respective repositories and stored in the ModelManager. It is during this step that the narrative is loaded into the AE. The next step is to create a new rule/scripting engine and to associate the narrative model with that engine so that when the scripting/rule engine is started it has a narrative to execute.

Having done this, the AE is ready to start the execution of the narrative. As the AE executes the narrative, the scripting/rule engine will need to access information contained in the metadata models. This is achieved through the calling of a 'custom function' that exposes the search functionality of the ModelManager to the scripting/rule engine. Through this mechanism, the scripting/rule engine has access to all of the available models.

During a typical narrative execution, a new model will be built in order to act as a persistent representation of the personalised activity that the engine generates. This model is built according to the rules in the narrative using the model manipulation custom functions that integrate the ModelManagers functionality with the scripting/rule engine. When narrative execution has completed and a new narrative model has been generated it is then stored in an appropriate data repository using the connections that the DataManager provides.

4.9 Delivery of User Oriented Services

In abstract terms a service can be considered to be something that provides value or meets a need. This definition encompasses a wide range of software on the internet. Commonly the term 'service' is used interchangably with the term 'Web Service', which in fact is a specific web technology that can be used to implement a service, while the term service is a more abstract concept.

As part of this research the term service is used more in the context of 'software as a service'. The term 'software as a service' or SaaS for short is used to mean an application that is made available to the user via the web. Such applications are commonly graphical in nature and intended to be used by end users through direct interaction. This differs from the more functional definition of a service that applies when talking about Web Services, which are not intended for users to interact with directly. The SaaS based interpretation of a service is more useful in the context of an eLearning application as the aim of activity based learning is to engage the learner in the learnign experience. In order to achieve this, the learner needs to be provided with tools that allow them to complete the required task. These tools could include, for example, email clients, rich text editors, instant messaging applications, etc.

A wide range of technologies can be used to implement such tools, for example web based technologies such as HTML and Javascript. However, although these technologies support service like features such as parameterisation they do not fit well with existing service orchestration technologies such as WSBPEL. For this system, the WSRP technology has been identified as an appropriate solution for the delivery of services. As a portlet based technology it meets the requirement for user centric, interactive services. It is also a Web Service based API and therefore compatible with existing WSBPEL workflow engines. As a WS based technology it has the benefit of allowing services to be hosted anywhere on the internet while still being available to be used as part of a personalised composition.

4.10 Unified User Interface

The delivery of a PWE to the user requires a user interface that is capable of providing access to both multimedia content and services. The environment provided by this interface must also allow the user to move between accessing content and interacting with services in a way that makes both content and services feel like parts of a single application. In addition to the core functionalities of content and service delivery, the user interface must also provide mechanisms that allow the user to affect and control the personalisation process.

In order to make the system easily accessible, the interface will be a web based application, allowing users to interact with the system using a standard web browser. This approach will minimise the prerequisite requirements that a user would have to meet in order to make use of the system and allow for greater use of the system in comparison to a standalone application.

The user interface 'portal', shown in figure 4.4, will be capable of retrieving an individual user's PWE model from a central repository and parsing it to generate a visual navigation structure for the PWE. The interface will also be capable of presenting content to the user that is stored both locally and on remote servers. The delivery of services to the user will be achieved through a portlet based approach and will require the portal to interact with a workflow engine that hosts the individual user's service compositions. This approach will allow services to be presented to the user using a window based paradigm within a single web page. This will allow the user to interact with both content and services as they require.

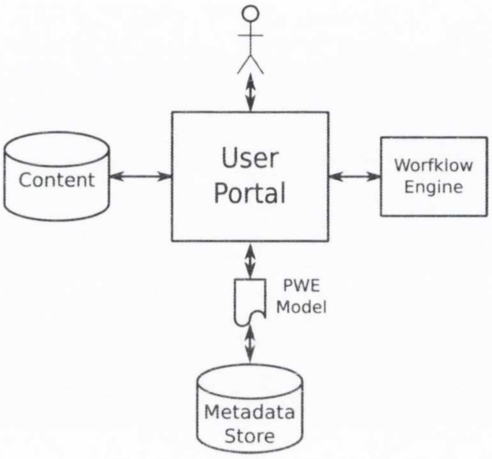


Figure 4.4: Architecture of the User Portal

Although services are visually presented as portlets, the underlying communication mechanism will be the WSRP specification. In order to deliver such services to the user, it is necessary for the portal to not only provide support for the portlet paradigm but also that it can handle the mapping between the user's interactions with the portlets and the appropriate WSRP web service calls.

4.11 Component Interaction

To better illustrate how PWEs can be delivered using the architecture presented in this chapter, the interactions between the components at run time are discussed in this section. To facilitate a clearer description of how the architecture can achieve the goal of delivering a PWE, the execution of the system has been broken down into two phases. The first of these phases is the personalisation phase, which is responsible for the generation of a PWE while the second phase, the experience phase, is concerned with the delivery of that PWE to the user. Figures 4.5 and 4.6 illustrate these two phases of execution respectively using UML component diagrams.

4.11.1 Personalisation Phase

In order to facilitate the discussion of how the various architecture components interact with each other during the personalisation phase, Figure 4.5 provides a component diagram that has been annotated so that the interactions between components are numbered.

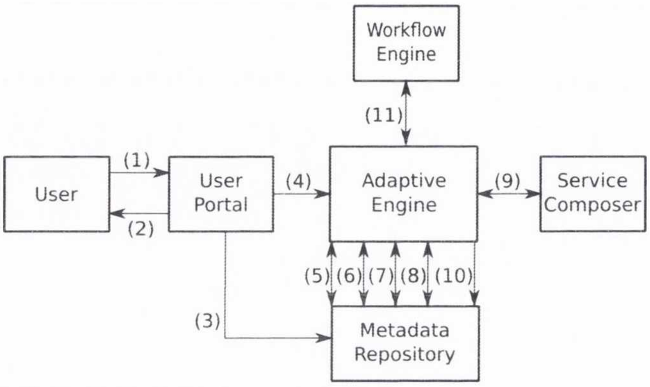


Figure 4.5: Component diagram illustrating the interaction of components during the personalisation phase

When a new user accesses the PWE Portal (1), they are presented with a form (2) that is used by the system to elicit a model of the user. The Portal then stores the newly acquired user model in a central repository for persistant storage (3). Once an appropriate model of the user is available to the system, it is then possible to generate a PWE for that user. The Portal invokes the Adaptive Engine in order to initiate this process (4). To carry out the personalisation process, the Adaptive Engine must first load the narrative model (5), which provides the adaptation rules that guide the generation of a PWE. The Adaptive Engine must also retrieve the user's learner model from a central repository (6) so that it can be used to influence the personalisation process.

When the necessary models (narrative and learner) are loaded into the Adaptive Engine, the narrative can be executed. As the Adaptive Engine creates the personalised sequencing of concepts and tasks in the PWE, it uses information from the learner model in order to satisfy the conditions associated with the sequencing rules from the narrative. The selection of appropriate content that instantiates the concepts in the PWE requires that the Adaptive Engine can not only reconcile the narrative rules with the learner model but also with the content model, which provides metadata about the content that is available to the system. This requires the Adaptive Engine to query the content model stored in the central metadata repository (7). Similarly, the selection of appropriate services to instantiate the tasks in the PWE requires that the Adaptive Engine retrieves information about the available services from the metadata repository (8).

Unlike the content selection process, which is carried out by the Adaptive Engine through the execution of the narrative, the selection of appropriate services is carried out by the service composer component. When an appropriate service is required by the Adaptive Engine to instantiate a task, it makes a request to the Service Composer (9). As part of this request, the Adaptive Engine provides the service composer with information about the available services as well as appropriate information about the user, which can be used to influence the selection process. As discussed previously in section 4.6.2, this allows the Adaptive Engine to influence the selection and composition carried out by the Service Composer not only by setting the requirements for the composition but also by controlling the set of services that the Service Composer is aware of. In this way the Adaptive Engine can implement a form of candidacy.

Once the Adaptive Engine has completed the sequencing of the PWE and selected appropriate content and services to instantiate the concepts and tasks that make up the PWE, the PWE is stored in the repository (10) for later retrieval and execution. In addition to storing the PWE, the Adaptive Engine also deploys the service composition aspect of the PWE to a workflow engine (11) in order to make the composition executable during the experience phase of execution.

4.11.2 Experience Phase

When an appropriate PWE has been composed, the Learner is then able to engage in that experience by logging into the Portal, which provides an integrated environment in which the user can interact with the personalised content and services that instantiate their PWE.

As for the Personalisation Phase, an annotated component diagram of the Experience Phase is provided in figure 4.6. When a user logs into the Portal (1), the Portal must first retrieve the model of the user's instantiated PWE from the repository (2) before it can provide the user with their PWE. This provides the Portal with details of the PWEs structure and references to the content that should be presented to the user. The delivery of content to the user requires that the Portal retrieve the necessary content resources from the server(s)[1] that host it (3). When the user interacts with a service, the Portal presents them with a graphical, web based interface. To fill this interface with the appropriate service, the Portal requests the service from the workflow engine that hosts the service component of the PWE as discussed previously (4). The workflow engine in turn retrieves the appropriate service from a service provider based on the user's current progress through the PWE (5). The retrieved service is then returned to the User Portal in response to the orignal request, which makes the service available to the user so that they can interact with it as part of the PWE.

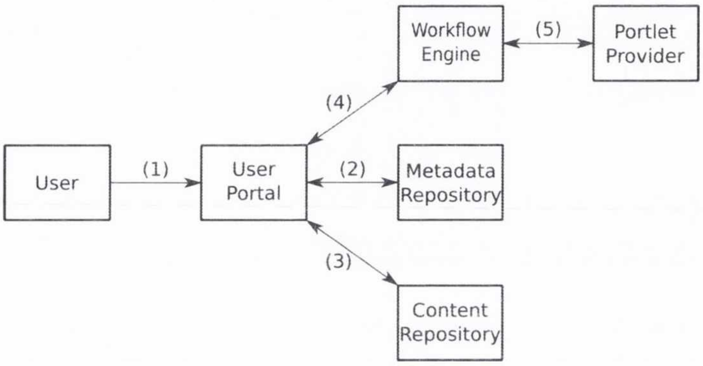


Figure 4.6: Component diagram illustrating the interaction of components during the experience phase

4.12 Summary

In this chapter, an approach capable of combining the adaptive selection and sequencing of multimedia content with the adaptive selection and sequencing of services in a unified manner in order to deliver personalised web based experiences has been outlined. This approach is influenced by the state of the art as discussed in chapters 2 and 3, which together have led to the definition of a set of functional and technical requirements.

Based on the defined requirements, the design of a system capable of delivering PWEs was outlined and the individual components of the design described in detail and their role in the delivery of PWEs was discussed. Finally, an outline of how the various components in the design would interoperate with each other at run time was presented.

[1] Content for one PWE can be retrieved from many different locations but for the sake of simplicity only one content server is considered in this discussion.

Chapter 5

Implementation

5.1 Introduction

In the previous chapter, the design of a system capable of delivering web based experiences that combine adaptively selected and sequenced multimedia content with adaptive composed services was outlined. This design was based on a set of technical and educational requirements. This chapter describes how this design were realised as a technical implementation.

This chapter is structured as follows, an overview of the technical realisation of the architecture is presented followed by detailed descriptions of how each of the components was implemented. Details of the data models used by the implemented system are also provided. An example, based on a real world scenario, of how the system operates in the delivery of a PWE is also presented in order to provide a clearer understanding of how the system operates

5.2 System Implementation

5.2.1 Overview

Figure 5.1 provides a component based view of the architecture for the PWE system as implemented. This differs from the architecture overview presented in section 4.3 of the previous chapter in that additional components, required to realise the design, are incorporated in this view of the architecture.

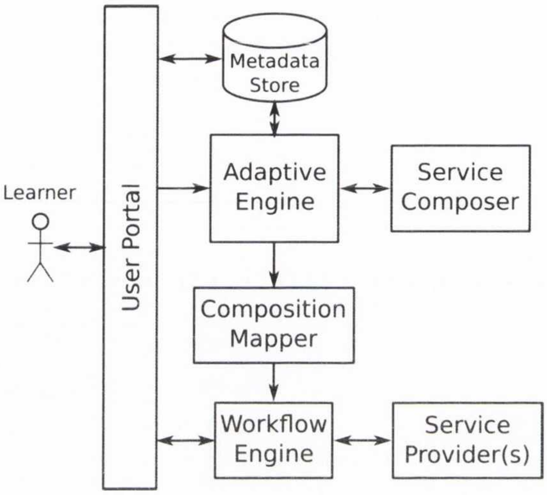


Figure 5.1: Detailed view of the system architecture

To satisfy the requirements outlined in the previous chapter for user centric, graphical services that support interactive tasks, a portlet based approach has been adopted as the basis for the implementation of the services composed by the system. The use of portlets alone is not sufficient for the implementation of services in this architecture. It should be possible for the adaptively generated PWE to contain services that are deployed remotely. To achieve this, the WSRP specification, which provides support for the delivery of portlets using a WS based protocol, has been incorporated into the system architecture.

The component through which the user will interact with the system is the User Portal, this is a web based interface that allows the user to engage in their PWE. The User Portal also carries out additional tasks within the system. It is responsible for eliciting models of the user and for initiating the adaptation process. In order to realise the User Portal, several existing technologies are leveraged to provide the necessary portlet functionality and WS support. The User Portal is built on top of the Apache Pluto portal engine [plu 11], which allows the User Portal to deliver portlets to the user. The functionality of the Apache Pluto system is extended to support for the WSRP specification by the addition of the WSRP4J Portlet Consumer.

When initiating the adaptation process, the User Portal interacts with the Adaptive Engine, a custom developed Java component that facilitates the adaptation process by providing an execution environment for the Narrative Model. The Adaptive Engine provides the Narrative Model with access to the necessary metadata models and other additional functionalities necessary to carry out the adaptation process. To support the data storage requirements of the various components in the architecture, a XML database is made available. This serves as a metadata repository for the different metadata models used in the adaptation process, e.g. the Learner Model, Content Model and Service Model.

During the adaptation process, the Adaptive Engine takes advantage of a special purpose component for the selection of appropriate services to satisfy the requirements of the user in accordance with the narrative. This component leverages AI Planning techniques to adaptively select an appropriate service or services. The services selected by the AI Planner are then incorporated into the PWE by the Adaptive Engine.

To make PWEs available to the user at run time, it is necessary to make the service composition component of the PWE available as an executable workflow. This is due to the complex nature of the control flow between composed services relative to that present in content compositions. To achieve this, the service composition component of the PWE is instantiated as a WSBPEL workflow and deployed to the ActiveBPEL workflow engine. The mapping of the XML description of the composition, generated by the Adaptive Engine during the adaptation process to a run time executable WSBPEL workflow is carried out by the Compositon2BPEL component. This component is responsible for generating a valid WSBPEL process that instantiates the compositions of services that were selected and sequenced for the user. Furthermore, the Composition2BPEL component also handles the deployment of the WSBPEL workflow to the ActiveBPEL workflow engine. An additional task carried out by the Composition2BPEL component is to register the newly deployed workflow with Pluto so that the workflow will be accessible to the user through the User Portal.

5.2.2 Adaptive Engine

As discussed in section 4.8 of the design chapter, the role of the Adaptive Engine (AE) is to execute the rules that guide the adaptation process in order to produce a PWE. To do so, the engine must not only provide an execution environment for an appropriate rule language but must also provide access from the environment to the metadata models that are used by the AE to inform the adaptation process.

Figure 5.2 provides a component based view of the Adaptive Engine architecture. As

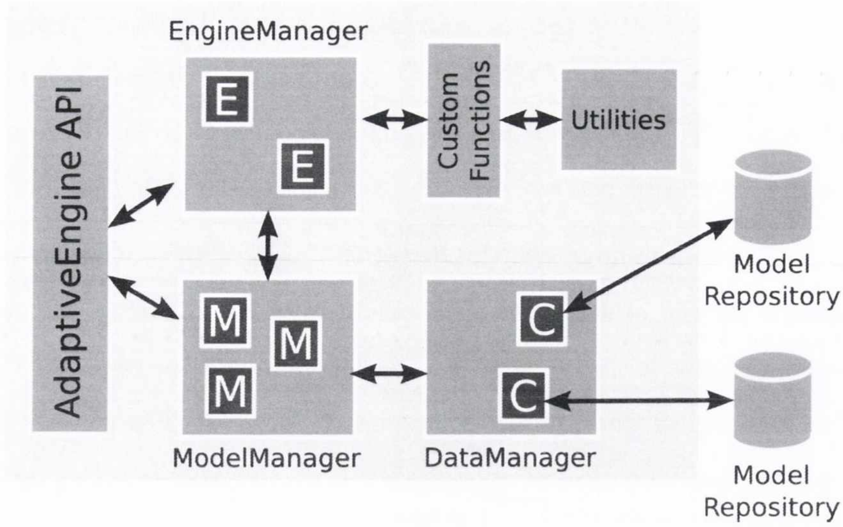


Figure 5.2: Component view of the Adaptive Engine (AE) Architecture

can be seen from the diagram, the AE consists of three primary subsystems, the EngineManager, ModelManager and DataManager. In addition to these components, the Adaptive Engine contains a set of utilities that provide functionalities required by the narrative.

5.2.2.1 AdaptiveEngine

The AdaptiveEngine component provides client applications that interact with the Adaptive Engine with a single point through which the functionality of the system can be accessed. It provides an API consisting of a set of functions that support the high level tasks that are necessary to configure and run the Adaptive Engine, for example creating data connections, loading models, initiating the execution of a narrative, etc. The AdaptiveEngine component is also responsible for the initialisation of the other components, such as the EngineManger, ModelManager and DataManager.

5.2.2.2 EngineManager

The AE must be able to execute a Narrative Model consisting of a set of sequencing rules that allow a PWE to be dynamically composed based on the information available to the AE from the metadata models that it has access to. These sequencing rules could be in the form of a rule base or alternatively a script consisting of if-else statements. The choice between these two approaches depends on the manner in which the adaptive behaviour of the system will be exposed to the user.

The generation of a PWE is carried out in an a priori manner that results in a complete PWE being composed in advance of the user's interaction with the personalised composition. This is in contrast to a just in time approach in which the next step in the composition is selected just as it is required by the user. The approach taken means that the AE can take into account all aspects of the activity during the personalisation process. For example, it removes the need to make assumptions about the availability of services to complete later tasks in the activity as would be the case using a just in time approach. Such assumptions could prove incorrect and could result in a learning being offered an activity that the system cannot fully realise.

The a priori nature of the adaptation process used to generate a PWE means that a scripting language is an appropriate means of defining a narrative model in this case. A rule-based narrative would be more suited when the adaptation is carried out at run time using a just in time selection mechanism where the state of the rule base changes between interactions with the system. The adaptive behaviour captured by the narrative model requires that the narrative is capable of handling various different types of data as well as supporting the functionalities described in section 5.3.4.

A further requirement is that the language used by the AE can be extended in order to add functionality that is specific to the use of the language to define narratives. Functionality is required that will enable the scripting engine to access the information that the AE has available to it in the form of metadata models. The narrative also needs to be able to manipulate the contents of the metadata models and to create new models. In addition to these core functionalities, it is necessary to be able to add additional functionalities to the narrative language that allow the AE to interact with external systems.

Based on the requirements and functionalities discussed, the Jatha [Hewett 07] scripting engine was selected as the basis for the execution environment for the Narrative Model. Jatha is a pure Java implementation of a Lisp interpreter, allowing the execution of Lisp programs. The use of Lisp to define narratives is a relatively low level approach, however it has the benefit of allowing a wide range of adaptive behaviours to be defined as well as providing features that are useful in narratives, for example functions, loosely typed variables useful APIs for manipulating strings, etc. The architecture of the Jatha interpreter also provides a straightforward mechanism for adding new functions, which can be accessed from the Lisp programs that it executes. New 'custom functions' can be added by writing a Java class that implements the appropriate Jatha interface and provides the necessary functionality. This class is then registered with the Jatha engine. Using this approach, a set of custom functions were developed and added to the Lisp interpreter to make it into a suitable narrative execution environment. These custom functions provide direct access from the narrative to the models that are loaded into the AE as well as providing additional functionalities, such as the ability to invoke Web Services.

5.2.2.3 Modelling

As discussed in section 4.8.2 of the Design chapter, the AE must satisfy two main requirements with respect to the handling of metadata models. It must be able to support the use of semantically rich metadata while allowing flexibility in the structure of that metadata. The AE must also be capable of interacting with multiple metadata models at the same time.

Section 4.4 described how XML will be used to capture and store the metadata necessary for the delivery of PWEs. This will allow the metadata schemas used in the various different metadata models to be flexible. In order for the AE to be able to access these models at run time, they are read from the persistent metadata repository where they are stored and parsed so that they can be stored in memory by the AE, ready to be accessed by the other AE subsystems.

This approach is taken so that it is possible for the AE to create/modify the XML models irrespective of the underlying data storage mechanism. XML query technologies do not yet have good support for modifying documents. XPath is only intended to support querying while the W3C are still working on an extension to XQuery that would support updates[1]. Similar functionality is provided by a limited number of tools but no common implementation is available. In scenarios where it is not desirable to load all of the necessary metadata, for example to search across the entire Content Model, it is still possible to run queries remotely using the narrative custom functions providing that the metadata store supports this functionality.

The XML data that is retrieved from the persistent storage is parsed using the native Java

[1] The initial working draft of this proposal was published in 2006 [?] but only became a W3C Recommendation in March 2011 [W3C 11]

XML parser and the output used to create a JDOM [JDOM Project 07] object. JDOM is a Java library that allows XML data to be represented using a Document Object Model (DOM) data structure. This allows the metadata models to be created, updated, modified and queried in memory. JDOM is a widely used library that can handle very large XML documents (several megabytes) that are far bigger than the documents that the AE is required to handle when accessing metadata models. In order to abstract away some of the complexity of the JDOM API and to provide a more convenient programming interface, the JDOM object is wrapped in a custom Model object.

The Model object provides an API that allows the system to create, update and search metadata models without the need to deal with the underlying implementation (JDOM). The Model object also provides an important functionality that is not present in JDOM, it provides a pointer to the current node in the DOM tree so that task such as retrieving data from the model or updating the structure of the model are simplified. This functionality allows the narrative to refer to models without the need to keep track of where in the model they were accessing previously. This is important not only because it reduces the complexity of the narrative syntax but also because a narrative could be accessing many different models at the same time. Tasks such as the handling of XML namespaces are also simplified by the Model object. The ability to create, access and query XML that contains multiple namespaces is important as without it, the AE would be restricted to only operating with custom model schemas that only used XML in a very simple way. It would not be possible for the AE to access information stored in models that confirm to standards such as IMS LIP or ADL SCORM.

As mentioned previously, the execution of a narrative can require access to many different models at the same time, therefore it is necessary for the AE to support this requirement. This is handled by the ModelManager component, this component uses a HashMap to store as many Model objects as are required by the AE. Each model is stored using a unique identifier that is used by the system to access that model.

5.2.2.4 DataManager

The DataManager is responsible for managing access to data sources through which metadata models are accessed. Its main tasks are the creation and management of data connections and the writing and reading of data to and from open data connections.

Other AE components can request that a new data connection is opened by making the appropriate function call and passing the URI for the connection as a parameter. The URI is used by the DataManager to identify the correct type of data connection to create. The current implementation supports the creation of connections to file system based data sources and network data sources that support XMLDB [The XML:DB Initiative 03] based communication. The URIs used to identify these two types of data source begin with the prefixes **file://** and **xmldb://** respectively. Once a data connection has been created it is stored in a HashMap using a human readable unique identifier, which is provided as a parameter when creating the connection. This allows multiple connections to be managed by the DataManager while making it easy for the developer to manage the connections since it is possible to give them meaningful names.

The connections created by the DataManager provide a simple set of functionalities, allowing the reading and writing of files/resources from an open connection.

To support the creation of XMLDB connections to a network based data source, the DataManager uses the XMLDB API along with the implementation of that API which is provided by the underlying implementation.

5.2.2.5 Custom Functions

As discussed, the Jatha Lisp engine was extended to provide additional functionality that was not provided by the standard Lisp functions supported by the engine. These custom functions turn the Lisp language supported by Jatha into a narrative language by integrating the scripting capabilities of the Jatha engine with the modeling and data components of the Adaptive Engine, allowing narratives to the key task of model reconciliation in order to adaptively instantiate the PWE.

The custom functions added to the Jatha engine fall into three categories, modeling, search and utility. The modeling functions allow the narrative to access information stored in the models that have been loaded into the Adaptive Engine as well as to manipulate the structure of those models. The search functions provide advanced search functionality to the narrative, allowing models to be searched using XPath or XQuery. In the case of XPath and XQuery the actual functionality is provided by existing libraries. In addition, the utility functions expose additional functionalities to the narrative language that are useful when building adaptive systems but that are not fundamental behaviours of the Adaptive Engine.

Modeling Functions

- **load-model** Load a model from an existing, open data connection
- **store-model** Write a model to an existing, open data connection
- **create-model** Create a new empty model
- **remove-model** Remove a currently loaded model from the AE
- **update-model** Add a new element/node to an open model
- **search-model** Carry out a simple node based search of an open model
- **get-text** Return the text data stored in the current element of a model
- **cd** Navigate the specified model using
- **get-attribute** Return the value of the specified attribute in the current element
- **add-attribute** Add an attribute to the current element of a model
- **model-to-string** Return the entire model as an XML string

Search Functions

- **xpath-query-model** Carry out a XPath query on a loaded model
- **xpath-query-collection** Carry out a XPath query on a set of models
- **remote-xquery-model** Carry out a XQuery on a model using the storage mediums underlying query support
- **remote-xquery-collection** Carry out a XQuery query on a collection of models using the storage mediums underlying query mechanisms

Utility Functions

- **transform-model** Carry out an XSLT transform
- **call-webservice** Make a SOAP based web service call

5.2.3 Data Storage

As discussed previously, the metadata models that drive the process of generating a PWE use XML to encode the information that they capture. In order to store these models persistently and to allow them to be accessed by the various components of the framework, it is necessary to store them in a central location that provides a suitable interface through which the models can be accessed. Storing the XML models on the file system is not a good solution due to the distributed nature of the system architecture. Furthermore, such an approach would bring with it performance issues due to the I/O characteristics of disk storage as well as the overhead of parsing the models whenever they are queried.

A possible solution would be to store the XML models in an SQL database. Many modern relational databases provide mechanisms through which XML can be stored and queried using their relational tables. Both MySQL [Oracle 10b] and Microsoft SQL Server 2008 [Microsoft 08] support the importing and exporting of relational tables in XML format. The limitation of this functionality is that the XML Schemas supported are very simple as they basically capture the relational table structure in XML format. Both databases also provide a more flexible mechanism of storing XML by including it in a relational table column.

The database based approach would facilitate the distributed access to the models as well as providing improved performance in comparison to file I/O. In addition, relational databases that provide support for XML storage commonly support mechanisms through which the XML models can be queried. In many cases, this support comes in the form of XQuery [W3C 07], a language for the querying of XML that provides functionality similar to that of SQL.

An alternative solution is the use of a native XML database. Such databases store the XML using data structures, such as B+ trees, that suit the inherent structure of XML documents. This means that the XML does not need to be mapped to an underlying data structure as is the case with relational databases. Native XML databases, such as eXist [eXist 10] and Apache Xindice [Apache Foundation 07], also provide search and query mechanisms such as XQuery.

For the implementation of this system, the eXist native XML database was selected. It is an actively developed open source database that is implemented in Java. It can also be deployed as a web application on a Java Servlet engine such as Apache Tomcat.

5.2.3.1 Database Access

Integration between the eXist database and the Java based components such as the AE is achieved through the use of the XMLDB API, which eXist implements. This API can be considered equivalent to the functionality that the Java Database Connectivity (JDBC) API provides when accessing a SQL based relational database from a Java application. Through the use of XMLDB API calls it is possible to connect to a remote eXist database using an XML/RPC based protocol and execute queries on the data stored in the database using XQuery or the simpler XPath [W3C 99a] query language. The API also supports functionalities such as the creation of collections as well as the retrieval of whole documents from the database.

5.2.4 Service Composer

As outlined in section 4.6.2 of the design chapter, the Service Composer component is responsible for selecting appropriate services for use in a PWE. It is also capable of dynamically composing services to meet a need when a suitable service in not available. As such it behaves as a candidate selector that is utilised by the AE during the execution of the narrative.

To make this service selection functionality available to the AE at run time, the Service Composer is deployed as a web service. The Service Composer WS interface is defined using WSDL and is designed to accept SOAP messages, allowing the caller of the service, in this case the Adaptive Engine, to provide the detailed information that the planner requires to carry out the selection and dynamic composition of services.

The AI Planner Service is based on the Java implementation of the Axis2 Web Service Framework [Apache Foundation 09]. As such, it runs as a service that is deployed on top of the Axis2 web application, which runs on a servlet engine [Sun Microsystems 03], in this case Apache Tomcat [Apache Foundation 10c].

Figure 5.3 provides an overview of the architecture for the Service Composer. The planning functionality of the service is provided by the LPG-td [Gerevini 04] planner, which uses AI planning techniques local search and planning graphs to solve planning problems. This is

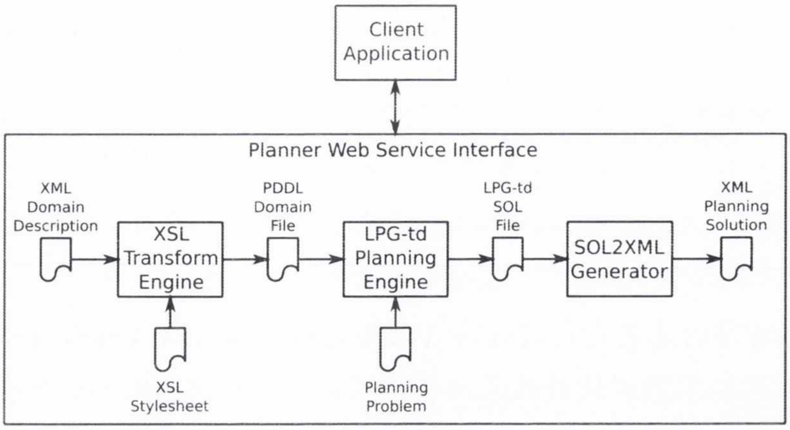


Figure 5.3: AI Planner Service Architecture

a freely available third party software component, written in C. Other Java based planners are available that would be easy to integrate with the existing Java infrastructure, however, the LPG-td planner was selected for its support for the PDDL specification. Existing Java based planners, such as JShop2, provide their own language for defining planning problems or offer only limited support for PDDL. To integrate the LPG-td x86 native binary application with the Java based components using the JVM's built in support for executing native processes on the host operating system is used.

To invoke the Service Composer, the Adaptive Engine provides information about all of the services that are available for selection and composition, this is known as the 'planning domain'. The planning domain is passed to the planner as an XML document containing a set of service models. This approach is taken in order to allow the AE to adaptively control the set of services from which the planner can choose. The structure of this model is shown in figure 5.4 in which single line boxes represent elements and double lined boxes represent attributes. This model is essentially an aggregation of the metadata models that describe all of the services available to the AE. A more detailed discussion of this model is provided in section implementation:servicemodel.

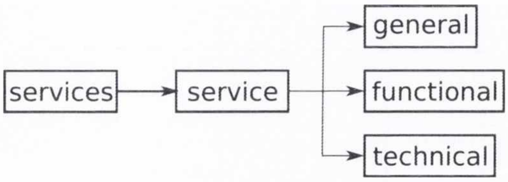


Figure 5.4: Element structure for XML model defining set of services available to Service Composer

In addition to the planning domain, the composition problem that the Service Composer should attempt to solve needs to be provided. This 'planning problem' is passed to the Service Composer as a XML document, the structure of which can be seen in figure 5.5.

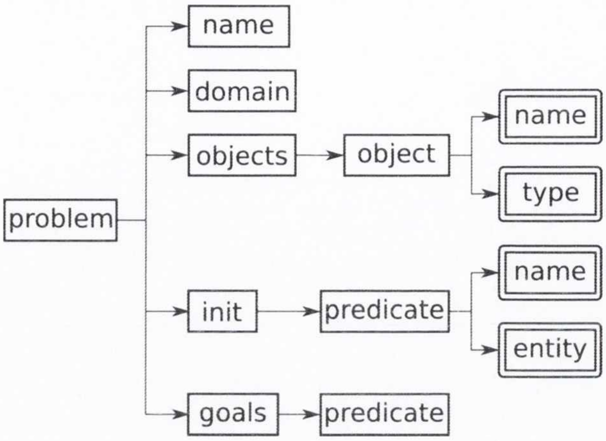


Figure 5.5: Element structure for XML model defining service composition problem

As mentioned previously, the inputs to the Service Composer web service are in the form of SOAP messages that contain XML documents. In order to make use of the information that is provided, it is necessary to first transform the XML documents into PDDL documents that can be processed by the LPG-td AI planner. Secondly, as LPG-td is designed to run as a stand alone component, it expects both of these inputs to be in the form of text files on the local file system, which are specified as parameters using LPG-td's command line interface. Therefore, it is necessary to write the files to the file system so that they are accessible.

The first of these steps is carried out by passing the planning problem XML document through an eXtensible Stylesheet Language Transformations (XSLT) [W3C 99b] transform engine along with a eXtensible Stylesheet Language (XSL) style sheet that describes how to extract the information from the XML document and transform it into appropriate PDDL syntax. An example of equivalent service definitions in XML and PDDL can be seen below. Figure 5.6 provides the definition of a service in XML while figure 5.7 shows the definition of the same service described using PDDL.

The newly generated PDDL domain definition and planning problem are temporarily stored on the file system, using unique identifiers as file names, so that the LPG-td planner can access them. The paths to the files are then passed as parameters to the planner when a new instance of the planner is started. When invoked, the LPG-td planner attempts to solve the planning problem and returns a solution to the problem as a textual

```xml
<service>
    <general>
        <name>EMAIL</name>
        <description>An email client portlet</description>
    </general>
    <functional>
        <parameters>
            <parameter>
                <name>id</name>
                <type>user</type>
            </parameter>
            <parameter>
                <name>passwd</name>
                <type>password</type>
            </parameter>
        </parameters>
        <preconditions>
            <predicate>
                <name>is-registered</name>
                <variables>
                    <variable>
                        <name>id</name>
                        <type>user</type>
                    </variable>
                </variables>
            </predicate>
        </preconditions>
        <effects>
            <predicate>
                <name>sent</name>
                <variables>
                    <variable>
                        <name>email</name>
                        <type>message</type>
                    </variable>
                </variables>
            </predicate>
        </effects>
    </functional>
    <technical>
        <type>wsrp</type>
        <portletIdentifier>
            emailClient
        </portletIdentifier>
        <getMarkupURI>
            http://server.cs.tcd.ie:8080/WSRPBaseService
        </getMarkupURI>
        <exitCondition>Submit=Done</exitCondition>
    </technical>
</service>
```

Figure 5.6: XML description of a Service

```
(:action EMAIL
    :parameters ( ?id - user
          ?passwd - password
          ?email - message
          ?id - user)

    :precondition (and (is-registered ?id))

    :effect (and (sent ?email))
)
```

Figure 5.7: PDDL description of a Service

report. Under LPG-td's normal execution environment, this report would be output to the command line but in this case the output is captured by the JVM. The captured output, a snippet of which is illustrated in figure 5.8, is then parsed using a series of regular expressions to extract the planning solution so that it can be encoded in XML. Figure 5.9 provides an example of a XML formatted solution to a planning problem, which is then returned by the Service Composer to the Adaptive Engine in a SOAP message.

```
Plan computed:
   Time: (ACTION) [action Duration; action Cost]
 0.0000: (AUTHENTICATE SUBMITTER PRIVATE-KEY) [D:1.0000; C:1.0000]
 1.0000: (WEBDAV-PUT DAV-FILE-SERVER SUBMITTER THESIS) [D:1.0000; C:1.0000]
 2.0000: (SEND-INSTANT-MESSAGE SUBMITTER SUBMITTER CONFIRMATION-MESSAGE) [D:1.0000; C:1.0000]
```

Figure 5.8: Snippet of the report generated by LPG-td showing the solution to a planning problem

```
<solution>
    <step time=”0.000”>
        <action>
            <name>AUTHENTICATE</name>
            <parameters>
                <parameter>SUBMITTER</parameter>
                <parameter>PRIVATE–KEY</parameter>
            </parameters>
        </action>
    </step>
    <step time=”1.000”>
        <action>
            <name>WEBDAV–PUT</name>
            <parameters>
                <parameter>DAV–FILE–SERVER</parameter>
                <parameter>SUBMITTER</parameter>
                <parameter>THESIS</parameter>
            </parameters>
        </action>
    </step>
    <step time=”2.000”>
        <action>
            <name>SEND–INSTANT–MESSAGE</name>
            <parameters>
                <parameter>SUBMITTER</parameter>
                <parameter>SUBMITTER</parameter>
                <parameter>CONFIRMATION–MESSAGE</parameter>
            </parameters>
        </action>
    </step>
</solution>
```

Figure 5.9: Example XML encoded solution to a planning problem

5.2.5 Composition Mapper

The role of the Composition Mapper is to convert the personalised service compositions generated by the AE into a format that can be deployed and then executed at run time by the user. The Service Composition Model, described in detail in section 5.3.6, which is generated by the AE is described using XML and supports the set of workflow patterns previously identified. This model contains all of the information necessary to invoke the selected services in the order intended but it is not executable. A detailed discussion of the structure of this model is provided in section 5.3.5 of this chapter.

As described previously, service compositions generated by the AE will be made available at run time as workflows defined using WSBPEL, therefore the Composition Mapper must take the generic Service Composition Model provided by the AE and transform it into an equivalent WSBPEL process. In addition, the Composition Mapper also needs to take into account the underlying technology that is used to deliver the services that are used in a PWE. The use of the WSRP protocol, see section 3.4.2, means that the generated workflow must not only orchestrate the services of the composition correctly but must also handle the protocol that they are based on. The details of how the composition of services based on the WSRP specification is achieved is described in section 5.4.

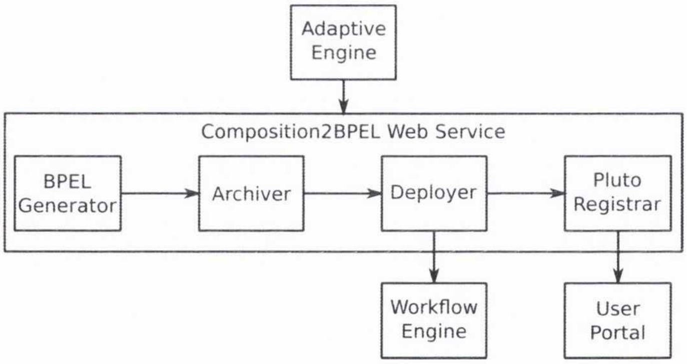


Figure 5.10: High level diagram of the Composition2BPEL Service

Figure 5.10 provides a high level view of the components that make up the Composition Mapper. As was the case with the Service Composer, the Composition Mapper is implemented as a web service using the Axis2 web service framework. The input to the Composition Mapper is an XML representation of a service composition that supports the workflow patterns that were previously identified in section 4.4.1.1. As shown in the diagram, the architecture for the Composition Mapper consists of a wrapper component that provides the interface and encapsulates the application logic that combines a set of functional components. To generate an executable process from the composition model, the Composition Mapper service iterates through the model making calls to the BPELGenerator component whenever an element representing a service or control flow pattern is found. The BPELGenerator provides an API that allows for the generation of a WSBPEL process through the use of API calls that correspond to the high level needs of the Composition2BPEL Service and not the specific details of WSBPEL.

When a complete WSBPEL process has been generated it is necessary to package the process definition as well as other related documents so that they conform to a specific package structure, which is compatible with the WSBPEL workflow engine that will execute the process. This is carried out by the Archiver Component. Once a suitable package containing the WSBPEL process has been generated, it can then be uploaded to the workflow engine using the Deployer Component. An addition task carried out by the Deployer Component is to register the new WSBPEL process with the User Portal so that it is accessible to the user.

5.2.5.1 BPELGenerator

The BPELGenerator component provides an API that supports the construction of a WSBPEL process without the need to deal directly with the complexities of WSBPEL. The functionality of the component is tailored to the specific task of generating WSBPEL processes that orchestrate WSRP based portlets. As such, it manages tasks such as the creation of the variables and the mapping of inputs and outputs of services without requiring the user to consider these issues. The API is designed to deal with services at the same conceptual level as in the composition model generated by the Adaptive Engine. That is, its methods allow a WSRP portlet to be added to the WSBPEL process in a single step, abstracting away how the WSBPEL process will handle the underlying WSRP communication protocol. Similarly, the addition of control flow constructs such as Sequences and Parallel Splits only requires a single method call.

The API provides support for the following main methods

- addService

- addSequence
- addParallelSplit
- addConditionalBranch
- stepOut
- BPEL2String

Not only does the BPELGenerator manage the generation of the WSBPEL process that orchestrates the services described in the composition model, it also auto generates a set of related models that are necessary for the successful deployment and execution of the workflow. The first of these is a second WSBPEL process to manage the ServiceDescription service that is required by the WSRP specification and which is used during the process of registering a portlet with the portal. The BPELGenerator also auto generates a set of documents that are specific to the ActiveBPEL workflow engine. The first of these documents is the Process Definition Document (PDD). This describes the 'partner links' that the WSBPEL process utilises as well as the WSDL documents documents that are used by the WSBPEL process. The second document that the BPELGenerator automatically generates is the Catalog, an XML document that lists all of the additional files that are used as part of the WSBPEL process definition. These include the WSDL documents that define the interface for the WSBPEL process and the external services orchestrated by the process.

The BPELGenerator uses DOM objects to represent all of the XML models that it manages during execution, the WSBPEL process models, PDD files and Catalog file. The JDOM library was used as the implementation for the DOM model. The use of this library allowed the code from the Adaptive Engine's Model implementation, see section 5.2.2.3, to be reused. This provided an API that supported the creation of the elements and attributes necessary to generate standard compliant WSBPEL. The code developed for the AE also provided good support for XML namespaces and so simplified the complexity of handling the many namespaces that are present in a WSBPEL document.

As the API provided by the BPELGenerator results in large portions of WSBPEL being added to the DOM structure in one step, for example to support the handling of a specific WSRP service instance, it is necessary for the BPELGenerator to be able to quickly move between key positions in the DOM structure. This is achieved by maintaining a stack of pointers to specific positions in the JDOM object that represents the main WSBPEL process for the service composition. When a new pattern implementation (sequence, parallel split, etc.) is added to the process, the position of the pointer is pushed onto the stack. This approach allows the BPELGenerator to move back up the structure of the service composition when necessary by popping the last stored pointer position off the top of the stack using the **stepOut** method. This functionality is necessary when, for example, all of the services in a sequence have been added to the WSBPEL process and so it is necessary to move back up to the parent of the sequence to continue adding constructs to the workflow.

The **BPEL2String** method is used to obtain a representation of the WSBPEL DOM model as a XML string so that it can be written to a file.

5.2.5.2 Archiver

The archiver component provides a high level API for the creation and management of zip archives. It is based on the zip libraries from the Java SDK (java.util.zip) and provides support for the management of the file structure inside of the zip archive including the addition and removal of files as well as the creation of directories within the archive.

The functionalities of this component are used to produce an archive that has the required structure to be compatible with the ActiveBPEL workflow engine and its deployment mechanisms. The directory structure of an ActiveBPEL process archive is shown in figure 5.11. For such an archive to be valid, it must at least contain a WSBPEL process file, an associated PDD and a Catalog file.

```
singleService
|-- META-INF
|   |-- catalog.xml
|   `-- pdd
|       `-- WSRPProcess
|           `-- Process.pdd
|-- bpel
|   `-- WSRPProcess
|       `-- BaseServiceProcess.bpel
`-- wsdl
    `-- WSRPProcess
        |-- stylesheets
        |   |-- messageFix.xsl
        |   `-- messageFix2.xsl
        `-- wsdl
            |-- BaseServicePartnerLink.wsdl
            |-- CorrelationSet.wsdl
            |-- WSRPBaseService.wsdl
            `-- WSRPProxyPartnerLink.wsdl
```

Figure 5.11: Structure of a WSBPEL process archive for the ActiveBPEL workflow engine.

In the specific case of the WSBPEL processes deployed as part of a PWE, the archive also contains the WSDL and XSL files that are used by the WSBPEL process itself. These are all contained in the **wsdl** directory of the archive, as shown in the example.

5.2.5.3 Deployer

As mentioned previously, the Composition2BPEL service deploys the generated WSBPEL process to the ActiveBPEL workflow engine and registers the service interfaces for the WSBPEL process with the User Portal. The deployment of the WSBPEL process itself is achieved through a web service call to the *ActiveBpelDeployBPR* operation, which is part of the management interface provided by ActiveBPEL. Using this web service, the zip archive generated by the Archiver Component can be uploaded to the server where it is unpacked, validated and deployed by the workflow engine.

Before the User Portal can access the deployed WSBPEL process, it must first know that the service exists and where it is located. To do this, the Composition2BPEL service registers the newly deployed WSBPEL process as a WSRP PortletProvider with the Pluto Portal engine that forms the basis of the User Portal. The next step is then to register with Pluto the portlet representing the activity. This then allows instances of the portlet to be incorporated into the User Portal providing an interface to the user's personalised service composition.

Technically, the process of registering the new PortletProvider that the WSBPEL process provides is achieved by making the appropriate HTTP requests to the Pluto engine in order to first register the portlet provider and then the portlet that it provides. This essentially simulates the requests that the Pluto Portal would receive if a user was interacting with the web based graphical management interface that it provides. This approach was required because it is necessary for the registration process to be automated. However, the Pluto Portal does not provide a management API that can be used by third parties.

5.2.6 Workflow Engine

As mentioned previously, the personalised service compositions generated by the AE are transformed into run time executable workflows by the Service Composer. Before the users can make use of these workflows, they must first be deployed to a workflow engine that supports their execution and which the User Portal can interact with.

The workflow engine used to execute the personalised service compositions must support the control flow patterns used to describe the PWE as well as supporting the deployment and parallel execution of many workflows (corresponding to the PWE of each individual user). This deployment mechanism also needs to be compatible with the existing infrastructure. Specifically, it must be possible to automatically generate the run time description of the workflows and to deploy those workflows as part of an automated process.

The first of these requirements, a workflow technology that can support the implementation of the necessary workflow patterns and which can be automatically generated, can be satisfied by the use of the Web Services Business Process Execution Language (WSBPEL).

The run time execution of a workflow defined using WSBPEL is carried out by compatible workflow engine. Many such workflow engines exist including several open source engines such as Apache ODE and ActiveBPEL. The ActiveBPEL workflow engine was selected as the workflow engine based on the maturity of the system, the completeness of its support for the WSBPEL standard and the flexibility that an open source license provides in terms of deployment and future use of the system.

ActiveBPEL is implemented in Java and can be deployed as a servlet based web application on a compatible servlet engine such as Apache Tomcat. As such, it integrates well with the platform used to run the other components and services that make up the system. Although ActiveBPEL provides a web based interface for managing the life cycle of workflows that are deployed to the server, it also provides a web service based management interface. As part of this interface, ActiveBPEL provides a web service that allows workflows, which have been packaged in a compatible archive format, to be deployed remotely. Through the use of this interface, the workflow component of a PWE can be automatically deployed. The details of the structure of this archive format and the deployment process have been discussed previously in this chapter.

5.2.7 User Portal

Once a user's PWE has been generated, through the execution of the narrative by the AE, and the run time instantiation of the personalised service composition has been deployed to the workflow engine, it is then possible for the user to interact with their PWE. This is achieved through the User Portal, which provides the user with a web based user interface that provides an environment in which they can interact with both content and services. The basis of the portal interface is a web application built using Java Server Pages (JSP) technology. JSP is a dynamic web development technology, similar to Java Servlets, that allows for the rapid development of dynamic web applications. The advantage of JSP over Servlets is that it allows Java and web technologies such as HTML to be combined more easily. The use of this Java based technology facilitates the integration of the portal with other core components of the architecture, which as discussed previously have also been implemented using various Java based technologies. The ability to develop an interface that can be modified dynamically is fundamental to the delivery of the type of personalised user experience required by PWEs. To deploy JSP based web applications a servlet engine that conforms to the Java Server Pages specification, examples of such engines include Apache Tomcat and Jetty [Eclipse Foundation 10]. For the deployment of the PWE User Portal, the Apache Tomcat servlet engine was chosen as it is a more mature and more widely adopted platform, which has a flexible open source license (Apache License version 2.0).

The User Portal web application itself is divided into a set of JSPs that are responsible for the various different tasks that the portal must perform, login, user model elicitation, invocation of adaptive behaviour, content delivery, and service delivery. In addition to these JSPs, there are a set of utility classes are libraries that implement some of the functionality that the JSPs require to carry out their function. Figure 5.12 provides an illustration of the navigation paths between the JSPs that make up the User Portal.

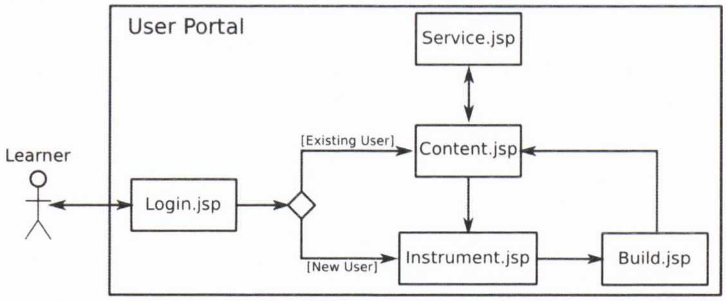


Figure 5.12: Navigation Paths through PWE User Portal

5.2.7.1 Login

When a user first attempts to access the portal, they are presented with a login page. The purpose of this page is to uniquely identify the user so that personalisations carried out by the system can be stored for later use. When a user enters their authorisation credentials (username and password) they are verified against the data stored in the database. If authorised, the user is logged into the system. Additionally, the login page checks whether or not the user has an existing PWE. If a PWE exists for the user then they are sent directly to the content page so that they can access the content and services of their PWE. Otherwise, the system will request additional information from the user in order to generate a new PWE for that user.

5.2.7.2 Instrument

The 'instrument' JSP provides the user user with a tool through which their user model can be elicited. The user fills in and submits a form containing information about their preferences with respect to the specific PWE. The nature of the form and the specific information elicited from the user is based on the PWE to be generated by the system. The information obtained from the user is used to populate a learner model that conforms to the XML schema of the learner model, as discussed in section 5.3.1 of this chapter. The newly generated user model is then stored in the database.

5.2.7.3 Build

When a model of the user is made available to the system through the use of the instrument JSP, it is then possible to generate a new PWE for the user. This is achieved through an automated call to the build JSP following the completion of the user model elicitation. The build JSP is responsible for invoking the Adaptive Engine and providing it with the information necessary (location of the database, names of the necessary models, etc.). The invocation of the Adaptive Engine is carried out by directly calling the Java API of the Adaptive Engine. This is possible because the Adaptive Engine is included in the Portal web application as a library. The process through which the AE generates a PWE is described in detail later in this chapter, see section 5.5.

5.2.7.4 Content

The primary functionalities of the Portal are the delivery of multimedia and services to the user. The first of these functionalities is provided by the content JSP. This JSP provides the user with a web page that allows them to navigate their PWE as well as presenting them with the content that has been selected for them.

As discussed previously, the output of the adaptation process is an XML model that describes the personalised composition of content and services that make up the user's PWE. To deliver this to the user it is necessary to extract the structure of the PWE from this model as well as to retrieve the content that the user is currently viewing. This is achieved through the use of XSL and related technologies to process the PWE model as shown in figure 5.13

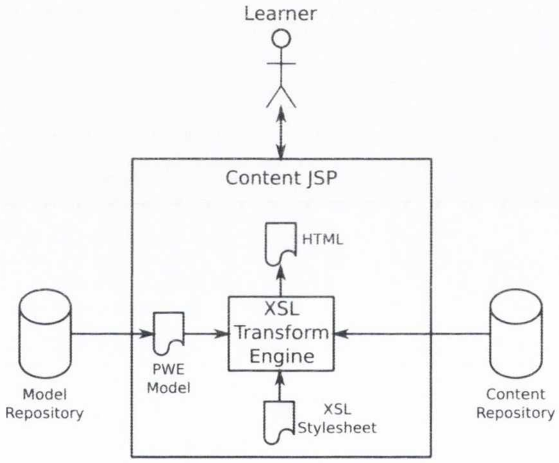


Figure 5.13: Extracting PWE navigation structure and content references from PWE model

The content JSP creates an instance of the Xalan XSLT engine[2] [Apache Foundation]. The transform engine is provided with the current user's PWE model, which is retrieved from the database along with the XSL stylesheet itself. The PWE model is then parsed into a Java Document Object Model (DOM) object. The DOM representation of the PWE model along with the XSL stylesheet are passed to the transform engine. The execution of the transform engine produces a HTML document that provides a navigational structure that is specific to the user's PWE. The resulting HTML document also contains the multimedia content that the user is currently accessing. This content was retrieved and

[2] The Xalan Engine is a Java based XSLT engine that, as version 1.5 is included in the default Sun Java installation as the reference implementation of the JAXP API[Oracle 10a]

included in the HTML document as part of the XSLT transform process.

As the user navigates through their PWE using the interface generated for them, the transformation is rerun to retrieve new content for the user and to update the navigation structure of the interface based on the user's changing context within the interface.

5.2.7.5 Services

The delivery of services to the user is handled by the service JSP, it is responsible for presenting a portlet based representation of the services that, in combination with the content, make up the user's PWE. The services used in a PWE are implemented as portlets using the WSRP protocol. In order to deliver the composition of these services to the user it is necessary for the portal interface to communicate with the workflow engine that hosts the run time executable workflow instances. The portal must also be capable of rendering the portlets as part of the user interface.

To provide support for the underlying portlet technologies, the PWE User Portal web application is built on top of the Apache Pluto Portal [Apache Foundation 08] framework. Apache Pluto is a reference implementation of JSR-168. As such Pluto provides a simple web application that allows the installation of portlets and their incorporation into a simple interface. Unlike other Java based portal environments, such as Liferay [Liferay Inc 10], which only allow portlets to be managed through the web interface to the portal, Apache Pluto provides an API that can be used to explicitly include a portlet in a JSP. This is necessary for the implementation of the PWE User Portal as the specific portlet to be included in the service JSP is only known at run time and therefore cannot be configured a priori. Furthermore, as new users are added to the system, it is necessary for the portal to include the corresponding portlets for those users. This can only be achieved using the flexible functionality provided by Apache Pluto.

Apache Pluto only provides support for portlets that are deployed locally and that are developed in accordance to the JSR-168 specification. To be able to include WSRP based portlets that are deployed remotely it is necessary to add support for the WSRP protocol to Apache Pluto. This is achieved using Apache WSRP4J [Apache Foundation 06], a Java based implementation of the WSRP protocol that is designed to work with any portal technology that uses the JSR-168 specification. For the PWE User Portal, it is only necessary to support the consumption of WSRP portlets. This is achieved by installing the WSRP4J PortletProvider web application along side the Pluto Portal web application. The functionality provided by WSRP4J can then be integrated with Apache Pluto by configuring Pluto to be aware of the two management portlets that WSRP4J provides. These management portlets allow remote portlet providers and the portlets that they provide to be registered with WSRP4J. After registration, it is then possible to integrate these WSRP portlets with Pluto through a third portlet that WSRP4J provides. Figure 5.14 illustrates how the Service JSP relates to Apache Pluto and WSRP4J.

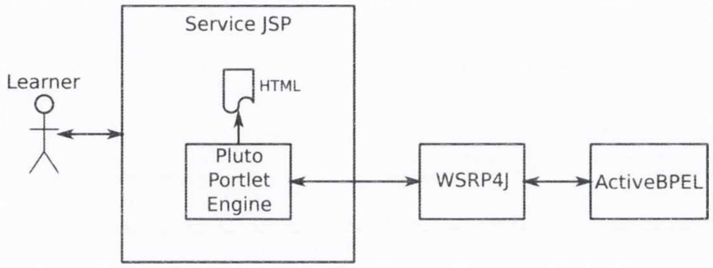


Figure 5.14: The delivery of services to the user requires the integration of several portlet technologies.

After combining Apache Pluto and WSRP4J it is possible for Apache Pluto to display WSRP based portlets by offloading the handling of the protocol to WSRP4J. All that Apache Pluto is required to do is to load the WSRP4J portlet.

5.2.8 Service Implementation

As described previously, the services used to instantiate a PWE are implemented as portlets delivered using the web service based protocol defined by the WSRP specification. The use of this protocol allows the system to take advantage of any portlets provided by third party providers that support the WSRP specification. As the communication mechanism used in WSRP is web service based, it also means that portlets can be used irrespective of the underlying implementation technology or platform. For example, a portlet developed using the Microsoft .NET platform[Microsoft 10a] and deployed on a Sharepoint Server[Microsoft 10b] can be integrated into a PWE, which is delivered using a Java based platform.

Even though the system can take advantage of existing portlets, some scenarios will require the development of services that meet needs that are specific to the PWE to be delivered. As mentioned, such services can be developed using any platform that supports the WSRP specification, such as the combination of Apache Pluto and WSRP4J used to implement the User Portal described previously in this chapter. Another platform that could be used is the OpenPortal Portal Server. OpenPortal is a Java based portal server that runs on the Glassfish[Glassfish Community 11] Java Enterprise Edition Application Server. In order to be deployed on the OpenPortal platform, a portlet must conform to the JSR-168 [Scott Nicklous 03] specification for Java based portlets. JSR-168 is a specification for portlets developed using Java, which was developed through the Java Community Process (JCP). It defines an API for the development of portlets as a means to promote interoperability between portals. Significantly, OpenPortal Portal Server also supports the WSRP specification and allows any portlet deployed on it to be exposed as a WSRP portlet automatically without the need to modify the portlet in any way. The portal server handles the mapping between the WSRP protocol and the actions that the portlet supports.

OpenPortal was selected as the platform for any custom developed portlets required in the testing and evaluation of the PWE framework for several reasons, it provides a mature platform that is compliant with the key specifications used in the system. As it is deployed on Glassfish, a full J2EE application server, it also allows for portlets to be developed that integrate with the other components, such as databases, that the application server provides. Importantly, this also means that there is good tool support for the development and deployment of portlets using the Netbeans IDE[Netbeans Community 11]. Both OpenPortal and Glassfish are open source projects made available under the Common Development and Distribution License (CDDL) [Sun Microsystems 06].

Alternatively, the Apache Pluto Portal Server, used in the development of the User Portlet component, could have been used as the platform for the deployment of custom portlets. However, the tool support for Pluto is not as mature as that of OpenPortal. Furthermore, both Pluto and WSRP4J are not as stable as OpenPortal[3].

The implementation of a JSR-168 based portlet follows a Model, View paradigm with a Java class used to implement the model and a set of JSPs to handle the view. To create a new portlet, a class that extends the GenericPortlet class (provided as part of the JSR-168

[3] Currently, it is not possible to get a stable build of WSRP4J, it must be checked out from the project's version control system and built from source

implementation) is written and the necessary methods overridden. The follow methods are the most significant:

- init
- processAction
- doView
- doEdit
- doHelp

The View component of the portlet is implemented as a set of JSPs, with at least one JSP for each mode that the portlet supports (view, edit, help).

The *init* method is used to initialise the portlet when it is loaded by the portal server for the first time. This can include tasks such as loading configuration data from the portlets configuration files.

When a request for the default view of a portlet is made, a call to the *doView* method is made, this carries out any necessary business logic and then specifies the JSP that should be used to generate the portlet UI. Information can be passed between the model implementation class and the JSP using Java Beans. If the portlet is one of the other supported states, Edit or Help, then the *doEdit* or *doHelp* methods would be called respectively.

When a user interacts with the portlet, by clicking a hyperlink or submitting a form component of the UI, the *processAction* method is called, allowing the portlet to process the information provided via the user interface. As a result of the business logic executed at this step, the next request to *doView* (or whichever view is currently active) could result in a change to the portlet presented to the user.

5.3 Model Implementation

Previously in this chapter the implementation of a model driven system capable of delivering personalised web experiences has been presented. This system requires a core set of models in order to drive the personalisation process consisting of the Narrative Model, Learner Model, Content Model and Service Model. The Narrative Model provide the strategy for adaptively composing a PWE based on the needs and preferences of the user, as defined in the Learner Model. The content and services that are available to the system for use in generating a PWE are described by Content and Service Models respectively.

In addition to the models used to inform the adaptation process, there are also a set of models generated by the Adaptive Engine as outcomes of the adaptation process. The Personalised Web Experience (PWE) Model is the instantiation of the abstract PWE described in the Narrative Model. The Composition Model is an additional model used as an intermediate step between the generation of the personalised composition of services produced by the Adaptive Engine and the run time instantiation of that composition.

This section provides technical details of how each of these models is implemented. Where models have been specifically designed for use in the system, a technical definition of the model structure is provided. For other models, such as the Learner Model and Content model, which are more flexible and open, the models are described in terms of their role within the system and the possible standards and specifications that can be used to implement them.

5.3.1 Learner Model

The Learner Model provides the system with information about the user, enabling it to personalise the composition of the PWE to meet the needs of the individual user. As such, it serves a critical role in the adaptation process. As discussed previously, the information captured in the Learner Model is dependent on the specific requirements of the PWE. So as not the restrict the designer of a PWE and to enable the system to adapt to a wide range of user properties, a schema for the Learner Model has not been explicitly defined. The designer is free to choose a schema that supports the specific requirements of the PWE that they wish to provide to the user, ranging from proprietary schema to open specifications such as IMS LIP. The only caveat on this flexibility is that the vocabulary used in the model should be consistent with the other models used in the information model.

An example of a Learner Model that could be used in the generation of a PWE is shown in figure 5.15. This model is a custom developed model that has been implemented as an overlay model of a domain covering the SQL database query language. It provides the

```
<learner>
    <general>
        <name></name>
    </general>
    <competencies>
        <competency level="advanced">DBConcepts</competency>
        <competency level="advanced">CreatingDB</competency>
        <competency level="novice">PopulatingDB</competency>
        <competency level="novice">DBRetrieval</competency>
        <competency level="novice">DBApplications</competency>
    </competencies>
</learner>
```

Figure 5.15: Example instance of a Learner Model

system with a value for the learner's competency for each concept in the domain.

5.3.2 Content Model

The Content Model is used to provide the system with information about the content resources that are available for selection and inclusion in an instantiated PWE in order to satisfy the requirements of the user. The Content Model is a composite model consisting of many individual metadata descriptions of the individual content resources. This approach makes the updating of the content model simpler in comparison to the difficulty in managing change in a model consisting of a single metadata resource. The metadata descriptions of individual content resources can be added, modified or removed without the risk of affecting the metadata relating to unrelated content resources.

A specific information model has not been defined for the Content Model used in the generation of PWEs. Instead, the designer of a PWE to be delivered by this framework is free to choose the information model that is appropriate for their needs as long as that information model is adhered to across all of the content metadata used by the system. An additional restraint on the Content Model is that the vocabulary it uses to describe the content should be consistent with the other models used by the system. At the very least, the vocabulary should be shared with the Narrative Model, which could carry out mappings between differing vocabularies if necessary.

This is important because the content used in the delivery of PWEs can differ greatly between instances. For example the granularity of the content could be different with one PWE using finely grained individual content resources described using IMS Metadata [Consortium 06] while another PWE could be using coarsely grained Learning Objects described using SCORM. The flexibility in the information model also means that the system can be used to deliver PWEs even in constrained environments, for example, where the designer of the PWE must use content provided from a repository that uses a specific metadata standard to the describe the content.

Although the information model used to describe the content is open, whatever model is used must provide sufficient information about the content in order to support the adaptation process. For example, the model must provide either technical properties, such as size and format, or pedagogical properties such as the objectives taught by the content. These properties are in addition to the basic information that the system requires in order to retrieve the content resources.

As discussed in the design chapter, the content model is implemented using XML, this provides a machine readable common format which can be easily processed by the system components. XML is also used by a wide range of standards used to provide metadata about content resources. Examples of such standards include Dublin Core [The Dublin Core Metadata Initiative 10], IMS Metadata, SCORM and IMS LOM.

5.3.3 Service Model

The role of the service model in the composition of a PWE is to make the Adaptive Engine aware of the services that are available for selection and inclusion. In addition to providing this discovery mechanism, the service model also provides the Adaptive Engine with information about the available services that will enable it to differentiate between the services based on both functional and non-functional properties.

To carry out this role in the system, the service model must provide information to the system about what a service does, where it is located and how it can be invoked. All of this information must be provided in a manner that is accessible to the system. In order for the Adaptive Engine to be able to comprehend the information in the Service Model, it is necessary for there to be a degree of commonality between the vocabulary used to describe the services and that used in the other metadata models used in the adaptation process. Unlike the learner and content models, an information model has been defined for the Service Model. This is necessary because the information captured in the Service Model is less flexible than that in the other models and is more closely tied to the operation of the system components that consume that information.

Common approaches to the description of services focus on the description of Web Services using semantic technologies such as OWL-S or RDF to provide additional information about the service that the WSDL description of the service does not provide. Although the services used in the delivery of a PWE are delivered using a WS based protocol, the 'service' that the service provides is not delivered at that level. Despite this, OWL-S could still be used to describe the services in this system as it provides mechanisms for the capturing of information important for the adaptation process, especially when using AI planning techniques as is the case in this system. The reason for this is that the development of OWL-S was heavily influenced by PDDL so it contains mechanisms for describing the preconditions and effects of services[4]. Although OWL-S would allow the necessary information to be captured, it also provides functionality that is not required in this system. The architecture presented in the Design chapter does not call for any semantic reasoning over the service model therefore the affordances of OWL-S in this respect are unnecessary.

Instead the Service Model is implemented using XML to encode the information stored in the model, this facilitates the system components in accessing the information as it is provided in a highly structured manner while still facilitating the encoding of all of the necessary information.

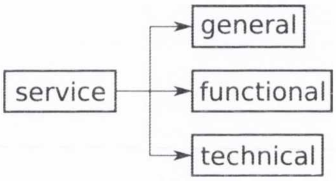


Figure 5.16: The <service> elements

Figure 5.16 provides an overview of the information model for the service model used in the system to describe a single service. The actual service model is a composite model consisting of the set of all of the available service metadata descriptions. As can be seen from the diagram, the model is divided into three parts, <general>, <functional> and <technical>. The <general> element is used to provide basic information about the service, such as the name of the service and a textual, human readable description of the service. The structure of this part of the model is shown in figure 5.17.

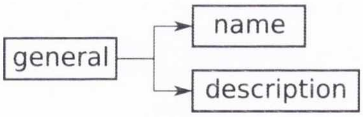


Figure 5.17: The <general> elements

The functional element of the model provides information about how the service operates and what the service does, providing details of the parameters that the service requires when being invoked. In order to capture information about how the service operates and what the service does, the model allows preconditions and effects for each service to be defined. These are concepts that are taken from the AI Planning domain in order to describe when it is appropriate or feasible to execute a service and what the outcome of

[4] At present this support is not well defined with only place holders for preconditions and effects described in the OWL-S specification. It is up the user to choose a means of describing the preconditions and effects

that service will be. An example of a precondition for an electronic document submission system, might be that the document exists. Similarly, the effect of executing that service would be that the document is submitted. Figure 5.18 shows the elements that are used to capture this information and how they relate to each other.

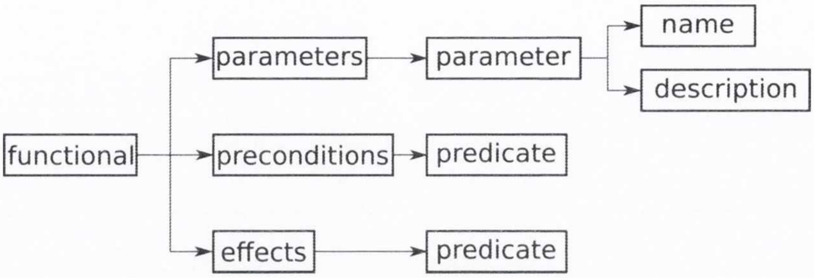


Figure 5.18: The <functional> elements

As can be seen from the diagram There are three child elements of the <functional> element, <parameters>, <preconditions> and <effects>. The <parameters> element contains zero or more <parameter> elements, each of which consist of a <name> and <type> element. The <preconditions> and <effects> elements contain zero or more <predicate> elements. The <predicate> element, shown in figure 5.19, allows predicates to be defined consisting of exactly one <name> element and one or more <variable> elements contained in a <variables> element.

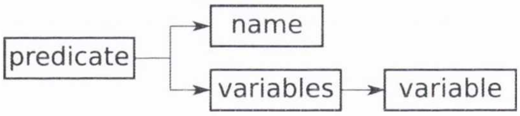


Figure 5.19: The <predicate> elements

The technical element of the model, shown in figure 5.20, describes the details of how the service can be executed. It proves a URL, which points to the location of the WSDL descriptions of the WSRP service interfaces that can be used to invoke the service. The 'handle' for the service that the model describes is also provided. This is necessary as a single WSRP service provider can host many different services.

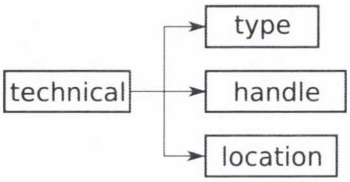


Figure 5.20: The <technical> elements

5.3.4 Narrative Model

The Narrative Model is very different to the other input models, such as the Learner, Content and Service models, that are consumed by the Adaptive Engine. Unlike those models, which are used as information sources to inform the Adaptive Engine about the context in which it is operating, the Narrative Model must capture the sequencing and selection rules that allow the PWE to be generated. Furthermore, the rules that make up the Narrative Model are based on the information contained in the other models. As part of the process of composing a PWE, the narrative must also be able to store the generated composition so that it persists beyond the execution of the narrative. As such the technique used to capture the Narrative Model has the following requirements:

- To describe the structure of a PWE and the possible branches in that structure
- To describe the rules that govern how the structure of the PWE can be adapted
- To facilitate access to the metadata models
- To support the generation of new models

The Adaptive Engine contains a scripting engine based on the Jatha, Java implementation of a Lisp engine, which is used to execute the narrative model. The use of Lisp as the basis for the narrative language allows the narrative to be very flexible in the types of behaviour that it can represent as it has all of the control flow and other constructs that are available in such a programming language. Other important features of a language such as Lisp, which are advantageous when writing a narrative are the use of variables to store information temporarily and the use of functions to encapsulate functionalities that are repeated throughout the narrative. The availability of libraries of basic functions such as those for basic arithmetic and string manipulation are also useful when retrieving and manipulating information from multiple data sources, such as the models used by the Adaptive Engine.

As mentioned previously, a set of extensions or 'custom functions' were added to the Lisp engine. These added additional functionality to the Lisp environment and allowed the integration of the different Adaptive Engine components. These functions are exposed to the Lisp based narrative, allowing it to interact with the available models. As with the functionality provided by the basic Lisp environment, these custom functions are designed to provide very low level functionality in order to maintain the flexibility of the Narrative

Model in terms of the behaviour that it is capable of capturing. Further details on the Narrative Model syntax and examples of its usage can be found in section 5.5 of this chapter.

5.3.5 PWE Model

The Personalised Web Experience (PWE) Model is the instantiation of the abstract PWE described in the Narrative Model. As such, it is generated as the outcome of the adaptation process and describes the personalised composition of content and services. The PWE model is used by the User Portal in order to generate the personalised navigation structure for the user. It also specifies the specific content resources that should be delivered to the user as well as specifying where the run time instantiation of the user's personalised service composition can be accessed. From this, four requirements of the PWE Model can be identified:

- The association of a PWE Model with a specific user
- The description of the structure of the user's PWE
- The identification of specific content resources to be included in the PWE
- The location of the service composition component of the PWE

As the role of the PWE Model is to describe the navigation structure for the activity, the schema for this model needs to remain flexible, allowing the designer to describe the structure of their activity as necessary. The only restriction that is placed on the PWE Model is that its schema should be compatible with the User Portal.

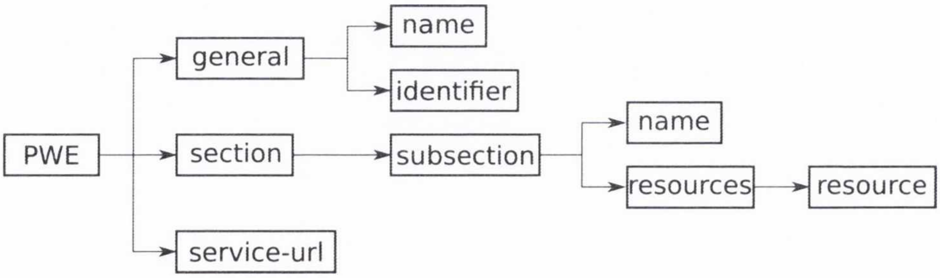


Figure 5.21: Example element structure of the PWE Model

An example of the information model for such a PWE Model can be seen in figure 5.21 (element names prefixed with a '*' indicate elements that can occur 1 or more times). In this example, the structure of the PWE consists of a section/subsection hierarchy in which subsections are populated with specific content resources to be provided to the user. The <services> element provides a URL that points to the user's run time service composition workflow.

5.3.6 Composition Model

The Composition Model is produced as an output of the adaptation process, as such it forms part of the PWE Model for an individual user. It is generated as a result of the execution of the narrative by the Adaptive Engine and describes the personalised composition of services that instantiates the tasks in the users PWE. As discussed previously, to make the composition of services available to the user at run time, it is necessary to deploy it to a workflow engine. As such, the Composition Model is an intermediate model that simplifies the task of defining the composition for the Adaptive Engine as it does not need to handle the complexity of the underlying workflow language, in this case WSBPEL. It also allows the adaptation process to be decoupled from the WSBPEL based delivery mechanism.

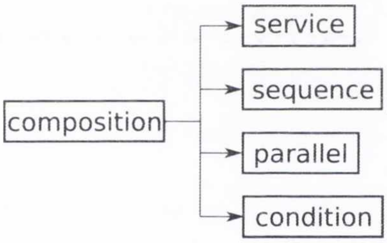


Figure 5.22: The <composition> elements

Figure 5.22 provides an overview of the basic structure of the Composition Model. The model allows services, described using the <service> element, to be composed using three basic constructs that correspond to the workflow patterns described, which have been identified as the basic control flow constructs used in a PWE as discussed in section 4.4.1.1 of the design chapter. The 'merge' pattern is not represented explicitly in the model but is implicit.

The <sequence> element can be used to represent a sequence of services or other control flow patterns. The <parallel> element is used to indicate the use of the Parallel Split pattern, allowing services to be defined as executing in parallel while the <condition> element is used to represent the user of an Exclusive Choice pattern. The use of this element implies that the choice will be between all of the child elements as there is no explicit definition of the possible choices. Both the <parallel> and <condition> elements support the inclusion of any of the elements representing control flow patterns as is the case with the <sequence> element.

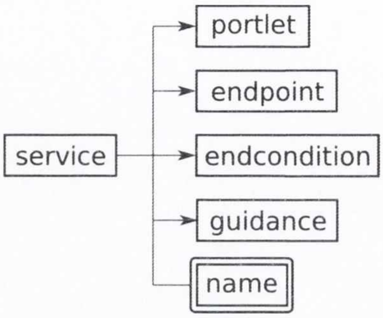


Figure 5.23: The <service> elements

Each service in the composition is described using the <service> element, shown in figure 5.23. This is a container for all of the information necessary to invoke a WSRP portlet based service. The name of the service, which corresponds to the 'handle' of the WSRP portlet, is specified using the 'name' attribute of the service element. The <endpoint> element provides the URI for the WSRP BaseService, which contains the getMarkup operation used to retrieve the interface code for the portlet while the <portlet> element contains the handle for the specific portlet to be included in the composition. The <endcondition> element describes the WSRP performBlockingInteraction message (parameter and corresponding value) that will be used to indicate when the user is finished with the service. The <guidance> element provides the instructional message to the user that will be presented to them while using the service.

5.4 Information Flow

Part of the instantiation of a PWE is the personalised service composition, which is generated by the Adaptive Engine along with the PWE Model. This composition consists of the adaptively sequenced services that were selected, and in some cases composed, by the Service Composer and is defined in terms of the workflow patterns that were identified previously. However, although this composition contains all of the information necessary to orchestrate and invoke the appropriate services, it is not executable. To achieve this, the composition is passed to the Composition Mapper, which generates an equivalent composition using the WSBPEL workflow language. The WSBPEL workflow can then be executed in order to provide the appropriate services to the user as part of their PWE.

As the WSBPEL workflow is orchestrating services that are based on the WSRP specification, it is necessary for them to handle the communication protocol of these services as well as the control flow between the services. To achieve this is is first necessary to have a WSBPEL process with a basic skeleton structure that will manage the life cycle of the service composition, allowing the system to provide the user with appropriate cues that the beginning and end of the composition. Once this is in place, the next requirement is for the WSBPEL process to be able to broker the communications between the User Portal and a WSRP compliant service provider. It is then necessary to be able to implement, in WSBPEL, the workflow patterns used in the composition, namely the Sequence, Parallel Split, Exclusive Choice, Simple Merge and Synchronisation patterns. In fact, it is not necessary to explicitly implement the Synchronisation and Simple Merge patterns as this functionality is implicit.

5.4.1 Skeleton WSBPEL Process

To transform a personalised composition of services, as generated by the AE, into a run time executable WSBPEL workflow, it is necessary to include some basic functionality that is common to all service composition workflows. This bootstrapping WSBPEL code provides three main functionalities, the creation of a new instance of the workflow, the management of the different message types used in the WSRP protocol and the notification of the user when the workflow has been completed. Figure 5.24 provides an illustration of how these three core functionalities are implemented in WSBPEL. As can be seen from the diagram, the functionalities are each provided by distinct structures, labelled 1-3 in the diagram.

The whole workflow is contained in a WSBPEL sequence element, this provides a linear sequencing between the three core components so that when the user has started a new workflow instance (step one in the diagram) the workflow will automatically progress to the main execution loop of the workflow (step two) and finally, upon completion of the services that make up the service composition, onto the final notification component that informs the user that they have finished their tasks (step three). These three steps will now be described in detail.

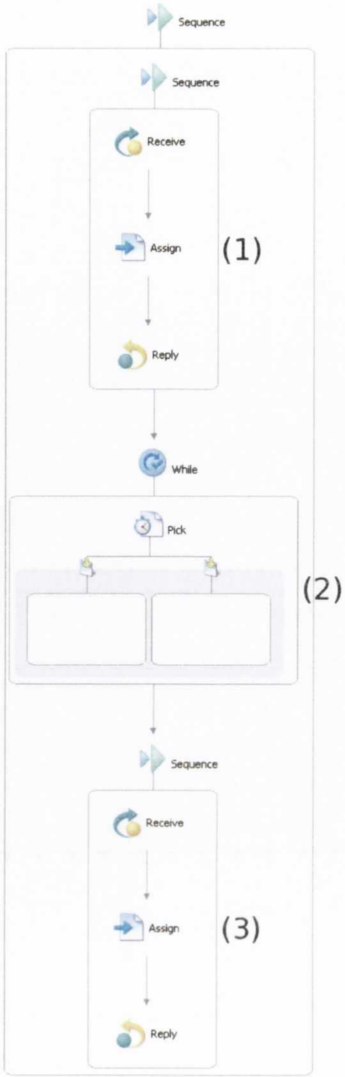


Figure 5.24: Visualisation of the WSBPEL code used to bootstrap a PWE Service Composition

To have a valid WSBPEL process, it is necessary for the first message received by the workflow process to occur outside of any control flow loops. This is because it will be used by the workflow engine as an indication that a new instance of the workflow should be created as well as providing information that the workflow engine will use to differentiate between different instances of the same workflow process[5]. For this reason, the first part of the WSBPEL process, which consists of a Receive, Assign, Reply sequence, waits for a WSRP GetMarkup message from the User Portal. It is this first message from the User Portal that results in a new instance of the process to be created by the workflow engine. The newly created process instance then responds to the User Portal with an appropriate

[5] The mechanism used in WSBPEL to differentiate between instances of a process is referred to as 'correlation'

message so that the user is presented with a simple 'welcome' portlet.

The second step in the workflow, labelled as step 2 in figure 5.24, is a container for the actual implementation of the personalised service composition. As such the WSBPEL code used to instantiate the personalised composition of services will be added to this section of the workflow process. The basis of this construct is a while loop containing a WSBPEL pick element. The while loop ensures that the process instance does not just end after the first interaction with the process, as would be the case without it. The while loop only exits successfully when all of the tasks that make up the service composition have been completed. The WSBPEL pick construct allows a WSBPEL process to make a branching decision based on the type of message received by the process. In this case, the pick construct allows the process to wait for one of the two WSRP message types used in this system (GetMarkup and PerformBlockingInteraction) and to execute the appropriate behaviour as necessary.

Once all of the necessary steps in the service composition have been executed successfully, the control flow will exit the while loop and move on to the final step in the WSBPEL process (3). This is another 'mock' WSRP service that responds to GetMarkup requests. When a user has completed their tasks, they are presented with a portlet that indicates to them that their activity has finished. Without this step in the WSBPEL process, the system could automatically restart the activity for the user, which is not the desired behaviour.

5.4.1.1 Brokering a WSRP service in WSBPEL

The basic construct used in the automatically generated WSBPEL process is one that allows for the execution of remote WSRP services in response to a user interaction with the WSBPEL process. This construct, as with the entire WSBPEL process, must be able to handle both the getMarkup messages that provide the information necessary to render the interface of a portlet and the performBlockingInteraction messages that notify the WSRP service of a user interaction. As the construct is acting as a proxy for WSRP messages between the User Portal and the WSRP service provider, it is necessary for the construct to be able to extract the information from the incoming request and to forward that information to the appropriate WSRP service implementation. The response from the WSRP provider then needs to be returned to the User Portal.

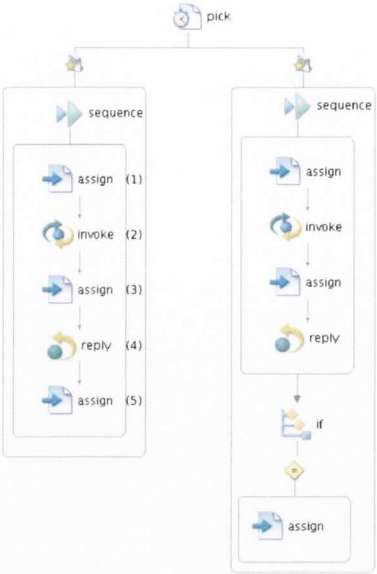


Figure 5.25: Illustration of WSBPEL code to manage the execution of a WSRP portlet

Figure 5.25 provides an illustration of a segment of a WSBPEL process illustrating how this functionality is implemented. The diagram show the WSBPEL for handling both the getMarkup message and the performBlockingInteraction message. As can be seen in the diagram, the getMarkup message is received into the WSBPEL process by the pick element and passed into the left hand side of the construct. The 'sequence' container is used simply to ensure the progression of the flow of execution through the necessary steps. The first element in the sequence is the assign (labelled as step 1 in the diagram). This copies the necessary information from the getMarkup message received by the pick to a variable that is used to store the message that will be sent to the WSRP service to be invoked. Step 2 in the actual invocation of the WSRP service, this is a synchronous operation that waits for the service to respond. The response from the WSRP service is then copied to another variable that is used to store the response that will be returned to the User Portal, step 4. The final step in the sequence, step 5, assigns a value to a variable to indicate that the service execution has finished. This sequence of steps allows a request from the User Portal for a portlet to be satisfied by a remote WSRP service. A typical set of interactions between the User Portal, the workflow engine and a service provider are shown in the UML sequence diagram below (Figure 5.26). The scenario represented by the diagram involves the user loading a new service in the User Portal and then interacting with that service, for example clicking a button in the service interface. The user then loads a second service.

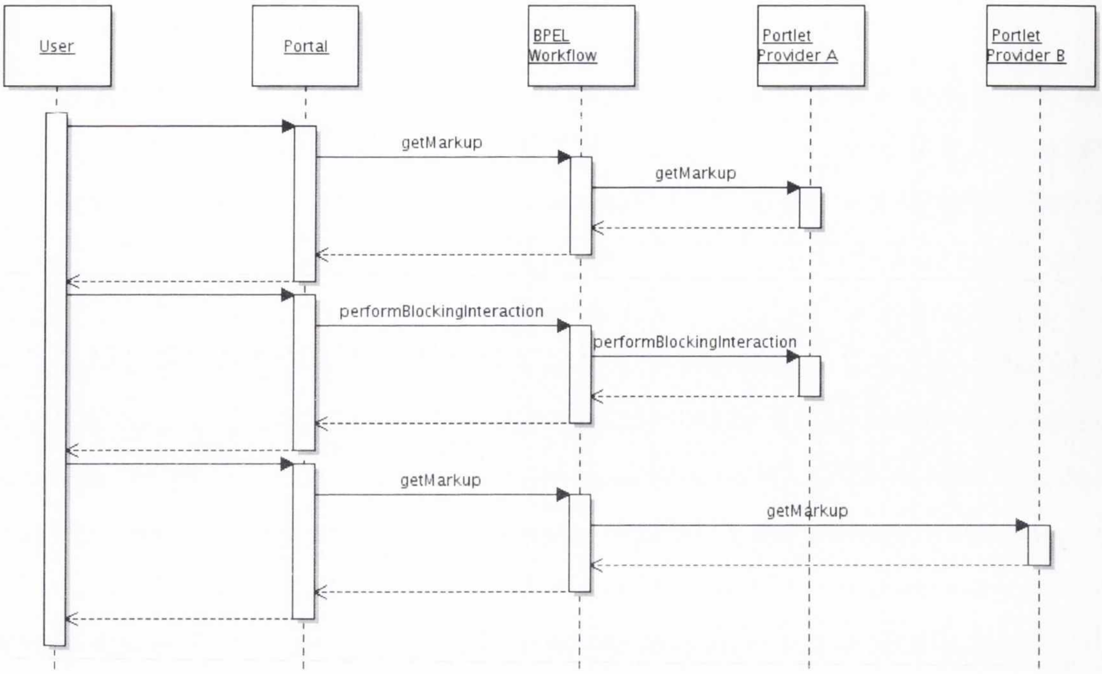


Figure 5.26: UML Sequence diagram illustrating the handling of requests for services from the User Portal

The handling of the WSRP performBlockingInteraction messages is illustrated on the right hand side of figure 5.25. The steps necessary are very similar to those required for a getMarkup message with a similar sequence of assign; invoke; assign; reply steps. An additional step that is carried out after handling a performBlockingInteraction message is that the message send to the WSRP service provider is compared to a predetermined message value in order to identify whether or not the user is finished with that specific service. It is assumed that all services used in a PWE will have some form of UI component that will allow the user to indicate that they are finished with the service. Without this functionality, it would not be possible to allow users to interact with a specific service over a prolonged period of time (for example, for more than one session instance).

5.4.1.2 Implementing the Sequence Pattern

Although WSBPEL already supports many workflow patterns [Wohed 02], including the sequence control flow pattern, it is necessary to design a new construct that can be used to manage the additional complexity of the WSRP protocol as well as fitting into the WSBPEL process as outlined previously. The mechanism used must also be flexible enough to allow sequences of arbitrary length to be defined as well as allowing the nesting of sequences and other control flow constructs.

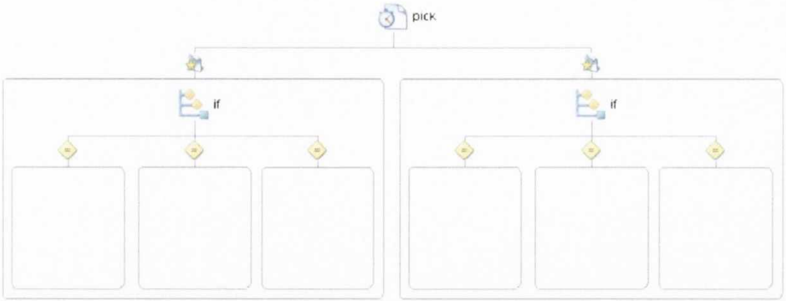


Figure 5.27: Illustration of the WSBPEL code to implement the Sequence Pattern

Figure 5.27 illustrates the basic structure used to implement a sequence of WSRP services. As shown, a sequence is implemented as a BPEL ‘if-else’ statement with a branch for each service in the sequence. The choice as to which branch is executed is made based on the evaluation of a simple counter variable, which is initially set to zero. Each branch in the ‘if-else’ statement evaluates against the counter, comparing its value against a pre set value that increments at each step in the sequence. For example, the first branch in the ’if-else’, representing the first service to be executed in the sequence, will compare the value of the counter variable against its pre set value, zero. When a user is finished with a service, the counter is incremented so that the next time control flow passes through the ‘if-else’ statement the test for the first branch will fail as the counter value is no longer zero but the test for the second branch will succeed and the second service in the sequence will be executed. In this way, each of the services in the sequence will be executed.

As illustrated in the diagram, the ‘if-else’ statement used to control which service in a sequence is executed is present for both the getMarkup and performBlockingInteraction branches of the WSBPEL process. The actual incrementing of the variable that controls the sequence is carried out based on the evaluation of the performBlockingInteraction messages sent from the User Portal. If the message contains a predetermined value for a specific variable, this indicates to the system that the user is finished with that service. If that service is part of a sequence then the counter variable controlling that sequence will be incremented.

This approach to the implementation of the sequence workflow pattern allows both simple services, described previously in this section, as well as other control flow constructs to be added at any point in the sequence. For example, when another sequence construct is added to an existing sequence, a different counter variable is used to control the new nested sequence construct. This means that the state of a different control flow constructs, such as sequences, are maintained independently of each other. This method of defining a sequence is also for automatic generation as required by the architecture of the system. The addition of new services or control flow constructs to an existing sequence only requires a new element to be added to the WSBPEL 'if-else' statement and does not affect any existing constructs in the workflow

5.4.1.3 Implementing the Parallel Split Pattern

The parallel split workflow pattern will allow the user to switch between multiple services that are running in parallel. It is used in a scenario in which the completion of a task requires the use of more than one service in a non linear fashion. From the user's perspective, it is necessary to have a visual mechanism through which the user can switch between services. Ideally, this would be achieved by leveraging the portal based approach to provide all of the services to the user at the same time. However, due to the limitations of the portal technologies used in this system, it is difficult to achieve an interface that can support a dynamically changing number of portlets.

As a solution to this problem, a tab based view is provided to the user in a single portlet. This allows multiple services to be provided to the user at the same time while providing a sense of integration between services. Figure 5.28 provides an example of a portlet that aggregates several portlets together while allowing the user to switch between them dynamically.

The implementation of the parallel split workflow pattern is based on the sequence pattern implementation described previously. As can be seen from the diagram in figure 5.29, the parallel split implementation uses the same 'if-else' based approach. It also uses a simple counter to keep track of which branch of the 'if-else' should be executed. The difference in how these two different patterns are implemented lies in how the value of the counter is updated. Unlike the sequence implementation, the value of the counter controlling the parallel split is set by the user through their interactions with the tab based interface. This interface is automatically generated from a template during the auto generation of the WSBPEL process and is added as a wrapper to the HTML source for the service portlet before it is returned to the User Portal and rendered to the user. It is possible to achieve

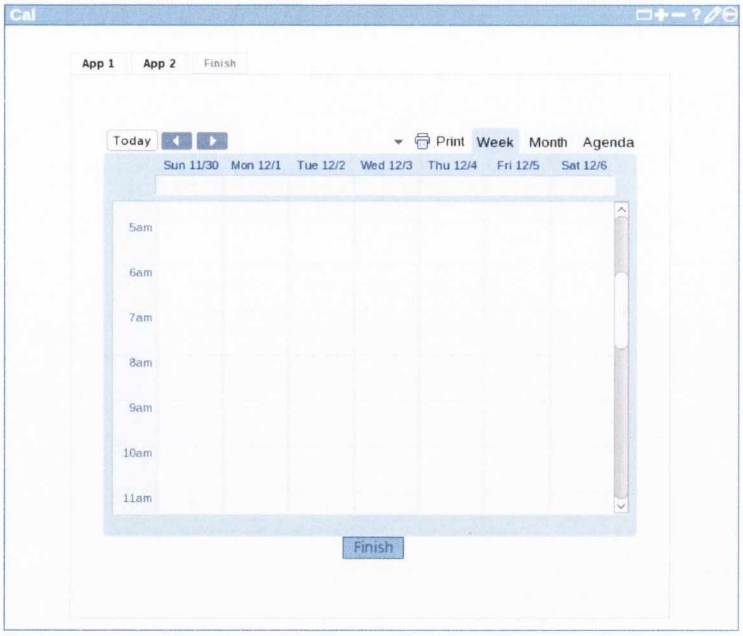


Figure 5.28: Screen shot showing the tab based metaphor used to implement the parallel split portlet

this wrapping functionality by carrying out an XSL transform as part of the execution of the WSBPEL process. The ActiveBPEL engine provides a function for carrying out XSL transforms that can be used when manipulating variables. The input to the transform is the unaltered code for the portlet that the user requested.

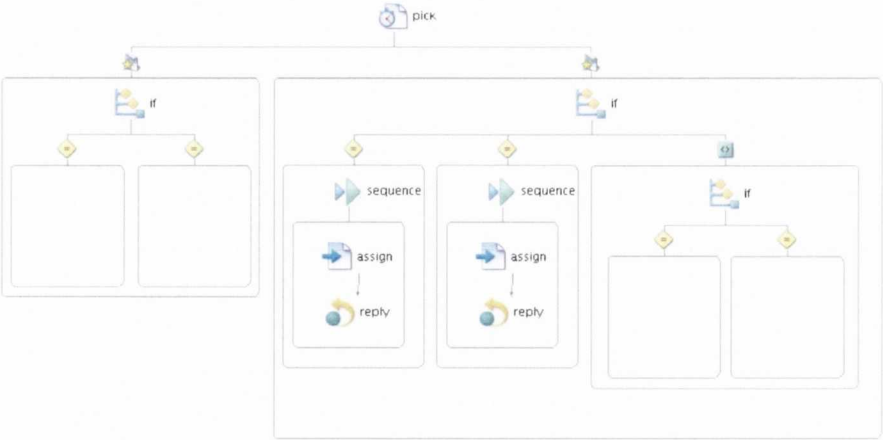


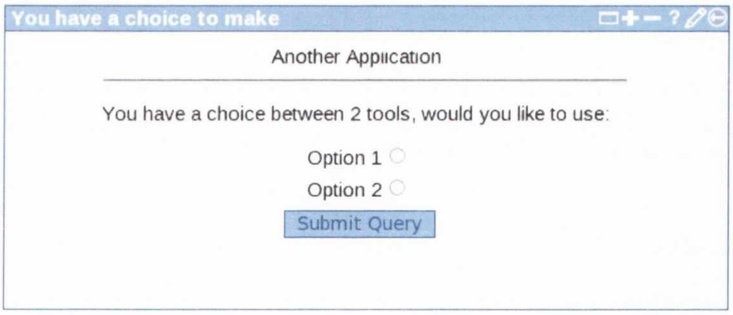
Figure 5.29: Illustration of the WSBPEL code to implement the Parallel Split Pattern

In order to capture the context switches requested by the user, it is necessary for the performBlockingInteraction part of the implementation (the right hand side of figure 5.29) to not only handle messages from the portlets but also from the tab switching mechanism. As can be seen in figure 5.29, compared to the implementation of the Sequence

Pattern, there is an additional 'if' element that wraps the constructs used to handle the messages from the portlets. This wrapper handles messages that control which portlet is the 'current' portlet buy looking at the parameter type that has been sent from the User Portal. If the parameter is of the appropriate type, indicating that a click of one of tabs was registered) then the first branch of the 'if' statement is executed and the appropriate value is assigned to the counter variable. If the parameter send from the User Portal does not correspond to a click on one of the tabs then the request is passed on to the appropriate WSRP portlet provider as described previously.

5.4.1.4 Implementing the Exclusive Choice Pattern

The Exclusive Choice workflow pattern allows control flow to be passed from a single incoming thread of execution to one outgoing branch based on a selection mechanism. In the context of a PWE, this selection mechanism is based on information that is only available at run time. This information could be based on the user's progress in the PWE or other such contextual information. An alternative application of this pattern is to allow the user to influence the path of execution by explicitly choosing the path that should be taken. The latter of these approaches has been implemented in the current implementation of the system.



You have a choice to make

Another Application

You have a choice between 2 tools, would you like to use:

Option 1 ○

Option 2 ○

Figure 5.30: Screen shot of a portlet used in the implementation of the Exclusive Choice Pattern

The implementation of this workflow pattern is similar to that of the parallel split workflow pattern described previously. The user is provided with a graphical mechanism through which they can select the appropriate branch to follow. An example of this visual instrument can be seen in figure 5.30. This instrument is implemented as a WSRP portlet that is delivered directly from the WSBPEL process. That is, the WSBPEL process contains the code for the WSRP portlet that allows the user to choose a branch by selecting the appropriate radio button in a traditional web based form. The code for this portlet is based on a template that is modified during the process of auto generating the WSBPEL workflow. This allows the number of possible choices presented by the portlet to be modified so that a choice can be made between any number of possible branches.

When the user submits their choice, a request is sent to the WSBPEL process in the same manner as any other WSRP performBlockingInteration message. When control flow enters an exclusive choice for the first time, the WSBPEL process is, by default, expecting a message containing the parameter for selecting the branch to be taken. Figure 5.31 provides a graphical visualisation of the WSBPEL implementation of the exclusive choice pattern. As can be seen from the diagram, the implementation is similar to that of the sequence pattern. However, in the Exclusive Choice implementation, the first service in the sequence is the management portlet that allows the user to choose which branch to follow. As was the case with the implementation of the other patterns, the selection of which branch is active is carried out using a WSBPEL 'if-else' that selects branches based on the value of a variable. It is this variable that is set based on the user's selection in the management portlet.

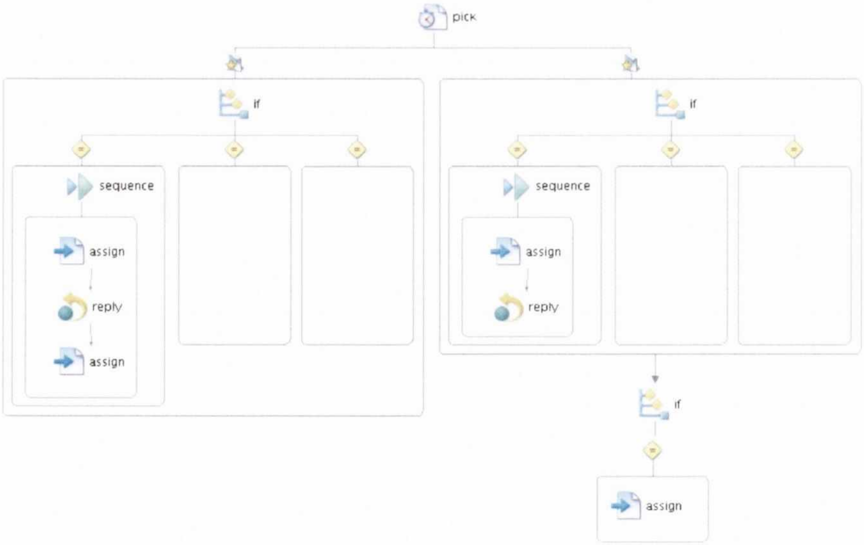


Figure 5.31: Illustration of the WSBPEL code to implement the Exclusive Choice Pattern

The Exclusive Choice Pattern implementation can support the nesting of any of the supported workflow patterns. As was the case for the Sequence Pattern, a dedicated variable in the WSBPEL process is used to control the selected branch of the Exclusive Choice.

5.5 Authoring a Narrative

In order to illustrate how a Narrative Model can be written using the extended Lisp language outlined previously, a simple adaptive system has been defined. Figure 5.32 illustrates a very simple concept space for the SQL subject domain. As shown in the diagram, the concept space consists of six concepts relating to the teaching of SQL.

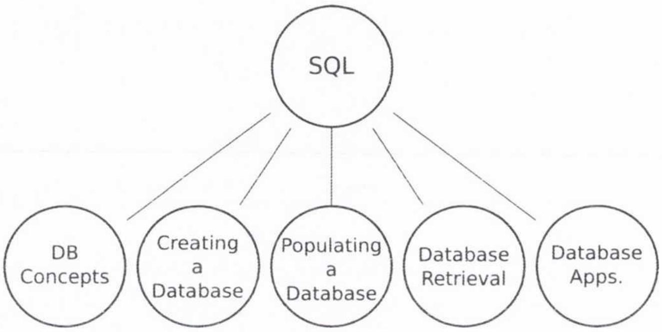


Figure 5.32: Simple Concept Space

As mentioned previously, there is no explicit domain or concept model used in the generation of a PWE. Instead the concept space is a tool used by the designer of the PWE while designing a PWE. During the execution of the system, the 'Concept Model' is implicit in the narrative which explicitly declares the concepts that make up the PWE and the relationships between them.

In this example, the information used to influence the adaptation process comes from a Learner Model. In this case, the learner model contains information about what concepts have been covered by the learner, as such it is a simple binary overlay model. Figure 5.33 shows an example instance of such a user model.

To adaptively select appropriate content, the narrative reconciles the Learner Model with the Content Model. This is only possible because the metadata models share a common vocabulary. By comparing the Learner model in figure 5.33 with the example Content Model in figure 5.34 this common vocabulary can be seen. For example, both models use the same terms for both concepts, e.g. DBConcepts, and educational level.

Figure 5.35 provides an example Narrative Model based on the concept space and metadata

```
<learner>
    <general>
        <name></name>
    </general>
    <competencies>
        <competency level="advanced">DBConcepts</competency>
        <competency level="advanced">CreatingDB</competency>
        <competency level="novice">PopulatingDB</competency>
        <competency level="novice">DBRetrieval</competency>
        <competency level="novice">DBApplications</competency>
    </competencies>
</learner>
```

Figure 5.33: Example Learner Model

```
<content>
    <id>00001</id>
    <file_name>DBConcepts1.html</file_name>
    <concepts>
        <concept>DBConcepts</concept>
    </concepts>
    <level>novice</level>
</content>
```

Figure 5.34: Example Content Model

models presented. As can be seen from the example, the narrative makes extensive use of the standard lisp functions to handle data and to implement the necessary adaptive behaviours. Custom Functions are only used when it is necessary to access metadata models or a specific functionality such as invoking a web service. In the example, the use of Custom Functions is highlighted using a bold typeface. This narrative adaptively sequences the concepts from the concept space based on the learners prior knowledge and then adaptively selects appropriate content to instantiate those concepts. The narrative also adds a service in the generated PWE.

The structure of the example narrative can be broken into 3 parts the first of which (lines 1-8) creates two models, one to store the structure of the PWE that the system generates and the second to store the service composition that is later transformed into a BPEL process. The second part of the narrative, lines 10-70, consists of a set of functions that carry out repetitive tasks such as adaptively adding concepts to the PWE, selecting appropriate content or adaptively adding services. The final part of the narrative, lines 71-102, describe the structure of the narrative in terms of concepts and services and makes use of the functions defined in the narrative to add them to the PWE.

```
(create-model PWE_MODEL)
(update-model PWE_MODEL "pwe")

(create-model COMPOSITION_MODEL)
(update-model COMPOSITION_MODEL "composition")
(add-attribute COMPOSITION_MODEL "name" "SQLActivity")

(create-model SERVICES_MODEL (first (remote-xquery-collection SERVICES_COLLECTION "/")))

(defun content-selector (concept)
    (setq learnerModelQuery
      (concatenate 'string "//competency[.='" concept "']/data(@level)"))
    (setq level (xpath-query-model "learnerModel" learnerModelQuery))
    (setq contentModelQuery
            (concatenate 'string "/content[concepts/concept='"
                                 concept
                                 "' and level='"
                                 level
                                 "']/filename/text()")

    (return (first (remote-xquery-collection "contentModel" query)))
)

(defun add-concept (concept)
    (if
       (member concept
          (xpath-query-model "learnerModel" "/learner/compentencies/competency/text()"))

       (update-model PWE_MODEL "section")
       (add-attribute PWE_MODEL "name" concept)
       (update-model PWE_MODEL "resource" (content-selector "concept"))
       (cd PWE_MODEL ".."))
)

(defun add-service ( name goal guidance )
  (setq PLAN (call-web-service PLANNER_WS_URI
                                        PLANNER_WS_URI
                                        PLANNER_OPERATION
                                        (list DOMAIN_NAME
                                              (model-to-string SERVICES_MODEL) goal)
                                        (list "domainName" "domain" "problem")))

  (create-model "solutionModel" PLAN)
  (setq actions (xpath-query-model "solutionModel" "//action/name/text()") )
  (remove-model "solutionModel")

  (dolist (action actions)
    (setq BASE_QUERY (concatenate 'string "/service[general/name='"
                                          action "']/technical"))
    (setq URI_QUERY (concatenate 'string BASE_QUERY "/getMarkupURI/text()"))
    (setq PORTLET_ID_QUERY (concatenate 'string BASE_QUERY "/portletIdentifier/text()"))
    (setq CONDITION_QUERY (concatenate 'string BASE_QUERY "/exitCondition/text()"))

    (list
      (setq URI
           (first (remote-xquery-collection SERVICE_MODEL_CONNECTION URI_QUERY)))
      (setq PORTLET
           (first (remote-xquery-collection SERVICE_MODEL_CONNECTION PORTLET_ID_QUERY)))
      (setq ENDCONDITION
           (first (remote-xquery-collection SERVICE_MODEL_CONNECTION CONDITION_QUERY)))

      (update-model COMPOSITION_MODEL "service")
      (add-attribute COMPOSITION_MODEL "name" action)
      (add-attribute COMPOSITION_MODEL "type" TYPE)
      (update-model COMPOSITION_MODEL "endpoint" URI)
      (update-model COMPOSITION_MODEL "portlet" PORTLET)
      (update-model COMPOSITION_MODEL "endcondition" ENDCONDITION)
      (update-model COMPOSITION_MODEL "guidance" guidance)
      (cd COMPOSITION_MODEL "..")))
)
```

```
71  (add-concept "DBConcepts")
72  (add-concept "CreatingDB")
73  (add-concept "PopulatingDB")
74  (add-concept "DBRetrieval")
75  (add-concept "DBApplications")
76
77  ;; Add services
78  (setq goal "<problem>
79                 <name>test-problem</name>
80                 <domain>test</domain>
81                 <objects>
82                     <object name='ian' type='user'/>
83                 </objects>
84                 <init>
85                     <predicate name='hasAirportInterest' value='ian'/>
86                 </init>
87                 <goals>
88                     <predicate name='takenDBConceptsQuiz' value='ian'>
89                 </goals>
90             </problem>")
91
92  (update-model COMPOSITION_MODEL "parallel")
93  (add-service "newService" goal1 "guidance")
94  (add-service "newService" goal2 "guidance")
95  (update-model COMPOSITION_MODEL "..")
96
97  (call-web-service "http://localhost:8080/axis2/services/CompositionMapper"
98                    "http://ws.kdeg.cs.tcd.ie"
99                    "urn:toBPEL"
100                   (list LEARNER (model-to-string COMPOSITION_MODEL)
101                   (list "username" "composition")
102 )
```

Figure 5.35: Narrative Model extract

When this narrative is executed by the Adaptive Engine, it will first create the new models as defined in lines 1-8 and then begin to adaptively add concepts to the narrative. Concepts are added to the PWE using the *add-concept* function, lines 71-75, which encapsulates the necessary adaptive behaviour. This function, defined on line 24, adaptively adds concepts to the PWE based on the learners competency in that concept. This is achieved using the *if* function to describe the conditional nature of the concept. The condition itself is implemented by testing whether or not the Learner Model contains the explicitly defined concept. The actual comparison is done through the use of the *member* function, which compares the specified concept against a list of concepts retrieved from the Learner Model. The list of concepts is retrieved from the Learner Model using the *xpath-query-model* custom function. This is a good example of how the functionality of the Lisp environment can be combined with the narrative specific functionality in order to achieve the desired outcome. This example also illustrates the importance of a shared vocabulary across the different models used by the system. Without this, it would not be possible to write such rules.

If the condition is determined to be true then the concept is added to the user's PWE Model. This is achieved by manipulating the model, previously created in the narrative, using the *update-model* custom function to add the appropriate elements to the output model.

The selection of appropriate content to instantiate the concept that has been added to the user's course is handled by the *content-selector* function defined on line 10. In this scenario the selection criteria are very simple so the candidate selector can be realised as a function that is defined as part of the narrative. In more complex scenarios this candidate selector can be implemented separately from the main Narrative Model. This can be achieved either using an additional 'sub-narrative', which can be included dynamically in the execution of the Narrative Model or by a dedicated system with specialised functionality that is completely independent of the Adaptive Engine.

The process of selecting and appropriate piece of content is similar to the sequencing of content. However, when working with the Content Model, it is not desirable to load the entire model into the Adaptive Engine as it is potentially very large. Instead the narrative uses the *remote-xpath-query-collection* custom function to query the XML database using its built in query functionality. If more than one piece of content covering the same concept is discovered then the selector compares the set of discovered resources based on non-functional properties of the content, for example, the suggested expertise required in order understand the content. As can be seen from the Learner Model in figure 5.33, the user is a 'novice' so the content selected by the Adaptive Engine should also be appropriate for that level. The content selection process is a good example of how the Narrative Model reconciles the other metadata models in order to make informed choices based on the needs of the user.

In the example narrative, lines 92-95, two services are added to the PWE, which the learner will be able to access in parallel. To specify that the two services should be part of a Parallel Split, a <parallel> element is added to the composition model, any services that are subsequently added to the composition will be placed inside of this element. The *add-service* function is then called in order to add new services. This function expects three parameters, the name of the service, the goals that the service should achieve and a *guidance* message.

The add-service function is responsible for selecting appropriate services to meet the requirements of the specified goals by invoking the Service Composer, as shown on line 36. The Service Composer is a web service so the narrative can interact with it using the *call-web-service* custom function, which allows the Adaptive Engine to invoke web services and to handle the XML data that they return. The service, or composition of services, that the Service Composer returns to the Adaptive Engine is then used to update the COMPOSITION_MODEL (lines 62-69 of in the example narrative). This requires the Adaptive Engine to get additional information about the service, such as the service URI and WSRP handle. This information is obtained from the Service Model by carrying out a set of XPath queries on the service model as shown in lines 54-60.

The adaptively generated composition of services is then deployed by the Adaptive Engine, line 97, by passing the COMPOSITION_MODEL to the Composition Mapper using its web service interface. As a result of this, the service composition part of the PWE is transformed into a WSBPEL process and deployed to a workflow engine so that it can be accessed by the learner.

5.6 Summary

This chapter discussed how the design, previously presented in the chapter 4, for a system capable of adaptively generating and delivering Personalised Web Experiences was realised as a set of software components and related information models. A technical discussion of how the implemented system operates was provided with a specific focus on how the individual software components inter operate with each other in order to generate and deliver a PWE. Each component was then described in detail with specifics of the technologies used to implement the component as well as how the core functionalities of the component were achieved. As part of the discussion of the system implementation, the metadata models that drive the adaptation process were presented with details of their information models and how these models were captured and stored.

Chapter 6

Evaluation

6.1 Introduction

In chapter 1, the following question was posed: "what are the appropriate techniques and technologies required to support the delivery of personalised web based experiences that combine adaptively sequenced and selected multimedia content with adaptively composed interactive services in a unified manner". Based on this research question, three objectives were identified as follows:

- A literature review of the state of the art investigating the technologies and techniques that can be used for the adaptive selection and composition of content and services for eLearning.
- To iteratively develop and test an integrated adaptive system suitable for the strategically driven composition of multimedia content and interactive services. The developed system will also be capable of delivering the generated compositions to the user.
- A detailed evaluation of the implemented system looking at the complexity, performance and usability of the system.

The first of these objectives was addressed by the analysis of the state of the art presented in chapters 2 and 3 while the second objective was addressed by the design and implementation discussed in chapters 4 and 5 respectively. This chapter focuses on addressing the third and final objective of this research through the presentation and analysis of a series of experiments that were carried out using the implemented system.

In section 4.2 of the Design chapter, a set of requirements were identified based on the objectives of this research, as set out in section 1.3 of the Introduction, and the analysis of the state of the art review. These requirements, listed below, were used as the basis for designing a set of four experiments. The aim of each of these experiments was to address specific aspects of the requirements so that, when taken as a whole, the experiments address all of the requirements that were derived from the objectives of this research.

1. The ability to sequence interactive services using appropriate Control Flow
2. The ability to Adaptively Select multimedia content
3. The ability to Adaptively Sequence multimedia content
4. The ability to Adaptively Select interactive services
5. The ability to Adaptively Sequence interactive services
6. The ability to generate a composition in accordance with a strategy
7. The ability to dynamically compose interactive services where appropriate
8. The ability to generate educationally appropriate activities
9. The ability to deliver compositions to the Learner

Table 6.1 shows which of the requirements are addressed by each of the four experiments carried out.

This chapter is structured as follows, section 6.2 provides an overview of the four experiments carried out. This is followed by a detailed discussion of each of the four experiments in sections 6.3 to 6.6. Each experiment is described in terms of the experimental procedure, the results obtained and an analysis of those results. The findings from the four experiments are then summarised at the end of the chapter.

6.2 Outline of Experiments

To evaluate how well the system implemented addresses the objectives that were drawn from the research question, a set of four experiments were designed and carried out. Each of these experiments focused on evaluating specific aspects of the system with respect

Requirement	Experiment One	Experiment Two	Experiment Three	Experiment Four
Sequence interactive services using appropriate Control Flow	X			
Adaptively Select multimedia content			X	
Adaptively Sequence multimedia content			X	
Adaptively Select interactive services		X		
Adaptively Sequence interactive services		X		
Generate a composition in accordance with a strategy			X	
Dynamcically compose interactive services where appropriate		X		
Generate educationally appropriate activities	X			
Deliver compositions to the Learner			X	X

Table 6.1: Mapping between requirements and experiments

to the research objectives. The experiments one, two and four are focused entirely on evaluating the compositional attributes of the engine with experiment three being the only experiment carried out that was intended to be educationally coherent.

Experiment One The first experiment, discussed in section 6.3, was designed to evaluate the representational power of the supported control flow patterns with respect to eLearning activities. As such it's aim was to evaluate the system's ability to support a wide range of eLearning activities based on the supported control flow patterns, these aims correspond to requirements 1 and 8. The methodology for this experiment involved the analysis of the LADiE use cases, discussed previously in the State of the Art chapter on Adaptive Content. These represent a set of independently developed use cases. The purpose of this analysis is to identify the workflow patterns that are necessary to sequence the appropriate services in each of the use cases. The necessary workflow patterns identified in this analysis were then compared with the patterns supported by the system. The purpose of this comparison was to identify how many of the use cases could be supported as well as the control flow patterns that would be necessary to implement the unsupported use cases.

Experiment Two As outlined, the first experiment was focused on evaluating the coverage that the supported workflow patterns provide with respect to the sequencing of services as part of realistic eLearning activities. However, it did not evaluate the ability of the system to adaptively sequence or select interactive services. Therefore the second experiment was designed to evaluate the system ability to adaptively select and sequencing services (requirements 4 and 5) as well as the ability of the system to dynamically compose services as per requirement 7. The methodology for this experiment was to implement 3 PWE scenarios based on the independently developed use cases from experiment one. These use cases were extended to incorporate both adaptive selection and sequencing of services as well as to make use of dynamic service composition. The implemented use cases were then run through a series of tests designed to exercise the required functionalities. The PWE models generated by the system based on these tests were then compared against each other to identify whether the correct adaptations were carried out.

Experiment Three The aim of the third experiment was to investigate the implemented system's support for the composition of an eLearning course through the adaptive selection and sequencing of multimedia content in accordance with a strategy or 'Narrative' as well as the delivery of that composition to learners. As such, this experiment addresses requirements 2, 3, 6 and 9 of system, evaluating how well they have been met by the system. It was designed to evaluate the systems ability to adaptively sequence content based on a Learner Model and in accordance with a Narrative Model. In addition, the evaluation looked at the systems ability to adaptively select appropriate content based on a Learner Model and finally it looked at the usability of the system from the perspective of the learner. The methodology used in this experiment was to develop a personalised eLearning course, using the Adaptive Engine that was developed as part of the PWE system, this course was then given to a group of undergraduate students who subsequently completed a user study questionnaire. This experiment evaluated the system based on how well the participants perceived the appropriateness of the content selected by the system as well as the overall sequencing of the selected content.

Experiment Four The fourth experiment, detailed in section 6.6, aimed to investigate the performance of the system during both the generation of a PWE and the runtime interactions of the end user with the system. As such it addressed requirement 9, the ability of the system to deliver PWEs to the learner. The experiment involved the benchmarking of the system through the execution of a set of simulated user interactions with the system to test both performance and scalability. The simulated user interactions represented a prerecorded set of real user interactions with the system, which are then used to load test the system with increasing numbers of users. As part of the load testing experiments, page load times as well as server performance metrics were recorded and subsequently analysised with respect to the usability of the system as the number of users increased and the potential for the system to be scaled to cope with larger numbers of users.

6.3 Experiment One - Control Flow Between Services

The aim of this experiment is to evaluate the expressive power of the 5 workflow patterns that the system supports for the specification of eLearning activities. As such, this experiment aims to evaluate requirements 1 and 8, namely the support of control flow between services and the support for eLearning activities. Unfortunately there is no comprehensive standard test set of eLearning activities against which to evaluate this system. Similarly, it is not possible to define a definitive list of activities based on a literature review as any list would still consist of arbitrarily selected activities, it would always be possible to conceive of new activity sequences.

To work around this limitation, the methodology for this experiment is instead based on the analysis of a set of use cases that have been developed independently of this research. The selected set of 16 use cases were developed by eLearning practitioners as part of the nationally funded LADiE project, discussed previously in the state of the art chapter on Adaptive Learning (Chp. 2), which was specifically tasked with capturing such learning activities. As part of this experiment, the LADiE use cases are analysed in order to identify the workflow patterns that would be necessary to sequence the appropriate services in each. The identified workflow patterns are then compared with the patterns supported by the system in order to evaluate whether or not the supported workflow patterns provide sufficient flexibility for the definition of eLearning activities. For each of the use cases it is then possible to identify whether or not the use case is supported by the available control flow patterns as well is identifying if any additional control flow patterns are necessary. The overall level of support for the use cases afforded by the supported patterns will provide a strong indication as to their expressive power when it comes to the specification of eLearning activities.

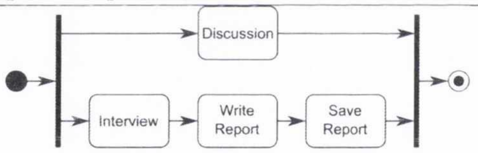
As the LADiE use cases can only be considered as a representative sample of the types of learning activities developed by eLearning practitioners and not an exhaustive list of all possible eLearning activities, the results of this experiment can only be taken as an indication of the flexibility and/or limitations of the system. Furthermore, as discussed in the state of the art chapter, the LADiE use cases show a bias towards assimilative, communicative and information handling type tasks with less coverage of other types of task. As such the results obtained from this experiment must be interpreted with this bias in mind.

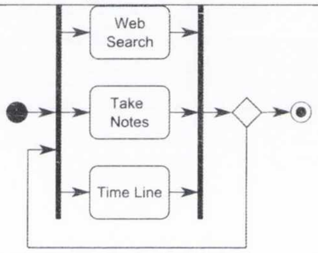
6.3.1 Methodology

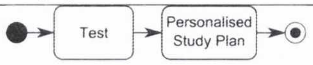
As defined by the LADiE project, the use cases incorporate the sequencing of tasks that must be carried out not only by the student but also the teacher and in some cases, support staff. In order to use these use cases as part of this evaluation, it is first necessary to represent the use cases entirely from the perspective of the student as the tasks carried out by the teacher and support staff are out of the scope of the system implemented. This process is quite straightforward as the actor involved in each task in a LADiE use case is explicitly specified. However, the LADiE use cases are written as a sequence of tasks and do not explicitly refer to the use of any other control flow constructs. Instead, the more complex control flow required by an individual use case is often provided simply in the description of a single task. For example, the first LADiE use case specifies that a discussion activity should run in parallel with two other activities by describing the discussion task as "Students... discuss the activity throughout steps 6-7". However, in some cases the descriptions of the tasks do not provide enough information from which to extrapolate the necessary control flow for the activity. In these cases it was necessary to apply some common sense in order to identify appropriate control flow that would be necessary to realise the activity based on the available information. As a result, such activities can be considered to be interpretations of the original LADiE use cases and are not definitive representations of them. The original definitions of the LADiE use cases can be found in appendix A.1

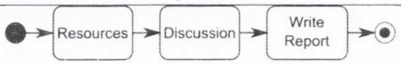
This process was carried out for each of the LADiE use cases and is described below. For each use case, the student centric tasks were identified based on the original descriptions of the tasks provided. A table was then produced for each of the modified use cases that specifies the tasks from the original use case as well as the description of each of those tasks.

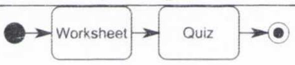
Based on this reduced set of tasks, a UML activity diagram was produced in order to explicitly define the control flow used in the activity. It is the control flow described in these diagrams that will be used for the purpose of evaluating the functionality of the system implemented.

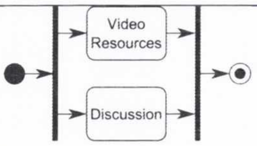
Use Case 1	
Summary: An interview based activity in which the student(s) carries out an interview with a third party and writes a report based on the interview. Use case consists of 11 tasks that are intended to be carried out by the student(s). The first task specified is a discussion task, which the use case specifies should be available to the student(s) while they take part in the other tasks in the use case. The other tasks are the interview task, the report writing task, and the report save/submission task. The appropriate control flow was identified as a sequence of tasks (interview, report writing and report submission), which is executed in parallel with the discussion activity.	
1	Students discuss activity throughout steps 2-4
2	Students carry out interview and record them as audio files
3	Students save interview recordings to system
4	Students write a report or presentation
	

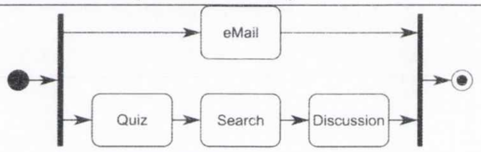
Use Case 2	
Summary: A search activity similar to a Webquest. Students search for information on historical events either online or in a content repository and plot those events on a timeline. Five of the eleven tasks specified in the original definition are for the student to complete with three services needed to implement the tasks: namely, a search service, note taking/bookmarking service and a time line service. Students need to have access to all of these services in parallel and as such this use case requires a parallel branch and simple merge to implement it. The use case also specifies that the students repeat the activity multiple times during their course. To achieve this it is also necessary to use a loop based control flow.	
1	Students search for and evaluate information, make notes and record references
2	Students record notes and references to system
3	Students access events in timeline, evaluate notes and enter comments and additional notes
4	Students interrogate timelines by aggregating two or more lines or by listing events as a timeslice
5	Students repeat steps 1-4 throughout course
	

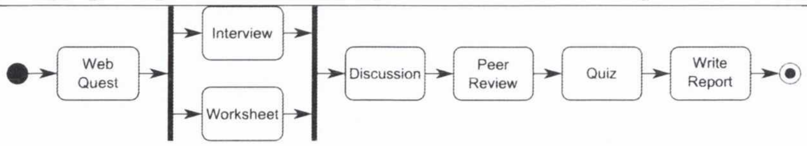
Use Case 3	
Summary: This is a pretest based scenario in which the student takes a pretest or quiz, which is followed by a personalised study plan based on their results in the pretest. This is a simple use case with only two tasks that the student takes part in, a test task followed by a study plan task. This use case can be implemented using a sequence workflow as illustrated.	
1	Students take a test
2	Students work through study plan based on test result
	

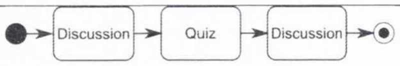
Use Case 4	
Summary: A discussion activity in which students are provided with a set of resources that they must discuss and write a report about. Three of the tasks from the original LADiE use case description have been identified as student tasks. The resource access aspect of the use case has been omitted from this analysis as it will be evaluated as part of experiment three, which focused on adaptive selection of content. By doing so only two tasks are left, the discussion and the write report tasks. In addition to these services it is also necessary to provide a service to support the saving of the report as specified in the original use case definition. As such, this use case includes three tasks that are carried out in sequence with the discussion followed by writing the report and finally the saving/submission of the report.	
1	Students access resources
2	Students discuss problem
3	Students write report and save to system
	

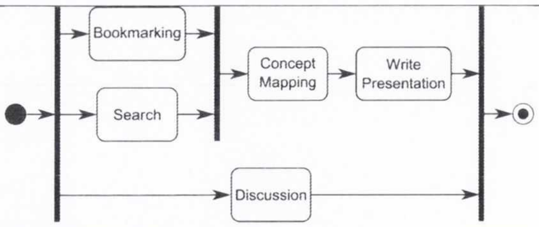
Use Case 5	
Summary: The student is provided with a worksheet on a topic followed by a diagnostic quiz, which tests their understanding of the topic(s) that the worksheet covered. This is a simple activity consisting of only two tasks, a worksheet task and a quiz task. To implement each of these tasks it is only necessary to have one service each, which are delivered in sequence.	
1	Students work through worksheet, do quiz and receive feedback
	

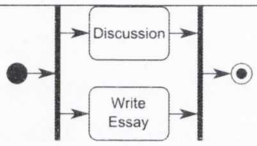
Use Case 6	
Summary: A critical thinking exercise in which students review a set of resources (in the LADiE use case description these resources are videos) and discuss the videos based on a framework that they have been provided with previously. The delivery of this framework can be supported by the content adaptation aspect of the system so the focus of this use case will be on the video and discussion. As described in the original use case description, the sequencing of student tasks is not explicitly defined and is open to interpretation. A simple approach would be to provide the video task followed by the discussion task, however the activity could also be implemented using a parallel split so that the students have access to the videos while they take part in the discussion. The latter approach would allow the students to review the videos based on the comments of other users and would be a more interesting use case from a technical perspective.	
1	Students access video clips and briefing material
2	Students send contributions to discussion forum
	

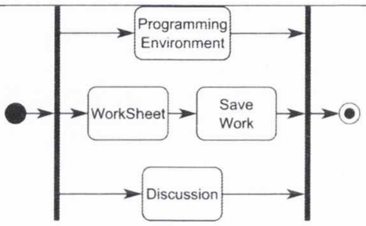
Use Case 7	
Summary: This use case first requires the delivery of a presentation to the students, in an online scenario this could be achieved through the use of a screencast of the presentation. Following this, the students are required to take part in a concept mapping task based on the presentation. This is then followed by a sequence of tasks consisting of a quiz, search and discussion task. During the quiz, search and discussion tasks the students are also able to communicate with their teacher using email. This activity can be realised using a combination of sequence, parallel split and simple merge workflow patterns. Two things to take note of in the interpretation of the use case is that a bookmarking service is included as well as a search service in order to realise the search task. This seems logical as the students need to be able to keep track of the information that they discover through the search service. Secondly, the search, bookmarking and discussion services are made available to the student in parallel. Again, although this is not specified in the use case, it would seem to be logical to provide the services in this way rather than providing the search task and discussion task in sequence.	
1	Students and teacher correspond by email throughout steps 2-4
2	Students do diagnostic quiz
3	Students search for resources online
4	Students discuss resources asynchronously
	

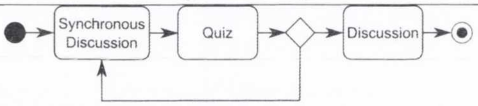
Use Case 8	
Summary: This is a relatively complex activity involving a range of tasks that the student must take part in. The first task is a Webquest task in which the students research a specific topic. This is followed by an interview task where the students interview a domain expert and use the information obtained to complete a worksheet. The interview task is then followed by a discussion about the students findings. The next task in the activity is a peer review task in which the students evaluate each other based on their contributions to the discussion. The students then complete a quiz task and finally produce a portfolio based on the Webquest, interview and quiz answers.	
1	Students take part in Webquest
2	Students carry out interview
3	Students enter and save worksheet
4	Students discuss findings in discussion forum
5	Students evaluate peers based on their contributions to the forum
6	Students do multiple choice quiz
7	Students prepare portfolio of Webquest, interview and quiz answers
 Web Quest Interview Worksheet Discussion Peer Review Quiz Write Report	

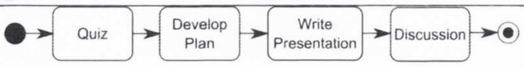
Use Case 9	
Summary: A discussion based activity consisting of three tasks, in which the students take part in a discussion about a topic followed by a diagnostic test on that topic. This is then followed by a discussion of the students results from the test and any issues that were raised in the previous discussion. This is a straight forward use case to implement consisting of a sequence of discussion and quiz activities.	
1	Students discuss course and study skills issues in asynchronous conference
2	Students do diagnostic test
3	Students discuss results of test and issues raised on asynchronous conference
	

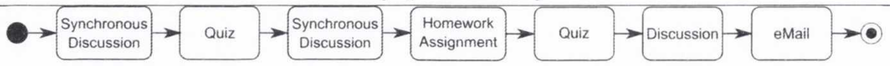
Use Case 10	
Summary: A project planning activity in which the students research the subject of the project using a Webquest style task, then organise their ideas in a concept mapping task and finally write a report on their plans. During all of these activities students have access to a discussion facility that allows them to discuss their findings. As shown in the activity diagram, the search activity is implemented as a parallel split providing access to both a search service and a bookmarking service at the same time. This is part of a sequence of services that includes a concept mapping service and a presentation writing service. This sequence of services is part of an outer parallel split that allows the students to access the discussion service as they utilise the other services in the activity.	
1	Students discuss their findings in the group wiki throughout steps 2-4
2	Students locate and record resources from web, etc.
3	Students use concept mapping software to map links between resources
4	Write up and present conclusions
 Bookmarking Search Concept Mapping Write Presentation Discussion	

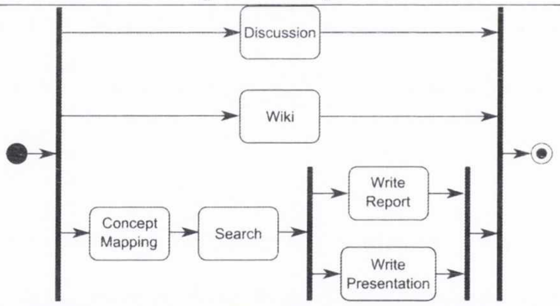
Use Case 11	
Summary: A reflective essay writing activity in which the students write an essay and discuss/reflect on the task. The students then submit their essays and the teacher provides feedback on the submitted essays. As shown in the activity diagram, this activity can be realised using a parallel split workflow pattern, allowing the students to have access to the discussion service while at the same time writing their essay and submitting it. The essay writing service and the submission service are delivered in sequence so that the students are only provided with the submission service when they have finished writing their essay.	
1	Students use forum for discussion throughout activity
2	Students write essay
3	Students submit/send essay
	

Use Case 12	
Summary: In this use case the students learn about a computer programming language through the use of a web based environment to complete a worksheet. In the original LADiE use case definition the activity is based on the Maple symbolic computational language but it could also be used with many other languages. As outlined below the revised task list for the activity consists of a programming task, a worksheet task, a save/submission task and a discussion task. Although the LADiE use case definition does not specify which tasks should be carried out in parallel the use case has been interpreted, as shown by the activity diagram, as making use of a parallel split so that the programming environment, discussion and worksheet tasks can be accessed in parallel by the student.	
1	Students experiment in SCP environment[1]
2	Students discuss exercise
3	Students fill in exercise sheet
4	Students save worksheet
 Programming Environment WorkSheet Save Work Discussion	

<table>
<tr><th colspan="2">Use Case 13</th></tr>
<tr><td colspan="2"><b>Summary:</b> Students are provided with introductory content and an open ended question, which they must then discuss. Following this discussion the students take part in a multiple choice quiz. Upon completion of the quiz the students are provided with feedback and prompted to repeat the discussion and quiz steps for a set of different scenarios. When the students have discussed and completed the quizzes for all of these scenarios they then take part in a further discussion. As shown in the activity diagram, this use case is quite similar to use case 9 except that it requires the loop in order to support the repetition of the discussion and quiz tasks for each of the scenarios.</td></tr>
<tr><td>1</td><td>Students watch video clip</td></tr>
<tr><td>2</td><td>Students discuss video clip for fixed period of time</td></tr>
<tr><td>3</td><td>Students take multiple choice quiz</td></tr>
<tr><td>4</td><td>Students repeat steps 1-3 for X different scenarios</td></tr>
<tr><td>5</td><td>Students discuss additional questions</td></tr>
<tr><td>6</td><td>Students contribute further thoughts in asynchronous discussion</td></tr>
<tr><td colspan="2">
</td></tr>
</table>

<table>
<tr><th colspan="2">Use Case 14</th></tr>
<tr><td colspan="2"><b>Summary:</b> Use case 14 is designed to allow students to develop their presentation skills through critical reflection. The first task is for the students to complete a self assessment quiz. Based on their performance in this quiz they then develop a plan for how they can improve their presentation skills, which they then present to the group. Following this the students develop a mini presentation. During this task they can discuss their progress and show their presentations to the group. When they have completed their mini discussion they present it to the group. The final tasks in the activity consist of the students taking part in a discussion to identify criteria for peer assessment of their presentations after which they give feedback based on these criteria and finally submit their presentations. This use case is primarily a sequence of tasks with the exception of the presentation writing task which is carried out in parallel with a discussion task as shown in the activity diagram</td></tr>
<tr><td>1</td><td>Students use quiz to identify their strengths and weaknesses</td></tr>
<tr><td>2</td><td>Students develop a plan for improving their presentation skills and present this to a group</td></tr>
<tr><td>3</td><td>Students and tutors give feedback on presentation</td></tr>
<tr><td>4</td><td>Students and tutors reflect on what they have learnt</td></tr>
<tr><td>5</td><td>Students use group discussion to develop mini presentations throughout steps 6-8</td></tr>
<tr><td>6</td><td>Students present mini-presentation, receive feedback and develop presentation further</td></tr>
<tr><td>7</td><td>Students discuss criteria for peer assessment of presentations</td></tr>
<tr><td>8</td><td>Students present final mini-presentation, give feedback according to assessment criteria</td></tr>
<tr><td>9</td><td>Students submit presentation to teacher</td></tr>
<tr><td colspan="2">
</td></tr>
</table>

Use Case 15	
Summary: A quiz activity designed to encourage the students to reflect on their knowledge. The students take part in a discussion on a topic followed by a multiple choice quiz. After receiving their results from the quiz the students take part in another discussion about the topic and the feedback from the quiz. Next the students discuss an assignment before completing that assignment. This is then followed by the students retaking the multiple choice quiz, continuing their discussion and finally they email their teacher about any points that have been raised in the discussion. As shown in the diagram this activity can be implemented as a sequence of tasks without the need for any branching.	
1	Students discuss their strengths and weaknesses in subject knowledge
2	Students do quiz
3	Students discuss subject knowledge and quiz feedback
4	Students synchronously discuss homework
5	Students do homework assignment and save work to system
6	Students retake quiz
7	Students continue discussion of points arising on discussion forum
8	Students email teacher about any points they want covered in next lesson
 Synchronous Discussion Quiz Synchronous Discussion Homework Assignment Quiz Discussion eMail	

<table>
<tr><td colspan="2"><b>Use Case 16</b></td></tr>
<tr><td colspan="2"><b>Summary:</b> The aim of this use case is to support students in collaboratively working on a design project. As such, they are provided with discussion and wiki services throughout the lifespan of the activity so that they can collaborate and record necessary information in the wiki. To support the design process the students are first provided with a concept mapping service to help then to arrange the structure of their project. Following this task they take part in a Webquest task to find appropriate resources for the project, which are stored in the wiki. The final task in the activity is to produce a report and presentation about the project. As can be seen from the activity diagram, the discussion and wiki services are made available throughout the activity through the use of a parallel split so that the user can take part in these tasks while they progress through the concept mapping, search and authoring tasks.</td></tr>
<tr><td>1</td><td>Students use concept mapping service and group discussion space to devise a structure for organising their work and resources on a wiki and to generate search terms</td></tr>
<tr><td>2</td><td>Students create directory structure on wiki</td></tr>
<tr><td>3</td><td>Students locate and record resources from the web, etc.</td></tr>
<tr><td>4</td><td>Students discuss the resources in discussion area</td></tr>
<tr><td>5</td><td>Students prepare a reflective report and presentation</td></tr>
<tr><td colspan="2">
</td></tr>
</table>

6.3.2 Results and Analysis

16 use cases were directly derived from the corresponding LADiE use cases as they were originally specified in that project. For each of the use cases, the original description has been analysed in order to identify the tasks in which the learner plays a role. Based on the identified tasks, an activity diagram was generated to illustrate the sequencing required to realise the activity as described by the revised specification. As the activity diagrams represent the control flow between the services necessary to realise each of the use cases, it was possible to identify the workflow patterns necessary to realise each use case, which have been listed in table 6.2 for comparison. This derivation process was necessary as the use cases, as defined by the LADiE project, did not provide explicit guidance on the sequencing of services instead providing a textual description of tasks.

As mentioned previously, the LADiE use cases were selected as the basis for this experiment based on their authenticity as eLearning activities, which were developed independently of this research by eLearning practitioners rather than more technology driven experts. In the absence of any standard test set of activities that fully represent the characteristics of activities in eLearning the LADiE use case have been used to evaluate to what degree the system's supported control flow patterns can be used to deliver eLearning activities.

In the State of the Art (Chapter 2), an analysis of the LADiE use cases based on the DialogPlus learning activity taxonomy was discussed. This analysis, carried out by the LADiE project, provides a means of evaluating how representative the LADiE use cases are of learning activities in general. Part of this analysis focused on the types of tasks from the DialogPlus taxonomy that were utilised in the LADiE use cases. These task types were:

- **Assimilative** Tasks that involve the processing of information such as reading, viewing or listening
- **Productive** Tasks that involve the generation of an artefact such as writing, drawing, synthesising, etc.
- **Communicative** Tasks that involve interact with others such as discussing, presenting, debating, etc.
- **Information Handling** Tasks that involve handling and/or analysing information such as gathering, selecting, sorting, etc.
- **Experiential** Tasks such as Case Studies, Experiments, Games or Role Play.

- **Adaptive** Tasks that involve simulation or modelling.

For convenience, the results of this analysis are shown in figure 6.1 for more details of this analysis, see section 2.2 of the State of the Art chapter on Adaptive Content. The graph shows the number of occurrences of each DialogPlus task type in the LADiE use cases. As shown the LADiE use cases show a strong representation of the Assimilative, Information Handling and Communicative task types and to a lesser degree the Productive task type. However, Experiential and Adaptive tasks are not well represented at all with only three experiential tasks and no adaptive tasks.

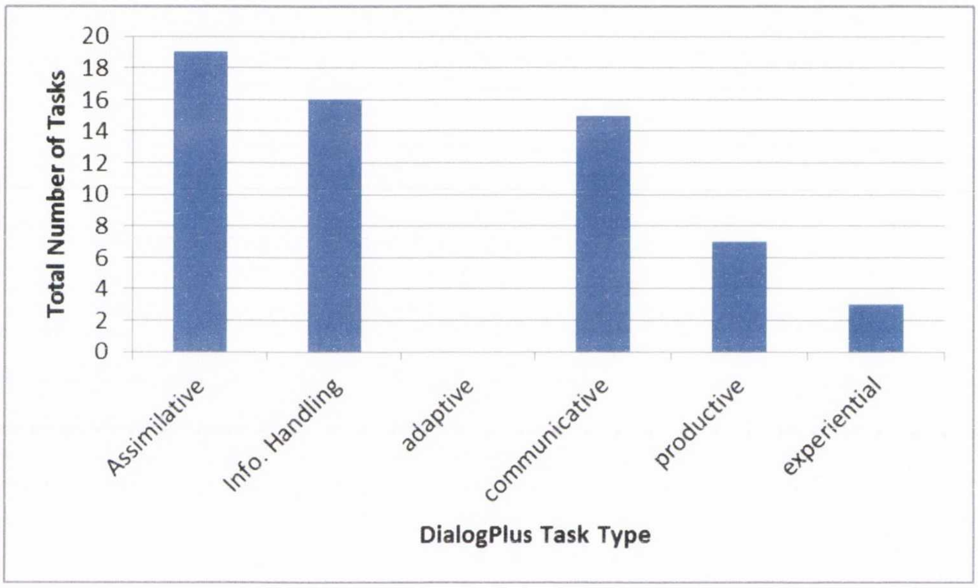


Figure 6.1: LADiE Use Case Gap Analysis - Breakdown of DialogPlus Task Usage

Use Case	Description	Sequence	Parallel Split	Exclusive Choice	Simple Merge	Sync
1	Interview a third party, discuss and write a report	YES	YES	NO	NO	YES
2	Note taking and commenting using time-line	YES	YES	NO	NO	YES
3	Quiz and remedial content	YES	NO	NO	NO	NO
4	Review resources, discuss and write a report	YES	NO	NO	NO	NO
5	Work through a worksheet and take a quiz	YES	YES	NO	NO	YES
6	Review resources and discuss	YES	YES	NO	NO	YES
7	Search for resources online, discuss findings, correspond with teacher	YES	YES	NO	NO	YES
8	WebQuest, interviews, worksheets and discuss	YES	YES	NO	NO	YES
9	Discussion followed by Quiz and further discussion	YES	NO	NO	NO	NO
10	Resource discovery, concept mapping, discussion, write and present report	YES	YES	NO	NO	YES
10a	Comment on findings from main task and commenting on other groups	YES	YES	NO	NO	YES
10b	Write report and presentation based on use case 10 and 10a	YES	YES	NO	NO	YES
11	Reflective essay writing, discussion, writing and feedback	YES	YES	NO	NO	YES
12	Use of simulation (programming env.), worksheet and discussion	YES	YES	NO	NO	YES
13	Review resources, discussion based on questions, quiz	YES	YES	NO	NO	YES
14	Use discussion forum to improve presentation skills	YES	YES	NO	NO	YES
15	Discussions, Quizzes, homework assignment	YES	NO	NO	NO	NO
16	Group collect resources on web, discuss and store in wiki, write a report	YES	YES	NO	NO	YES

Table 6.2: Supported workflow patterns required to implement LADiE use cases

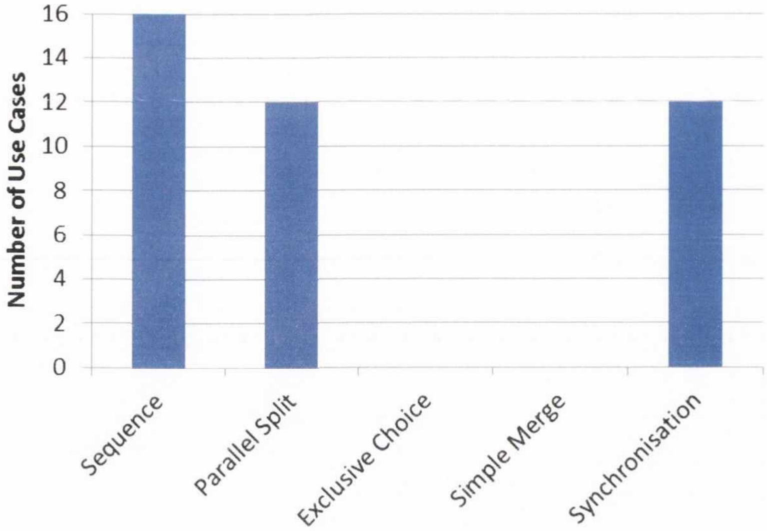


Figure 6.2: Supported workflow patterns required to implement LADiE use cases

The first thing to note is that all 16 of the use cases make use of the Sequence pattern as shown in table 6.2, which lists the workflow patterns necessary to implement each use case from the set of 5 supported patterns. This is not surprising as a linear progression of tasks would seem to be the most basic requirement for any activity consisting of multiple tasks. Looking at the use of the Parallel Split workflow pattern, we can see that 75% of the use cases require this pattern. Similarly, the same set of use cases utilise the Synchronisation pattern. The correlation between the Parallel Split and Synchronisation patterns is to be expected as they are inherently related to each other.

From this table it is clear that none of the use cases require the use of either the Exclusive Choice or Simple Merge patterns. This is because none of the use cases offer choices to the user as to the path they take through the activity. In some cases, such as use case 3, the content delivered to the user as part of the activity is adapted based on the results of a test or quiz but the same functionality is not applied to services.

As pointed out in the LADiE Project final report [Jeffery 06], the LADiE use cases could not be all encompassing and may only reflect the practices of educators in specific domains. As such, this result should not be taken as an indication that the Exclusive Choice and Simple Merge workflow patterns are not necessary but only that they are not necessary for the LADiE use cases.

Table 6.3 shows which of the 16 use cases can be implemented by the system using the

Use Case	Description	Support	Notes
1	Interview a third party, discuss and write a report	Full	All 3 required patterns supported
2	Note taking and commenting using time line	Partial	Basic structure supported through Parallel Split but requires loop pattern for full support
3	Quiz and remedial content	Full	Only requires sequence pattern
4	Review resources, discuss and write a report	Full	Only requires sequence pattern
5	Work through a worksheet and take a quiz	Full	Only requires sequence pattern
6	Review resources and discuss	Full	Supported through use of Parallel Split and Simple Merge
7	Search for resources online, discuss findings, correspond with teacher	Full	Supported through use of Sequence,Parallel Split and Simple Merge
8	WebQuest, interviews, worksheets and discuss	Full	Supported through use of Sequence,Parallel Split and Simple Merge
9	Discussion followed by Quiz and further discussion	Full	Only requires sequence pattern
10	Resource discovery, concept mapping, discussion, write and present report	Full	
10a	Comment on findings from main task and commenting on other groups	Full	
10b	Write report and presentation based on use case 10 and 10a	Full	
11	Reflective essay writing, discussion, writing and feedback	Full	Supported through use of Parallel Split/Simple Merge
12	Use of simulation (programming env.), worksheet and discussion	Full	
13	Review resources, discussion based on questions, quiz	Partial	Basic structure supported through use of Sequence pattern but requires loop pattern for full support
14	Use discussion forum to improve presentation skills	Full	Only requires sequence pattern
15	Discussions, Quizzes, homework assignment	Full	Only requires sequence pattern
16	Group collect resources on web, discuss and store in wiki, write a report	Full	

Table 6.3: Level of support for each of the LADiE use cases using the 5 workflow patterns

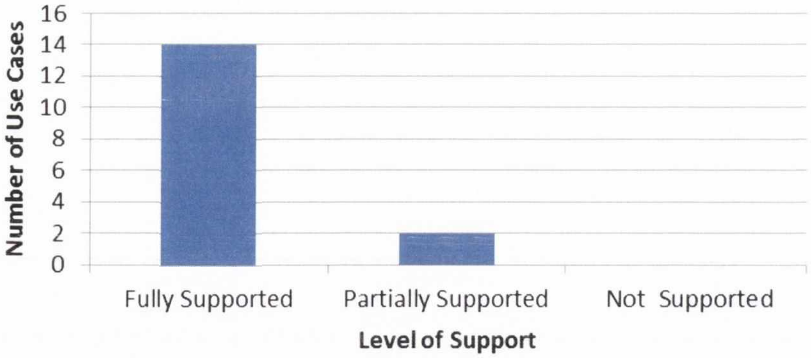


Figure 6.3: Bar chart illustrating the level of support for each of the LADiE use cases based on the 5 workflow patterns

5 supported workflow patterns, this information is also visualised in figure6.3. Each of the use cases has been graded into one of three categories: Fully Supported, Partially Supported and Not Supported. Each of these categories has been defined as follows:

- **Fully Supported** All workflow patterns necessary to implement an activity are supported.
- **Partially Supported** Required services can be composed into coherent blocks but complete definition of use case is not possible.
- **Not Supported** Not possible to support use case either in it's entirety or partially.

As shown 14 of the 16 use cases can be supported fully by the system while 2 of the user cases can only be partially supported. This is because, in addition to the 5 workflow patterns supported by the system, use cases 2 and 13 require the use of additional patterns that are not currently supported. In the cases of both of these use cases it is possible to support the main structure of the activity, however both require some form of loop control flow, such as the Structured Loop pattern [Russell 06] in order to be realised. It is still possible to characterise these use cases as partially supported because in both use cases the loop pattern is used to allow the students to repeat the activity a number of times. Without support for the loop pattern it is still possible to implement an activity that allows the students to carry out the task once.

Taking the LADiE use cases as a set of eLearning activities that are representative of the types of activities that are developed by eLearning practitioners and based on the analysis of those use cases carried out as part of this experiment it is valid to argue that the set of 5 workflow patterns supported by the system can be used to implement the a wide range of eLearning activities. This high degree of support for eLearning activities, despite the seemingly narrow and simplistic range of workflow patterns supported by the system, can be attributed to the nature of activities in eLearning. Unlike business process, from which much of the work on workflow patterns stems, eLearning does not make use of many of the complex patterns that have been identified. Although this experiment has shown that there is a need for at least one additional pattern in order to support all of the LADiE use cases, it is difficult to foresee the need for the full set of 42 control flow patterns that van der Aalst identified in his work.

6.3.3 Conclusions

The analysis of the results of this experiment have shown that the system developed, through the 5 workflow patterns that it supports, can support a wide range of eLearning activities. Taking the LADiE use cases as a benchmark, it has been shown that the 5 supported workflow patterns are sufficient to implement 87.5% of use cases. In the two use cases that it was not possible to support a full implementation, it was found that a partial implementation was still possible and that with the support for an additional loop pattern these use cases could also be fully supported.

However, as discussed previously the LADiE use cases, although developed by a community of practice consisting of eLearning practitioners, do not provide a comprehensive test set of learning activities. They are biased towards specific types of tasks as shown by the gap analysis carried out by the LADiE project. Due to this bias in the LADiE use cases it is possible to have a high degree of confidence that the system can support activities that are based on Assimilative, Information Handling and Communicative tasks and to a lesser degree Productive tasks. However, due to the low or non-existent level of representation of Experiential and Adaptive tasks in the LADiE use cases it is not possible to have such a degree of confidence that the system is capable of supporting activities based on such tasks.

6.4 Experiment Two

The aim of this experiment is to evaluate the system's ability to adaptively select and sequence services (requirements 4 and 5 as outlined in the introduction to this chapter) as well as the ability of the system to dynamically compose services as per requirement 7. To achieve this, the experiment provides a full implementation of three adaptive scenarios[2], complete with required narratives, user interfaces, etc., that exercise the various requirements. The scenarios developed were based on LADiE use cases used previously in experiment one. Some of the more complex LADiE use cases, Use Case 8, 12 and 16, were selected and where necessary extended to provide the necessary features for this experiment.

Although the implementation chapter describes in detail the implementation of all of the components of the PWE framework, it does not describe the implementation of a specific adaptive application. To carry out this experiment, it was first necessary to implement such an application to act as a sort of experiment workbench. The technical details of any application specific components of this experimental workbench are provided as part of the experiment methodology.

The LADiE use cases, provide independently developed activities in which the sequencing and in some cases the adaptive behaviours are specified. As such, they provide a good framework for the evaluation of the adaptive selection and sequencing of services. Each scenario requires the use of different subsets of the functionality of the system with respect to the adaptive selection and sequencing of interactive services. In order to create each scenario, a LADiE use case was taken as the basis and then extended where necessary so that the scenario included the functionality that was to be investigated. When looked at as a whole, the scenarios make use of the full range of functionalities supported.

To evaluate whether or not the system was able to support the three requirements investigated by this experiment, the implemented scenarios were run through a series of tests designed to exercise the required functionalities. The PWE models generated by the system based on these tests were then compared against each other to identify whether the correct adaptations were carried out and whether they resulted in the expected adaptations.

[2] The three experimental activities developed are referred to as 'scenarios' rather than use cases to try to avoid confusion between the experimental implementation and the LADiE use case definitions

This experiment is presented as follows, the methodology and evaluation metrics are first described followed by a detailed description of the application implementation for the scenarios. As mentioned previously this includes the web application that was developed along with all of services, metadata models and adaptation rules that were developed in order to implement the scenarios. The methodology section also provides details of how the adaptation process was simulated for each of the scenarios based on a set of different Learner Models. Following this, the results of the simulations are presented and analysed.

6.4.1 Methodology

As mentioned previously in Chapter 2, the LADiE use cases not only describe the sequence of tasks that make up an activity but also how alternative services can be used to implement specific tasks within an activity based on various different adaptive axes, for example time constraints, disabilities of the learner, available infrastructure, etc. As such, they are ideal activities for the purpose of this experiment as they require the same set of requirements, namely the process driven adaptive selection and sequencing of services, as we are interested in evaluating.

The methodology for this experiment first required that an experimental workbench was developed. As mentioned previously this involved the development of an adaptive web application that could be used as the basis for the implementation of three experimental scenarios based on the LADiE use cases. As the workbench provided all of the supporting functionality and prerequisite metadata models, the development of each of the scenarios was only focused on authoring of the Narrative Model that would attempt to implement each of the scenarios.

The three scenarios implemented were based on the LADiE use cases, with modifications made to the use case sequencing in some cases in order to increase the complexity of the control flow or to introduce a requirement for adaptive sequencing as part of the activity. Table 6.4 provides details of the relationships between the three scenarios and the LADiE use cases that they are based on.

When the experimental workbench was in place, the Narrative models for each of the three scenarios could then be implemented. The experimental procedure then consisted of executing each of the scenario Narrative models with varying Learner Models so that

Scenario	LADiE Use Case(s)
One	Use Case 8
Two	Modified Use Case 8
Three	Combination of Use Cases 12 and 16

Table 6.4: Mapping between scenarios and LADiE use cases

the outcome of the adaptation process, the PWE Model, could be compared against each other as a means of evaluating a series of different evaluation criteria.

6.4.1.1 Evaluation Criteria and Metrics

To evaluate the ability of the system to address the second objective of this research, the adaptive selection and sequencing of interactive services, it was necessary to devise a set of criteria that could be used to analyse the implementation of the scenarios as part of this experiment. The aim of these criteria is to identify which aspects of the research objective each of the scenarios makes use and to what degree the system supports the required functionality.

- To what degree did the system support the control flow constructs necessary to implement the learning activities
- To what degree did the system support the use of the necessary adaptation techniques:
  - Adaptive selection of services
  - Adaptive sequencing of services
- Where necessary, did the system dynamically compose services to meet requirements for which a service did not already exist
  - To what degree did the compositions meet the requirements of the task
  - To what degree did the compositions the needs of the learner.

6.4.1.2 Experimental Workbench

To implement the adaptive system, a basic web application was first developed using JSPs. This application was essentially in line with the outline of the portal interface described in section 5.2.7 of the implementation chapter, consisting of a basic login page, user model elicitation page and service presentation/interface page.

In order for the Adaptive Engine and AI Planner components to be aware of the available services, it was necessary to provide a metadata description of each service, in line with the metadata schema described in section 5.3.3 of the implementation chapter. The three use cases required 14 different types of service, listed below, to be available. These services were identified based on both the primary and secondary tasks described in the LADiE use cases. In total 20 service instances were made available to the system, each of which corresponds to one of the service types listed and providing different non functional properties. The metadata descriptions of each of these services can be found in appendix ??.

- Web Search
- Library Search
- Bookmarking
- Note Taking
- Interview
- Worksheet
- Discussion Forum
- Peer Review
- Quiz
- Programming Environment
- Submission
- Wiki
- Concept Mapping
- Document/Report Writing

As the focus of this experiment was not on the user interactions with the system but rather on a functional test of its capabilities, the details of this web application are not described as part of the experiment methodology. The details of the web application, including screenshots of the user interface, Narrative models and related metadata models, are provided in Appendix for reference.

6.4.1.3 Scenarios

Scenario One The first scenario implemented was based on the LADiE use case 8 as described in the previous section. For convenience the activity diagram for this use case has been repeated below, see figure 6.4. As illustrated, this use case requires the use of 3 of the 5 supported workflow patterns:

- Sequence
- Parallel Split
- Synchronisation

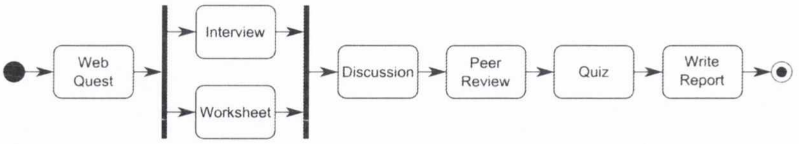


Figure 6.4: Activity Diagram for Scenario One

In addition to demonstrating the ability of the system to support these three workflow patterns, the use case also demonstrates the use of the adaptive service selection mechanism to adaptively select services that are appropriate based on the literacy skills of the learner as well as the learner's ability to access the internet. The requirement for these two adaptive axes is specified in the extensions to the original LADiE definition of use case 8. In addition, this use case also demonstrates the dynamic composition of services. As shown in the use case activity diagram, the first task specified is a Webquest. There is no single service in the set of services made available to the system that can satisfy the requirements of a Webquest and as such, the system will need to dynamically compose a set of services that can be used to realise the task.

The narrative model used to describe the activity and the adaptive behaviours discussed is provided in Appendix 6.23 with appropriate segments presented here for illustration. The fragment of the Narrative Model shown in figure 6.5 contain rules that adaptively specify the literacy skills of the learner as well as other important aspects of the learner such as their access to the internet. These rules are informed by the Learner Model, which is queried using an XPath expression to retrieve the necessary information from the model. By controlling the objects that are declared to exist in the planning problem, the planning process itself can be influenced by the Adaptive Engine so that the most appropriate service or composition of services is selected.

```
(setq LITERACY_XPATH_QUERY
      "//accessibility/language[typename/tyvalue='English']/proficiency[@profmode='Read']/text()")

(setq LITERACY_QUERY_RESULT (first (xpath-query-model LEARNER_MODEL LITERACY_XPATH_QUERY)) )

(if (string-equal LITERACY_QUERY_RESULT "excellent")
    (setq PROBLEM (concatenate 'string PROBLEM "(isLiterate)")))

(setq GOOGLE_XPATH_QUERY "boolean(//securitykey[typename/tyvalue='GoogleAccount'])" )

(setq GOOGLE_QUERY_RESULT (first (xpath-query-model LEARNER_MODEL GOOGLE_XPATH_QUERY)) )

(if (string-equal GOOGLE_QUERY_RESULT "true")
    (setq PROBLEM (concatenate 'string PROBLEM "(hasGoogleAccount)")))
```

Figure 6.5: Narrative Model for Scenario One

6.4.1.4 Scenario Two

The purpose of the second scenario is to demonstrate the use of the Exclusive Choice and Simple Merge workflow patterns in an activity. As shown in the previous analysis of the LADiE use cases, none of the use cases require either of these patterns. In order to demonstrate and evaluate the use of these patterns as part of a realistic activity a LADiE use case was modified to incorporate both an Exclusive Choice and Simple Merge. The activity diagram for this modified use case is shown in figure 6.6. As can be seen from the diagram, the modified activity is based on LADiE use case 8, which was implemented previously in its original form. The use case has been modified so that instead of the learner first taking part in the Webquest and then the interview tasks, they are instead given a choice between either activity. For example, the learner can choose to obtain the necessary information by searching the internet or alternatively by interviewing an expert in the subject domain.

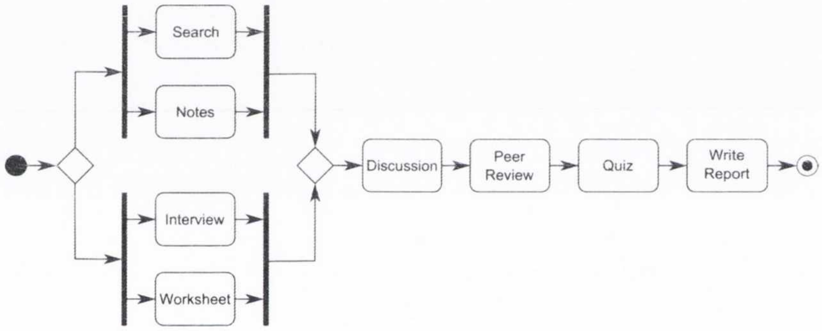


Figure 6.6: Activity Diagram for Scenario Two

6.4.1.5 Scenario Three

The third scenario implemented is designed to demonstrate the use of the adaptive sequencing of services as part of a real world activity. Although the LADiE use cases can easily be modified to make use of adaptive selection of services by taking into account the extensions to each use case there is no such scope for adaptive sequencing within the definitions of the use cases. As such, it was necessary to modify an existing use case so that it incorporates this functionality.

To achieve this, two LADiE use cases have been used as the basis for a single new activity in which the path through the activity is based on the Learner Model. The basis for this activity is LADiE use case 12 in which the learner uses an online programming service to complete a worksheet. During this activity a discussion forum is available to the learner so that they can discuss the activity with their classmates. Although the original LADiE use case was based on the learner working with a SCP[3], it has been modified in this instance so that the SCP environment is replaced with an interpreter for the Python programming language. As specified in the use case, this is a static activity without any adaptive sequencing. To incorporate adaptive sequencing into this activity, the sequencing of task in LADiE use case 12 has been extended to incorporate aspects of use case 16. This results in a more complex activity in which alternative branches exist, which can be adaptively selected by the AE based on the Learner Model. LADiE use case 16 is a project planning activity in which students collaborate on how they will carry out their project and write a report detailing their plans.

By combining these two activities a new adaptive activity has been defined in which inexperienced or intermediate students are presented with a worksheet based activity, as they would in the original definition of LADiE use case 12. In contrast, users that are experienced in the subject matter (ie. the python programming language) are instead provided with a project based activity in which they plan and research a project. In both cases the learners are provided with a discussion service and a programming environment service.

To illustrate these two alternative service compositions, two activity diagrams have been provided, which illustrate the resulting compositions. Figure 6.7 is an activity diagram representing the activity for an inexperienced or intermediate level student while figure 6.8 represents the activity of an expert level student.

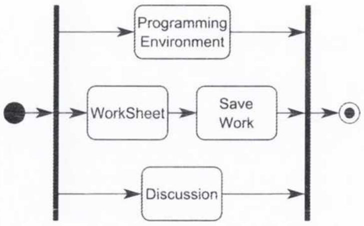


Figure 6.7: Activity Diagram for first possible composition from Scenario Three

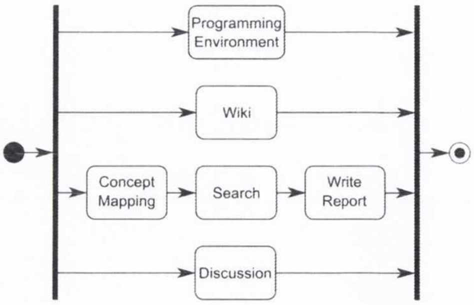


Figure 6.8: Activity Diagram for second possible composition from Scenario Three

[3] Symbolic Computational Program

6.4.2 Results

Based on the evaluation criteria outlined previously, the three use case scenarios have been analysed and the results summarised in table 6.5. In some cases, a scenario is designed to demonstrate a specific functionality, which is not demonstrated by other scenarios. In these cases, scenarios where the evaluation criteria do not apply are marked as *NA*.

Criteria	**Scenario 1**	**Scenario 2**	**Scenario 3**
Support for control flow constructs	Yes	Yes	Yes
Support for adaptation techniques:			
Adaptive Selection	Yes	NA	NA
Adaptive Sequencing	NA	NA	Yes
Generated composition as expected			
Appropriate service selection	Yes	Yes	Yes
Services dynamically composed by Planner	Yes	NA	NA
Was AI Planner generated composition appropriate	Yes	NA	NA

Table 6.5: Breakdown of evaluation criteria as they apply to specific experiment scenarios

6.4.3 Analysis

6.4.3.1 Support for Control Flow

As can be seen from the discussions of the three use cases that make up this evaluation, the full set of 5 control flow patterns supported by the system are not utilised. In fact only *Sequence*, *Parallel Split* and *Synchronisation* are necessary to be able to fully implement the three scenarios. As shown in this evaluation, see table 6.6, it was possible to fully implement the three scenarios using the range of control flow patterns supported by the system. As these three scenarios are exemplary of the type of activities that are contained in the set of LADiE use cases it can be argued that the ability to fully implement them illustrates the systems ability to support the LADiE activities in general.

6.4.3.2 Support for Adaptation Techniques

Adaptive Selection of Services Of the three scenarios implemented only Scenario One employs the adaptive selection of services based on the Learner Model. Of course, through the use of the AI Planner, all service selections are carried out dynamically based

Scenario	Pattern					Level of Suppor
	Sequence	Parallel Split	Exclusive Choice	Simple Merge	Sync	
One	YES	YES			YES	Full
Two	YES	YES	YES	YES	YES	Full
Three	YES	YES			YES	Full

Table 6.6: Patterns required and level of support for implemented scenarios

on the Service Model but these selections are only influenced by the descriptions of the available services in the Service Model and not by the Learner Model. This none adaptive form of service selection is discussed in the next section.

	Adaptively Selected Service	Selection Made	Appropriate Selection
1	WebQuest (a)	Yes	Yes
		Yes	Yes
2	WebQuest (b)	Yes	Yes
		Yes	Yes
3	Discussion	Yes	Yes
		Yes	Yes
4	Write Report	Yes	Yes
		Yes	Yes

Table 6.7: Results for Adaptive Selection of Services in Scenario One

As part of this experiment, the adaptive selection of services was based on two properties of the Learner Model, the learner's literacy skill captured by the **accessibility** element of the Learner Model and the whether or not the learner has a Google account, captured by the **securityKey** element of the Learner Model. Based on the two different Learner Models, provided in appendix 6.4.1.2, two different PWE Models were generated. The services selected by the system to instantiate these PWE Models are summarised in table 6.8.

	User 1	User 2
Web Quest	GoogleSearch, GoogleNotebook	Youtube Search, SimpleBookmarking
Interview	Interview	Interview
Worksheet	Worksheet	Worksheet
Discussion	GmailClient	MyEmailClient
Peer Review	PeerReview	PeerReview
Quiz	SimpleQuiz	SimpleQuiz
Write Report	Google Docs	PresentationTool

Table 6.8: Services selected to instantiate tasks in Scenario One

As shown by the comparison of the two PWEs in table 6.8, by modifying the two properties of the Learner Model the selection of services can be influenced so that appropriate services are selected in each case. Specifically, the Web Quest task is instantiated for test user 1 using two Google services, GoogleSearch and Google Notebook where as for test user 2 two alternative services, the Youtube search service and a simplebookmarking service are selected. This difference is caused by a combination of the two Learner Model properties. Services that require a Google account are not selected for test user 2 because this user does not have a Google account. This user's Learner Model also specifies that they have a poor level of literacy and so they are given a more visual service (Youtube) with which to carry out the Webquest. Another service selection that shows a difference is that of the Discussion task, the GmailClient service is selected for test user 1 while an alternative email client, MyEmailClient service, is selected for test user 2 again because they do not have a Google account.

Adaptive Sequencing of Services Scenario Three is designed to demonstrate the ability of the system to generate an adaptively sequenced PWE. In this case the sequencing of the services that make up the PWE is adapted based on the learner's level of competency in the Python programming language, captured by the **qcl** element of the Learner Model. As shown in figures **??** and **??**, which can be found in appendix 6.4.1.2, the generated compositions have different structures depending on the Learner Model. The sequencing of the tasks in the two activity diagrams is consistent with the two PWE Models generated for the test users (see figures **??** and **??**). The PWE Model for test user 1 is consistent with the activity diagram in figure 6.7 while the PWE Model for test user 2 is consistent with the activity diagram in figure 6.8. This illustrates that different values for properties captured in the Learner Model can result in changes in the sequencing of services in the generated PWE Model based on the execution of the Narrative Model.

6.4.3.3 Appropriate Non-Adaptive Service Selection

Every service incorporated into a PWE must be selected by the system, based on the abstract description of the service required and any information that is relevant from the Learner Model. This means that, the quality of the PWE generated is heavily dependent on the ability of the AI Planner to appropriately select services. In this

User	Composition Generated	Appropriate Adaptation	Notes
1	Yes	Yes	PWE Model was generated by system. Sequencing was adapted for inexperienced learner
2	Yes	Yes	PWE Model was generated by system. Sequencing was adapted for inexperienced learner

Table 6.9: Services selected to instantiate tasks in Scenario One

evaluation the planner was responsible for selecting services for 17 different tasks across the three scenarios. Even in scenarios where the selection mechanism was not influenced by adaptive behaviours the planner itself still introduces a degree of dynamism into the selection process. To be better able to identify whether or not the planner was able to successfully select appropriate services for each of the three scenarios the results of the service selections have been summarised in table 6.10. The first column of the table lists all of the tasks that make up the three scenarios while the second column shows the service that the planner selected from the available services to satisfy the requirements of the corresponding task. The third column indicates whether or not the planner made a selection (this could also be inferred from the second column) irrespective of the correctness of the selection while the fourth column indicates whether or not the selection made by the planner, if any, can be considered appropriate based on the requirements of the task.

As shown in table 6.10, the planner was able to make a service selection for every one of the 17 service selections requests that were made to it. Furthermore, the planner selected a service that could be considered appropriate in each case.

6.4.3.4 Dynamic Composition by AI Planner

Another of the criteria that was investigated was the ability to have the AI Planner dynamically select and sequence services to satisfy the requirements of an individual task where a single appropriate service is not available. This behaviour was demonstrated by Scenario One in which the Web Quest task could not be implemented using a single service, instead requiring the AI Planner to compose services to satisfy the goals specified in the narrative.

By comparing the abstract activity diagram of the scenario in figure 6.9 with an activity

Task	Service Selected	Selection Made	Appropriate Selection
Web Quest	GOOGLESEARCH GOOGLENOTEBOOK	Yes	Yes
Interview	INTERVIEW	Yes	Yes
Worksheet	WORKSHEET	Yes	Yes
Discussion	GMAILCLIENT	Yes	Yes
Peer Review	PEERREVIEW	Yes	Yes
Quiz	QUIZ	Yes	Yes
Write Report	GOOGLEDOCUMENTS	Yes	Yes
Search	GOOGLESEARCH	Yes	Yes
Note Taking	SIMPLEBOOKMARKING	Yes	Yes
Programming	PYTHONINTERPRETER	Yes	Yes
Worksheet	WORKSHEET	Yes	Yes
Save Work	FTPCLIENT	Yes	Yes
Discussion	SIMPLEIRC	Yes	Yes
Wiki	WIKI	Yes	Yes
Concept Mapping	BUBBLUS	Yes	Yes
Search	GOOGLESEARCH	Yes	Yes
Write Report	GOOGLEDOCUMENTS	Yes	Yes

Table 6.10: Services selected to instantiate tasks in Scenario One

diagram representation of the generated PWE for test user 1, figure 6.10, we can see that to satisfy the requirements of the Web Quest task the AI Planner generated a plan consisting of two services, GoogleSearch and GoogleNotebook which run in parallel.

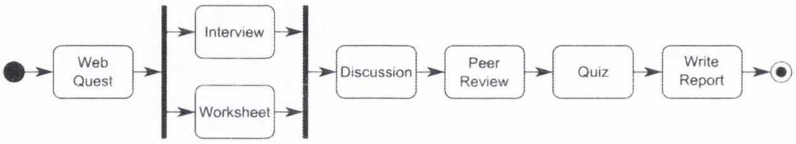


Figure 6.9: Abstract Activity Diagram for Scenario One

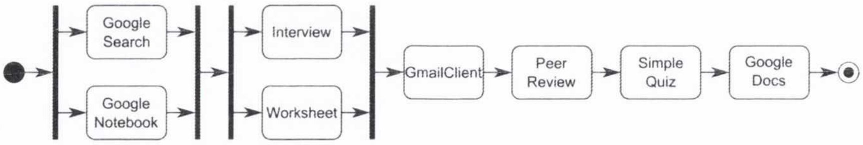


Figure 6.10: Scenario One Activity Diagram representing instantiated PWE

Three aspects of the dynamic composition are important:

- Whether or not a composition is generated
- Whether or not the composition generated meets the requirements (goals) specified for the task

- Whether or not the composition was appropriate for the individual learner.

Findings for htese three factors are shown in figure 6.11. As can be seen from the table, for both learners, the system was able to generate a dynamic conposition consisting of type types of service, a search service and a bookmarking service. Taken together, these services meet the requirements specified in the narrative for a search and bookmarking service. Furthermore, for the learning with a high literacy score and a google account the GOOGLESEARCH GOOGLENOTEBOOK services were selected where as for the learner with a low literacy level and no Google account the YOUTUBESEARCH, SIMPLEBOOKMARKING services were selected.

User	Composition Generated	Composition meets task requirements	Composition meets user requirements	Notes
1	Yes	Yes	Yes	
2	Yes	Yes	Yes	

Table 6.11: Summary of findings for Dynamic Composition of services

6.4.4 Conclusions

This experiment was designed to evaluate 5 key aspects of the system:

- Ability to implement learning activities and to sequence services using the available control flow patterns
- Ability to select services
- Ability to adaptively select services
- Ability to adaptively sequence services
- Ability to dynamically compose services

It was shown that, for the three scenarios used as part of the experiment, it was possible for the system to generate PWEs using the 5 supported control flow patterns. Furthermore, the system was able to select appropriate services using the service composition component (AI planner).

The adaptive behaviours supported by the system were shown to result in appropriate adaptations to the PWE in all cases. As part of scenario one, 8 adaptive service selections were carried out by the system and in all cases these were shown to be appropriate for the individual learners. Similarly, the adaptive sequencing of scenario one was also shown to be appropriate for the learner.

Finally, it was shown that the system wa capable of dynamically generating service compositions to realise tasks for which a service did not exist. Not only did the system respond in such cases and generate such a composition of services but it was shown that the generated compositions were appropriate based on the requirements of the task and that they were appropriate for the individual learners.

This experiment has shown that the system not only provides the necessary functionalities to adaptively select and sequence services in order to compose a PWE but also that the system can bring together these functionalities successfully as part of an integrated system to generate PWEs.

6.5 Experiment Three - Adaptive Selection and Adaptive Sequencing

The aim of this experiment is to investigate the implemented system's ability to deliver coherent and personalised eLearning. As outlined previously, a series of requirements for this system were drawn up based on the objectives of the research, see chapter 6.23 section 6.23. Of theses requirements, this experiment aims to addresses the following five requirements:

- The ability to Adaptively Select multimedia content (Objective 2)
- The ability to Adaptively Sequence multimedia content (Objective 3)
- The ability to generate a composition in accordance with a Strategy (Objective 4)
- Delivery of compositions to the user (Objective 9)

To evaluate these requirements, a user based trail was carried out in which a class of undergraduate students were given access to an adaptive eLearning course built on top of the PWE System. The details of this trial are provided in the following section. In order to focus primarily on the first two requirements listed above, the course consisted only of multimedia content with no interactive services and so only made use of techniques for the adaptive selection and sequencing of content. As part of the user trail the participants were given a questionnaire to complete and the answers that they provided were then aggregated and are provided in section 6.5.2. The aggregated results are divided into three key areas: **Support for Adaptive Behaviours**, **Quality of Content** and **Usability**. An analysis of the results obtained is then presented in section 6.5.3.

6.5.1 Methodology

To facilitate this evaluation, a personalised eLearning course, designed to teach Structured Query Language (SQL) as part of a course on databases, was developed. This course was designed to be delivered using the Adaptive Engine, which is at the core of the PWE system. The structure and content for this course had been used previously for several years both as part of a non adaptive eLearning course and an adaptive eLearning course. The previous adaptive course had been delivered using an older version of the AE and was the subject of several publications [Conlan 02, Brady 04, Conlan 04]. As such the educational soundness of this course have already been validated. The Narrative model and example Learner Model used in this adaptive course can be found in Appendix C.1.1 and ?? respectively while an overview of the user interface can be found in Appendix ??.

The eLearning course allowed a user study to be carried out in which a class of 3rd and 4th year undergraduate students from three degree courses, Computer Science, Computer Science and Linguistics and Computer Engineering, were given access to the adaptive eLearning course. The adaptive online SQL course was the only formally delivered source of information on the subject provided to the students as the topic was not covered as part of the lectures. Following a period of five weeks in which the students had access to the course the students were given an exam on SQL as part of their course work. Subsequently, students from the class were offered the opportunity to take part in the experiment. Participants were asked to complete a questionnaire consisting of 30 questions. The questions were designed to elicit the opinion of the participants on a range of properties of the course such as whether or not the users felt that the course generated by the system accurately reflected the course they wanted and whether the content selected by the system was appropriate. The questionnaire consisted mainly of multiple choice questions although some of the questions asked the users to rate certain aspects of the system. In addition to these questions, the users were also offered the chance to provide additional comments or feedback. The questionnaire form provided to participants in the experiment is provided in Appendix C.2. In addition to this questionnaire, the participants were asked to complete a System Usability Scale (SUS) [Brooke 96] questionnaire, see Appendix C.2.

To illustrate how the questions from the questionnaire correspond to the aims of the experiment, the questions have been grouped in accordance the three areas of interest as well as an additional group of questions used to gauge the participants prior experience, which are shown below:

Prior Experience Questions

- How much experience do you have using online learning resources?
- Are you comfortable learning new course material via the Web?
- How much experience in SQL did you have before commencing the Online SQL Course?

Support for Adaptive Behaviours

- After completing the initial online questionnaire, approximately how many times did you rebuild the course?
- Did the course(s) generated by the system reflect the answers you gave
- Did the course(s) generated by the system reflect the course(s) you wanted?
- Did the course sections contain the content you expected?

Quality of Content

- Were the courses generated easy to navigate?
- Did the course content of the generated course(s) appear disjoint?
- 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)
- Please rate the course sections on how effectively you felt they represented the subject matter.

Usability

- Were the objectives of the generated course(s) clear to you?
- Upon completion of the online course did you feel you had completed the objectives?
- Was the quantity of content on each page satisfactory?
- Would you have found the ability to modify the web interface beneficial? (i.e. placement of buttons, number and type of hyperlinks)

6.5.2 Results

This section presents the aggregated results from the user evaluation along with some comments on the significance of those results. The answers to the individual questions from the user survey are grouped together in accordance with the three evaluation criteria, Support for Adaptive Behaviours, Content Quality and Usability, as outlined previously. The full set of results from the evaluation can be found in appendix C.3, including the results of the questions relating to prior knowledge and experience with elearning, which are not presented in this section.

In total 32 students participated in the evaluation. From these 32 participants it was possible to obtain between 24 and 30 responses for each of the questions asked in the evaluation questionnaire. Although some observations about the results are provided in this section, a more thorough analysis of the results is presented later in section 6.5.3.

6.5.2.1 Support for Adaptive Behaviours

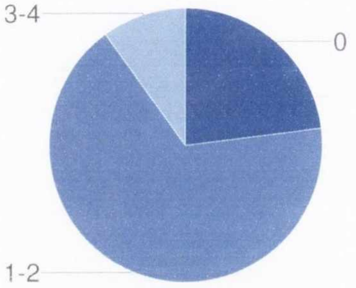
The aim of this part of the evaluation was to identify how well the adaptive behaviours implemented by the system were able to meet the needs of the individual Learners. Four of the questions in the evaluation questionnaire were specifically designed to address this criteria.

How many times did you rebuild the course? The participants in the evaluation were asked how many times they rebuilt the course subsequent to the initial build of their personalised course, which occurred the first time that they logged into the system. Table 6.11a provides the percentage breakdown of the responses of the 30 participants in the evaluation that answered this particular question. This data is also represented visually as a pie chart in figure 6.11b.

As shown, almost a quarter of the participants did not rebuild the course at all while the majority of the participants, 67%, only rebuilt the course a further one or two times. The remaining 10% of the participants rebuilt the course three to four times.

Response	%
0	23
1-2	67
3-4	10
5-6	0

(a) Q4 Table



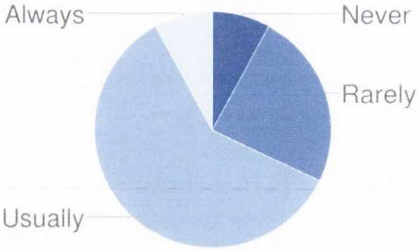
(b) Q4 Graph

Figure 6.11: How many times did you rebuild the course?

Did the course generated reflect the answers given in the online questionnaire? The second question relating to the adaptive behaviour of the system was Q6 in the evaluation questionnaire, "Did the course(s) generated by the system reflect the answers you gave in the online questionnaire". 29 participants responded to this question, the results of which are listed in table 6.12a and visualised as a pie chart in figure 6.12b. As shown, the majority (68%) of participants felt that the personalised course generated by the system was consistent with the answers given by them in the initial course questionnaire.

Response	%
Never	8
Rarely	24
Usually	60
Always	8

(a) Q6 Table



(b) Q6 Graph

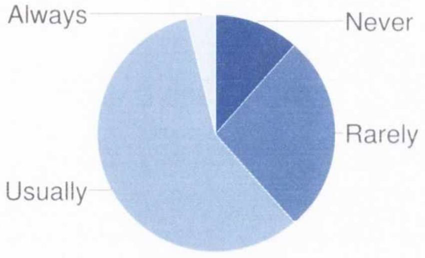
Figure 6.12: Did the course generated reflect the answers given in the online questionnaire?

Did the course generated reflect the course you wanted? Question 7 in the evaluation questionnaire asked the participants if the course generated by the system reflected the course they wanted. As was the case in the answers to the previous question, a similarly large proportion of the participants, 61.6%, responded positively, as can be seen from the results table and corresponding chart, see Table 6.13a and Figure 6.13b respectively. This indicates that the course generated by the system was the course that these participants wanted. Of those that responded negatively, one participant commented that they had previous knowledge of SQL and only really wanted a summary of the syntax rather than the detailed discussion provided by the course content.

Did the course sections contain the content you expected? The final question relating to the adaptive behaviour of the system, question 13 of the evaluation questionnaire, asked the participants if the course sections contained the content that they expected. As shown in table 6.14a, 86.67% of the 30 participants that answered this question responded positively, indicating that the content was appropriately selected by

Response	%
Never	11.5
Rarely	26.9
Usually	57.7
Always	3.9

(a) Q7 Table



(b) Q7 Graph

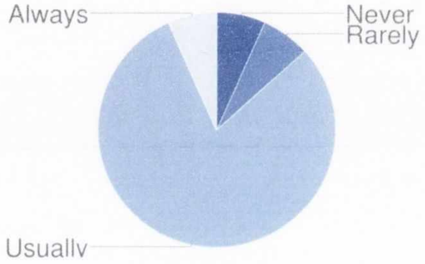
Figure 6.13: id the course generated reflect the course you wanted?

the system.

Interestingly, this is a significantly more positive response compared to the results for the previous two questions. Perhaps this can be explained by the slightly less subjective nature of the question.

Response	%
Never	6.67
Rarely	6.67
Usually	80.00
Always	6.67

(a) Q13 Table



(b) Q13 Graph

Figure 6.14: Did the course sections contain the content you expected?

6.5.2.2 Quality of Content

The second aspect of the system looked at in this experiment was that of content quality. In order to evaluate the quality of the content delivered by the system two issues of content quality were investigated, the appropriateness of the content for a given user and the flow/sequencing of the content as selected and delivered by the system.

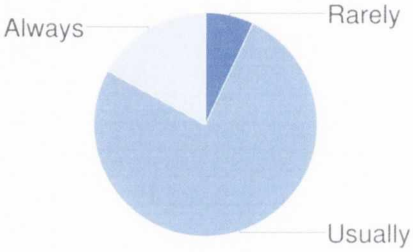
Flow/Sequencing of Content

Were the courses generated easy to navigate? To evaluate the quality of the flow/sequencing of the content, three specific questions were asked of the participants as part of the evaluation questionnaire, the first of these questions, Q10, was "Were the courses generated easy to navigate?". The participant's responses to this question shown in table 6.18a and visualised in figure 6.15b were, for the majority, positive with 93% of the 29 respondents saying that the course was *always* or *usually* easy to navigate.

One aspect of the system that contributed to the negative responses to this question was that some participants had issues with the visual layout of the tabs across the top of the page (see figure **??** in Appendix **??**).

Response	%
Never	0
Rarely	7
Usually	76
Always	17

(a) Q10 Table



(b) Q10 Graph

Figure 6.15: Were the courses generated easy to navigate?

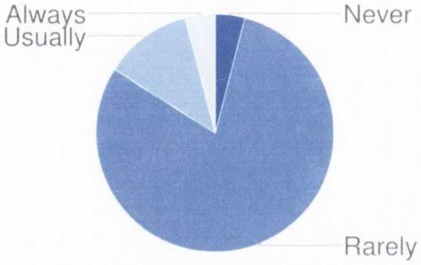
Did the course content appear disjoint? The second question that participants were asked with respect to the quality of the flow of content asked the participants if the course appeared disjoint. As can be seen from the data in table 6.16a, 84% of the 25 participants that responded to this question indicated that it was *rarely* or *never* disjoint in their opinion. These results can also be visualised in the chart shown in figure 6.16b.

Some of the negative responses to this question can be attributed to, for example, participants that were more familiar with SQL and subsequently felt that there was too much repetition in the course content. Where as the participants that were less familiar with SQL also felt the content to be disjoint because in some areas a topic from later in the course was intentionally highlighted.

Unfortunately, the one participant that indicated that the course was *always* disjoint did not provide any comments as to why they felt this was the case. However this participant did comment that they did not use the course to learn SQL. This raises the question as to how much time they spent using the course and subsequently how accurately they could judge the quality of the content. Due to the ethical requirement for participants in the experiment to be anonymous, it is not possible to actually check how much the participant in question did use the course.

Response	%
Never	4
Rarely	80
Usually	12
Always	4

(a) Q11 Table



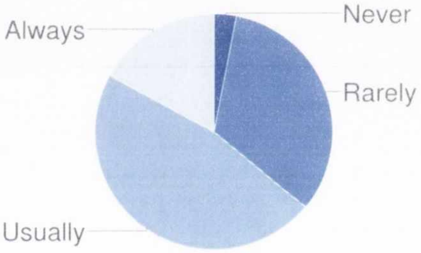
(b) Q11 Graph

Figure 6.16: Did the course content appear disjoint?

Would have liked greater control of content structure? Question 15 of the questionnaire asked the participants if they would have liked a greater level of control over the structuring of the content. From the participant's responses to this question, summarised in table 6.17a, it can be seen that 64% of the participants (29 participants responded to this question) would like a greater level of control compared to 31% who did not.

Response	%
Never	3
Rarely	33
Usually	47
Always	17

(a) Q15 Table



(b) Q15 Graph

Figure 6.17: Would have liked greater control of content structure?

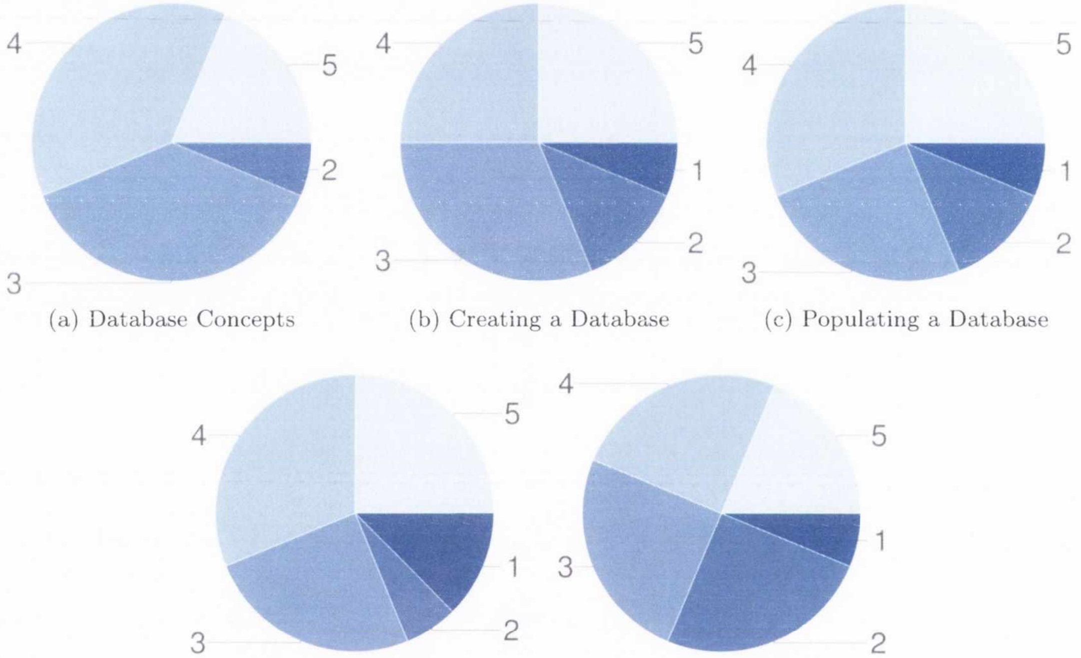
Appropriateness of Content To gauge the appropriateness of the content delivered by the system, the participants were asked to rate, on a scale from 1 to 5, how effectively they felt each of the course sections represented the subject matter. A score of 1 indicates that the subject matter was not represented at all while a score of 5 indicates that the subject matter was represented completely. The results obtained from the 26 participants that answered this question are shown in table 6.12. The table shows the percentage breakdown of results in each of the 5 rating categories for each of the course sections.

Figure 6.18 consists of a set of pie charts that illustrate the proportions of participant's responses that fell into each of the five rating categories for each of the individual course sections. As can be seen from the pie charts and the corresponding data from table 6.12,

Course Section	Rating Score Percentages				
	1	2	3	4	5
Database Concepts	0.0%	3.8%	38.5%	38.5%	19.2%
Creating a Database	7.7%	11.5%	30.8%	26.9%	23.1%
Populating a Database	7.7%	11.5%	26.9%	30.8%	23.1%
Database Retrieval	15.4%	3.8%	23.1%	30.8%	26.9%
Database Applications	7.7%	26.9%	23.1%	23.1%	19.2%

Table 6.12: Effectiveness of Subject Matter Representation Results (Q17)

for most of the course sections, the majority of participants (between 50% and 57.7%) gave the effective representation of subject matter a score of 4 or 5. The only exception was for the Database Applications section, which only received a rating of 4 or 5 from 42.3% or participants.



(a) Database Concepts (b) Creating a Database (c) Populating a Database

(d) Database Retrieval (e) Database Applications

Figure 6.18: Effectiveness of Subject Matter Representation

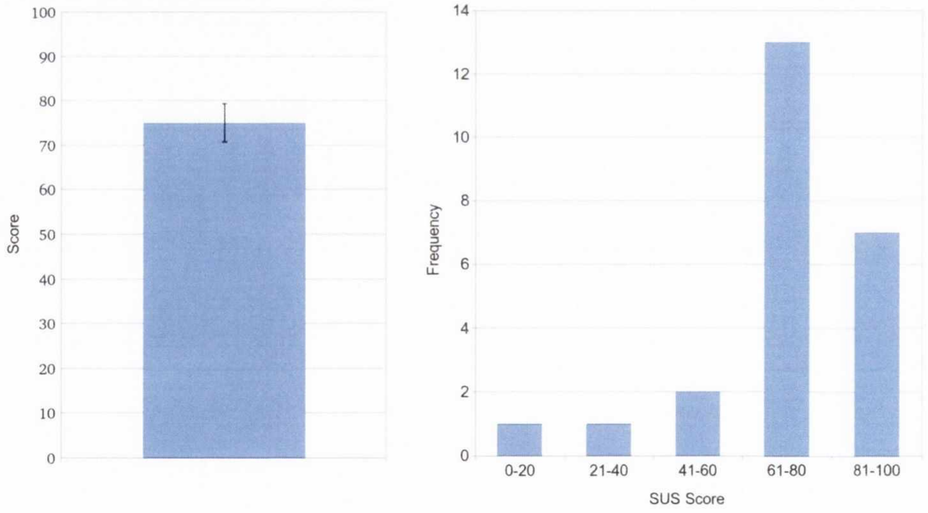
If the threshold for an acceptable rating is expanded from 4 to 3, the percentage of participants giving the content a positive rating increases to above 80%. Again the Database Applications section has a much lower level of acceptance with only 65% of participants rating it 3 or higher. This analysis of the data is based on the assumption that 3 is a neutral score indicating that the participant considered the subject matter representation to be neither totally incomplete nor totally complete.

6.5.2.3 Usability

In order to evaluate the usability of the system, the participants in the evaluation were asked to fill in a standard SUS questionnaire. In addition to this, several of the questions from the main questionnaire were focused on different aspects of the usability of the system, namely the educational usability of the system and the visual usability of the system.

System Usability Of the 32 participants that took part in the evaluation, only 24 answered all of the questions on the SUS questionnaire (5 participants did not complete the SUS questionnaire at all while a further 3 participants did not answer all 10 questions). After calculating the SUS score for each participant, a mean SUS score of 70.94 with a 95% confidence interval of 6.99 was calculated, see figure 6.19a.

To better understand this mean SUS score, the individual SUS scores calculated were used to generate a frequency distribution chart as shown in figure 6.19b. As can be seen from the graph, the majority of participants, 83.33%, scored the system highly (above 60). In fact, the lowest two scores given to the system equate to a 4.06 point fall in the SUS score. These two low scores also account for 1.7 points of the confidence interval of 6.99.



(a) Mean SUS Score

(b) Frequency Distribution of SUS Scores from Evaluation

Figure 6.19: Graphs of SUS Scores

The mean SUS score of 70.94 can be interpreted, according to [Tullis 08], as being above average based on an evaluation of 50 SUS studies, which had an average score of 66.

Similarly, this score can be interpreted, according to Bangor et al's [Bangor 09] 7 point adjective rating scale as equating to a 'good' system in terms of usability.

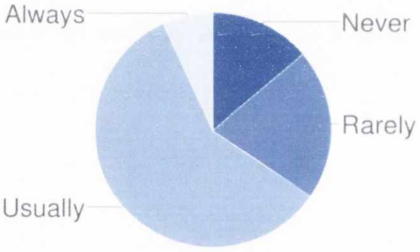
In addition to the SUS based evaluation, several questions in the main questionnaire were focused on the evaluation of the system usability

Educational/Instructional Usability To evaluate the usability of the system from an educational perspective two aspects of the system were investigated, the user's understanding of the course objectives and the ability to meet those objectives and the user interface itself. These aspects of the system under investigation were evaluated by means of a series of questions as part of the user questionnaire discussed previously.

Were the objectives of the generated course(s) clear to you? The responses of the participants to question 5, as shown in table 6.20a and visualised in figure 6.20b are generally positive with 65.5% of participants indicating that the objectives were clear. However, a significant number of participants 34.5% found the objectives unclear.

Response	%
Never	13.8
Rarely	20.7
Usually	58.6
Always	6.9

(a) Q5 Table



(b) Q5 Graph

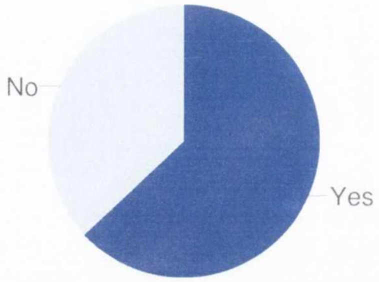
Figure 6.20: Were the objectives of the generated course(s) clear to you?

Upon completion of the online course did you feel you had completed the objectives? 63% of the participants that responded to question 8 of the questionnaire indicated that they did feel that they had met the objectives as shown in table 6.21a and figure 6.21b. Unsurprisingly, there was a strong correlation between the participants responses to this question and those of the previous question as shown in Appendix 6.23. It is to be expected that if the participants could clearly identify the objectives of the course then they would be in a better position to meet those requirements.

Visual Usability

Response	%
Yes	63
No	37

(a) Q8 Table



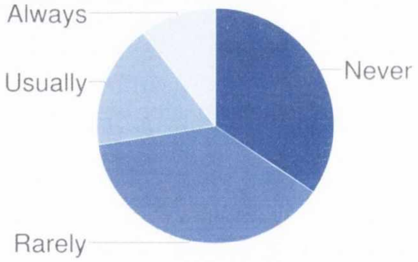
(b) Q8 Graph

Figure 6.21: Upon completion of the online course did you feel you had completed the objectives?

Would you have found the ability to modify the web interface beneficial? To gauge the visual usability of the system, the participants were asked whether they would have liked to be able to modify the UI to which 72.4% of the participants that responded to the question indicated that they did not feel the need for this, stating that they would either *never* or *rarely* find this beneficial. The breakdown of the results for this question are shown in table 6.22a as well as visually in figure 6.22b. These results can be taken as an indication that they found the UI sufficiently well designed.

Response	%
Never	34.5
Rarely	37.9
Usually	17.2
Always	10.4

(a) Q16 Table



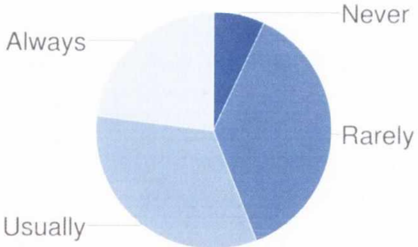
(b) Q16 Graph

Figure 6.22: Would you have found the ability to modify the web interface beneficial?

Was the quantity of content on each page satisfactory? In question 14 participants were asked if they would have liked to make changes to the UI. The results, as shown in table 6.23a and figure 6.23b, indicate that 56% of the participants found the amount of content sufficient with only 7% indicating that they amount of content was *never* sufficient.

Response	%
Never	7
Rarely	37
Usually	33
Always	23

(a) Q14 Table



(b) Q14 Graph

Figure 6.23: Was the quantity of content on each page satisfactory?

6.5.3 Discussion and Analysis

As outlined previously, the aim of this experiment was to evaluate the ability of the system to support adaptive content based on three criteria, the ability of the system to adaptively select and sequence content, the quality of that content and the usability of the system. This section provides an analysis of the results based on these three criteria. As part of this analysis any significant correlations were identified between the answers given by the participants to each question. This was achieved by using Spearman's method[4] for calculating the correlation coefficients and comparing them against the standard table of significant values. Due to the nature of the results in which not all questions were answered by all participants, pairwise deletion was used to calculate the correlation matrix. Based on this it was possible to identify the correlations that were statistically significant. A table outlining the Spearman correlation coefficients that were calculated is provided in Appendix C.4.2. For convenience, a table of the significant coefficient values is also provided in Appendix C.4.1. Any significant correlations that are of interest are discussed as part of this analysis.

From the results reported in section 6.5.2.1 of this chapter it is clear that the majority of participants in the evaluation did not rebuild their courses frequently. To be able to understand why this was the case it is necessary to look at the results for the other questions in the evaluation questionnaire that were targeted at evaluating the systems support for adaptive behaviour. To facilitate this analysis, the results from the three questions are summarised in table 6.13.

The results shown in the table have been grouped based on whether the response of the participants were positive or negative. From this, it can be seen that the majority of

[4] Spearmans method was chosen over Pearsons due to the ordinal nature of the data

Did the course generated reflect the:	Never/Rarely	Usually/Always
answers given in questionnaire	32%	68%
course you wanted	38.4%	61.6%
content you expected	13.33%	86.67%

Table 6.13: Summary of results from evaluation questions on adaptive behaviours

participants found that the courses generated by the system *usually* or *always* reflected both the answers they gave to the questionnaire and the course they wanted. However 32% of participants did not feel that the course reflected the answers that gave to the questionnaire while 38.4% did not feel that the course was the one that they wanted. These both represent significant percentages of the participants to have been dissatisfied by the personalised course generated for them. One question that is raised by such high numbers is whether or not the composition, as guided by the Narrative Model, failed for these numbers. Based on the design of the system and how the Narrative Model for the SQL course is written it would seem an unlikely answer. Each of the five questions directly correspond to sections of the course resulting in a very coarse grained form of adaptation. The result of answering 'yes' to a question is that the corresponding section of the course is removed. Based on this design it is not possible for the system to have generated an incorrect structure. There are two likely causes for this result. Based on the low level of use that the personalisation feature of the course received, it is possible that participants simply didn't see the system adapting to their needs. Alternatively, the coarse grained nature of the adaptation might not have provided enough control of the course structure for participants. This would be consistent with the results for question 15 in which 64% of participants indicated that they would have liked more control over the course structure.

As might be expected, there was a significant correlation between the participants comfort with online learning and whether or not they felt the courses generated reflected the answers they gave. With 83% of participants stating that they were either *quite* or *very* comfortable with online learning this is not surprising although it does mean that participants that were less comfortable with online learning were also less likely to consider the course generated to correspond with their answers. Similarly, there was a significant negative correlation between the participants prior experience in SQL and whether the course generated was the course they wanted. This means that participants with little or no prior experience, who represented 73% of all participants, were much happier with the course generated. It is possible to attribute this result to the design of the SQL course and the content used. The course is designed to primarily support beginner or intermediate level students as it does not provide in depth discussions or more advanced topics that would be appropriate for stronger students. As such, the content that makes up the course is less useful to these stronger students and so they are more likely to be dissatisfied by the course.

Interestingly there was also a correlation between prior experience and whether the course contained the content expected. It would seem that the more experienced participants felt that the courses generated were as they expected but at the same time did not feel that they reflected their answers to the personalisation questionnaire. It is difficult to interpret this result as the questions posed in the SQL course personalisation instrument, see figure ?? in Appendix ??, directly corresponded to high level concepts within the SQL domain and explicitly referred to these concepts in the questions. Perhaps the more experienced participants expected more topics or advanced concepts to be covered in certain sections of the course based on their prior knowledge of the domain.

Furthermore, a significant majority considered the generated course to contain the content they expected. Based on these results it can be argued that the reason that most participants did not rebuild the course, after it was initially generated for them, was that the course already accurately matched their needs and expectations and so they did not feel the need to rebuild it a second time.

When looking at the quality of the content in the course, the experiment focused on the appropriateness of the content and the flow or sequencing of the content. The results for this part of the experiment were reported in section 6.5.2.2 of this chapter and are summarised in table 6.14.

	Never/Rarely	**Usually/Always**
Were courses easy to navigate?	7%	93%
Did course content appear disjoint?	84%	16%
Would you have liked greater control of content structure?	36%	64%

Table 6.14: Summary of results from evaluation questions on quality of content flow

As shown by the results in the table, the majority of the participants found that the course was easy to navigate (93%) and that the content was not disjoint (84%). These are very positive results that clearly indicate that the structure of the course and the sequencing of the content are both well designed. Importantly, it clearly indicates that the adaptive behaviours of the system do not impact adversely on the flow of the course. Interestingly, when the participants are asked if they would like greater control over the structure of the course they respond less positively with 64% indicating that they would like more control. One reason for this might be the background of the participants, who were all computer science or engineering students who are therefore both technically experienced and curious. This assertion is backed up a comment that one participant provided that indicated they would find it 'cool' to be able to modify the course layout in this way.

The results for the evaluation of the appropriateness of the content were presented in section 6.5.2.2. Table 6.15 shows the percentage breakdown of results in each of the 5 rating categories for each of the course sections. Also listed in the table are the mean scores for each of the course sections and the 95% confidence interval for each of those mean scores. The numerical breakdown of the rating scores used to generate the percentages in the table can be found in Appendix C.3.1.

Course Section	**Rating Score Percentages**					**Mean Score**	**Confidence Interval**
	1	**2**	**3**	**4**	**5**		
Database Concepts	0.0%	3.8%	38.5%	38.5%	19.2%	3.73	0.32
Creating a Database	7.7%	11.5%	30.8%	26.9%	23.1%	3.46	0.46
Populating a Database	7.7%	11.5%	26.9%	30.8%	23.1%	3.50	0.46
Database Retrieval	15.4%	3.8%	23.1%	30.8%	26.9%	3.50	0.52
Database Applications	7.7%	26.9%	23.1%	23.1%	19.2%	3.19	0.49
Average	7.7%	11.5%	28.5%	30.0%	22.3%		

Table 6.15: Effectiveness of Subject Matter Representation Results (Q17)

As can be seen the data in the table, for most of the course sections, the majority of participants (between 50% and 57.7%) gave the effective representation of subject matter a score of 4 or 5. The only exception was for the Database Applications section, which only received a rating of 4 or 5 from 42.3% or participants.

If the threshold for an acceptable rating is expanded from 4 to 3, the percentage of participants giving the content a positive rating increases to above 80%. Again the Database Applications section has a much lower level of acceptance with only 65% of participants rating it 3 or higher. This analysis of the data is based on the assumption that 3 is a neutral score indicating that the participant considered the subject matter representation to be neither totally incomplete nor totally complete.

The mean scores shown in table 6.15 were used to generate a bar chart, figure 6.24, in which each of the mean scores is shown along with the corresponding confidence interval. As shown, it is possible, with a high degree of confidence, to say that the mean score for each of the course sections is 3 or higher. Again, the values for the Database Applications section are less straight forward although the calculated mean of 3.19 is still relatively consistent with the values calculated for the other course sections.

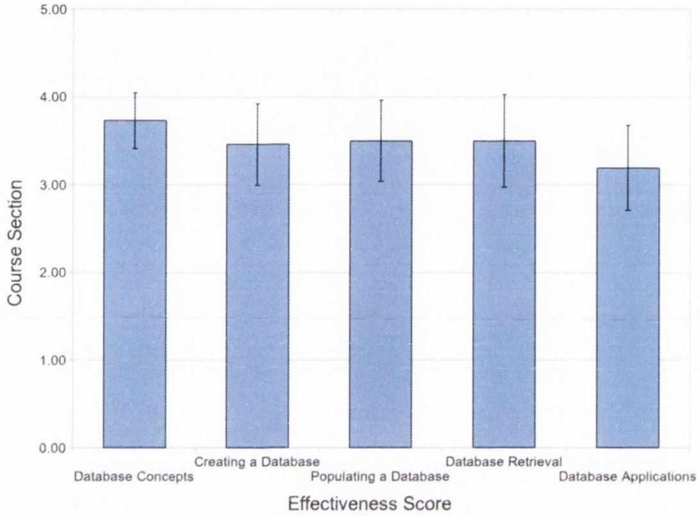


Figure 6.24: Mean Effectiveness Scores by Course Section (Error bars represent 95% confidence interval)

The difference in ratings for the Database Applications section in comparison to the other course sections can be attributed to the fact that this is a relatively short section of the course, which has shortcomings with respect to the range of topics it covers. For example, it discusses Embedded SQL and PL/SQL but does not cover language specific details such as how to write a PHP application using Embedded SQL or how to write a Trigger. This is an intentional aspect of the course, which is intended to encourage the students that use the course to find this information for themselves as part of a self directed learning activity. However, with the increasing familiarity of students with web development they are beginning to expect a lot more detail on topics such as Embedded SQL. This assertion is backed up by some of the comments provided by participants in the experiment, which made this exact point.

To gain a better insight into the results obtained, the frequency distribution of ratings for each of the course sections were plotted as a stacked line graph as shown in figure 6.25. An analysis of this graph seems to back up the interpretation of the results. It can be seen from the plots for each of the course sections that the distribution tend towards scores of 3 or higher with a noticeable peak to the curves in the 3-4 range.

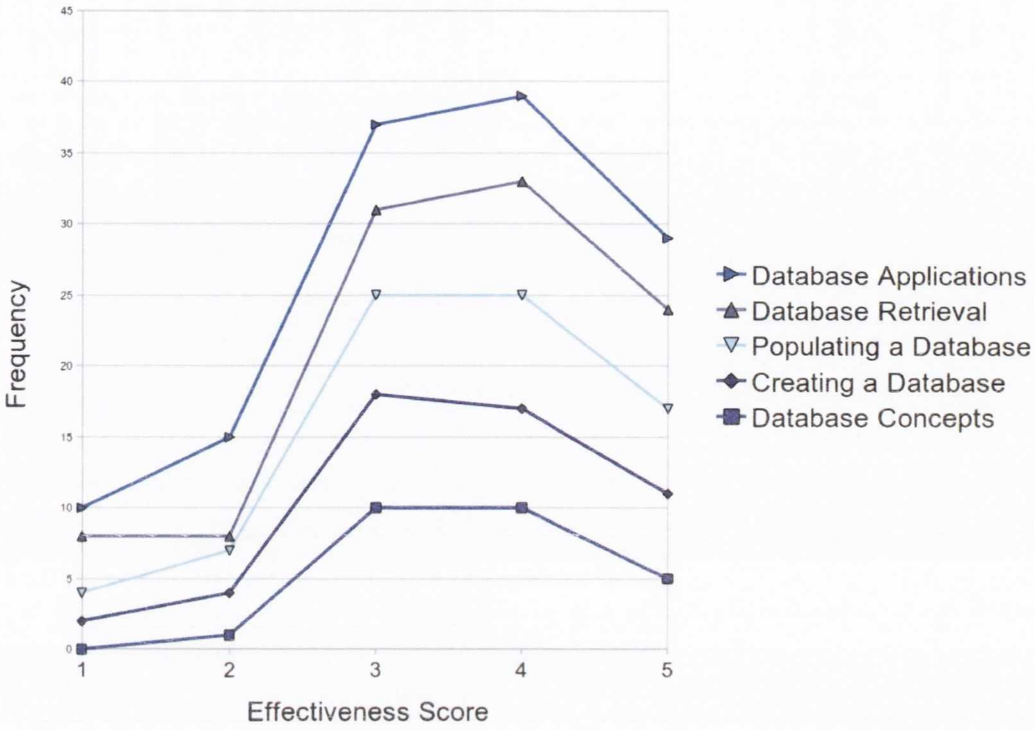


Figure 6.25: Frequency Distribution of Effectiveness Scores by Course Section

The results of the usability aspect of this experiment are reported in section 6.5.2.3 of this chapter. An initial analysis of the overall usability of the system by means of the SUS score indicates that the system is pretty average in terms of usability with a score that can be interpreted as indicating that the system is 'good'. This rating would seem to indicate that although the system cannot be considered to be exceptional it is not bad either and this is an important point. This result can be seen to indicate that the adaptive behaviours implemented by the system do not have a significant adverse effect on the usability of the system.

Looking more closely at the usability of the system from an educational and visual perspective we can see from the results that the majority of users found the objectives of the system clear (65.5%) with 63% of participants feeling that they had completed the objectives of the course. Unsurprisingly, there is a strong correlation (see Appendix C.4.2 for more details on how this correlation was identified) between these two results indicating that participants that found the objectives of the course clearly identifiable were more likely to feel that they were able to meet the objectives of the course.

Aside from the expected correlation between the participants being able to identify the objectives of the course and feeling that they had completed the objectives at the end of the course there were some other interesting correlations. It was found that participants that were not able to identify the objectives of the course also found the course more difficult to navigate and the content more disjoint. The obvious question arising from this is in which direction this relationship goes, intuitively it would seem that the ability of the participants to identify the objectives of the course had an impact on how easy they found the course to navigate. The inverse of this would seem less likely. Interestingly, no correlation was found between the participant's ability to identify the objectives of the course and their prior experience either in SQL or with online learning in general. Another interesting observation was that the participants that found the objectives of the course clear also felt that the amount of content on each page was satisfactory.

However, 38.5% of participants found the objectives of the course unclear. This represents a very high proportion of the participants and indicates that there is an issue with how the objectives of the course are conveyed to learners. This issue, although not directly related to the adaptive nature of the course can have a serious impact on the educational effectiveness of the course. There exists the potential to significantly improve this through further investigating the problems that participants had with this aspect of the course.

The most significant of all correlations between answers in the evaluation indicated that the participants who felt they had completed the objectives of the course also wanted more control over the content included. This observation makes sense as it implies that those participants who found the course easiest and so were able to achieve the objectives, were less burdened by the content and so were more likely to consider other related topics. For such learners a more freedom with respect to the course content is clearly desirable.

As discussed previously, the results for the visual usability of the system indicate that the participants found the UI to be 'good'. Some of the more negative reactions can be attributed to specific issues that were identified by the participants with the user interface. Specifically, some participants had issues with some of the font sizes used while others made valid points about the placement of the navigation buttons. By addressing some of these issues it might be possible to improve the usability of the system even further.

Another issue that some participants in the study had was with the volume of content displayed on each page. There was quite an even split between participants that felt the volume was sufficient and those that did not. Of those that did not like the small 'pagelet' style presentation of content, some commented that it felt like a presentation while others felt that they had to interact too much with the UI to navigate the content. This type of issue is always going to be present in any aspect of the system that is one size fits all. There will always be users who do not want to access the content online and would rather download a single document containing all of the content. Perhaps the results obtained for this aspect of the system imply that the adaptive behaviours of the system need to be extended to the volume of content displayed to individual users.

6.5.4 Conclusions

The analysis of the results from this experiment have shown that the system was capable of adaptively selecting and sequencing appropriate eLearning content. This is illustrated by three key findings:

- Only 20% of participants considered the content selected by the system to be ineffective.
- 84% of participants found the content sequencing was not disjointed. Furthermore, 93% said that they found the composed courses easy to navigate.
- The overall usability of the system was rated as above average.

An additional finding of this experiment is that there was a significant correlation between participant's prior experience and how effective they felt the personalisation of the course was. This seems to suggest that the personalisation applied by the system was most effective for less experienced participants compared to the more experienced participants.

6.6 Experiment Four - Performance and Scalability

The aim of this experiment is to evaluate the performance and scalability of the PWE system (objective 9) during both the adaptation process and the delivery of the PWE to the user. Specifically the objectives of this experiment are to evaluate the impact of load on the time it takes to carry out the adaptation process and to evaluate whether or not the added complexity that the use of the workflow engine introduces into the architecture adversely affects the usability of the system from the perspective of page load times. A final objective of this experiment was to evaluate the impact of load on the delivery of services to the user.

To achieve this, the experiment is broken into two parts, the first focuses on the performance of the system during the adaptation process with the second looking at runtime performance. The simulated user interactions represented a prerecorded set of real user interactions with the system, which are then used to load test the system with increasing numbers of users. As part of the load testing experiments, page load times as well as server performance metrics were recorded and subsequently analysed with respect to the usability of the system as the number of users increased and the potential for the system to be scaled to cope with larger numbers of users.

6.6.1 Environment Configuration

The invocation of the system will be handled by the Apache JMeter [Apache Foundation 10a] application, which is configured to send the appropriate HTTP requests to the adaptive system in order to trigger the adaptation process. The time taken by the system to process each request is recorded by JMeter based on the time from the initial request by JMeter until a response is received by JMeter from the PWE system indicating a successful execution.

The experiments were run using a set of three virtual machines[5] running on a single server with 2 quad core Xeon processors running at 2.5 GHz and 16GB ram. The software configuration of the three virtual machines is provided in tables 6.16, 6.17 and 6.18.

All three servers were connected to a gigabit (1000Mb/sec) ethernet switch which was then connected to the LAN. However the machine used to run the benchmarks using the JMeter

[5] The Xen [Citrix 06] virtualisation technology was used to create the virtual machines.

Memory	2GB ram	
Operating System	64 bit Ubuntu Linux 8.04	
Java Virtual Machine	64 bit Sun JVM v1.5.0_16-b02	
JVM Configuration	-Xms	768m
	-Xmx	768m
	-Xss	513k
	-PermSize	256m
	-MaxPermSize	256m
Servlet Engine	Apache Tomcat 5.5.20	
Hosted Applications	PWE Web Appliation (LADiE Use Cases)	
	Planner Web Service (Apache Axis 1.x)	
	Apache Axis 2.x	
	Composition2BPEL Web Service	
	WSRP Proxy Web Service	
Database	eXist Native XML Database v1.4.0	

Table 6.16: Software Configuration of Host Server One

Memory	1GB ram	
Operating System	64 bit Ubuntu Linux 8.04	
Java Virtual Machine	64 bit Sun JVM v1.5.0_16-b02	
JVM Configuration	-Xms	768m
	-Xmx	768m
Servlet Engine	Apache Tomcat 5.5.28	
Servlet Engine Configuration	maxThreads	150
	minSpareThreads	25
	acceptCount	100
	connectionTimeout	2000
Hosted Applications	ActiveBPEL 5.0.2	

Table 6.17: Software Configuration of Host Server Two

Memory	2GB ram	
Operating System	64 bit Ubuntu Linux 8.04	
Java Virtual Machine	64 bit Sun JVM v1.5.0_16-b02	
JVM Configuration	-Xms	1024m
	-PermSize	256m
J2EE Applicaiton Server	Sun Glassfish v2.1	
Application Server Configuration	maxThreads	200
Hosted Applications	OpenPortal Portlet Container v2.0	
	OpenPortal WSRP Implementation v1.0	

Table 6.18: Software Configuration of Host Server Three

tool was not connected to the same switch. The benchmarking machine was located in the same building as the servers and connected to the LAN via a gigabit ethernet connection.

6.6.2 Evaluation of Adaptation Process Performance

The adaptation process was carried out by the Adaptive Engine in conjunction with the AI Planner service and supported by the eXist XML database, which provides access to the necessary metadata models. In addition to the process of generating a PWE model, it is necessary to deploy the PWE so that it can be executed. This secondary process involves an additional set of components, the BPELGenerator Service, the WSBPEL workflow engine and the user portal.

To evaluate the performance of the system during these phases of execution, a set of benchmarks were generated based on the adaptive generation of a PWE that realises one of the use cases used in experiment two. The system was configured, as described in experiment three, with the necessary metadata models and services available. The benchmarking of the system was carried out using the number of concurrent users as a mechanism for scaling the load applied to the system, with individual samples being taken for 1, 5, 10, 20, 50 and 100 simulated users. For runs of more than a single user, a set of example user models will be used to simulate the different properties of individual users, resulting in the system generating different PWEs.

6.6.2.1 Evaluation Set Up and Execution

The Apache JMeter load testing application was used to carry out the benchmarking of the adaptation process. This allowed the benchmarking to be carried out in a realistic manner with the adaptation taking place in the context of the action web application that was used to demonstrate the functionality of the system in the previous section of this chapter.

To achieve this, a JMeter Test Plan was developed that allowed a number of independent users to be logged into the webapp and for them to invoke the adaptation process in the same way as a real user would when accessing the system through a web browser. A screenshot illustrating the components of the developed Test Plan can be seen in figure 6.26. The Test Plan consisted of a Thread Group that allowed the number of users logged into the system to be controlled. The Thread Group contained a series of HTTP request handlers that made the necessary requests to the appropriate JSPs. The username and password used by each 'user' to log into the system were dynamically generated as part of the Test Plan execution in accordance with a naming convention. The generated user credentials corresponded to accounts that were set up on the system prior to the running of the benchmarking. In addition to the Thread Group, the Test Plan contained a Listener that allowed the response times as well as the actual responses returned by the webapp to be captured. An XML description of this JMeter Test Plan can be found in appendix D.1.

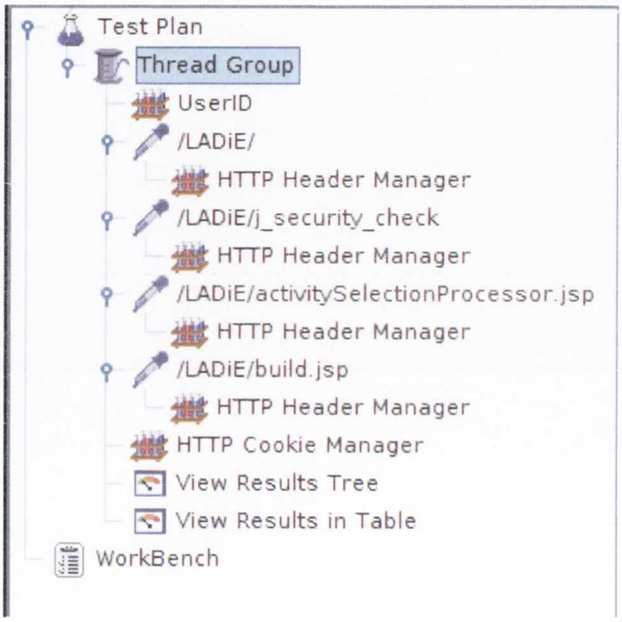


Figure 6.26: Screenshot of Apache JMeter Test Plan for personalisation process benchmark

6.6.2.2 Results

The benchmarking process involved a series of samples being taken corresponding to an increasing number of users accessing the system. Samples were taken for 1, 10, 20, 50 and 100 concurrent users accessing the system. The average time taken for the build process was then calculated for each sample. The raw data for these benchmarks can be found in appendix D.2 while the average times calculated from this data are provided in table 6.19

The average build times were then used to generate a line graph, see figure 6.27, plotting the average build time against the number of parallel build processes (number of concurrent

# Users	1	10	20	50	100
Average Response Time	3.883	15.4043	26.1495	70.3759	161.47733

Table 6.19: Average build times (in seconds) for personalisation process

users).

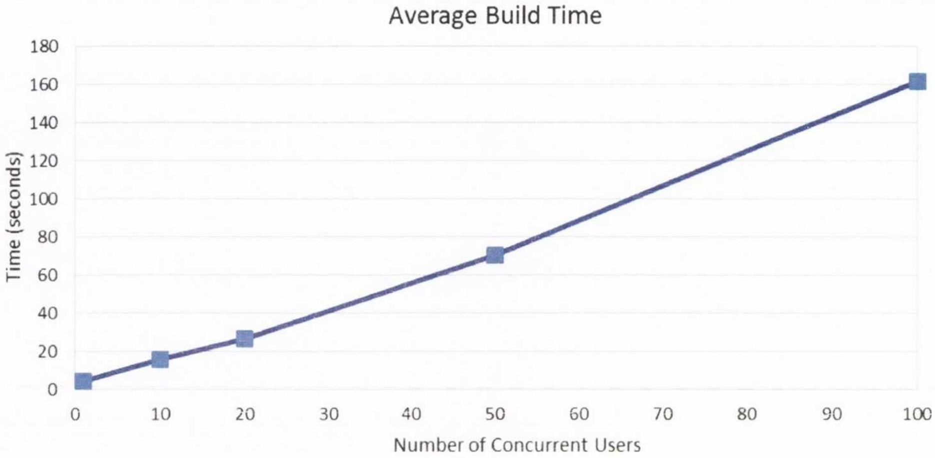


Figure 6.27: Line graph plotting average time taken to generate PWE

In order to be able to interpret the average build time data captured as part of this experiment, the performance of the server that the system was running on was also monitored over the time period that the benchmarking was carried out. Using the Dstat [Wieers 10] command line utility, which was configured to capture data relating to the CPU usage and Load average on the server at 1 second intervals. The raw data captured during the benchmark is provided in Appendix D.3.

Using the data captured a set of graphs were generated, which provide a visualisation of the data captured during the 50 user benchmark. Figure 6.29 shows a plot of the load average on the server over time. The load data in this case is the average load for a one minute period recorded on the server at every sample point. The load on the server represents the number of processes that are either running or waiting for the CPU, known as the ready queue. In this case the graph plots the value of the average load over the previous one minute period. Figure 6.29 plots the percentage CPU usage over the same period of time. In the graph the percentage CPU usage for both User, System and I/O Wait are stacked on top of each other so that the height of the graph at any point represents the sum of the three CPU usage values.

The build time data presented in figure 6.27 helps to illustrate what happens to the

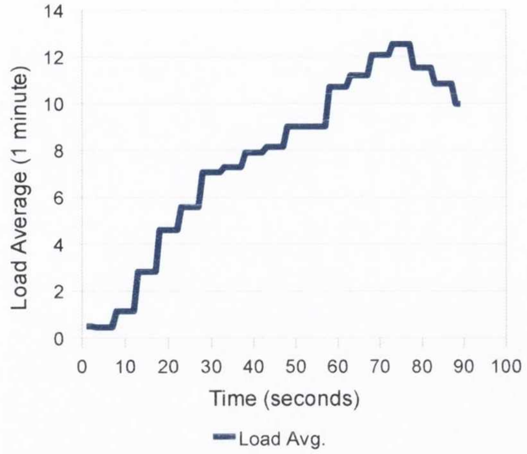


Figure 6.28: 1 Minute Load Average on build server during 50 user benchmark

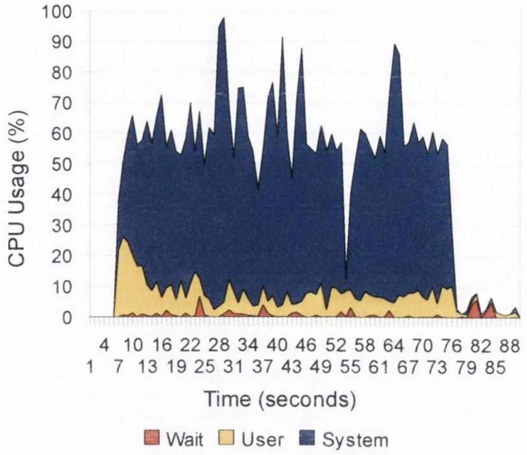


Figure 6.29: Percentage CPU Usage on build server during 50 user benchmark

performance of the system as load (the number of users) increases but it does not tell us anything about why the performance changes. In order to get a better understanding of what aspects of the build process are having the most significant effect on the performance of the system the system was instrumented so that a log of the time taken to carry out each discrete step in the Narrative execution could be monitored. The discrete steps in the Narrative that were monitored were the generation of the domain model used as part of the planning problem, the invocation of the AI Planner Web Service, the processing of the solution returned by the planner and the querying of metadata models stored in the database. A log of all of these activities was generated by repeating the 50 user benchmark that was carried out previously. From the log the average time spent carrying out each of these discrete tasks, per user, was calculated. The calculated averages are shown in table 6.20 and a bar graph representation provided in figure 6.30.

Task	Avg. Time (sec)	Percentage Avg. Time
Processing Solution	11.175	21.63%
Querying Models	7.622	14.76%
Planning (Service Selection)	26.259	50.84%
Generating Domain	6.596	12.77%
Total Time	51.652	100%

Table 6.20: Breakdown of average time spent per user during 50 user benchmark

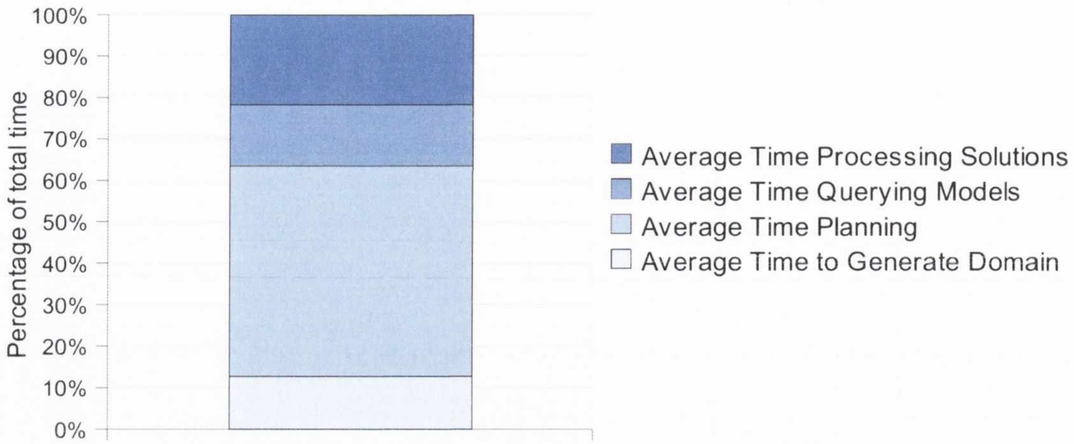


Figure 6.30: Bar chart showing breakdown of average time spent for 50 user benchmark

6.6.2.3 Analysis

The time taken to build a PWE for a single user, 3.883 seconds, although not instantaneous, is acceptable for a web based application. Considering that a relatively complex web page such as the home page of the BBC News website (http://news.bbc.co.uk) takes over 4 seconds to be retrieved by a web browser running on a PC on a network with a high speed internet connection (10 Gigabit/sec). Looking at often cited response times from HCI research this time is almost twice a long as the 2 second minimum Tolerable Wait Time (TWT) recommended by the early work on HCI [Miller 68] and the 1 second maximum time recommended by Neilsen [Nielsen 93] so as not to interupt the user's flow of thought. However, this delay only occurs during the personalisation process, prior to the student beginning the learning activity. Looking at the recommended upper bounds on TWT, time taken is within the 10 second limit recommended by Neilsen [Nielsen 93] so as not to lose the user's attention on the task and also Scheiderman's earlier 8 second recommendation [Schneiderman 84]. More recent HCI research on TWT for information retrieval suggests that the 2 second time frame still holds true [Nah 04].

Looking at how the system scales as the load increases we can see that the response time increases at a steady rate that closely follows a linear scale. However as the load increases from 50 to 100 users the average time taken to build the PWE starts to deviate from the relatively linear scaling seen for lower loads. An average build time of 70 seconds for a load of 50 users increases by approximately 130% to 161 seconds. Although these build times are very long, they are broadly in line with how we would expect the system to scale and significantly there is no indication of exponential growth in the average built time, at least within the sample sizes used for this benchmark. This benchmark also puts the system under a very high load in a short space of time, 100 simulated users requesting PWE's to be generated in a 1 second period.

The server performance data shown in figures 6.28 and 6.29 provides an indication of where the bottleneck, in terms of performance, lies. We can see that during the 50 user benchmark sample the load on the server reaches a peak value of 12.54 and was at a value of 8 or more for a period of 60 seconds. This is significant because the server used to carry out the benchmarking contained 8 CPU cores and as such could easily handle a load value of up to 8. Above this level, processes would have to wait before being allocated time on the CPU. Considering the service oriented nature of the system architecture and the limitations of the approach taken to integrate the native AI Planner component with the Java based WS wrapper, it is easy to see how so many processes could be generated. However, this service oriented architecture is also a means of addressing the issue. If the system was to be scaled horizontally by moving the database and other services to separate servers then the load could be spread across several machines. Additionally, a simple load balancing approach could be taken by taking advantage of the built in support of the application server.

As mentioned, another factor that impacts on the performance is the AI Planner. We can see from figure 6.30 that 50% of the average time taken to generate a PWE is spent waiting for a response from the AI Planner WS. A large part of this time is likely to be saved simply by the horizontal scaling of the system as discussed. However, the need for AI Planner WS to write files to disk and spawn native threads is not the most efficient approach. If the native AI Planner were to be replaced with a Java implementation of an appropriate planning algorithm, which fully supported the PDDL language, then further significant performance benefits could be achieved and the time taken to generate a personalised activity could be brought closer to the 2 second time frame with this change alone.

6.6.3 Evaluation of Runtime Performance

The evaluation of the runtime performance of the system was focused on the impact on the system of the addition of the workflow engine as a proxy for the actual services used by the end user. The time taken by the system to deliver portlets as part of the UI and to react to user interactions is critically important in order for the system to be usable and not act as an impediment to the activities being delivered. In addition, the system needs to be capable of supporting a significantly large group of users accessing the system at the same time so that, for example, a class of students can make use of the system without the performance of the system degrading to an extent that makes it unusable.

To evaluate this aspect of the system's performance two sets of benchmarks were carried out. The first designed to compare the time taken to deliver a portlet using the WSRP protocol with the delivery of the same portlet with the workflow engine acting as a proxy. The second set of benchmarks aim to investigate the scalability of the system by simulating the interactions of groups of users with the PWE system.

6.6.3.1 Part One

To evaluate the effect of adding the workflow engine as a proxy between the portlet consumer and the portlet provider, two sets of benchmarks were carried out. The first measured the time taken, from the moment a request is sent by the consumer to the time that the consumer receives the response from the producer. In this initial benchmark, the request from the consumer is sent directly to the portlet producer.

The second benchmark is carried out by measuring the same request-response time but with the messages being proxied by the workflow running on the workflow engine. In this way, the time taken by the workflow engine to carry out any processing of the message and/or control flow within the process is taken into account.

These two benchmarks are carried out for both the getMarkup and the performBlockingInteraction requests.

The four benchmarks are carried out using Apache JMeter acting as the consumer as is the case in the previous experiment. This allows the number of parallel requests to be scaled while still allowing the timing to be recorded.

Methodology The benchmarking was carried out by first creating an appropriate JMeter Test Plan consisting of a thread group, used to control the number of requests made by JMeter in a specified period of time and a Listener to capture the data for analysis. In order to make the necessary requests to the WSRP services, a SOAP/XML-RPC

Request Sampler was added to the Thread Group. This type of sampler allows the URL and SOAP message to be specified as part of the Test Plan definition. Four different test plans were generated, one for each of the of the 4 benchmarks to be carried out as listed below. The XML definitions for these test plans can be found in appendix appendix:runtimeBenchmarkJMeterPlan.

- getMarkup (direct request)
- getMarkup (request via WSBPEL process)
- performBlockingInteraction (direct request)
- performBlockingInteraction (request via WSBPEL process)

When running the Test Plans for each of the benchmark, the number of threads created by JMeter in the specified period of time was increased in order to generate samples for a range of request volumes from 1 to 200 threads. In these benchmarks, these separate threads are used to represent individual users.

The **getMarkup** and **performBlockingInteraction** operations used in these benchmarks are that of a simple HTML iframe portlet hosted on the Glassfish application server using the OpenPortal portlet and WSRP implementations as described previously. This is a very simple portlet that is well suited to the purpose of evaluating the impact of the proxying of WSRP requests through a WSBPEL process on the overall performance of the system with respect to the users interactions with services. The suitability of this portlet is based on its simplicity as it requires very little computation to generate the necessary HTML markup fragment. As such it will provide a consistent basis for the benchmarks without introducing any variability that might result from the use of a more complex portlet.

The WSBPEL process used for the benchmarking of the proxied requests is a simple PWE BPEL process that contains only one activity/service, the iframe service. This will allow a comparison to be made between the results for both the WSBPEL proxied request and direct request benchmarks.

Results The results of the benchmarking of the getMarkup operation are provided in table 6.21. For each sample size, the average response time was calculated from a set of all the responses captured. A log of all of the responses for the getMarkup operation can be found in appendix D.4. The table compares the average response time, for a given sample size, of SOAP requests that are proxied through the WSBPEL process with those that are sent directly to the Portlet Container.

# threads	Direct	Proxied via WSBPEL	% Difference
1	53	94	77%
10	40.7	92.4	127%
25	43.48	121.44	179%
50	372.1	533.02	43%
100	1241.42	1443.34	16%
200	3007.75	3084.49	3%

Table 6.21: Average response times (in milliseconds) for getMarkup requests

To better illustrate these results, a line graph was generated, see figure 6.31, which plots the results for both the direct and proxied requests against each other. In the graph, the data points represented by diamonds correspond to the direct getMarkup requests while the triangles mark data points for the getMarkup requests proxied via the WSBPEL process. Due to the scale of this graph it does not clearly illustrate the results for sample sizes below 50. For this reason a second line graph, figure 6.32 is provided. This graph is based on the same data set but only shows the average response times up to a sample size of 50.

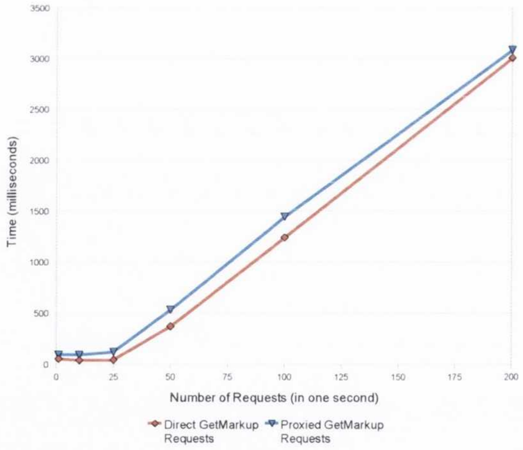


Figure 6.31: Line graph plotting average response times for getMarkup requests (1-200 requests)

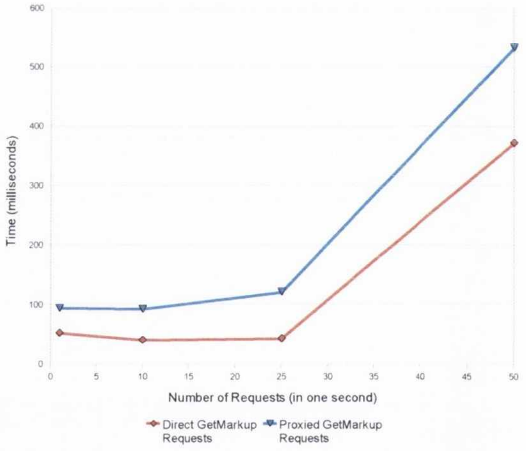


Figure 6.32: Line graph plotting average response times for getMarkup requests (1-50 requests)

The benchmark results for the performBlockingInteraction operation are provided in table 6.22. In the same manner as the results for the getMarkup samples, the average response time of both the proxied and direct versions of the performBlockingInteracation sample sets was calculated from the raw data, which can be found in appendix D.5.

# threads	Direct	Proxied via WSBPEL	% Difference
1	41	93	127%
10	41.7	79	89%
25	40.88	103.96	154%
50	389.84	602.32	55%
100	1245.76	1358.75	9%
200	3019.42	3145.49	4%

Table 6.22: Average response times (in milliseconds) for performBlockingInteraction requests

The results from table 6.22 were then plotted against each other in a line graph as shown in figure 6.33. The data points in the graph marked with a diamond correspond to the direct performBlockingInteraction requests and the data points marked with a triangle correspond to the proxied requests. As was the case for the getMarkup results, a second graph was also generated for the performBlockingInteraction benchmark results, figure 6.34, which provides a better visualisation of the benchmark results between 1 and 50.

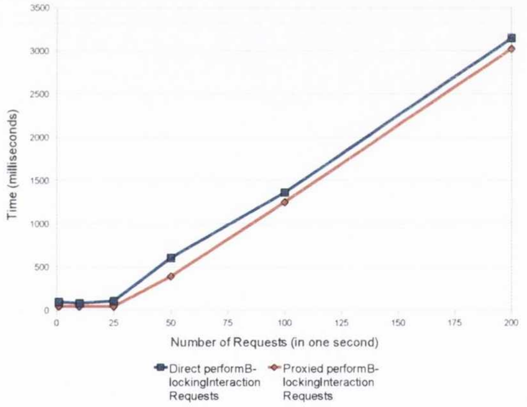


Figure 6.33: Line graph plotting average response times for performBlockingInteraction requests (1-200 requests)

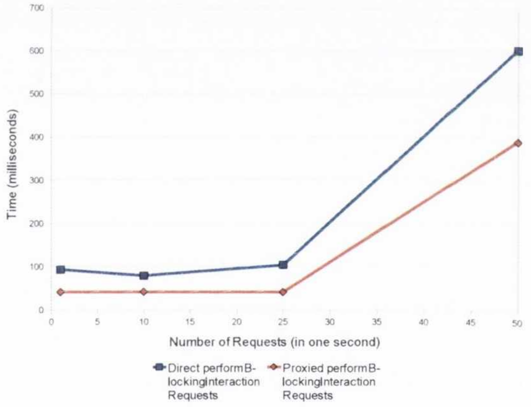


Figure 6.34: Line graph plotting average response times for performBlockingInteraction requests (1-50 requests)

Analysis As expected, the introduction of a WSBPEL process acting as a proxy between the portal and the portlet provider does introduce a significant overhead. For a single request, the response time more than doubles from 41ms to 93ms, an increase of 127%. However, as the results show, the increase in the average response time, although significant with respect to the baseline of the direct request to the portlet provider, it is relatively insignificant when viewed in the context of an internet application. Users on the web accept page load times that are relatively long. For instance, the Google home page (http://www.google.com) takes 500ms to load when requested from the same machine used as the client in these benchmarks. Similarly, the home page for the BBC News website (http://news.bbc.co.uk) takes over 4 seconds to load. In this context, an increase in response time of 50ms is not as significant as it might initially seem.

More importantly, the response time for a WSBPEL proxied request for a single user is still less that 0.1 seconds, which according to Neilsen [Nielsen 93] is the limit for a response that the user will perceive as instantaneous. In fact performance did not begin to degrade significantly until the number of parallel threads increased beyond 25 users.

Looking at the affect on response time as the load on the system increases, we can see that the average response time does increase significantly as the number of users increases but this is the case for both the direct requests and the requests that are proxied through the WSBPEL process. In fact, the average response time for the proxied requests in the 200 request sample is only 126.07ms longer, an increase of 4%, than the average response time for the same number of direct requests.

6.6.4 Conclusions

This experiment aimed to evaluate the performance of the system both during the personalisation process for an activity and at run time as the user interacts with the activity. Through a series of benchmarks it was shown that the time taken to generate a personalised learnign activity was acceptable for a process that only occured prior to the user engaging in the activity. The benchmarks also provided an indication of where performance bottlenecks existed in the system and how the peformance could be improved. The run time performance of the system was also shown to remain below the threshold for responses to be considered instantaneous until the number of parallel threads increased beyond 25. This illustrates that the application of load balancing techniques to scale the system horizontally would allow the system to support even larger numbers of users with acceptable response times.

6.7 Summary

This chapter has discussed the evaluation of the PWE system. As part of this discussion, the methodology, results and analysis for four experiments were presented. The aim of these four experiments were to evaluate the implemented systems ability to satisfy the objectives of this thesis.

The first experiment was designed to evaluate the ability to describe eLearning activities using the 5 workflow patterns that the system supports for the specification of control flow between services. As such it aims to evaluate the system's ability sequence services to deliver a wide range of eLearning activities. This experiment consisted of an analysis of a set of independently developed eLearning activities with respect to the functionality provided by the system implemented. The results of this analysis showed that the workflow patterns supported by the system provided a sufficient range of control flow in order to support the implementation of such activities.

The aim of the second experiment was to evaluate the ability of the system to adaptively sequence or select interactive services. This was achieved through the implementation of three use cases, based on the learning activities defined by the LADiE project. The subsequent analysis of the implemented use cases verified that the implemented system could deliver personalised learning activities.

The third experiment was designed to investigate the implemented system's support for the adaptive selection and sequencing of multimedia content. It consisted of a user based trial that was run using a personalised course designed to teach SQL. This results obtained from this experiment showed that the system and the functionality it provided were sufficient to support the effective delivery of such a personalised course, making use of both adaptive selection and sequencing.

The fourth experiment investigated the performance of the system during both the generation of a PWE and the runtime interactions of the end user with the system. This experiment involved the benchmarking of both processes, the results of which showed that the system could handle the type of loads that might be expected in real world deployment.

The results of these four experiments show that the objectives of this research as outlined in chapter 1 and the requirements specified in chapter 4 have been met.

Chapter 7

Conclusion

7.1 Introduction

The research question posed in this thesis asked what are the appropriate techniques and technologies required to support the delivery of personalised web based learning experiences. Such experiences were characterised, in this research, as combining adaptively sequenced and selected multimedia content with adaptively composed interactive services in a unified manner while taking into account the needs of the individual.

This chapter discusses the objectives of this thesis and how those objectives were met. This discussion is then followed by a summary of the contribution to the state of the art that this represents. This chapter concludes with a discussion of possible directions in which this research can be taken in the future. This discussion includes potential extensions to the research platform as well as new research questions that have arisen based on the research carried out for the purpose of this thesis.

7.2 Objectives and Achievements

7.2.1 State of the Art

The first objective of this research was:

"To carry out a literature review of the state of the art in those research areas that are of significance to this work. Specifically, this review will investigate the technologies and

techniques that can be used for the adaptive selection and composition of content and services for eLearning".

The review of Adaptive Learning was carried out in two parts, the first focused on the learning theory that forms the basis of modern learning pedagogy. Specifically, the three perspectives of Associationism/Empiricism, Cognitivism and Situativism. This review provided a theoretical justification for the research question as applied in the context of eLearning as the idea of engaging the Learner in the learning process through activity is promoted in all three perspectives although the motivation for doing so differed in each respective perspective.

Following this, a review of the techniques and technologies that can be applied for the purpose of adaptively delivering multimedia content was carried out. As part of this review, a set of prominent systems for the adaptive delivery of content were compared, with a focus on the area of Adaptive Hypermedia, such as AHA!, ADAPT^2 and APeLS as well as the ActiveMath system and the IMS Learning Design specification and its various implementations. A comparison of these adaptive systems and approaches was carried out based on a set of criteria that were considered important in adaptive systems, namely the adaptive techniques applied in each system and their architectural characteristics. From an analysis of this comparison, it was possible to identify common features, which successful adaptive content systems, as characterised by the systems presented in the review, possess. These common features can be seen as fundamental to any adaptive system to be developed. A secondary finding that can be drawn from this review is the limited support for the selection and sequencing of services in the current generation of adaptive content composition systems.

The second part of the state of the art review, which focused on the orchestration and dynamic composition of services was subsequently carried out. As mentioned, this first looked at technologies and systems that are capable of orchestrating services as part of a manually authored service workflow. In the context of this research, the manual authoring of workflows is important as it is necessary to be able to manage the control flow between services in a PWE, for instance to enforce pedagogical constraints. From this review it was clear that although the state of the art provided flexible support for control flow between services it did not address the need for adaptivity within the workflow. Although some of the systems surveyed, such as YAWL and the CAWE Framework, did provide basic support for adaptivity, this was very rudimentary. In order to identify techniques that could be used to address this lack of support for adaptivity within the service composition, the state of the art in dynamic service composition techniques was surveyed...

7.2.2 Unified Architecture

The second objective of this research was:

"To research, iteratively develop and test an integrated adaptive system suitable for the pedagogically driven composition of multimedia content and interactive services."

A model driven framework was developed, which applied many of the design principles of Adaptive Hypermedia as identified from the state of the art review. The core component of this architecture, the Adaptive Engine (AE), facilitated the strategy driven selection and sequencing of content as well as the sequencing of services based on the needs and preferences of the Learner. The adaptive selection of services was achieved through the integration of an AI Planner with the AE. This allowed services to be adaptively selected based on the needs of the Learner through the configuration of the planning environment by the AE while taking into account the necessary parameters and other prerequisites of the services. The use of an AI Planner for this purpose also facilitated the dynamic composition of services in order to meet the needs of the learner in the event that a service did no exist that satisfied the necessary requirements. An additional benefit of the AI Planner, from the perspective of the PWE author, is that it allows them to be less explicit about the definition of certain parts of a PWE where there is no pedagogical need for the sequencing rules to be explicitly stated. In such cases, the AI Planner can generate the necessary intermediate steps in the composition.

The delivery of the personalised composition was achieve through the novel integration of Web Service based portlet APIs and a WSBPEL based workflow engine. By dynamically transforming the personalised composition of services generated by the AE into an executable workflow it was possible to deliver the services selected for the Learner at the appropriate points in the composition, while maintaining the PWE author's intended sequencing.

Through a series of evaluations the implemented system was shown to be capable of successfully delivering personalised multimedia content through both adaptive selection and adaptive sequencing techniques. Similarly, the evaluation showed that the system was capable of generating personalised service compositions consistent with the design of the PWE author. It was also shown that the system was sufficiently flexible so as to support the delivery of a range of eLearning activities that were independently developed by eLearning practitioners. The evaluation also showed that the system implemented had sufficient performance so that it would be usable from an end user perspective in the context of a web application. The performance evaluation also showed that the system was scalable vertically to a large degree with the system's service oriented architecture facilitated further horizontal scaling.

7.2.2.1 Mechanisms and Techniques for Adaptive Behaviours

As part of the design and implementation of the system, a set of techniques were developed that could be applied, as part of the narrative, to take advantage of the design of the system and as such realise the requirement for adaptive selection and sequencing of services. The first of these techniques focused on the adaptive sequencing of services. As discussed previously, the sequencing of services in an eLearning activity is not inconsequential and is generally by design. As such, it is important that while allowing for the adaptive sequencing of services the integrity of the activity is not compromised. It is for this reason that a composition generated in a single step by an AI Planner is not desirable as the planner does not take into account the order in which things should happen, only the desired outcome. In contrast, the strategy driven narrative is ideally suited to the task as its driving principle is to embody the strategy while allowing for adaptivity to be layered on top of the scaffolding that the strategy based narrative elements provide. In the adaptive composition of activities the strategy is the sequencing of the tasks and services that make up the activity.

The problem of adaptively sequencing services as part of an eLearning activity was addressed by taking a hybrid approach that combined AH inspired adaptation techniques with service orchestration. An Adaptation Engine is utilised to carry out the adaptation of the PWE with the Narrative Model playing a key role as the embodiment of the pedagogical strategy or sequencing of the services. This is akin to the role of Narrative in content based personalisation as in both cases the pedagogy directly influences the structure of the composition. This common underlying principle also facilitates the unification of content and service composition, allowing the sequencing of both content and services to be carried out in a consistent manner that takes into account the the needs of both in the context of a single over arching pedagogy. By using the AE to generate compositions that can be delivered using standard workflow technologies, it is then possible to ensure that the sequencing of the PWE can be enforced as the Learner interacts with the activity, thus maintaining the pedagogical integrity of the learning activity from design through personalisation to delivery.

The mechanism through which adaptive selection is achieved is also significant. As discussed in this thesis, AI Planning techniques are ideally suited to the dynamic selection of services as they take into account both the parameters of a service and also the service's preconditions and effects. It can be argued that it is the preconditions and effects that are of most significance in the composition of services as part of a PWE. However, as mentioned previously AI Planning in the context of this research also has its limitations. Based on these two observations, the use of AI Planning has been confined to the role of selecting appropriate services to instantiate a single task within an activity. As such the issues associated with the sequencing of services are mitigated. However, the use of AI Planning as the mechanism through which selection is achieved only affords us with dynamism but it can be argued that it is not adaptive. The adaptive aspect of the selection mechanism is achieved through the configuration of the AI Planner by the AE. By leveraging the capabilities of the AE to reconcile the various metadata models available to the system it is possible to influence the selection process of the planner and thus achieve adaptive selection.

The limitations of AI Planning have been discussed previously and have been cited as the justification for some of the design choices made in this research. However, those same capabilities of AI Planning can be used to the advantage of the system in specific scenarios. When faced with a set of criteria for a service that cannot be satisfied by a single available service, the AI Planner can overcome this by dynamically composing two or more services to meet the specified criteria. A system that used a selection mechanism akin to that used for content would fail if faced with the same scenario. As mentioned previously, this approach can also be used to dynamically generate less explicitly defined sequences of services.

7.2.3 Evaluation

The third objective of this research was to carry out a detailed evaluation of the implemented system. This evaluation investigated the complexity and performance of the system as well as its usability from the perspective of the relevant stakeholders and was achieved through the execution of four experiments.

The first of these experiment was designed to evaluate the ability to describe eLearning activities using the 5 workflow patterns that the system supports for the specification of control flow between services. As such it aims to evaluate the system's ability sequence services to deliver a wide range of eLearning activities. This experiment consisted of an analysis of a set of independently developed eLearning activities with respect to the functionality provided by the system implemented. The results of this analysis showed that the workflow patterns supported by the system provided a sufficient range of control flow in order to support the implementation of such activities.

The aim of the second experiment was to evaluate the ability of the system to adaptively sequence or select interactive services. This was achieved through the implementation of three use cases, based on the learning activities defined by the LADiE project. The subsequent analysis of the implemented use cases verified that the implemented system could deliver personalised learning activities.

The third experiment investigated the system's support for the adaptive selection and sequencing of multimedia content. It consisted of a user based trial that was run using a personalised course designed to teach SQL. This results obtained from this experiment showed that the system and the functionality it provided were sufficient to support the effective delivery of such a personalised course, making use of both adaptive selection and sequencing.

The fourth experiment investigated the performance of the system during both the generation of a PWE and the run-time interactions of the end user with the system. This experiment involved the benchmarking of both processes, the results of which showed that the system could handle the type of loads that might be expected in real world deployment.

7.3 Contribution to the State of the Art

The major contribution of the research described in this thesis to the state of the art is an integrated approach to the combination of multimedia content based selection and composition with workflow based service selection and composition for personalisation. This approach has been realised in the form of a system capable of the unified delivery of adaptively selected and sequenced multimedia content and adaptively selected and sequenced services. As shown in the state of the art review of adaptive content composition systems, adaptivity can bring real benefits to the learner yet such systems do not take into account the need for services to be included as fundamental components of an eLearning experience. The motivation for this was that the current, content centric, adaptive systems do not facilitate the educator in applying activity based pedagogical strategies. By supporting both the adaptive content and services, the system described in this thesis allows the power of personalisation to be applied to activity based eLearning.

The second minor contribution is the novel use of AI planning techniques for the adaptive selection of services. In this thesis, it was observed that a pure AI planning approach is not suitable for the composition of services to realise an eLearning activity due to the requirement for a pedagogically inspired control flow that is often predetermined at design time. To address this an approach was taken in which AI Planning techniques were used as a service selector, akin to the notion of a candidate content selection. Through the manipulation of the planning domain, the Adaptive Engine is able to extend the inherent adaptive behaviour of the AI Planner while still taking advantage of the planner's ability to handle the features of the services to be composed such as their input and output parameters.

- O'Keeffe, I., Staikopoulos, A., Rafter, R., Walsh, E.,Yousuf, B., Conlan, O. & Wade, V. (2012). Personalized Activity Based eLearning. Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies (pp. 2:1-2:8). ACM New York.

- O'Keeffe, I., & Wade, V. (2012). Personalised Educational Activities. 12th IEEE International Conference on Advanced Learning Technologies (ICALT 2012) (pp. 460-461). IEEE. DOI: 10.1109/ICALT.2012.178

- Staikopoulos, A., OKeeffe, I., Rafter, R., Walsh, E., Yousuf, B., Conlan, O. & Wade, V. (2012). AMASE: A Framework for Composing Adaptive and Personalised Learning Activities on the Web. Advances in Web-Based Learning-ICWL 2012, (pp. 190-199). Springer Berlin / Heidelberg.

- O'Keeffe, I., & Wade, V. (2009). Personalised Web Experiences: Seamless Adaptivity across Web Service Composition and Web Content. User Modeling, Adaptation and Personalization (pp. 480-485). Springer Berlin / Heidelberg.

- Conlan, O., O'Keeffe, I., Brady, A., & Wade, V. (2007). Principles for Designing Activity-based Personalized eLearning. Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007) (pp. 642-644). IEEE. DOI: 10.1109/ICALT.2007.209.

- O'Keeffe, I., Conlan, O., & Wade, V. (2006). A Unified Approach to Adaptive Hypermedia Personalisation and Adaptive Service Composition. In V. Wade, H. Ashman, & B. Smyth (Eds.), Adaptive Hypermedia and Adaptive Web Based Systems (pp. 303-307). Springer Berlin / Heidelberg.

- Conlan, O., O'Keeffe, I., & Tallon, S. (2006). Combining Adaptive Hypermedia Techniques and Ontology Reasoning to Produce Dynamic Personalized News Services. In V. Wade, H. Ashman, & B. Smyth (Eds.), Adaptive Hypermedia and Adaptive Web Based Systems (pp. 81-90). Springer-Verlag.

- O'Keeffe, I., Conlan, O., Wade, V., & Brady, A. (2006). Just-in-time Generation of Pedagogically Sound, Context Sensitive Personalized Learning Experiences. IJeL, 5(1), 113127.

7.4 Future Work

7.4.1 Collaborative Activities

The framework implemented as part of this research made the assumption that the learning experience was one in which Learners were individuals. Group based activities such as those described in the LADiE activities, which placed a significant importance on the interactions of Learners as part of a group were not taken into account when considering how personalisation should be achieved. The assumption was that any collaborative functionality would be handled by the application built on top of the framework. However, in such an application careful consideration would have to be given to where adaptive behaviours were applied so that the PWE generated for each individual in the group would be compatible with the PWE of the other members of the group.

An example of where this might break down might be if the system adaptively selected an instant messaging service for one Learner in a group while at the same time selecting a VOIP service for another Learner in the same group. Obviously this could result in a significant problem as the two communication mechanisms would not be compatible. Two possible ways in which this issue could be addressed might be to provide an interoperation mechanism for the services or alternatively to extend the framework so that it was capable of taking into account the context of the Learner, which in this case would be the group that they were a member of. The first approach of providing services to support interoperation could be achieved through the provision of services that were able to transform the inputs and outputs of services so that they were compatible. This approach would require a range of such transform services to be made available but once this was done the inherent functionality of the AI planner would be able to incorporate the necessary services into the composition automatically. The second approach of supporting adaptation in the context of a group would be an interesting problem as a balance would need to be found between the needs of the individual and the constraints placed upon the composition by the group.

7.4.2 Support for Authoring of Personalised Web Experiences

The development of personalised activities is currently a process that requires the development of the necessary metadata models and adaptation rules by hand. This is a complex process requiring technical expertise. For personalised activities to be adopted by eLearning practitioners they need to be easily developed by non technical users.

Authoring tools such as the ACCT have been developed to support the authoring of educationally sound content based personalied eLearning by non technical users. Similarly, tools such as the LAMS and DialogPlus authoring tools have been developed to support the authoring of non adaptive eLearning activities. To make personalised activies, such as those developed as part of this research, accessable to eLearning practitioners a next generate authoring tool is needed that takes the pedagogical and personalisation capabilities of tools such as ACCT and combines them with support for activities.

Bibliography

[Active Endpoints 10] Active Endpoints. *ActiveVOS.* http://www.activevos.com/, 2010.

[Adams 06] Michael Adams, Arthur H. M. ter Hofstede, David Edmond & Wil M. P. van der Aalst. *Worklets : A Service-Oriented Implementation of Dynamic Flexibility in Workflows.* In Proceedings of the 14th International Conference on Cooperative Information Systems (CoopIS 06), pages 291–308, 2006.

[ADL Initiative 09] ADL Initiative. *The Sharable Content Object Reference Model (SCORM) 4th Edition.* http://www.adlnet.gov/Technologies/scorm, 2009.

[Apache Foundation] Apache Foundation. *Xalan.* http://xml.apache.org/xalan-j/.

[Apache Foundation 06] Apache Foundation. *WSRP4J.* http://portals.apache.org/wsrp4j/, 2006.

[Apache Foundation 07] Apache Foundation. *Xindice.* http://xml.apache.org/xindice/, 2007.

[Apache Foundation 08] Apache Foundation. *Pluto 1.1.5.* http://portals.apache.org/pluto/, 2008.

[Apache Foundation 09] Apache Foundation. *Axis2.* http://ws.apache.org/axis2/, 2009.

[Apache Foundation 10a] Apache Foundation. *Apache JMeter.* http://jakarta.apache.org/jmeter, 2010.

[Apache Foundation 10b] Apache Foundation. *Apache ODE.* http://ode.apache.org, 2010.

[Apache Foundation 10c] Apache Foundation. *Tomcat.* http://tomcat.apache.org, 2010.

[Ardissono 10] Liliana Ardissono, Roberto Furnari, Anna Goy, Giovanna Petrone & Marino Segnan. *The Context Aware Workflow Execution Framework.* International Journal Autonomous and Adaptive Communications Systems, Special Issue on: Context awareness for Self-Managed Systems, pages 1–21, 2010.

[Arpinar 05] I. Budak Arpinar, Ruoyan Zhang, Boanerges Aleman-Meza & Angela Maduko. *Ontology-driven Web services composition platform.* Information Systems and e-Business Management, vol. 3, no. 2, pages 175–199, July 2005.

[Bailey 06] Chris Bailey, Mohd T Zalfan, Hugh C Davis, Karen Fill & Gráinne Conole. *Panning for Gold : Designing Pedagogically-inspired Learning Nuggets.* Educational Technology & Society, vol. 9, no. 1, pages 113–122, 2006.

[Bangor 09] Aaron Bangor, Philip Kortum & James Miller. *Determining What Individual SUS*

Scores Mean : Adding an Adjective Rating Scale. Journal of Usability Studies, vol. 4, no. 3, pages 114–123, 2009.

[Barah 00] Sasha A Barah & Thomas M Duffy. *From Practice Fields to Communities of Practice.* In David H. Jonassen & Susan M. Land, editors, Theoretical Foundations of Learning Environments, chapter 2, pages 25–55. Lawrence Erlbaum Associates, Mahwah, New Jersey, 2000.

[Barrows 80] Howard S. Barrows & Robyn M. Tamblyn. *Problem-Based Learning: An Approach to Medical Education*, 1980.

[Berlanga 08] Adriana J Berlanga & Francisco J García-Peñalvo. *Learning Design in Adaptive Educational Hypermedia Systems.* Journal of Universal Computer Science, vol. 14, no. 22, pages 3627–3647, 2008.

[Berners-Lee 05] Tim Berners-Lee. *Berners-Lee on the read/write web.* http://news.bbc.co.uk/2/hi/technology/4132 2005.

[Blackboard Inc. 10] Blackboard Inc. *Blackboard.* http://www.blackboard.com, 2010.

[Blum 97] Avrim Blum & Merrick Furst. *Fast planning through planning graph analysis.* Artificial In, vol. Artificial, no. 90, pages 281–300, 1997.

[Boyle 94] Craig Boyle & Antonio O. Encarnacion. *MetaDoc: An Adaptive Hypertext Reading*

System. User Modeling and User-Adapted Interaction, vol. 4, pages 1–19, 1994.

[Brady 04] Aoife Brady, Owen Conlan & Vincent Wade. *Dynamic Composition and Personalization of PDA-based eLearning - Personalized mLearning.* In Griff Richards, editor, E-Learn 2004, World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education, pages 234–242. AACE, 2004.

[Brooke 96] John Brooke. *SUS: a "quick and dirty" usability scale.* In Patrick W. Jordan, Bruce Thomas, Bernard A. Weerdmeester & Ian L. McClelland, editors, Usability Evaluation in Industry, pages 189–194. Taylor and Francis., 1996.

[Brusilovsky 96a] Peter Brusilovsky. *Methods and Techniques of Adaptive Hypermedia.* User Modeling and User-Adapted Interaction, vol. 6, no. 2-3, pages 87–129, 1996.

[Brusilovsky 96b] Peter Brusilovsky, Elmar Schwarz & Gerhard Weber. *ELM-ART: An intelligent tutoring system on world wide web.* In Claude Frasson, Gilles Gauthier & Alan Lesgold, editors, Intelligent Tutoring Systems, Proceedings of the Third International Conference, ITS '96, pages 261–269. Springer, 1996.

[Brusilovsky 98] Peter Brusilovsky, John Eklund & Elmar Schwarz. *Web-based education for all: a tool for development adaptive courseware.*

Computer Networks and ISDN Systems, vol. 30, no. 1-7, pages 291–300, April 1998.

[Brusilovsky 01] Peter Brusilovsky. *Adaptive Hypermedia.* User Modeling and User-Adapted Interaction, vol. 11, no. 1, pages 87–110, March 2001.

[Brusilovsky 02] Peter Brusilovsky & Riccardo Rizzo. *Using maps and landmarks for navigation between closed and open corpus hyperspace in Web-based education.* The New Review of Hypermedia and Multimedia, vol. 9, 2002.

[Brusilovsky 04a] Peter Brusilovsky. *KnowledgeTree: A distributed architecture for adaptive e-learning.* In Proceedings of the 13th International World Wide Web Conference, WWW, pages 104–113, 2004.

[Brusilovsky 04b] Peter Brusilovsky, Girish Chavan & Rosta Farzan. *Social Adaptive Navigation Support for Open Corpus Electronic Textbooks.* In Wolfgang Nejdl & Paul De Bra, editors, Proceedings of the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, pages 24–33. Springer, 2004.

[Brusilovsky 05a] Peter Brusilovsky, Sergey Sosnovsky & Olena Shcherbinina. *User Modeling in a Distributed E-Learning Architecture.* In Liliana Ardissono, Paul Brna & Antonija Mitrovic, editors, Proceedings of the 10th International Conference on User Modeling (UM 2005), pages 387–391. Springer, 2005.

[Brusilovsky 05b] Peter Brusilovsky, Sergey Sosnovsky & Michael Yudelson. *Ontology-based framework for user model interoperability in distributed learning environments.* In G. Richards, editor, Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, pages 2851–2855. AACE, 2005.

[Brusilovsky 05c] Peter Brusilovsky & Michael Yudelson. *NavEx: Providing Navigation Support for Adaptive Browsing of Annotated Code Examples.* In Proceedings of the 12th International Conference on Artificial Intelligence in Education, AI-Ed'2005, pages 710–717, 2005.

[Brusilovsky 08] Peter Brusilovsky & Michael Yudelson. *From WebEx to NavEx: Interactive Access to Annotated Program Examples.* Proceedings of the IEEE, vol. 96, no. 6, pages 990–999, June 2008.

[Chan 08] Pat Pik Wah Chan & Michael R. Lyu. *Dynamic Web Service Composition: A New Approach in Building Reliable Web Service.* In Proceedings of the 22nd International Conference on Advanced Information Networking and Applications (AINA 2008), pages 20–25. IEEE Computer Society, March 2008.

[Citrix 06] Citrix. *Xen Hypervisor.* http://xen.org, 2006.

[Conklin 87] Jeff Conklin. *Hypertext: a Survey and*

Introduction. IEEE Computer, vol. 20, no. 9, pages 17–41, September 1987.

[Conlan 02] Owen Conlan, Vincent Wade, Catherine Bruen & Mark Gargan. *Multi-model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning.* In Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, pages 100–111, London, UK, 2002. Springer-Verlag.

[Conlan 03a] Owen Conlan, Cord Hockemeyer, Vincent Wade & Dietrich Albert. *Metadata Driven Approaches to Facilitate Adaptivity in Personalized eLearning Systems.* Journal of the Japanese Society for Information and Systems in Education, vol. 1, no. 1, pages 38–45, 2003.

[Conlan 03b] Owen Conlan, Dave Lewis, Steffen Higel, Declan O'Sullivan & Vincent Wade. *Applying adaptive hypermedia techniques to semantic web service composition.* In Proceedings of the Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, The Twelfth International World Wide Web Conference (WWW2003), pages 53–62, 2003.

[Conlan 04] Owen Conlan & Vincent Wade. *Evaluation of APeLS An Adaptive eLearning Service Based on the Multi-Model, Metadata-Driven Approach.* In Paul

De Bra & Wolfgang Nejdl, editors, Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2004), pages 291–295. Springer Berlin / Heidelberg, 2004.

[Conlan 05] Owen Conlan. *The Multi-Model, Metadata driven approach to Personalised eLearning Services.* Phd, Trinity College Dublin, 2005.

[Conole 05] Gráinne Conole & Karen Fill. *A learning design toolkit to create pedagogically effective learning activities.* Journal of Interactive Media in Education, vol. 2005, no. 8, pages 1–16, 2005.

[Consortium 06] IMS Global Learning Consortium. *IMS Metadata.* http://www.imsglobal.org/metadata, 2006.

[Crews 97] Thad Crews, Gautam Biswas, Susan Goldman & John Bransford. *Anchored Interactive Learning Environments.* International Journal of AI in Education, vol. 8, 1997.

[Curbera 01] Francisco Curbera, Sanjiva Weerawarana, Erik Christensen & Greg Meredith. *Web Services Description Language (WSDL) 1.1.* March 2001.

[D 05] Keclercq D & M. Poumay. *The 8 Learning Events Model.* Release 2005.1. LabSET, University of Liège, 2005.

[Dagger 03] Declan Dagger, Owen Conlan & Vincent Wade. *An Architecture for Candidacy in Adaptive eLearning Systems to Facilitate the Reuse of Learning Resources.* In A. Rossett, editor, Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2003, pages 49–56, Chesapeake, VA, 2003. AACE.

[Dagger 05] Declan Dagger, Owen Conlan & Vincent Wade. *Personalisation for All: Making Adaptive Course Composition Easy.* Special issue of the IEEE Journal of Educational Technology and Society, vol. 8, no. 3, pages 9–25, July 2005.

[Dalziel 03] James Dalziel. *Implementing Learning Design: The Learning Activity Management System (LAMS).* In Geoffrey Crisp, Di Thiele, Ingrid Scholten, Sandra Barker & Judi Baron, editors, 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE), pages 593–596, Adelaide, December 2003. ASCILITE.

[DAML-S Coalition 02] DAML-S Coalition, Anupriya Ankolekar, Mark Burstein, Jerry R Hobbs, Ora Lassila, Drew Mcdermott, David Martin, Sheila A Mcilraith, Massimo Paolucci, Terry Payne & Katia Sycara. *DAML-S : Web Service Description for the Semantic Web.* In Proceedings of the 1st

International Semantic Web Conference (ISWC), pages 348–363, 2002.

[De Bra 98] Paul De Bra & Licia Calvi. *AHA! An open Adaptive Hypermedia Architecture.* New Review of Hypermedia and Multimedia, vol. 4, no. 1, pages 115–139, 1998.

[De Bra 99] Paul De Bra, Peter Brusilovsky & Geert-Jan Houben. *Adaptive hypermedia: from systems to framework.* ACM Computing Surveys, vol. 31, no. 12-4es, December 1999.

[De Bra 00] Paul De Bra. *Pros and Cons of Adaptive Hypermedia in Web-based Education.* Journal on Cyber Psychology and Behavior, vol. 3, no. 1, pages 71–77, 2000.

[De Bra 02a] Paul De Bra, Ad Aerts & Brendan Rousseau. *Concept Relationship Types for AHA ! 2 . 0.* In M. Driscoll & T. Reeves, editors, Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, pages 1386–1389, Chesapeake, VA, 2002. AACE.

[De Bra 02b] Paul De Bra, Ad Aerts, David Smits & Natalia Stash. *AHA! Version 2.0: More Adaptation Flexibility for Authors.* In Proceedings of the World Conference on eLearning (eLearn), pages 240–246. AACE, 2002.

[De Bra 03] Paul De Bra, Ad Aerts, Bart Berden & Barend De Lange. *Escape from the*

Tyranny of the Textbook : Adaptive Object Inclusion in AHA ! In Proceedings of the AACE ELearn 2003 Conference, pages 65–71, 2003.

[De Bra 06] Paul De Bra, David Smits & Natalia Stash. *The Design of AHA!* In Proceedings of the seventeenth conference on Hypertext and hypermedia, pages 171–195, New York, NY, USA, 2006. ACM.

[Del Cid 07] Jose Pablo Escobedo Del Cid, Luis de La Fuente Valentín, Sergio Gutiérrez, Abelardo Pardo & Carlos Delgado Kloos. *Implementation of a Learning Design Run-Time Environment for the .LRN Learning Management System.* Journal of Interactive Media in Education, special edition on Adaptation and IMS Learning Design, vol. 2007, no. 01, pages 1–12, 2007.

[Dewey 38] John Dewey. Experience and education. Collier Books, New York, 1938.

[Dias 99] Paulo Dias, Maria Joao Gomes & Ana Paula Correia. *Disorientation in Hypermedia Environments: Mechanisms to Support Navigation.* Educational Computing Research, vol. 20, no. 2, pages 93–117, 1999.

[Dufresne 00] Aude Dufresne & Gilbert Paquette. *ExploraGraph : A Flexible and Adaptive Interface to Support Distance Learning.* In Jacqueline Bourdeau & Rachelle Heller, editors, Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications, pages 304–309. AACE, 2000.

[Dustdar 05] Schahram Dustdar & Wolfgang Schreiner. *A survey on web services composition.* International Journal Web and Grid Services, vol. 1, no. 1, pages 1–30, 2005.

[Eclipse Foundation 10] Eclipse Foundation. *Jetty.* http://www.eclipse.org/jetty/, 2010.

[EL-Manzalawy 04] Yasser EL-Manzalawy. *JPlan: Java GraphPlan Implementation.* http://sourceforge.net/projects/jplan/, 2004.

[Elf 10] *eLearning Framework (ELF).* http://www.elframework.org/, 2010.

[Erol 94] Kutluhan Erol, James Hendler & Dana Nau. *HTN Planning : Complexity and Expressivity.* In Proceedings of the Twelfth National Conference on Artificial Intelligence, pages 1123—-1128. AAAI Press, 1994.

[eXist 10] eXist. *eXist.* http://exist.sourceforge.net/, 2010.

[Falconer 06] Isobel Falconer & Grinne Conole. *LADiE Gap Analysis.* http://www.elframework.org/refmodels/ladie/guides/I 2006.

[Fikes 71] Rirchard Fikes & Nils Nilsson. *STRIPS: A new approach to the application of theorem proving to problem-solving.* Artificial

Intelligence, vol. 2, no. 3, pages 189–208, 1971.

[Fox 03] Maria Fox & Derek Long. *PDDL2.1: An extension to PDDL for expressing temporal planning domains.* Journal of Artificial Intelligence Research, no. 20, pages 61–124, 2003.

[Friedman-Hill 08] Ernest Friedman-Hill. *JESS Expert System.* http://herzberg.ca.sandia.gov/jess/, 2008. Accessed October 2011.

[Fuentes 06] José María Fuentes, Jorge E López de Vergara & Pablo Castells. *An Ontology-Based Approach to the Description and Execution of Composite Network Management Processes for Network Monitoring.* In Proceedings of 17th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM), pages 86—-97, 2006.

[Gagné 85] Robert Gagné. The Conditions of Learning. Holt, Rinehart and Winston, New York, 4th edition, 1985.

[Gerevini 04] Alfonso Gerevini, Alessandro Saetti, Ivan Serina & Paolo Toninelli. *LPG-TD : a Fully Automated Planner for PDDL2.2 Domains.* In Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS-04), 2004.

[Ghedira 06] Chirine Ghedira & H Mezni. *Through Personalized Web Service Composition Specification: From BPEL to C-BPEL.* Electronic Notes in Theoretical Computer Science, vol. 146, no. 1, pages 117–132, 2006.

[Glassfish Community 11] Glassfish Community. *GlassFish Server.* http://glassfish.java.net/, 2011.

[Goguadze 05] Giorgi Goguadze, Alberto González Palomo & Erica Melis. *Interactivity of Exercises in ActiveMath.* In Proceedings of the 13th International Conference on Computers in Education, numéro 1, pages 107–113. IOS Press, 2005.

[Greeno 96] James G. Greeno, Allan M. Collins & Lauren B. Resnick. *Cognition and learning. Handbook of educational psychology.* In David C. Berliner & Robert C. Calfee, editors, Handbook of educational psychology, pages 15–46. Macmillan Library Reference USA; Prentice Hall International, New York, 1996.

[Gutierrez-Santos 08] Sergio Gutierrez-Santos, Abelardo Pardo & Carlos Delgado Kloos. *Authoring Courses with Rich Adaptive Sequencing for IMS Learning Design.* Journal of Universal Computer Science, vol. 14, no. 17, pages 2819–2839, 2008.

[Halasz 90] Frank Halasz & Mayer Schwartz. *The Dexter Hypertext Reference.*

Communications of the ACM, vol. 37, no. 2, pages 30–39, 1990.

[Hart 68] Peter Hart, Nils Nilsson & Bertram Raphael. *A Formal Basis for the Heuristic Determination of Minimum Cost Paths.* IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pages 100–107, 1968.

[Hatzi 09] Ourania Hatzi, Georgios Meditskos, Dimitris Vrakas, Nick Bassiliades, Dimosthenis Anagnostopoulos & Ioannis Vlahavas. *PORSCE II : Using Planning for Semantic Web Service Composition.* In Proceedings of the International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS), in conjunction with the International Conference on Automated Planning and Scheduling (ICAPS-09), pages 38–45, 2009.

[Hendrix 09] Maurice Hendrix, Alexandra Cristea & Daniel Burgos. *Comparative Analysis of Adaptation in Adaptive Educational Hypermedia and IMS- Learning Design.* Rapport technique, Technical Report RR443, Department of Computer Science, The University of Warwick, UK, 2009.

[Henze 05] Nicola Henze. *Personal Readers: Personalized Learning Object Readers for the Semantic Web.* In Proceeding of the 12th International Conference on

Artificial Intelligence in Education, pages 274–281. IOS Press, 2005.

[Hewett 07] Micheal Hewett. *Jatha - Common LISP library in Java.* http://jatha.sourceforge.net/, 2007.

[IBM 08] IBM. *IBM WebSphere.* www.ibm.com/websphere, 2008.

[IBM 10] IBM. *IBM Mashup Center.* http://www-01.ibm.com/software/info/mashup-center/, 2010.

[IMS 03] IMS. *Learning Design Information Model, Version 1.0.* http://www.imsglobal.org/learningdesign/, 2003.

[IMS 05] IMS. *IMS Learner Information Package (LIP).* http://www.imsglobal.org/profiles, 2005.

[JackBe 10] JackBe. *JackBe Presto Server.* http://www.jackbe.com/products/server.php, 2010.

[JAX 10] *Java API for XML Web Services (JAX-WS).* https://jax-ws.dev.java.net/, 2010.

[JBOSS Community 11] JBOSS Community. *JBOSS Riftsaw.* http://www.jboss.org/riftsaw, 2011.

[JBoss 11] JBoss. *jBPM.* http://www.jboss.org/jbpm, 2011.

[JDOM Project 07] JDOM Project. *JDOM 1.1.* http://www.jdom.org/, 2007.

[Jeffery 06] A. Jeffery, Gráinne Conole & Isobel Falconer. *LADiE Project Final Report.* Rapport technique April, University of Southampton, 2006.

[Klusch 05] Matthias Klusch, Andreas Gerber & Marcus Schmidt. *Semantic Web Service Composition Planning with OWLS-Xplan.* In Proceedings of the 1st International AAAI Fall Symposium on Agents and the Semantic Web, pages 55—-62, 2005.

[Knutov 09] Evgeny Knutov, Paul De Bra & Mykola Pechenizkiy. *AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques.* New Review of Hypermedia and Multimedia, vol. 15, no. 1, pages 5–38, 2009.

[Kobsa 94] Alfred Kobsa, Dietmar Müller & Andreas Nill. *KN-AHS : An Adaptive Hypertext Client of the User Modeling System BGP-MS.* In Proceedings of the Fourth International Conference on User Modeling, pages 99–105, 1994.

[Koch 02] Nora Koch & Martin Wirsing. *The Munich Reference Model for Adaptive Hypermedia Applications.* In Paul De Bra, Peter Brusilovsky & Ricardo Conejo, editors, Adaptive Hypermedia and Adaptive Web-Based Systems, volume 2347 of *Lecture Notes in Computer Science*, pages 213–222, Berlin, Heidelberg, May 2002. Springer Berlin Heidelberg.

[Kohlhase 06] Michael Kohlhase. OMDoc An Open Markup Format for Mathematical Documents [version 1.2], volume 4180 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006.

[Lave 91] Jean Lave & Etienne Wenger. Situated Learning. Legitimate peripheral participation. University of Cambridge Press, Cambridge, 1991.

[Leymann 01] Frank Leymann. *Web Services Flow Language (WSFL 1.0)*. Rapport technique, IBM Software Group, 2001.

[Liferay Inc 10] Liferay Inc. *Liferay*. http://www.liferay.com/, 2010.

[Martens 05] Harrie Martens & Hubert Vogten. *A Reference Implementation of a Learning Design Engine*. In Rob Koper & Colin Tattersall, editors, Learning Design, chapter 6, pages 91–108. Springer-Verlag, Berlin/Heidelberg, 2005.

[Martin 04] David Martin, Anupriya Ankolekar, Mark Burstein, Grit Denker, Daniel Elenius, Jerry Hobbs, Lalana Kagal, Ora Lassila, Drew McDermott, Deborah McGuinness, Sheila McIlraith, Massimo Paolucci, Bijan Parsia, Terry Payne, Marta Sabou, Craig Schlenoff, Evren Sirin, Monika Solanki, Naveen Srinivasan, Katia Sycara & Randy Washington. *OWL-S: Semantic Markup for Web Services, version 1.1.*

http://www.daml.org/services/owl-s/1.1/, November 2004.

[Melis 01] Erica Melis, Eric Andres, Jochen Budenbender, Adrian Frischauf, Giorgi Goguadze, Paul Libbrecht, Martin Pollet & Carsten Ullrich. *ActiveMath: A Generic and Adaptive Web-Based Learning Environment.* International Journal of Artificial Intelligence in Education, vol. 12, pages 385–407, 2001.

[Meyes 04] Terry Meyes & Sara De Freitas. *JISC e-Learning Models Desk Study Stage 2: Review of e-learning theories, frameworks and models.* Rapport technique, 2004.

[Micarelli 96] Alessandro Micarelli & Filippo Sciarrone. *A case-based system for adaptive hypermedia navigation.* In Proceedings of the Fifth International Conference on User Modeling UM-96, pages 129–136, 1996.

[Microsoft 08] Microsoft. *SQL Server.* http://www.microsoft.com/sqlserver, 2008.

[Microsoft 09] Microsoft. *Biztalk Server 2009.* http://www.microsoft.com/biztalk, 2009.

[Microsoft 10a] Microsoft. *Microsoft .NET Framework.* http://www.microsoft.com/net, 2010.

[Microsoft 10b] Microsoft. *Microsoft Sharepoint 2010.* http://sharepoint.microsoft.com, 2010.

[Microsoft 10c] Microsoft. *.NET Framework.*

http://www.microsoft.com/net/, 2010. Accessed October 2011.

[Miller 68] R.B Miller. *Response time in man-computer conversational transaction.* In AFIPS Fall Joint Computer Conference, pages 267–277, 1968.

[Moodle Trust 10] Moodle Trust. *Moodle.* http://moodle.org/, 2010.

[Nah 04] Fiona Nah. *A study on tolerable waiting time: how long are Web users willing to wait?* Behaviour & Information Technology, vol. 23, no. 3, pages 153–163, 2004.

[Nau 03] Dana Nau, Tsz-Chiu Au, Okhtay Ilghami, Ugur Kuter, J. William Murdock, Dan Wu & Fusun Yaman. *SHOP2: An HTN Planning System.* Journal of Artificial Intelligence Research, vol. 20, pages 379–404, 2003.

[Netbeans Community 09] Netbeans Community. *Netbeans SOA Project.* http://soa.netbeans.org/, 2009.

[Netbeans Community 11] Netbeans Community. *Netbeans IDE.* http://www.netbeans.org, 2011.

[Nielsen 93] Jakob Nielsen. *Usability Engineering.* chapter 5. Morgan Kaufmann, San Francisco, 1993.

[Oasis Consortium 07] Oasis Consortium. *Web Service Business Process Execution Language.* http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.(2007.

[Oasis Consortium 08] Oasis Consortium. *Web Services for Remote Portlets (WSRP).* http://docs.oasis-open.org/wsrp/v2/wsrp-2. 2008.

[Open Mashup Alliance 09] Open Mashup Alliance. *Enterprise Mashup Markup Language (EMML).* http://www.openmashup.org/omadocs/v1.0/ 2009.

[Oracle 09] Oracle. *BPEL Process Manager.* www.oracle.com/technology/bpel, 2009.

[Oracle 10a] Oracle. *Java API for XML Processing (JAXP).* http://jaxp.java.net/, 2010.

[Oracle 10b] Oracle. *MySQL.* http://www.mysql.com/, 2010.

[Piaget 70] Jean Piaget. Science of Education and the Psychology of the Child. Orion Press, New York, 1st edition, 1970.

[plu 11] *Apache Pluto.* http://portals.apache.org/pluto, 2011. Accessed October 2011.

[Rao 04] Jinghai Rao & Xiaomeng Su. *A survey of automated web service composition methods.* In Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition, SWSWPC. Springer-Verlag, 2004.

[Reload Project 10] Reload Project. *Reload Editor.* http://www.reload.ac.uk/, 2010.

[Rey-lópez 08] Marta Rey-lópez, Maram Meccawy & Peter Brusilovsky. *Resolving the Problem* *of Intelligent Learning Content in Learning Management Systems.* International Journal on E-Learning, vol. 7, no. 3, pages 363–381, 2008.

[Russell 06] Nick Russell, Arthur H M ter Hofstede, Wil M. P. van der Aalst & Nataliya Mulyar. *WORKFLOW CONTROL-FLOW PATTERNS A Revised View.* Rapport technique, BPM Center, 2006.

[Sakai Foundation 10] Sakai Foundation. *Sakai.* http://sakaiproject.org/, 2010.

[Sampson 05] Demetrios Sampson, Pythagoras Karampiperis & Panayiotis Zervas. *Developing Web-Based Learning Scenarios Using the IMS Learning Design : The ASK-LDT Environment.* In Mike Dean, Yuanbo Guo, Woochun Jun, Roland Kaschek, Shonali Krishnaswamy, Zhengxiang Pan & Quan Sheng, editors, International Workshop on Web Information Systems Engineering (WISE), pages 104 – 113. Springer Berlin / Heidelberg, 2005.

[Schneiderman 84] Ben Schneiderman. *Response Time and Display Rate in Human Performance with Computers.* Computing Surveys, vol. 16, no. 3, pages 265–285, 1984.

[Scott Nicklous 03] Martin Scott Nicklous. *JSR-168 Portlet Specification.* http://jcp.org/ja/jsr/detail?id=168, 2003.

[Shen 05] Jun Shen, Yun Yang, Chengang Wan & Chuan Zhu. *From BPEL4WS to OWL-S: Integrating E-Business Process Descriptions.* Proceedings of the 2005 IEEE International Conference on Services Computing - Volume 01, pages 181–190, 2005.

[Sirin 04] Evren Sirin, Bijan Parsia, Dan Wu, James Hendler & Dana Nau. *HTN Planning for Web Service Composition using SHOP2.* Web Semantics: Science, Services and Agents on the World Wide Web, vol. 1, no. 4, pages 377–396, October 2004.

[Smits 11] David Smits & Paul De Bra. *GALE: a highly extensible adaptive hypermedia engine.* In 22nd ACM Conference on Hypertext and Hypermedia, Eindhoven, The Netherlands, June 6-9, pages 63–72, 2011.

[Sosnovsky 04] Sergey Sosnovsky. *Adaptive Navigation for Self-Assessment Quizzes.* In Wolfgang Nejdl & Paul De Bra, editors, Proceedings of the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, pages 366–371, 2004.

[Sun Microsystems 03] Sun Microsystems. *Java Servlet Specification 2.4.* http://jcp.org/aboutJava/communityprocess 2003.

[Sun Microsystems 06] Sun Microsystems. *Common Development and Distribution License (CDDL).*

http://www.opensource.org/licenses/CDDL-1.0, 2006.

[Telecommunication Development Bureau 11] Telecommunication Development Bureau. *Measuring the information society.* Rapport technique 3, International Telecommunication Union (ITU), 2011.

[The Dialog Plus Team 04] The Dialog Plus Team, Michael Freeston & Hugh Davis. *DialogPlus: Digital Libraries in Support of Innovative Approaches to Learning and Teaching in Geography.* In Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries (JCDL04), page 385, 2004.

[The Dublin Core Metadata Initiative 10] The Dublin Core Metadata Initiative. *Dublin Core.* http://dublincore.org/specifications/, 2010.

[The XML:DB Initiative 03] The XML:DB Initiative. *XMLDB.* http://xmldb-org.sourceforge.net/, 2003.

[Tullis 08] Tom Tullis & Bill Albert. Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics. Morgan Kaufmann, Amsterdam, 2008.

[Ullrich 08] Carsten Ullrich & Paul Libbrecht. *Educational Services in the ActiveMath Learning Environment.* In S. Salerno, M. Gaeta, P. Ritrovato, N. Capuano, F. Orciuoli, S. Miranda & A. Pierri, editors, The Learning Grid Handbook Concepts, Technologies and Applications, numéro Section 1, pages 219–244. IOS Press, Amsterdam, 2008.

[Ullrich 09] Carsten Ullrich & Erica Melis. *Pedagogically founded courseware generation based on HTN-planning.* Expert Systems with Applications: An International Journal, vol. 36, no. 5, pages 9319–9332, 2009.

[van der Aalst 98] Wil M. P. van der Aalst. *The Application of Petri Nets to Workflow Management.* Journal of Circuits, Systems, and Computers, vol. 8, no. 1, pages 21–66, 1998.

[van Der Aalst 03] Wil M. P. van Der Aalst, Arthur H. M. ter Hofstede, Bartek Kiepuszewski & Alistair P. Barros. *Workflow Patterns.* Distributed and Parallel Databases, vol. 14, no. 3, pages 5–51, 2003.

[van der Aalst 04] Wil M. P. van der Aalst. *Design and implementation of the YAWL system.* Systems Engineering, 2004.

[Vygotsky 78] Lev Vygotsky. Mind in Society: The development of higher psychological processes. Harvard University Press, Cambridge, MA, 1978.

[W3C 99a] W3C. *XML Path Language (XPath) Version 1.0.* http://www.w3.org/TR/xpath/, 1999.

[W3C 99b] W3C. *XSL Transformations (XSLT)*

Version 1.0. http://www.w3.org/TR/xslt, 1999.

[W3C 04] W3C. *Resource Description Framework (RDF).* http://www.w3.org/RDF/, 2004.

[W3C 07] W3C. *XQuery 1.0: An XML Query Language.* http://www.w3.org/TR/xquery/, 2007.

[W3C 10] W3C. *XML Schema.* http://www.w3.org/XML/Schema, 2010.

[W3C 11] W3C. *XQuery Update Facility.* http://www.w3.org/TR/xquery-update-10/, 2011.

[Watson 25] John B. Watson. Behaviourism. People's Institute Publishing Company, New York, 1925.

[Wieers 10] Dag Wieers. *Dstat.* http://dag.wieers.com/home-made/dstat, 2010.

[Wohed 02] Petia Wohed, Wil M. P. van der Aalst, Marlon Dumas & Arthur H. M. ter Hofstede. *Pattern Based Analysis of BPEL4WS.* Rapport technique, Queensland University of Technology, 2002.

[WSO2 11] WSO2. *WSO2 Mashup Server.* http://wso2.com/products/mashup-server/, 2011.

[Yahoo! 11] Yahoo! *Yahoo! Pipes.* http://pipes.yahoo.com/pipes/, 2011. Accessed October 2011.

Appendix

Appendix A

Evaluation Experiment One

A.1 LADiE Use Case Definitions

Authors

Author 1	Kathy Trinder
Author 2	Janice West

Use Case Summary

The teacher assists students to consolidate their understanding of a theory through an experiential approach based on interviewing a third party

Narrative

Students, in pairs, interview a third party about the theory underlying a particular topic, and then discuss their interview findings with the rest of the class in order to consolidate their understanding of the theory.

Primary Actor (and goal)

Teacher	To use the system to deliver learning materials, facilitate discussion and administer the activity

Other Actors (and goals)

Students	To complete the task successfully
Interviewee	To have an interview that they are comfortable with

Stakeholders and Interests

Quality assurers	To ensure that the students have achieved the learning outcomes
Administrators	To use the system to track student progress etc

Preconditions

1	
2	
3	
4	

Teaching Approach

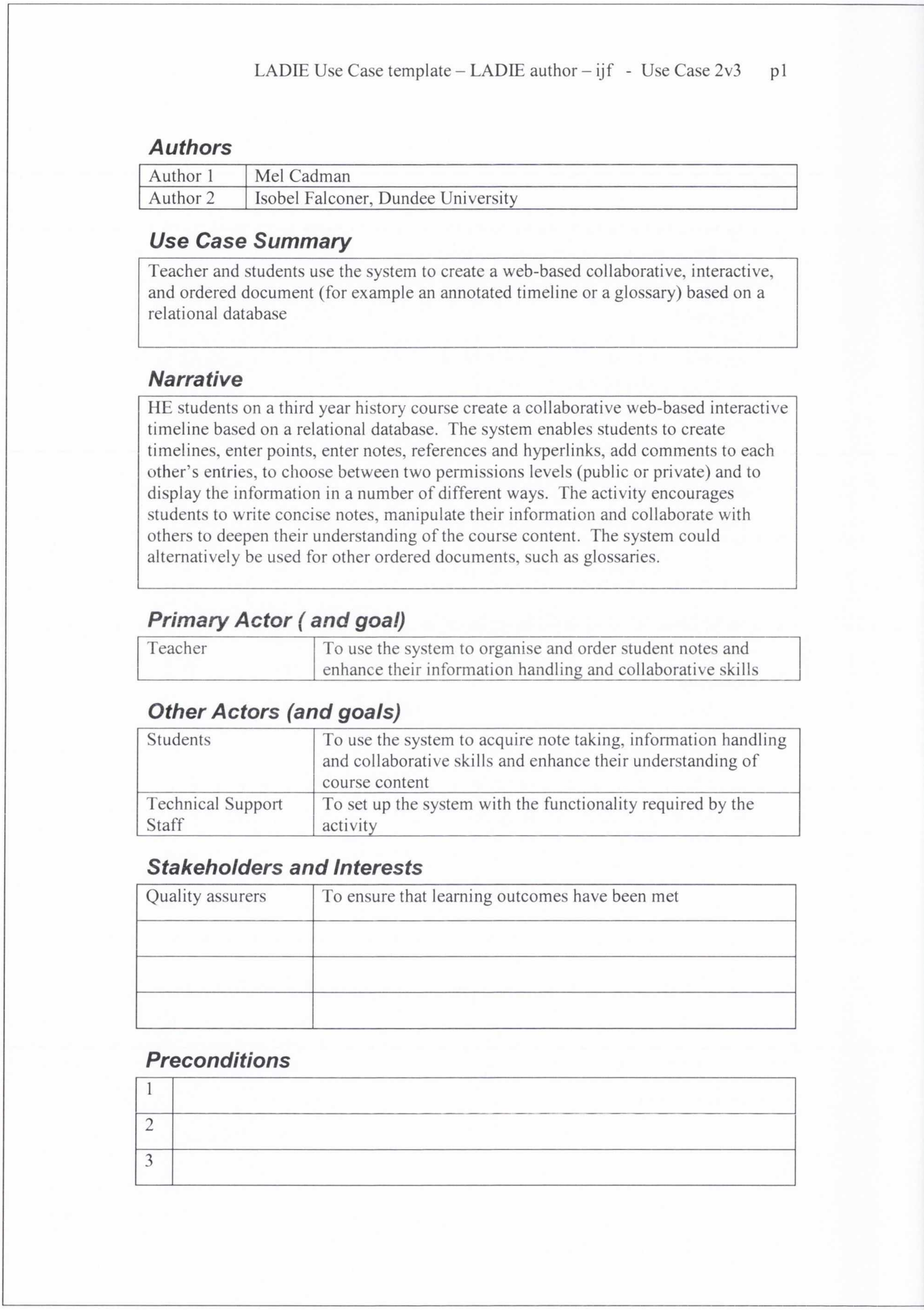
Vicarious learning

Main Success Scenario

1	Teacher creates learning resources (interview guidelines etc) and defines services required and sequence of tasks and saves them to the system
2	Teacher enters student, pair, and class information into the system and defines access permissions for discussion groups
3	Teacher makes arrangements for interviews and student access to recording equipment
4	Teacher briefs students on the activity, shows them how to use the recording equipment, and refers them to resources and interview guidelines in the system
5	Students log into system to retrieve resources, and discuss the activity throughout steps 6-7 (see usecase 'discussion')
6	Student pairs carry out interviews and record them as audio files
7	Students save interview recordings, and documentary records to the system
8	Students write a report or presentation based on interview records and save it to the system
9	Teacher accesses the system throughout the activity to monitor student progress and the discussion
10	Teacher retrieves student work and discussion records for assessment
11	Teacher archives records of the activity for quality assurance, evaluation and future reuse

Extensions

1a	Teacher searches for resources from a repository and imports them
1b	Teacher searches for resources from the web and creates links to them
1c	Teacher retrieves complete learning activity from a repository and updates it
4a	Teacher triggers the activity using an email message to students, referring them to briefing materials on the system
6a	Students record interviews on magnetic tape



Authors

Author 1	Mel Cadman
Author 2	Isobel Falconer, Dundee University

Use Case Summary

Teacher and students use the system to create a web-based collaborative, interactive, and ordered document (for example an annotated timeline or a glossary) based on a relational database

Narrative

HE students on a third year history course create a collaborative web-based interactive timeline based on a relational database. The system enables students to create timelines, enter points, enter notes, references and hyperlinks, add comments to each other's entries, to choose between two permissions levels (public or private) and to display the information in a number of different ways. The activity encourages students to write concise notes, manipulate their information and collaborate with others to deepen their understanding of the course content. The system could alternatively be used for other ordered documents, such as glossaries.

Primary Actor (and goal)

Teacher	To use the system to organise and order student notes and enhance their information handling and collaborative skills

Other Actors (and goals)

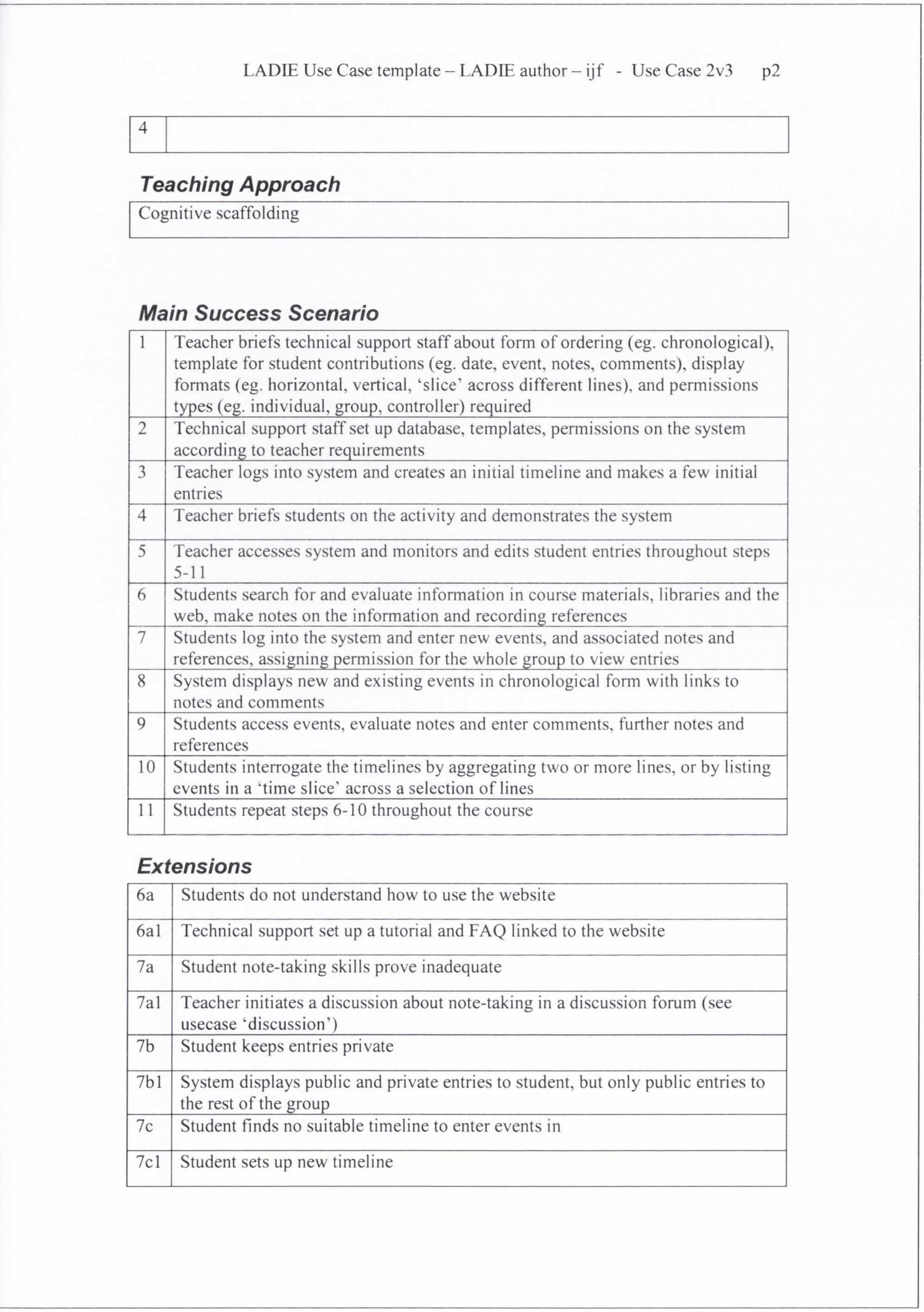
Students	To use the system to acquire note taking, information handling and collaborative skills and enhance their understanding of course content
Technical Support Staff	To set up the system with the functionality required by the activity

Stakeholders and Interests

Quality assurers	To ensure that learning outcomes have been met

Preconditions

1	
2	
3	



4	

Teaching Approach

Cognitive scaffolding

Main Success Scenario

1	Teacher briefs technical support staff about form of ordering (eg. chronological), template for student contributions (eg. date, event, notes, comments), display formats (eg. horizontal, vertical, 'slice' across different lines), and permissions types (eg. individual, group, controller) required
2	Technical support staff set up database, templates, permissions on the system according to teacher requirements
3	Teacher logs into system and creates an initial timeline and makes a few initial entries
4	Teacher briefs students on the activity and demonstrates the system
5	Teacher accesses system and monitors and edits student entries throughout steps 5-11
6	Students search for and evaluate information in course materials, libraries and the web, make notes on the information and recording references
7	Students log into the system and enter new events, and associated notes and references, assigning permission for the whole group to view entries
8	System displays new and existing events in chronological form with links to notes and comments
9	Students access events, evaluate notes and enter comments, further notes and references
10	Students interrogate the timelines by aggregating two or more lines, or by listing events in a 'time slice' across a selection of lines
11	Students repeat steps 6-10 throughout the course

Extensions

6a	Students do not understand how to use the website
6a1	Technical support set up a tutorial and FAQ linked to the website
7a	Student note-taking skills prove inadequate
7a1	Teacher initiates a discussion about note-taking in a discussion forum (see usecase 'discussion')
7b	Student keeps entries private
7b1	System displays public and private entries to student, but only public entries to the rest of the group
7c	Student finds no suitable timeline to enter events in
7c1	Student sets up new timeline

Authors

Author 1	Mhairi McAlpine
Author 2	Mary McCulloch

Use Case Summary

A teacher asks a system to assess a candidate in relation to a particular curricular domain and present appropriate learning resources to the student until mastery is demonstrated

Narrative

The teacher uses the system to generate a multiple choice quiz automatically from a repository of questions, answers and feedback suggestions, and to deliver it to students. The system tracks student progress, and recommends remedial study plans followed by retaking the test until the student demonstrates mastery by 'passing' the test. If the student fails more than a pre-defined number of times they are referred to the teacher for extra help.

Primary Actor (and goal)

Teacher	To use the system to deliver assessment questions and feedback to the students until mastery is demonstrated

Other Actors (and goals)

Students	To use the system to gain skills and knowledge defined by the teacher

Stakeholders and Interests

Preconditions

1	A repository of assessment questions, study plans and learning resources
2	
3	
4	

Teaching Approach

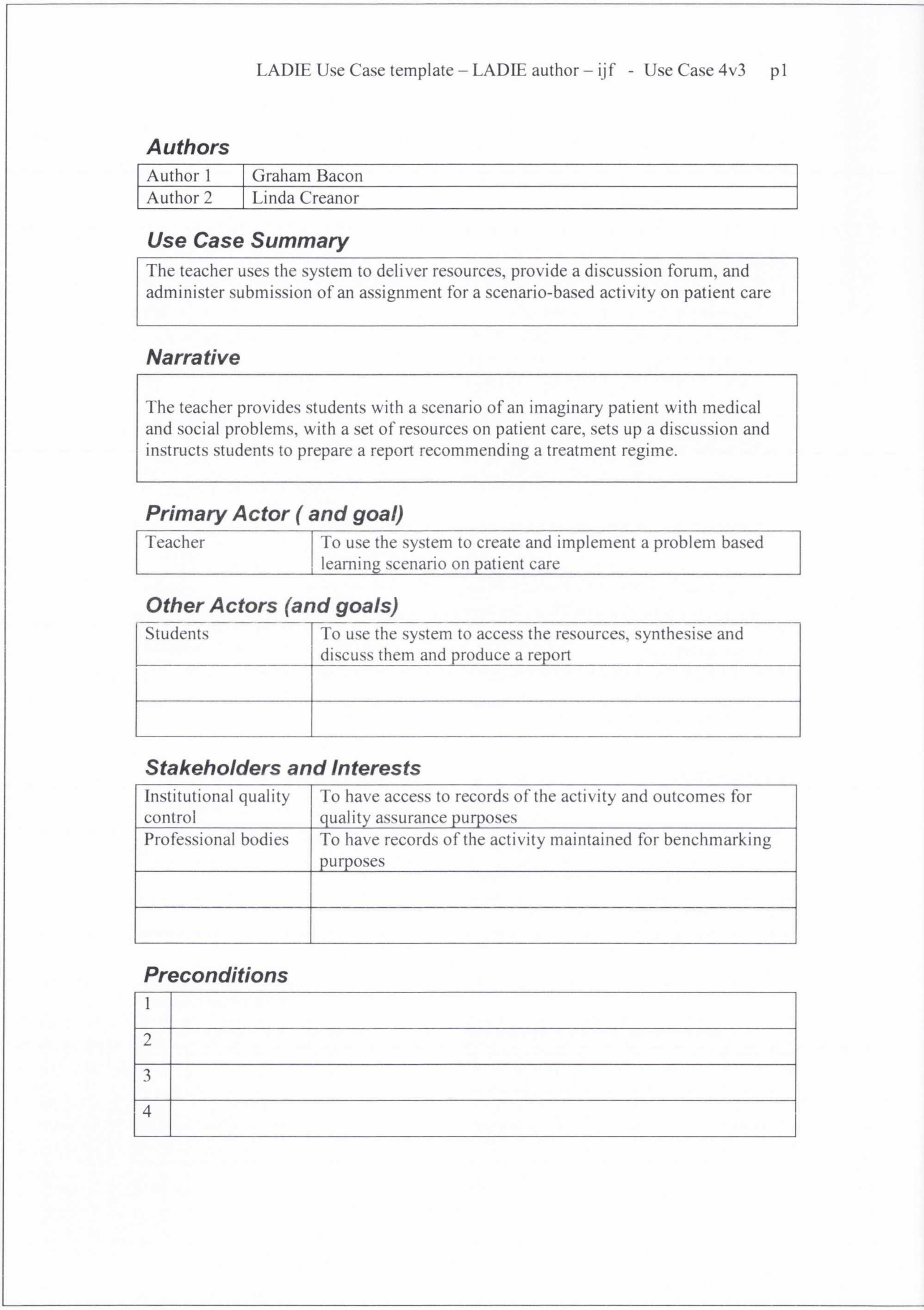
Intelligent tutoring

Main Success Scenario

1	Teacher tells the system the discipline and subdiscipline areas of the test and the learning outcomes to be tested
2	System generates criteria upon which it will select questions from the repository, and presents the criteria to the teacher
3	Teacher edits the criteria and adjusts the weightings if necessary
4	System generates the criteria upon which it will suggest study plans and learning resources to the students based on their answers to questions
5	Teacher edits the criteria and adjusts the weightings if necessary, and defines the maximum number of times students should take the test
6	Teacher monitors activity throughout steps 7-10 and changes tagging of questions, selection criteria for questions and study plans if necessary
7	Student logs into system, which tracks their progress through the test, recording how many times they have taken it, which questions they were presented with, their answers, and the study plans suggested
8	System selects questions from the repository and presents them to the student, ensuring that they are different from those the student has previously been presented with
9	System records student answers to the questions and compares them with the criteria from step 4, and delivers appropriate study plan to student
10	Student works at study plan

Extensions

9a	Student has 'failed' test more than the defined maximum number of times
9a1	System tells student to go and see the teacher and terminates the activity
10a	Study plan recommends that student retakes the test
10a1	Student returns to step 7



Authors

Author 1	Graham Bacon
Author 2	Linda Creanor

Use Case Summary

The teacher uses the system to deliver resources, provide a discussion forum, and administer submission of an assignment for a scenario-based activity on patient care

Narrative

The teacher provides students with a scenario of an imaginary patient with medical and social problems, with a set of resources on patient care, sets up a discussion and instructs students to prepare a report recommending a treatment regime.

Primary Actor (and goal)

Teacher	To use the system to create and implement a problem based learning scenario on patient care

Other Actors (and goals)

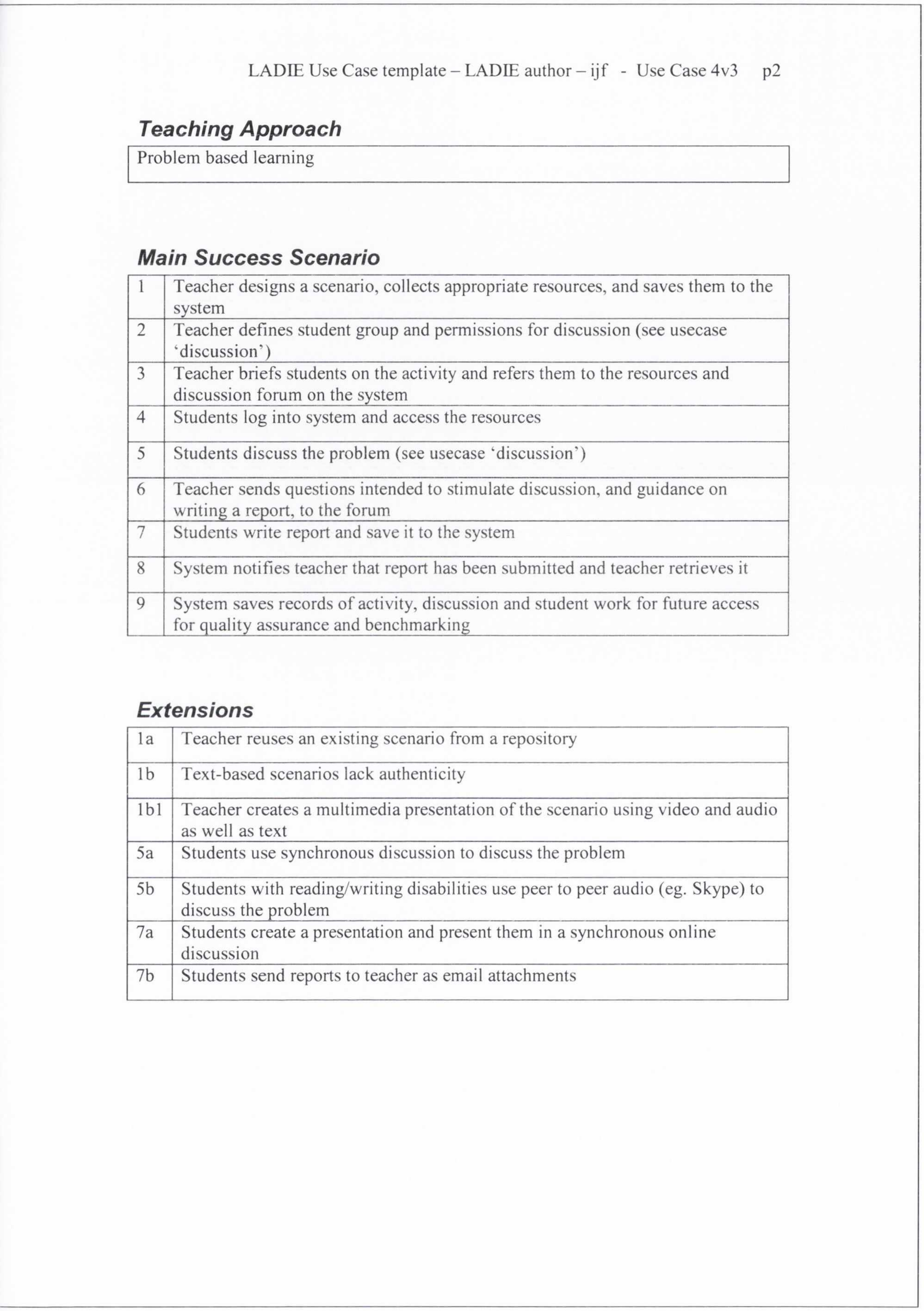
Students	To use the system to access the resources, synthesise and discuss them and produce a report

Stakeholders and Interests

Institutional quality control	To have access to records of the activity and outcomes for quality assurance purposes
Professional bodies	To have records of the activity maintained for benchmarking purposes

Preconditions

1	
2	
3	
4	



LADIE Use Case template – LADIE author – ijf - Use Case 4v3 p2

Teaching Approach

Problem based learning

Main Success Scenario

1	Teacher designs a scenario, collects appropriate resources, and saves them to the system
2	Teacher defines student group and permissions for discussion (see usecase 'discussion')
3	Teacher briefs students on the activity and refers them to the resources and discussion forum on the system
4	Students log into system and access the resources
5	Students discuss the problem (see usecase 'discussion')
6	Teacher sends questions intended to stimulate discussion, and guidance on writing a report, to the forum
7	Students write report and save it to the system
8	System notifies teacher that report has been submitted and teacher retrieves it
9	System saves records of activity, discussion and student work for future access for quality assurance and benchmarking

Extensions

1a	Teacher reuses an existing scenario from a repository
1b	Text-based scenarios lack authenticity
1b1	Teacher creates a multimedia presentation of the scenario using video and audio as well as text
5a	Students use synchronous discussion to discuss the problem
5b	Students with reading/writing disabilities use peer to peer audio (eg. Skype) to discuss the problem
7a	Students create a presentation and present them in a synchronous online discussion
7b	Students send reports to teacher as email attachments

Authors

Author 1	Sheila NcNeill
Author 2	Neill Ballantyne

Use Case Summary

Teacher uses the system to deliver a worksheet on interpreting graphical data to the students, to administer a quiz to test their understanding, and to provide appropriate feedback automatically

Narrative

Students are presented with a worksheet designed to help them interpret data presented in graphical format. They take a diagnostic quiz to check their understanding and receive automatically generated diagnostic feedback on their analysis

Primary Actor (and goal)

Teacher	To use the system to administer the activity

Other Actors (and goals)

Students	To receive learning resources and appropriate feedback from the system

Stakeholders and Interests

Quality Assurance	To have access to records of the activity for QA purposes

Preconditions

1	
2	
3	
4	

Teaching Approach

Active learning

Main Success Scenario

1	Teacher uses the system to create a worksheet presenting data in graphical format, and an associated multiple choice quiz and feedback study plans

2	Students log into the system, which delivers the worksheet and quiz
3	Students work through the worksheet, do the quiz and receive feedback (see usecase 'quiz')
4	System records student progress and results for teacher monitoring and for quality assurance

Extensions

1a	Teacher retrieves worksheet and quiz from a repository and edits them to suit their context
4a	System sends student results to ePortfolio
4b	Teacher archives activity in a repository for future use and sharing

Authors

Author 1	Kevin Brosnan, Stirling University
Author 2	Linda Lafferty, University of Paisley

Use Case Summary

Teacher compiles a collection of resources (video clips), provides students with a framework for analysing the clips, and moderates student discussion

Narrative

The Learning Activity introduces students to an aspect of critical thinking - argument analysis. Students are presented with a series of digitised video clips from a film and asked to analyse/evaluate the clips using a framework that has been presented to them previously. Students are asked to post their analyses to the discussion area and comment on the contributions made by other students in an attempt to encourage reflective thinking.

Primary Actor (and goal)

Teacher	To develop students' critical thinking and reflective skills

Other Actors (and goals)

Moderator	To access the system and moderate student discussion
Student	To complete the task successfully
System	To manage and store teacher and student generated materials and discussion contributions, and to manage teacher and student access to the materials

Stakeholders and Interests

Technical support	To have a system that runs smoothly
Technical developers	To have a system that meets teacher and student requirements
Quality control	To have access to a record of the activity for evaluation purposes
Institution	To deliver courses in a cost and time efficient manner at a sufficient level of quality
Staff developers	To ensure staff have necessary skills for effective teaching

Preconditions

1	

Teaching Approach

Constructivist/Dialogic

Main Success Scenario

1	Teacher locates suitable video clips by searching library catalogue
2	Teacher accesses, edits and saves the clips in an appropriate form for teaching
3	Teacher logs into system
4	System authenticates teacher and gives them controller access to the course area (materials store and discussion forum)
5	Teacher uploads clips to the course
6	Teacher produces briefing materials, including analytical framework, using word processor
7	Teacher uploads briefing materials to the course area
8	Students log in to system and register for the course
9	System assigns student access to the course materials and discussion forum.
10	Teacher chooses a student to moderate the discussion and tells the system
11	System assigns moderator permissions
12	Teacher sends message to forum to trigger activity
13	Students log in to learning activity
14	System authenticates students, and delivers clips and briefing materials
15	Students send contributions to discussion forums
16	Teacher monitors discussion forum and email and responds to student questions, encourages discussion, etc.
17	Teacher logs into discussion forum, reads and may edit, all messages to prevent inappropriate postings etc
18	Teacher summarises the activity
19	Teacher sends a message to forum, formally ending activity
20	System saves all forum contributions as read only for students to refer back to, and for evaluation and quality control
21	Teacher analyses postings for evaluation purposes

Extensions

1a	Teacher cannot find suitable clips in the library.
1a1a	Teacher searches online databases for clips
1a1b	Teacher purchases clips from a database
1b	University network breaks down and teacher cannot access catalogue from own computer
1b1a	Teacher goes to library and tries access from dedicated catalogue terminal

1b2a	Teacher postpones search until another day
2a	Teacher doesn't know how to access, edit or save clips in digital format
2a1	Teacher seeks technical support
3a	Network or computer breaks down and teacher cannot log in
3a1	Teacher postpones work
4a	System does not recognise teacher
4a1	Teacher seeks technical support
4b	System has lost teacher details and will not assign controller access
4b1	Teacher seeks technical support
5a	System cannot handle management and storage of video clips
5a1	Technical developers upgrade system
6a	Word processor drawing facility not sufficiently sophisticated for diagrams teacher needs in briefing materials
6a1	Teacher asks technical support to install drawing software that is compatible with word processor
6a2	Teacher hand draws diagrams and scans them into computer in a format that the word processor will recognise
7a	System cannot store or handle file format of briefing materials
7a1a	Teacher seeks advise from technical support on conversion to a suitable format
7a1b	Teacher converts materials to a suitable format and uploads them
7a2	Teacher prints out hard copy of briefing materials and distributes them to students
7a3	Teacher emails copy of briefing materials to students as an attached file
8a	Some visually impaired students register
8a1a	Teacher locates, edits and uploads alternative audio clips
8a1b	Teacher modifies analytical framework to be suitable for audio clips
8a1c	Teacher seeks technical support to ensure that student-system interface is compatible with screen readers
8b	Too many students register for online discussion to be successful (in the teacher's opinion)
8b1a	Teacher asks system to divide students into groups for discussion and to provide an appropriate number of discussion areas
8b1b	System sets up group discussion fora and assigns students to them
8b1c	System gives moderator permissions for one student in each forum, and controller permissions for teacher in all fora
8c	Students find they cannot log in or register

8c1a	Students seek technical support
8c1a1	Students adjust cookie settings and are able to log in
8c1a2	Students update browser and are able to log in
8c1a3a	Technical support checks and repairs system
8c1a3a1	Students postpone registration until system is sorted
8c1a3a2a	Students go to see teacher and register face to face
8c1a3a2b	Teacher maintains details of who has registered on own computer or in hard copy
8c1a3a2c	Teacher seeks technical support to repair system
8c1a3a2d	Teacher gives student details to technical support for upload into repaired system
9a	System assigns students to the wrong course area
9a1a	Students complain to teacher
9a1b	Teacher seeks technical support
9a1c	Support staff reassign students
10a	System does not recognise the role of 'moderator' in its discussion forums
10a1	Teacher asks technical support staff what the equivalent term is that the system does recognise
10a2	Teacher asks technical developers to create a moderator role for system's discussion forums
10b	System has lost registration details of the student chosen to be moderator, and does not recognise them
10b1a	Teacher asks technical support to check registration and records management system
10b1b	Technical support check and repair registration and records management systems
10b1b	Teacher tells student moderator to re-register
10b1c	Teacher suspects that other student registrations may have been lost, and sends a circular email to all students advising them to check their registrations and re-register if necessary
10b1d	Teacher tells the system about the moderator again
11a	System does not have a permissions level suitable for moderators who are students
11a1	Teacher asks technical developers to create a permissions level for student moderators
11a2	Teacher asks technical support to assign special permission to the student moderator
12a	Students are not logging in to forum and don't see message
12a1	Teacher sends a group email to students to trigger activity

12a2	Teacher sends a text message from their computer to student mobiles to trigger activity
12b	Teacher message does not reach forum
12b1a	Teacher accesses permissions system and finds that their controller permission has disappeared
12b1b	Teacher asks technical support to reinstate controller permissions and investigate system failure
12b1c	Teacher postpones start of activity until problem is fixed
12b2a	Teacher tries sending messages to other forums and finds that none of them get through
12b2b	Teacher asks technical support to investigate system failure
12b2c	Teacher sends group email to students to trigger activity
13a	System is down and students cannot log in
13a1a	Teacher asks technical support to investigate and fix problem
13a1b	Technical support fail to diagnose and fix problem within a reasonable time
13a1c	Teacher calls a face to face meeting by group email and hard copy poster on notice board
13a1d	Teacher downloads video clips etc and burns them onto DVDs
13a1e	Teacher prints hard copy briefing materials
13a1f	Teacher holds face to face meeting, talks through briefing materials and uses DVD player to show clips
13a1g	Teacher appoints a student to record group discussion
13a1h	Students discuss clips in face to face group
13a1i	Student recorder uses PDA to make brief notes on discussion contributions
13a1j	Student recorder summarises discussion, synchronises PDA with computer and mails notes and summary to teacher and other students
13a1k	Teacher saves notes and summary with course records on system for evaluation
14a	System fails to authenticate students
14a1a	Students complain to teacher
14a1b	Teacher asks technical support to fix system
14a1c	Teacher advises students to keep personal backup copies of all their work on separate systems as a precaution
14a1d	Technical investigate and fix system
14b	System fails to deliver clips and briefing materials
14b1	Teacher downloads materials, and advises via text message to come and save them onto memory sticks

14c	Student computers do not have application or plugin to play video
14c1	Technical support staff load necessary applications onto student computers
14c2	Students download necessary applications
15a	Discussion forum system fails
15a1a	Teacher sets up alternative face to face discussion
15a1b	Teacher records and makes notes on discussion contributions on laptop
15a1c	Teacher saves notes to course records on system for evaluation
15a1d	Teacher keeps personal backup copy of notes
15a2a	Teacher sets up alternative synchronous discussion via commercial system (eg. MSN)
15a2b	Teacher saves records of synchronous discussion as a text file and saves it in course records on system for evaluation
15a2c	Teacher keeps personal backup copy of records
15a3a	Teacher uses own personal web space to set up a blog for the group
15a3b	Teacher saves copy of blog contributions as a text file and saves in course records on system for evaluation
15a3c	Teacher keeps personal backup copy of records
15a4a	Teacher sets up telephone conference for discussion
15a4b	Teacher asks a student to record discussion
15a4c	Teacher asks a student to chair discussion
15a4d	Students discuss clips
15a4e	Recorder and Chairperson compile record and summary of discussion using word processor, private email and phone
15a4f	Recorder emails record and summary to teacher and other students as an attached text file
15a4g	Teacher saves record and summary to course records on system for evaluation
15a4h	Teacher keeps personal backup copy of record and summary
15b	Students find that forum will not support inclusion of screen shots taken from the video clips that they want to refer to in discussion
15b1a	Teacher asks technical support for advice
15b1b	Technical support converts video files to a format that will display play time and/or frame number
15b1c	Students refer to time or frame number to identify shots in discussion
15b1c	Teacher asks technical developers to develop support for screen shots for following year's cohort of students

16a	Teacher finds that students are not using the discussion forum
16a1a	Teacher contacts students individually (email, text message, notes in pigeonholes, or face to face) and establishes that technical problems are not an issue
16a1b	Teacher arranges a synchronous chat conference session to boost motivation
16a1c	Teacher saves records of chat session to course records on system, and keeps a personal backup
16b	Teacher finds that discussion forum does not support replying to individual students from messages posted to discussion
16b1a	Teacher copies text of discussion message and pastes into email to individual student
16b1b	Teacher asks technical developers to develop a better integrated system for future years.
16c	Teacher finds that they do not have access to forum
16c1a	Teacher asks technical support to fix problem
16c1b	Teacher contacts moderator (email, face to face or phone) to confirm that forum is going OK
16c1c	Technical support assign teacher a temporary login and permissions while they sort the problem
17a	Moderator finds inappropriate postings
17a1a	Moderator edits or deletes posting
17a1b	Moderator generates email from posting to author of posting advising them about netiquette
17a1c	Moderator keeps a record of original posting and action taken
17a2	Moderator observes that student persists in posting inappropriate messages
17a2a1	Moderator tells teacher
17a2a2	Teacher examines record of postings and action
17a2a3	Teacher warns student about possibility of disciplinary action
17a2a4	Teacher changes permissions so that contributions from that student require prior moderator or teacher approval
18a	System does not support copying of individuals into message to forum
18a1	Moderator copies and pastes text of message into separate email to teacher
20a	System crashes and loses all records
20a1a	Teacher locates their personal backups of materials and any messages sent, aggregates them into a record file for the activity, and backs it up
20a1b	Teacher asks students to send him/her any personal backup records they may have for inclusion into the record file
20a1c	Teacher compiles as complete a record of the discussion postings as they can and emails to students for reference

20a1d	Teacher is unable to authenticate student contributions to record of activity so assigns less reliance on them in any evaluation
20a1e	Quality control and evaluation is not robust or reliable
20a2	System has been backing up at all stages and teacher, students and quality controllers are able to access backup copies

Authors

Author 1	John Edmonstone, Cardonald College
Author 2	Ann Jeffery, University of Southampton

Use Case Summary

Teacher uses concept mapping and a discussion forum, followed by a diagnostic test to introduce students to a VLE and enhance their understanding of a concept

Narrative

A highly motivated 1st year Higher National Journalism class (in Scotland) studying the UK print industry. Introductory session on structure of industry. Twenty five percent of the class are distance students taking part online. The session would begin with a brief exposition on industry, followed by a brainstorm on categorising newspapers within the industry framework. The brainstorm would have the purpose of building dialogue and enabling validation of prior learning. as well introducing a conceptual and practical map of the UK print sector. A testing activity would follow to establish understanding and then a research based task. The session has the additional purpose of introducing new students to the VLE

Primary Actor (and goal)

Teacher	To administer the activities successfully via the VLE

Other Actors (and goals)

Students	To acquire skills and concepts to pass the course To complete task successfully
VLE	To manage teacher and student activities and records
Examiners	To have access to records of coursework
Quality control	To have access to a record of the activity for evaluation

Stakeholders and Interests

Technical support	To have a system that runs smoothly
Technical developers	To have a system that meets teacher and student requirements
Institution	To deliver courses in a cost and time efficient manner at a sufficient level of quality
Staff developers	To ensure staff have necessary skills for effective teaching

Preconditions

1	Computer classroom available for hands on session
2	VLE containing course resources (text, diagrams, diagnostic quiz (questions and answers may contain diagrams)), test, and conferencing facilities

Teaching Approach

Blended learning, conceptualisation, social constructivist, dialogic, community of practice, reciprocal

Main Success Scenario

1	Teacher leads face to face session and shows presentation (powerpoint or similar)
2	Teacher leads brainstorming activity using concept mapping software and saves results in VLE
3	Teacher leads hands on session and shows students how to access VLE
4	Students log in to VLE
5	Students and teacher correspond by email throughout steps 6-8
6	Students do diagnostic quiz (see usecase 7a)
7	Students search for resources online and in electronic databases
8	Students discuss resources in asynchronous small group conference (see usecase 7b)
9	Teacher, examiners and quality controllers accesses records of activities for assessment and evaluation

Extensions

1a	Teacher uses web-based video conferencing and shared whiteboard to allow distance students to view session
2a	Teacher uses web-based video conferencing and shared whiteboard to allow distance students to view session
3a	Teacher uses remote assistance (eg. in MSN) to help distance students if necessary
7a	Students save resources in a portfolio or database
7b	Students bookmark links to resources, or save as hyperlinks in a text file

USECASE 7A (QUIZ)

Authors

Author 1	John Edmonstone, Cardonald College
Author 2	Ann Jeffery, University of Southampton

Use Case Summary

Teacher uses a diagnostic test to introduce students to the facilities of a VLE and enhance their understanding of a concept

Narrative

1st year Higher National Journalism class, with campus-based and distance students, are being introduced to the formative assessment facilities in the VLE. A quiz follows a brainstorming activity in which they have begun to develop a concept map of the UK print sector.

Primary Actor (and goal)

Teacher	To administer the quiz successfully via the VLE

Other Actors (and goals)

Students	To acquire skills and concepts to pass the course To complete task successfully
VLE	To manage teacher and student activities and records
Examiners	To have access to records of coursework
Quality control	To have access to a record of the activity for evaluation

Stakeholders and Interests

Technical support	To have a system that runs smoothly
Technical developers	To have a system that meets teacher and student requirements
Institution	To deliver courses in a cost and time efficient manner at a sufficient level of quality
Staff developers	To ensure staff have necessary skills for effective teaching

Preconditions

1	System with online multiple choice quiz facilities
2	Existing quiz saved in system (questions may contain diagrams, images, audio, video files)

Teaching Approach

behaviourist

Main Success Scenario

1	Students access system, locate and do diagnostic quiz
2	System gives students their score, and answer profile
3	System saves records of student quiz
4	Teacher accesses student quiz results and intervenes with individual students if necessary
5	Teacher, examiners and quality controllers access quiz and discussion records for assessment and evaluation

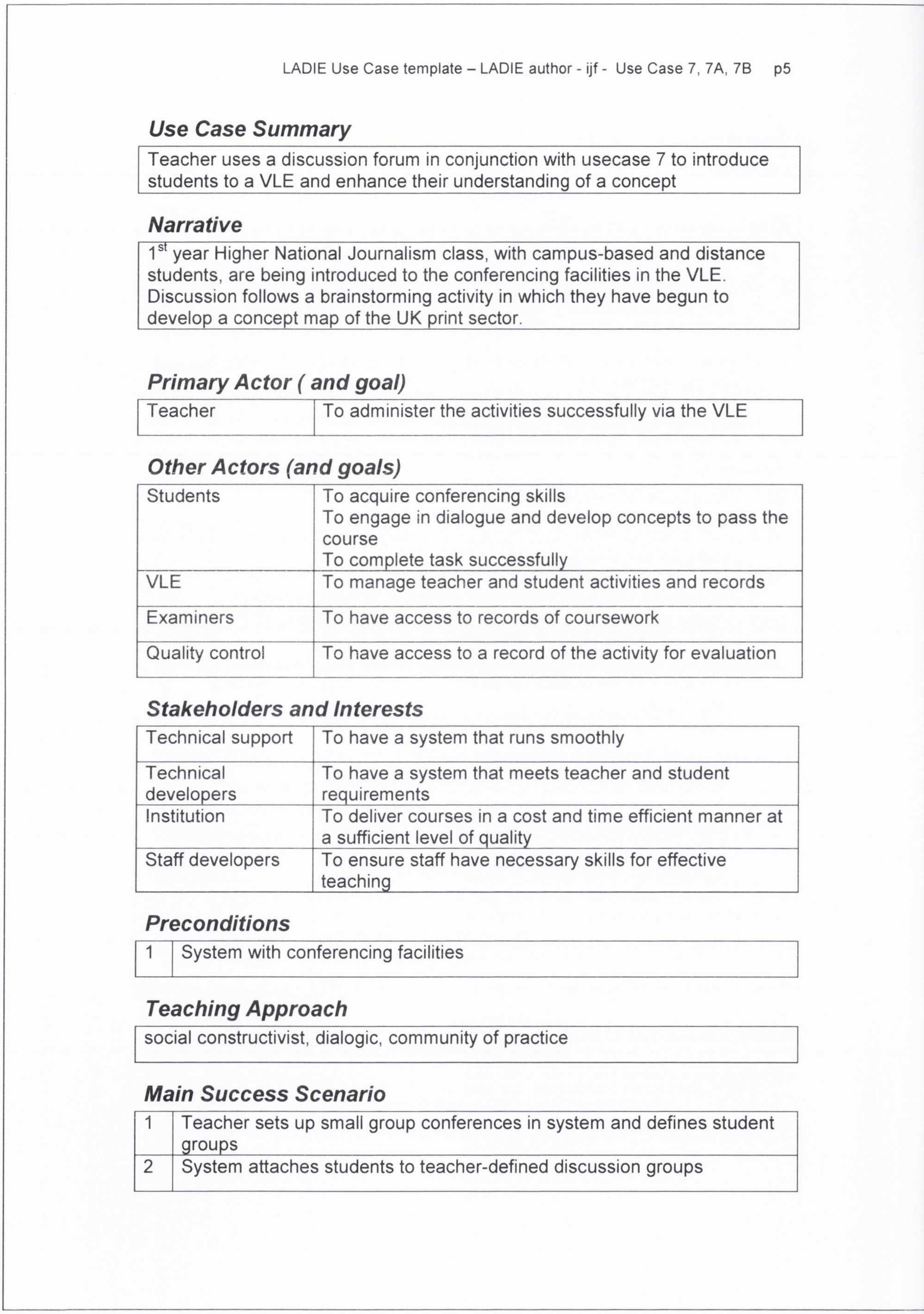
Extensions

1a	Teacher wants to use quiz in a face to face session
1a1a	Teacher presents diagnostic quiz at face to face session with students using personal response systems
1a1b	System collates student responses and gives teacher a histogram of responses, as well as recording individual student responses
1a2a	Distance students access the session via video conference and respond to quiz using mobile phone text messages
1a2b	System collates student text message responses and gives teacher a histogram of responses, as well as recording individual student responses
2a	System also gives students correct answers
3a	System collates student responses and gives teacher a histogram of responses, as well as recording individual student responses
4a	System provides students with individualised study plans based on their response profiles
4b	Teacher or system tells students with scores below the threshold to repeat steps 1 & 2 after further study
4b1	System generates new questions for repeat tests (see usecase 3)

USECASE 7B (DISCUSSION)

Authors

Author 1	John Edmonstone, Cardonald College
Author 2	Ann Jeffery, University of Southampton



Use Case Summary

Teacher uses a discussion forum in conjunction with usecase 7 to introduce students to a VLE and enhance their understanding of a concept

Narrative

1st year Higher National Journalism class, with campus-based and distance students, are being introduced to the conferencing facilities in the VLE. Discussion follows a brainstorming activity in which they have begun to develop a concept map of the UK print sector.

Primary Actor (and goal)

Teacher	To administer the activities successfully via the VLE

Other Actors (and goals)

Students	To acquire conferencing skills To engage in dialogue and develop concepts to pass the course To complete task successfully
VLE	To manage teacher and student activities and records
Examiners	To have access to records of coursework
Quality control	To have access to a record of the activity for evaluation

Stakeholders and Interests

Technical support	To have a system that runs smoothly
Technical developers	To have a system that meets teacher and student requirements
Institution	To deliver courses in a cost and time efficient manner at a sufficient level of quality
Staff developers	To ensure staff have necessary skills for effective teaching

Preconditions

1	System with conferencing facilities

Teaching Approach

social constructivist, dialogic, community of practice

Main Success Scenario

1	Teacher sets up small group conferences in system and defines student groups
2	System attaches students to teacher-defined discussion groups

3	Students discuss resources asynchronously in small group conferences
4	Teacher monitors small group conferences
5	System saves records of conferences
6	Students access saved conferences for future work
7	Teacher, examiners and quality controllers access saved conferences for evaluation and quality control

Extensions

1a	Teacher defines a student moderator for the whole group
1b	Teacher defines whole group conference and small group subconferences
1b1	Teacher defines student moderators for whole conference and subconferences
1b2	Teacher defines whole conference as read -only for students and subconferences as contributory for students (or vice versa)
2a	VLE attaches students to whole conference and relevant subconferences
3a	Some messages are inappropriate
3a1	Moderators edit or remove messages
3b	Students need to embed special characters, hyperlinks, tables, diagrams, images or audio files in messages
3c	Students need to attach image, audio, video, concept maps, spreadsheets, text, html, diagrams files to conference messages
3d	There are too many messages for students to keep track of easily
3d1	Students follow message threads
3d2	Moderators organise messages by date, subject, author, last posted, or unread
3d3	Moderators organise messages into folders or sub subconferences
4a	Student participation is low
4a1	Teacher sends messages to conference and subconferences encouraging participation
4a2	Teacher changes conference structure so new messages are more easily visible (eg. by doing away with subconferences)
4a3	Teacher holds group synchronous chat session to boost motivation
4a4	Teacher examines history of posted conference messages to distinguish students who are not participating at all from those who are reading but not posting
4b	Teacher finds persistent inappropriate messages
4b1	Teacher requires messages from certain students to be approved by teacher or moderator before appearing on conference or

	subconference
5a	Teacher or moderator extends expiry date on messages to ensure student, teacher, or quality controller access as long as necessary
5b	Teacher defines archive subconferences which are read-only for students
6a	Students request text file summary of selected messages for inclusion in ePortfolio

Authors

Author 1	Colleen Hurren, Cumbernauld College
Author 2	Andy McKenzie, SFEU. Emerge

Use Case Summary

Teacher sets up a 'webquest' , interviews, worksheets and a discussion forum for students to gain understanding of workplace legislation.

Narrative

Higher National Certificate in Office Administration (now Administration and Information Management) unit on Health and Safety (SCQF level 7). The class is of mixed age (16-50), from a narrow social class, and some have literacy needs. Historically there has been a high drop out rate and teaching materials need to be activity rich. The activity uses a webquest to explore Health and Safety Legislation and its application in the workplace, in order to achieve SQA HN Unit which includes Health and Safety, It is taught by blended learning with use of discussion boards on a VLE

Primary Actor (and goal)

Teacher	To use the system to support a webquest and associated discussion

Other Actors (and goals)

Students	To complete task successfully To acquire knowledge and skills defined by teacher To get credit for the unit
Technical support	To facilitate system side of the activity
Interviewees	To have a well managed and appropriately conducted interview
Quality Control	To have access to a record of the activity for evaluation purposes

Stakeholders and Interests

SQA	To ensure unit meets subject benchmarks
Technical support	To have a system that runs smoothly
Technical developers	To have a system that meets teacher and student requirements
Institution	To deliver courses in a cost and time efficient manner at a sufficient level of quality
Staff developers	To ensure staff have necessary skills for effective teaching

Preconditions

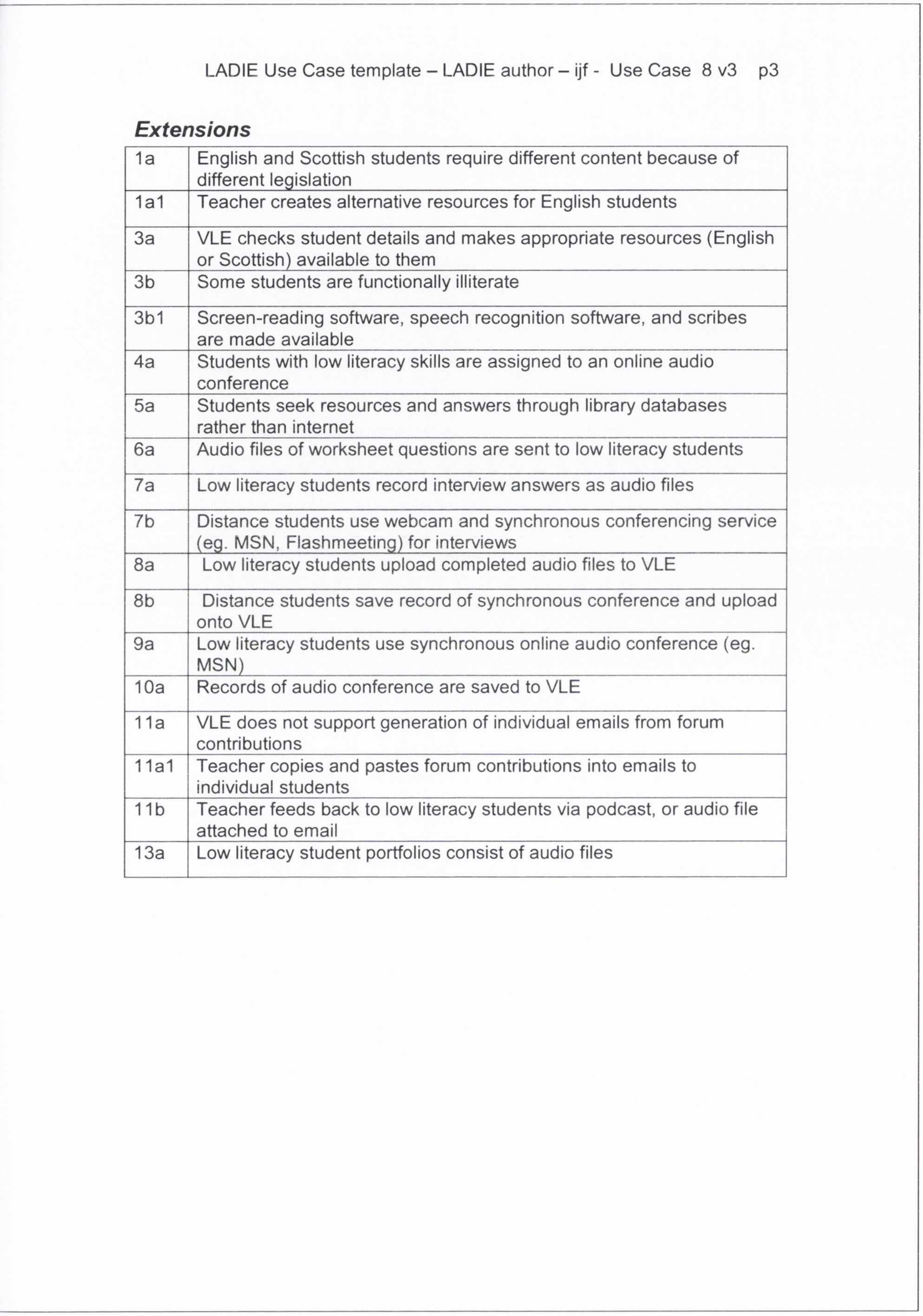
1	Students have necessary IT skills
2	Internet connected computers, one per student
3	VLE
4	Tagged resources
5	Basic activity set up a previous year
6	Rooms with computers and internet access for interviews

Teaching Approach

Problem based learning, situative, activity theory

Main Success Scenario

1	Teacher logs into VLE, locates and accesses previous year's resources, updates and saves them into course area
2	Teacher assigns students to pairs, and each pair to a larger discussion group and records pairs and groups on VLE
3	Teacher triggers activity in face to face session
4	Students log into VLE, which assigns them to pairs and discussion fora
5	Students access webquest materials and undertake internet research
6	Students print out worksheets from VLE
7	Pairs of students interview staff face to face about health and safety issues and fill in worksheets
8	Students enter and save worksheet answers in VLE
9	Students discuss findings in VLE discussion forum (usecase 'Discussion') which saves records
10	Students access forum records and evaluate peers based on their contributions to the forum
11	Teacher accesses discussion fora and completed worksheets on VLE and generates feedback messages to group fora and emails to individual students
12	Students log into VLE and do multiple choice quiz for formative assessment and get feedback (see usecase 'Quiz')
13	Students prepare portfolio of webquest, interview and quiz answers and load it into ePortfolio on VLE for future reference
14	Teacher and quality controllers access records of discussion, quiz, worksheets and portfolios



LADIE Use Case template – LADIE author – ijf - Use Case 8 v3 p3

Extensions

1a	English and Scottish students require different content because of different legislation
1a1	Teacher creates alternative resources for English students
3a	VLE checks student details and makes appropriate resources (English or Scottish) available to them
3b	Some students are functionally illiterate
3b1	Screen-reading software, speech recognition software, and scribes are made available
4a	Students with low literacy skills are assigned to an online audio conference
5a	Students seek resources and answers through library databases rather than internet
6a	Audio files of worksheet questions are sent to low literacy students
7a	Low literacy students record interview answers as audio files
7b	Distance students use webcam and synchronous conferencing service (eg. MSN, Flashmeeting) for interviews
8a	Low literacy students upload completed audio files to VLE
8b	Distance students save record of synchronous conference and upload onto VLE
9a	Low literacy students use synchronous online audio conference (eg. MSN)
10a	Records of audio conference are saved to VLE
11a	VLE does not support generation of individual emails from forum contributions
11a1	Teacher copies and pastes forum contributions into emails to individual students
11b	Teacher feeds back to low literacy students via podcast, or audio file attached to email
13a	Low literacy student portfolios consist of audio files

Authors

Author 1	Stephen Draper, Glasgow University
Author 2	Gerard Graham, Paisley University

Use Case Summary

Teacher sets up a diagnostic test to determine gaps in mastery of course and work plans and discussion groups for remedial work

Narrative

On a level 2 course, a lowered entry point means there is a danger of a high drop out rate among lower ability students. The teacher sets up a diagnostic test to assess mastery of learning skills taught during the level 1 course, and uses the results to prompt remedial work, followed by re-testing and more remedial work if necessary , in order to reduce the drop out rate.

Primary Actor (and goal)

Teacher	To use the system to administer a diagnostic test and support a discussion of remedial work

Other Actors (and goals)

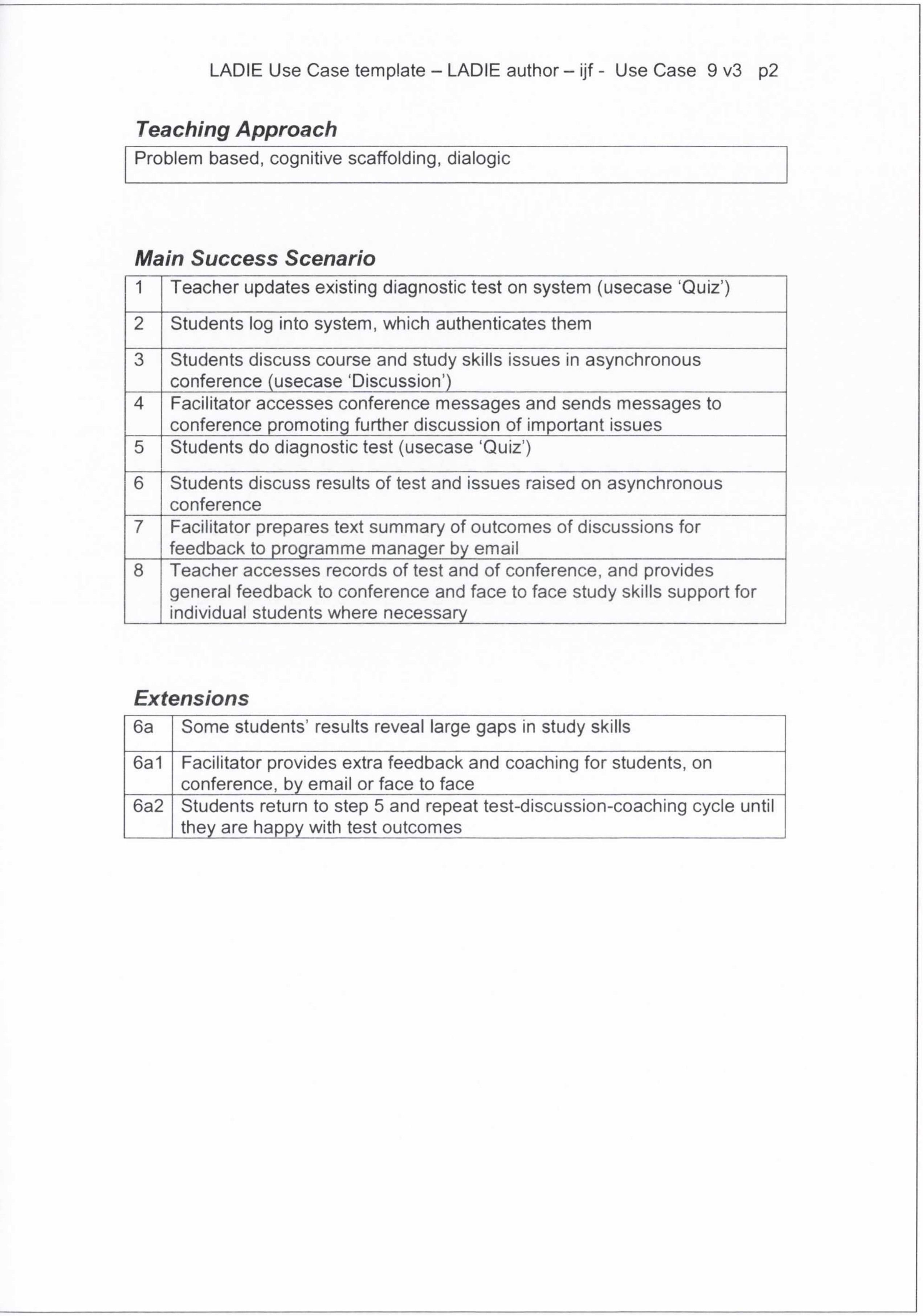
Students	To improve learning skills To pass the course
Facilitator (a student)	To enable learners' discussions To earn some money by acting as facilitator
Programme manager	To ensure quality and smooth running of entire course

Stakeholders and Interests

Technical support	To facilitate teacher and student use of the system
Technical developers	To ensure that system meets teacher and student requirements
Quality control	To have access to a record of the activity for evaluation purposes
Institution	To deliver courses in a cost and time efficient manner at a sufficient level of quality
Staff developers	To ensure staff have necessary skills for effective teaching

Preconditions

1	Funding for facilitators
2	Software for creating and administering diagnostic tests
3	Computer lab for students' to use when taking the test



Teaching Approach

Problem based, cognitive scaffolding, dialogic

Main Success Scenario

1	Teacher updates existing diagnostic test on system (usecase 'Quiz')
2	Students log into system, which authenticates them
3	Students discuss course and study skills issues in asynchronous conference (usecase 'Discussion')
4	Facilitator accesses conference messages and sends messages to conference promoting further discussion of important issues
5	Students do diagnostic test (usecase 'Quiz')
6	Students discuss results of test and issues raised on asynchronous conference
7	Facilitator prepares text summary of outcomes of discussions for feedback to programme manager by email
8	Teacher accesses records of test and of conference, and provides general feedback to conference and face to face study skills support for individual students where necessary

Extensions

6a	Some students' results reveal large gaps in study skills
6a1	Facilitator provides extra feedback and coaching for students, on conference, by email or face to face
6a2	Students return to step 5 and repeat test-discussion-coaching cycle until they are happy with test outcomes

USECASE 10

Authors

Author 1	Hilary Grierson, University of Strathclyde
Author 2	Sue Milne, Glasgow Caledonian University

Use Case Summary

Teacher sets up and monitors an activity in which students collect, manage, discuss, and evaluate resources as the first stage in a group design project

Narrative

Third year campus based engineering students meet at least twice a week. They may meet at other times and are in email contact with each other and the teacher. As the first stage in a group design project they gather and store information and design ideas. After an induction session with the librarian, teams of 4 students compile a resource collection, evaluating resources for quality, usability, etc and storing and managing the resources in an online environment (LauLima) and using a concept map to link the resources. Students use a wiki to reflect on their resources and share ideas with other groups. Each group compiles a report and a presentation for the other groups. The collection forms a resource for the next stage of the project.

Primary Actor (and goal)

Teacher	To improve student skills in sourcing, assessing, evaluating and managing information, and use information to generate design ideas

Other Actors (and goals)

Students	To acquire skills and design ideas required by teacher and necessary to progress on to next stage of project
Librarian	To improve student skills in discovery, management and archiving of information
Technical support staff	To facilitate running of activity
Public	To be treated in an appropriate manner and receive due recognition if used as a resource
Examiners	To have access to records of the activity for assessment
Quality contollers	To have access to records of the activity for quality control

Stakeholders and Interests

Technical developers	To have a system that meets teacher and student requirements
Institution	To deliver courses in a cost and time efficient manner at a sufficient level of quality

Staff developers	To ensure staff have necessary skills for effective teaching

Preconditions

1	Computer lab with terminals for students for face to face briefing sessions
2	Teacher has defined design task precisely
3	Technical support to create databases
4	

Teaching Approach

Problem based learning

Main Success Scenario

1	Teacher writes briefing materials to be presented via system and face to face, and saves them onto the system
2	Teacher organises students into groups of four and tells system
3	Technical support creates a database for each group
4	Teacher briefs students in a face to face session, using presentation services
5	Librarian briefs students in a face to face session, using a networked computer and data projector to demonstrate information management services
6	Teacher is in contact with students, face to face or by email or wiki, throughout steps 8-12
7	Students log into system, which attaches them to correct groups
8	Students discuss their findings in the group wiki (usecase 10a) throughout steps 10-12
9	Students locate and record resources from web, members of public, library databases, existing (physical) products etc, in text, image, audio and video formats
10	Students save resources in their group's database in a variety of formats (text, images, audio, video)
11	Students use concept mapping software to map links between the resources in their group database, and save maps on the system
12	Student rapporteurs and presenters write up and present group conclusions (usecase 10b)
13	Teacher, examiners and quality controllers accesses databases, discussion records, reports and presentations for evaluation
14	Students archive all records (report, presentation, database, wiki) for their own use later on in the project, and for use by other students in subsequent years

Extensions

1a	Teacher records audio materials for visually impaired students
1b	Technical support install screen readers and voice recognition software for student with disability
3a	A tool or 'wizard' on the system enables teacher to set up database him/herself
4a	Teacher briefs distance students in a video or audio session using webcam, shared whiteboard and file sharing
5a	Librarian briefs students in a video or audio session using webcam, shared whiteboard and file sharing
10a	Students have collected some physical objects as part of their resources
10a1a	Teacher assigns physical space in classroom for storage of objects
10a1b	Students enter brief description of object, and note of its location, into database
10a2	Students make a video of object and save video to database
11a	Students use diagram drawing software instead of concept maps
14a	Students save records in ePortfolio

USECASE 10A

Authors

Author 1	Hilary Grierson, University of Strathclyde
Author 2	Sue Milne, Glasgow Caledonian University

Use Case Summary

Teacher sets up and monitors discussion, evaluation and reporting of resources collected in Usecase 10.

Narrative

Groups of third year engineering students have compiled a resource collection, evaluating resources for quality, usability, etc and storing and managing the resources in an online environment (LauLima) and using a concept map to link the resources. Students use a wiki to reflect on their resources and share ideas with other groups.

Primary Actor (and goal)

Teacher	To facilitate student discussion and evaluation of resources collected

Other Actors (and goals)

Students	To acquire skills and design ideas required by teacher and necessary to progress on to next stage of project
Technical support staff	To facilitate running of activity
Examiners	To have access to records of the activity for assessment
Quality contollers	To have access to records of the activity for quality control

Stakeholders and Interests

Technical developers	To have a system that meets teacher and student requirements
Institution	To deliver courses in a cost and time efficient manner at a sufficient level of quality
Staff developers	To ensure staff have necessary skills for effective teaching

Preconditions

1	Database set up for management of student-discovered resources (from usecase 10)
2	Teacher has defined purpose of discussion
3	
4	

Teaching Approach

Social constructivist, dialogic, community of practice

Main Success Scenario

1	Technical support create a wiki for each group of four students
2	Teacher defines groups and sets access permissions so that each group can contribute to both their own and other group wikis
3	Students comment on their findings (from usecase 10) in the group wiki using text and concept maps, and linking to records in database
4	Students access other group wikis to comment on findings of other groups
5	Teacher accesses system to monitor discussion throughout

Extensions

1a	Teacher wants to run course for remote students, some with poor internet connections

1a1	Technical support set up an asynchronous conference with offline reader facility with subconferences for each group of four students instead of wiki
1b	A tool or 'wizard' on the system enables teacher to set up wiki or conference him/herself
2a	Teacher sets up access permissions for groups in asynchronous conference
3a	Students comment on their findings (from usecase 10) in the group subconference using text and concept maps, and linking to records in database
3b	Students use diagrams instead of concept maps in their messages to the wiki
4a	Students access other group subconferences to comment on findings of other groups

USECASE 10B

Authors

Author 1	Hilary Grierson, University of Strathclyde
Author 2	Sue Milne, Glasgow Caledonian University

Use Case Summary

Students evaluate and report on resources collected in usecase 10 and discussed in usecase 10a.

Narrative

Groups of third year engineering students have compiled a resource collection, evaluating resources for quality, usability, etc and storing and managing the resources in an online environment (LauLima) and using a concept map to link the resources. Students have used a wiki to reflect on their resources and share ideas with other groups. Each group compiles a report and a presentation for the other groups.

Primary Actor (and goal)

Teacher	To facilitate evaluation and reporting of resources collected and to use information to generate design ideas

Other Actors (and goals)

Students	To acquire skills and design ideas required by teacher and necessary to progress on to next stage of project
Rapporteurs (students)	To be able to access information and discussion records necessary for writing report
Presenters (students)	To access suitable presentation writing and presenting facilities

Technical support staff	To facilitate running of activity
Examiners	To have access to records of the activity for assessment
Quality contollers	To have access to records of the activity for quality control

Stakeholders and Interests

Technical developers	To have a system that meets teacher and student requirements
Institution	To deliver courses in a cost and time efficient manner at a sufficient level of quality
Staff developers	To ensure staff have necessary skills for effective teaching

Preconditions

1	Database set up for management of student-discovered resources (from usecase 10)
2	Group discussion in a wiki (from usecase 10a)
3	
4	

Teaching Approach

Problem based learning

Main Success Scenario

1	Student rapporteurs access their group information database (created in usecase 10) and wiki discussion (usecase 10a), write report, and save it on the system
2	Student presenters access reports, write presentations based on them, and save presentations on the system
3	Student presenters give face to face presentation on their group's work using presentation software

Extensions

1a	Rapporteur includes concept maps, diagrams and images as well as text in the report
2a	Presenter includes concept maps, diagrams and images as well as text in the report
3a	Distance student presenters save presentation file on system and link it to a message to the wiki

Authors

Author 1	Arthur Loughran, University of Paisley
Author 2	Michael McCarney, Glasgow Caledonian University

Use Case Summary

Teacher creates and supports online discussion between students around a theme of reflective essay writing

Narrative

The students are new to online discussion and to the ideas of reflective essay writing. The teacher creates and supports online discussion by providing mentoring support and feedback comments to individual reflective essays

Primary Actor (and goal)

Teacher	To use the system to support a discussion of reflective essay writing skills

Other Actors (and goals)

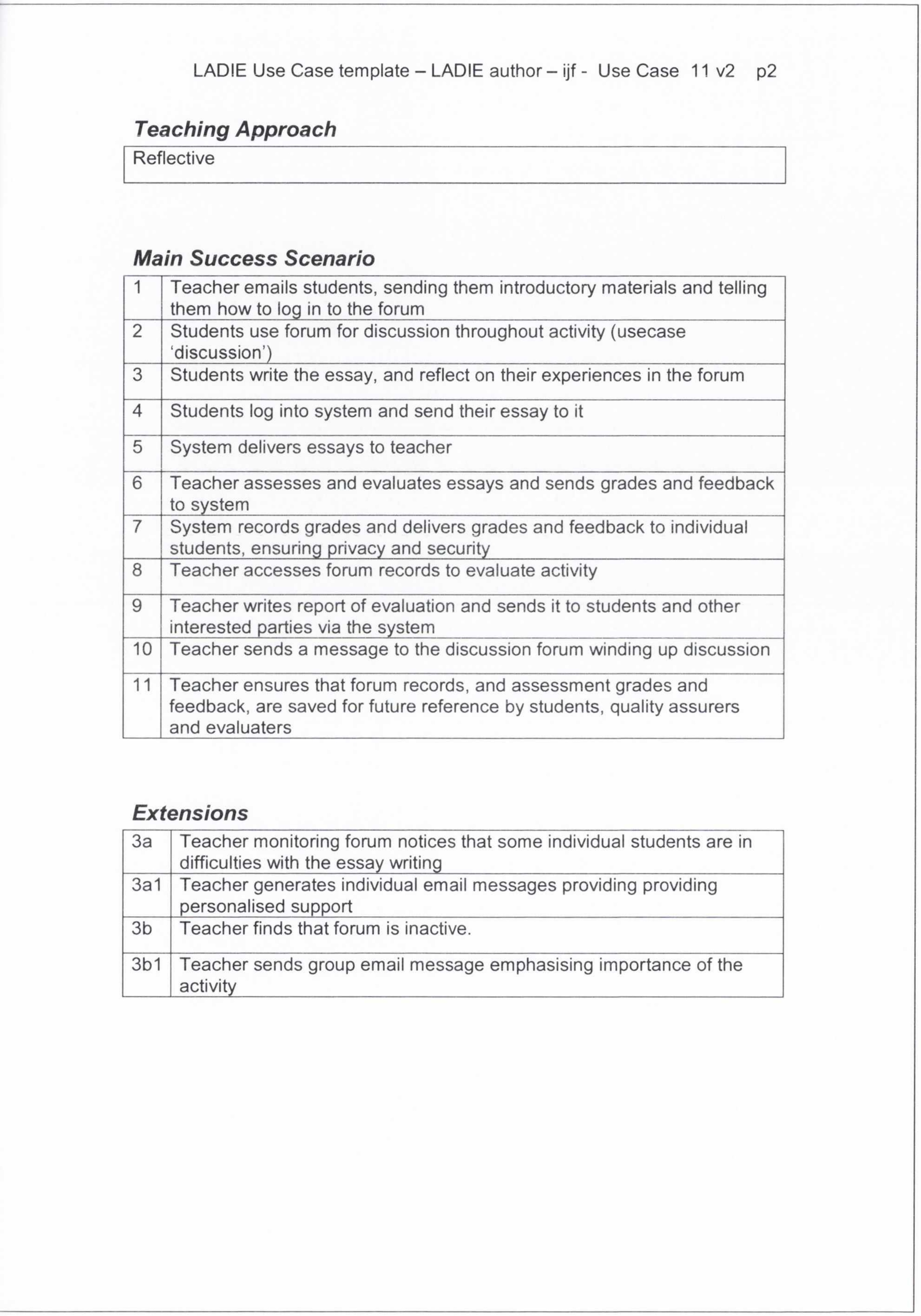
Students	To improve skills defined by teacher
Technical support	To make sure systems support teacher and student discussion activity
System	To manage and record communications between teachers, students and evaluators, one to one, one to many, and many to many

Stakeholders and Interests

Technical support	To have a system that runs smoothly
Technical developers	To have a system that meets teacher and student requirements
Quality control	To have access to a record of the activity for evaluation purposes
Institution	To deliver courses in a cost and time efficient manner at a sufficient level of quality
Staff developers	To ensure staff have necessary skills for effective teaching

Preconditions

1	A discussion forum with links to discussion help files in forum
2	Student groups defined in the registration system
3	Briefing materials are saved on the system and linked to the forum
4	



Teaching Approach

Reflective

Main Success Scenario

1	Teacher emails students, sending them introductory materials and telling them how to log in to the forum
2	Students use forum for discussion throughout activity (usecase 'discussion')
3	Students write the essay, and reflect on their experiences in the forum
4	Students log into system and send their essay to it
5	System delivers essays to teacher
6	Teacher assesses and evaluates essays and sends grades and feedback to system
7	System records grades and delivers grades and feedback to individual students, ensuring privacy and security
8	Teacher accesses forum records to evaluate activity
9	Teacher writes report of evaluation and sends it to students and other interested parties via the system
10	Teacher sends a message to the discussion forum winding up discussion
11	Teacher ensures that forum records, and assessment grades and feedback, are saved for future reference by students, quality assurers and evaluaters

Extensions

3a	Teacher monitoring forum notices that some individual students are in difficulties with the essay writing
3a1	Teacher generates individual email messages providing providing personalised support
3b	Teacher finds that forum is inactive.
3b1	Teacher sends group email message emphasising importance of the activity

Authors

Author 1	Kenneth Falconer, St Andrews University
Author 2	

Use Case Summary

The teacher uses the system to administer and run a student activity in a symbolic computational language (eg. Maple)

Narrative

The teacher uses the system to introduce first year (HE) mathematics students to a symbolic computational program (SCP) (eg. Maple) and to the maths department's file management system (MFM). Students have one session per week in a networked computer lab, where they access SCP via browser interface. Students may also carry on the activity on personal computers at home provided they have SCP nstalled. The aims are to remind students of basic mathematical relationships learnt in school, to get students set up on MFM, and to enable students to acquire sufficient knowledge of Maple to use with any mathematics course they take later on (and in future employment)

Primary Actor (and goal)

Lecturer	To administer the activity successfully using the system

Other Actors (and goals)

Students	To learn to use the system
	To undertake the activity successfully
Tutors	To receive information on students' progress and performance to enable them to taylor tutorial teaching appropriately
Markers	To use the system for marking student work
Demonstrators	To help the students use the system in computer lab sessions

Stakeholders and Interests

Technical staff	To ensure system is running smoothly and provides services actors require
Quality control	To have access to records for evaluation and quality control

Preconditions

1	Networked computer lab for students' use in lab sessions

2	SCP available on network via browser interface
3	File management system (FMS)
4	Technical support available, especially during lab sessions

Teaching Approach

constructive

Main Success Scenario

1	Lecturer logs into file management system and locates a SCP worksheet (the SCP equivalent of a wordprocessor 'document' - it contains text commentary, SCP commands, and automatically generated SCP output)
2	Lecturer edits worksheet commentary, mathematics commentary, and SCP commands using relevant menus from the SCP interface
3	Lecturer saves files in the correct course area of the file management system
4	Lecturer locates and updates wordprocessed student exercise sheets and saves them in the course area on the file management system
5	Students register for the course on the file management system, which allocates them to markers and tutors
6	Lecturer triggers the activity with a face to face session in a computer lab, using an interactive whiteboard, which is also networked in, to display the SCP worksheets
7	Students log in to file management system, which delivers SCP worksheet, and wordprocessed exercise sheets to their desktops
8	Lecturer explains the worksheet, demonstrating with the whiteboard how to modify SCP commands and displaying the generated SCP output
9	Students experiment with modifying SCP commands on their personal worksheets, and examine the generated output
10	Students discuss the exercise with each other face to face
11	Students fill in exercise sheets
12	Students save original worksheet, modified worksheet, and exercise sheets on file management system
13	Markers log into file management system, which delivers the appropriate group of completed exercise sheets
14	Markers mark and grade exercise sheets and save them in the system for access by tutors and students
15	Tutors log into the system and receive marked exercise sheets for their tutorial group
16	Tutors provide tailored feedback to students at face to face tutorials

Extensions

1a	Lecturer is not confident in use of the file management system

1a1	Lecturer uses their personal computer, which has SCP installed, to edit worksheet
1a2	Lecturer gives SCP files to technical support for loading into file management system
4a	Lecturer has experience of students copying each others' exercise sheets electronically, and of incompatibility problems between file management system and students home wordprocessing and operating systems (see 9b)
4a1	Lecturer prints out hard copy exercise sheets, instead of using file management system to deliver them
5a	Students register for course on university central records system, which passes registration on to department file management system
9a	Activity runs over several computer lab sessions
9a1	Subsequent computer labs are lead by a demonstrator who logs into the system and helps students with using the SCP
9a2	Students want to work on worksheets and exercise sheets at home between computer lab sessions
9a2a	SCP made available for students to install on their personal computers
9a2b	Students log into file management system from home and use installed SCP to work on worksheets
9ab3	Students save modified worksheets back onto file management system from home
12a	Students hand in completed hard copy worksheets which are distributed to markers
14a	Markers record grades on file management system and pass marked hard copy exercise sheets on to tutors for return to students

Authors

Author 1	Julie Watson – University of Southampton
Author 2	Adam Warren – University of Southampton

Use Case Summary

Teacher uses the system to set up and run an online self-study resource for students to study in pairs

Narrative

The institution (as teacher) uses the system to run a short (45 min) online self-study course for HE tutors (as learners) to help them deal with inter-cultural student issues. The aim of the course is to raise awareness and to practice and discuss solutions. Tutors can work individually or (preferably) in pairs to encourage reflection. This is a compulsory staff development course that all tutors must take.

Primary Actor (and goal)

Teacher (Institution)	To use the system to administer deliver the self-study course

Other Actors (and goals)

Learners (tutors)	To use the system in pairs to work through the course materials
Technical support	To be able to access the system to set up and maintain the course
Institutional managers	To access records of the activity for audit purposes

Stakeholders and Interests

Preconditions

1	A system that tracks usage, and can cope with streamed video and a control bar, allows for branches and options in study patterns.
2	Discussion forum
3	Activity and its resources set up in advance by technical staff
4	

Teaching Approach

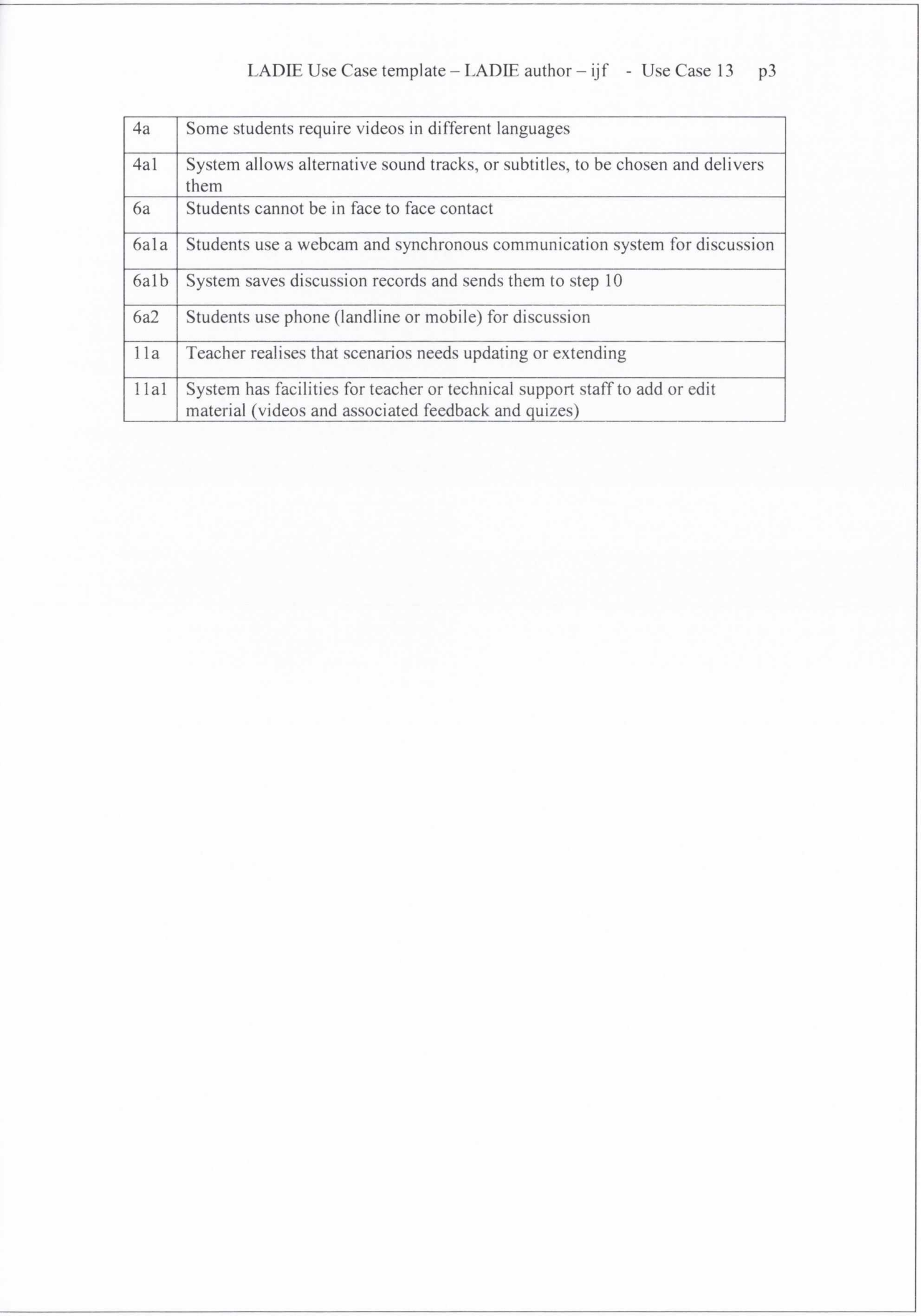
Scenarios/storytelling, dialogic, adaptive, communicative, experiential

Main Success Scenario

1	Teacher triggers the activity by an email to all students, but students may start at any time within the next x months (where x is defined by the teacher)
2	Students organise themselves into pairs and access system
3	System demands two usernames, links them together, and subsequently tracks the pair of users through the activity
4	System delivers introductory material (text) and offers a choice of video clips illustrating different scenarios
5	Students choose a clip and the system presents it to them with a facility for students to start, pause, stop, and move on to the next screen of instructions
6	Students watch clip and move on to next screen, which presents an open ended question for discussion in pairs for a minimum of y minutes (where y has been defined by the teacher in advance)
7	Onscreen timer shows students how long they have been discussing, and system tracks time before students move on to 8
8	System presents students with a multiple choice quiz which has a branching structure, so that each answer leads to a different range of subsequent questions (see usecase 'quiz')
9	System delivers quiz feedback to students and tells them to complete steps 5 to 8 for z different scenarios (where z has been defined by the teacher in advance and is less than the number of possible scenarios)
10	On completion of z scenarios, system records usage and time taken for each scenario and delivers a text summary and further discussion questions to students
11	Students, as individuals, contribute further thoughts in asynchronous discussion forum (see usecase 'discussion')
12	Teacher, and institutional managers, access system records to check user statistics for institutional audit

Extensions

3a	Pair of students are geographically separated on different computers in different locations (which may be outside the institution's network)
3a1a	System requires one login from each computer, links them together, and subsequently tracks the pair of users through the activity, delivering the same material to each student
3a1b	Students coordinate their activity by phone or by synchronous internet-based communication
3a2a	Students work individually
3a2b	System requires only one login, and tracks students as individuals



4a	Some students require videos in different languages
4a1	System allows alternative sound tracks, or subtitles, to be chosen and delivers them
6a	Students cannot be in face to face contact
6a1a	Students use a webcam and synchronous communication system for discussion
6a1b	System saves discussion records and sends them to step 10
6a2	Students use phone (landline or mobile) for discussion
11a	Teacher realises that scenarios needs updating or extending
11a1	System has facilities for teacher or technical support staff to add or edit material (videos and associated feedback and quizes)

Authors

Author 1	Anne Irving, University of Surrey
Author 2	Conci Maduli-Bush, University of Surrey

Use Case Summary

Teacher uses discussion, resource repository, and quiz services to deliver and administer online elements of a blended learning exercise on improving presentation skills

Narrative

Students in their first year at university may be unused to having to present their work in public. This activity is designed to develop their presentation skills, using critical reflection on a provided video, self assessment, development and practice of a presentation in small groups, and peer feedback, followed by an assessed presentation. Ideally it will be blended into a module containing online and face to face sessions, but the material can be used for self study. Different departments may use the material differently.

Primary Actor (and goal)

Teacher	To use the system to administer and deliver the activity

Other Actors (and goals)

Students	To use the system to study the course material, and for collaboration and discussion
Tutors	To provide feedback to students in discussion forum and face to face sessions
Teacher (designer)	To use the system to design and write the activity
Technical designer	To ensure technical functioning and consistent look/design of course materials and user interface

Stakeholders and Interests

Institution	Quality control

Preconditions

1	Video of someone doing a presentation
2	VLE or discussion forum, resource repository, and quiz services
3	Multimedia teaching lab if delivering by blended learning

4	

Teaching Approach

Situative, collaborative, activity based, assimilative, adaptive, communicative, productive, blended learning

Main Success Scenario

1	Teacher accesses course resources, edits and updates them if necessary, and saves them in course area on system
2	Teacher briefs tutors about the activity by email
3	Teacher shows video to students and shows them what courses resources are on the system
4	Students log into system which delivers self assessment quiz questions (see usecase 'quiz')
5	Students use quiz to identify their strengths and weaknesses and to select relevant advice from the course resources on the system
6	Teacher divides students into small groups, allocates each group to a tutor, and enters group membership and tutor information into system
7	Students develop a plan for improving their presentation skills, based on step 5, and present this to their group face to face
8	Students and tutors give constructive feedback on presentations
9	Students and tutors use group discussion forum to reflect on what they have learnt and comment on others' reflections (see usecase 'discussion)
10	Teacher uses video and presentation software (eg. Powerpoint) to deliver a session on presentation structure to start students off preparing mini-presentations for their groups
11	Students and tutors use group discussions on discussion forum, including uploading draft presentation files, to develop their mini-presentations through steps 11-14 (see usecase 'discussion')
12	Students deliver mini-presentation to group, receive feedback, and develop mini-presentation further
13	Teacher delivers a session on relaxation techniques (NLP[*]), presentation software tips, etc
14	Students discuss face to face and develop agreed criteria for peer assessment of whole class presentations
15	Students present to class, give and receive feedback according to assessment criteria, and identify where they have succeeded and areas to work on
16	Students give assessed presentation to teacher (and class?)
17	Teacher evaluates records of quiz and discussion and modifies activity for next presentation if necessary

[*] NLP = neuro linguistic programming

Extensions

1a	Course resources do not exist
1a1a	Teacher (designer) collects suitable resources
1a1b	Teacher (designer) and technologist (designer) develop online learning materials (quiz, advice documents)
1a1c	Teacher pilots activity
2a	Teacher briefs tutors face to face
3a	Students cannot meet face to face
3a1a	Teacher checks that course materials are not copyright and makes videos available via the system
3a1b1	Teacher uses synchronous conferencing system, with webcam, and shared whiteboard to show video and deliver face to face sessions
3a1b2	Teacher uses asynchronous discussion forum, with a link to the videos, instead of face to face sessions
5a	System automatically selects appropriate advice to deliver to students, based on students quiz answers (see usecase 'quiz)
7a	Distance students use synchronous conferencing system with webcam in their groups, to deliver presentations and give feedback to rest of group
7b	Distance students make video of their presentation and link it to group discussion forum for feedback from the rest of the group
14a	Distance students use discussion forum to agree criteria for peer assessment of class presentation (see usecase 'discussion')
15a	Distance students use synchronous conferencing system with webcam to deliver presentation to whole class and assess others' presentations
15b	Distance students make video of their presentation and link it to discussion forum for assessment by the rest of the class

Authors

Author 1	Paul Brown, John Hanson Community School
Author 2	John Adams, John Hanson Community School

Use Case Summary

Teacher uses system to deliver a multiple choice quiz that students can do at home or in class, to reinforce learning from a previous lesson

Narrative

During a lesson on any secondary school subject, pupils do an online multiple choice quiz and receive feedback. Following the lesson, they study online follow-up material, participate in a discussion forum and/or online chat, and repeat the multiple choice quiz (with questions and answers in a changed order) at home. The teacher monitors discussion, chat, and quiz results and modifies their next lesson to target any problem areas. The activity encourages pupils to reflect on their knowledge, and develop independent learning skills and the ability to use various e-learning tools

Primary Actor (and goal)

Teacher	To use the system to deliver and administer the activity

Other Actors (and goals)

Pupils	To use the system to take part in the activity
Head of Department	To use the system to monitor pupil and teacher activity

Stakeholders and Interests

School	Improved student performance, and quality control

Preconditions

1	A system to deliver the activity
2	Sufficient computers for classroom use

Teaching Approach

Reflective

Main Success Scenario

1	Teacher assembles self study materials, and quiz, in course area on the system
2	Teacher sets up chat and discussion forum

3	Teacher runs face to face session
4	Pupils discuss in pairs their strengths and weaknesses in subject knowledge
5	Pupils in classroom log into system and do quiz (see usecase 'quiz')
6	Pupils discuss subject knowledge and quiz feedback in face to face whole class discussion
7	Pupils at home log into system and take part in synchronous chat session to discuss homework questions
8	Pupils do homework assignment and save completed homework onto the system
9	System delivers homework to the teacher, who marks it and writes feedback
10	Teacher saves homework marks and feedback to the system, which delivers it to the pupils
11	Pupils retake the quiz (with order of questions and choices changed) (see usecase 'quiz')
12	Pupils continue discussion of points arising on discussion forum (see usecase 'discussion')
13	Pupils email teacher about any points they would like covered in the next lesson
14	Teacher reviews records of chat session, homework submissions, discussion forum, emails, and statistics from quiz and modifies next lesson accordingly
15	Head of department accesses records for tracking pupil progress and quality control

Extensions

1a	Teacher saves links to external resources into course area
3a	Some pupils are at home - ill or excluded
3a1	Teacher sets up webcam and synchronous conferencing system with shared whiteboard to enable absent pupils to participate
3a2	Teacher sets up telephone conference and speaker phone to enable absent students to participate
4a	Absent students use synchronous conferencing to discuss lesson in pairs
4b	Absent students use telephone to discuss lesson in pairs
5a	Pupils can choose one out of a number of quiz, designed to target the areas of knowledge that they have decided they need to work on
13a	Pupils send text message to teacher about points they would like covered in the next lesson

Authors

Author 1	Lou McGill, JISC
Author 2	

Use Case Summary

Teacher sets up and monitors an activity in which students collect, manage, discuss, and evaluate resources, using a modified wiki, as the first stage in a group design project

Narrative

Third year campus based engineering students meet at least twice a week and are in online or face to face contact with each other and their tutor at other times. As the first stage in a design project to produce a 'proof-of-concept' model of a domestic ice crushing machine, teams of 4 students search for and manage relevant resources, and write a reflective team report and presentation for other teams and tutors using the wiki pages.

The online environment offered to support the activity includes a range of tools to support communication – email, skype, communication features inside the wiki environment (shoutbox, discussion forums, email, blogging). The wiki tool also provided a workspace for students to share content with team members. The wiki (LauLima) had been extensively modified with a permissions system which allowed selective sharing of content. Students, therefore, had control of who could access their content. The wiki had also been modified to allow hierarchical file storage systems (file galleries) rather than a flat file structure.

Students were asked to store their files (photographs, scanned drawings, word documents) within an organised file gallery area. Students used concept mapping to devise a folder structure for the gallery that had meaning for them, thus reducing the need for search and index facilities, although these were also provided

Following the activity selected resources were archived for future teaching

Primary Actor (and goal)

Teacher	To use the system to run the activity and to enable the creation of useful student-generated resources for future teaching and learning

Other Actors (and goals)

Students	To acquire information management literacies and develop design ideas for the next stage of the project. To gain skills and experience in design engineering
Librarian	To improve students information management literacies and to archive resources for future teaching and learning
Tutors	To support student groups

Stakeholders and Interests

Preconditions

1	Modified wiki
2	Computer lab with terminals, or studio with laptops and web access for students for face to face briefing sessions
3	
4	

Teaching Approach

Project based learning, blended learning

Main Success Scenario

1	Teacher writes briefing materials to be presented via wiki and facc to face and saves them onto the wiki
2	Teacher organises students into groups of four, allocates groups to a tutor, and tells the wiki
3	Wiki allocates discussion areas and workspace to the groups
4	Teacher and librarian brief students about the activity at a face to face session, focusing on use of concept maps to support the planning stages of information searching, including identifying search terms, using appropriate sources, modifying searches, evaluating resources, copyright issues, organising information, assimilating found information into their own design concepts and referencing
5	Students log into wiki which attaches them to the correct group workspace and discussion forum
6	Student groups use concept mapping service which allows saving to a server (eg. Cmap), and group discussion space (see usecase 'discussion') within the wiki, to devise a folder structure for organising their work and the resources collected, and to generate search terms
7	Tutors monitor group workspace and are in online communication with students throughout steps 6-17
8	Student groups create folder structure in their workspace on the wiki

9	Students save concept map, and future individual and group work, in appropriate folders on the wiki,, throughout steps 9-17 File types include text, graphic, photos, video, audio, presentation, spreadsheets
10	Students record search terms for each file saved, using terms generated in step 6
11	Students set access permissions for each file saved onto the wiki, permitting individual, group, tutor, whole class, teacher, reading, adding to, or editing
12	Teacher and librarian give students further help with accessing information/conducting research into the design problem at a face to face session, including copyright issues, and referencing as well as quality, formats and size of content loaded into the wiki environment.
13	Students locate and record resources from the web, members of the public, library databases, existing (physical) products etc, in text, image, audio and video formats
14	Students save resources on wiki in folders created in step 8, setting access permissions for each file
15	Students access other group members' shared files in the file gallery and view or edit them according to permissions
16	Students discuss the resources in the discussion area of the wiki
17	Groups use discussion forum and group workspace to prepare a reflective report and presentation for the rest of the class, based on group wiki pages
18	Teacher accesses shared wiki pages and selects resources and student work for archiving for reuse
19	Teacher saves selected resources in the 'for approval' area of a second 'repository' wiki, tagging them with free text and controlled metadata descriptions
20	Teacher emails librarian to alert them to resources for approval
21	Librarian accesses resources in approval area, checks them for quality, copyright etc, completes tagging if necessary, and saves them into the file gallery of the 'repository' wiki
16	

Extensions

6a	Students use skype and webcams to share, discuss and develop concept maps
6b	Sudents use synchronous conferencing system with shared whiteboard or remote assist to share, discuss, and develop concept maps
6c	Students use pencil, paper, and face to face meetings to share, discuss, and develop concept maps
21a	Librarian rejects resource and does not save it to file gallery

Appendix B

Evaluation Experiment Two

B.1 Personalised Learning Activity Wevb Application

B.1.0.1 Models

Service Model An example of a metadata description for the discussion forum service is provided in figure B.1

Learner Model As part of the three use cases, adaptation is carried out, to varying degrees, based on the system's view of the learner. To support this, the AE is provided with a Learner Model containing information about the individual learner that is necessary for the adaptation process. In the case of the system developed to support these use cases, the Learner Model is based on the IMS Learner Information Package (LIP) specification. Three aspects of the learner are modelled for the system, their competency/skill in a specific subject area, accessibility requirements, for example their literacy, and whether or not the user has a Google account. To support these aspects of the learner, three elements of the LIP specification have been identified as suitable for modelling this information:

- qcl (qualification)
- accessibility
- securityKey

Figures B.2 and B.3, show example instances of such a Learner Model. As can be seen in the examples, the skills of the learner are captured in the **qcl** element of the Learner

```
<service>
    <general>
        <name>SIMPLEFORUM</name>
        <description>
            A portlet, which allows the user to use a simple forum service
        </description>
    </general>
    <functional>
        <parameters>
            <parameter>
                <name>id</name>
                <type>username</type>
            </parameter>
            <parameter>
                <name>passwd</name>
                <type>password</type>
            </parameter>
        </parameters>
        <preconditions>
            <predicate>
                <name>isliterate</name>
            </predicate>
        </preconditions>
        <effects>
            <predicate>
                <name>discussion</name>
                <variables>
                    <variable>
                        <name>asynchronous</name>
                        <type>discussionType</type>
                    </variable>
                </variables>
            </predicate>
        </effects>
    </functional>
    <technical>
        <type>wsrp</type>
        <portletIdentifier>ForumPortlet.ForumPortlet</portletIdentifier>
        <getMarkupURI>
            http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer
        </getMarkupURI>
        <exitCondition>Finish=yes</exitCondition>
    </technical>
</service>
```

Figure B.1: Metadata description of a discussion forum service used in experiment four.

Model, lines 8 to 13 of figure B.2. The valid values for this application are one of three levels: novice, intermediate or expert. The literacy of the learner is captured in the **accessibility** element, lines 14 to 22 of figure B.2. The valid values for both the read and write proficiency values are poor, intermediate, good and excellent. Finally, the learner's Google account details, if any, are captured in the **securityKey** element, lines 23 to 43 of figure B.2.

```
1   <learnerinformation xmlns = "http://www.imsglobal.org/xsd/imslip_v1p0"
2               xsi:noNamespaceSchemaLocation="/home/okeeffei/Desktop/imslip_v1p0.xsd"
3               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4               xsi:schemaLocation = "http://www.imsglobal.org/xsd/imslip_v1p0
5                                     http://www.imsglobal.org/xsd/imslip_v1p0.xsd">
6       <identification>
7           <name></name>
8       </identification>
9       <qcl>
10          <title>python</title>
11          <level>
12              <text>expert</text>
13          </level>
14      </qcl>
15      <accessibility>
16          <language>
17              <typename>
18                  <tyvalue>English</tyvalue>
19              </typename>
20              <proficiency profmode="Read">Excellent</proficiency>
21              <proficiency profmode="Write">Excellent</proficiency>
22          </language>
23      </accessibility>
24      <securitykey>
25          <typename>
26              <tyvalue>GoogleAccount</tyvalue>
27          </typename>
28          <keyfields>
29              <fieldlabel>
30                  <typename>
31                      <tyvalue>username</tyvalue>
32                  </typename>
33              </fieldlabel>
34              <fielddata>user@gmail.com</fielddata>
35          </keyfields>
36          <keyfields>
37              <fieldlabel>
38                  <typename>
39                      <tyvalue>password</tyvalue>
40                  </typename>
41              </fieldlabel>
42              <fielddata>secret</fielddata>
43          </keyfields>
44      </securitykey>
45  </learnerinformation>
```

Figure B.2: Example Instance of a Learner Model

```
<learnerinformation xmlns = "http://www.imsglobal.org/xsd/imslip_v1p0"
            xsi:noNamespaceSchemaLocation="/home/okeeffei/Desktop/imslip_v1p0.xsd"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation = "http://www.imsglobal.org/xsd/imslip_v1p0
                                  http://www.imsglobal.org/xsd/imslip_v1p0.xsd">
    <identification>
        <name></name>
    </identification>
    <qcl>
        <title>python</title>
        <level>
            <text>novice</text>
        </level>
    </qcl>
    <accessibility>
        <language>
            <typename>
                <tyvalue>English</tyvalue>
            </typename>
            <proficiency profmode="Read">Poor</proficiency>
            <proficiency profmode="Write">Poor</proficiency>
        </language>
    </accessibility>
</learnerinformation>
```

Figure B.3: Example Instance of a Learner Model

The two Learner Models shown in figures B.2 and B.3 will be used as part of the use case implementations. As such they represent learners with properties from different ends of the scale of possible values, as illustrated in table B.1. This will allow the range of adaptive behaviours in the use cases to be illustrated based on just these two Learner Model instances.

Property	**Learner 1**	**Learner 2**
qcl (python)	expert	novice
accessability	excellent	intermediate
securityKey (google a/c)	yes	no

Table B.1: Comparison of example Learner Model properties

Narrative Model In addition to the metadata descriptions of the services (collectively referred to as the service model), it is also necessary to develop a Narrative Model for each of the use cases being implemented. This model describes the sequencing of the activities in accordance with the description of the user case as well as describing the rules that govern the adaptive sequencing within the constraints of the activity. As specified in the implementation chapter, this narrative is written in Lisp.

B.1.0.2 Adaptive Web Application Implementation

Services As the focus of the implementation of the three use cases is on the system's ability to support the functionality required by the use cases and not on the usability of the system from an end user perspective, the necessary services were only implemented in the form of 'mock-ups'. That is, individual portlets were developed for each of the required services and those portlets were made available using the WSRP protocol. However, the functionality that the portlets provided varied greatly in its completeness as they were not designed to allow a user to take part in a complete 'end to end' activity. In some cases, such as the eMail and Quiz services, complete portlet applications were implemented while other portlets, such as the simple bookmarking portlet, were obtained from the Openportal portlet repository. Further portlets were implemented as simple HTML iframes that allowed a third part web application to be embedded in the portlet window so that the functionality required from the portlet could be simulated. Examples of such portlets include an email portlet that embeds Googles Gmail service, a document authoring portlet that embeds Google's Docs service and a concept mapping portlet that embeds the Bubbl.us service.

The necessary portlets were developed in accordance with the JSR-168 portlet specification and deployed to a Glassfish application server running the Openportal portal container and WSRP implementation. Using the functionality provided by Openportal the implemented portlets were exposed using the WSRP protocol. Figure B.4 provides a screenshot of a simple eMail portlet running on the Glassfish application server.

Basic Web Application A single web application was developed to support the execution of all three of the use cases. This 'webapp' closely follows the design of a typical PWE application as specified in the Implementation chapter, section 5.2.7. As such it provides five key functionalities:

Login/Logout The basic authentication mechanism allows the system to uniquely identify the users so that the appropriate user model and PWE can be loaded, figure B.5.

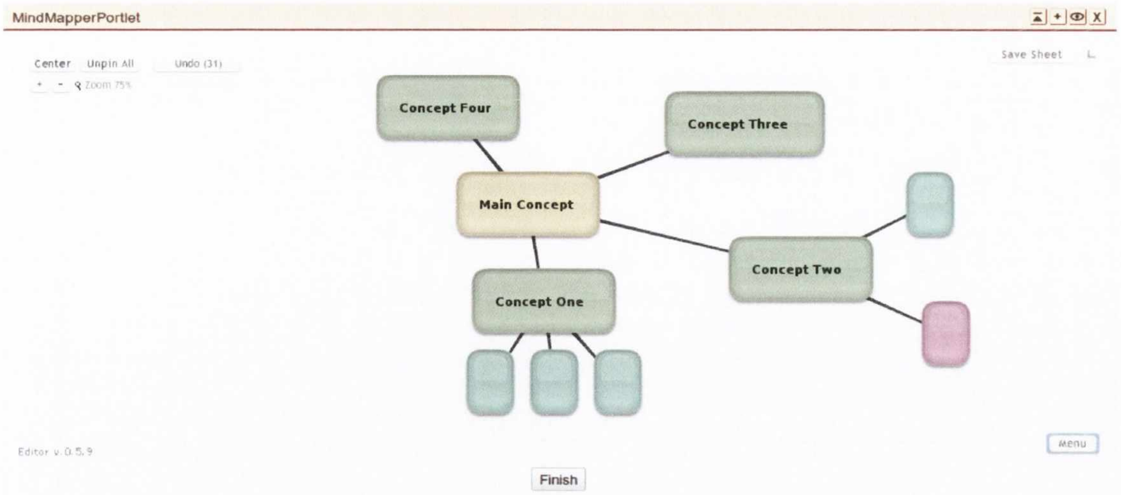


Figure B.4: Concept Mapping portlet running on Glassfish application server

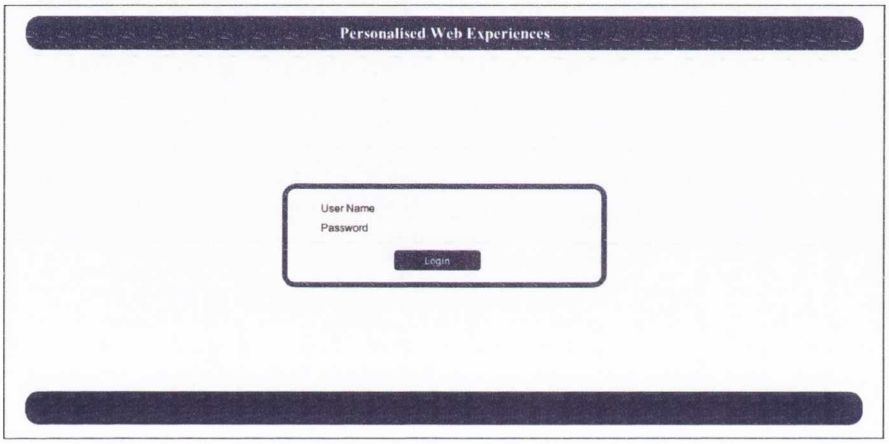
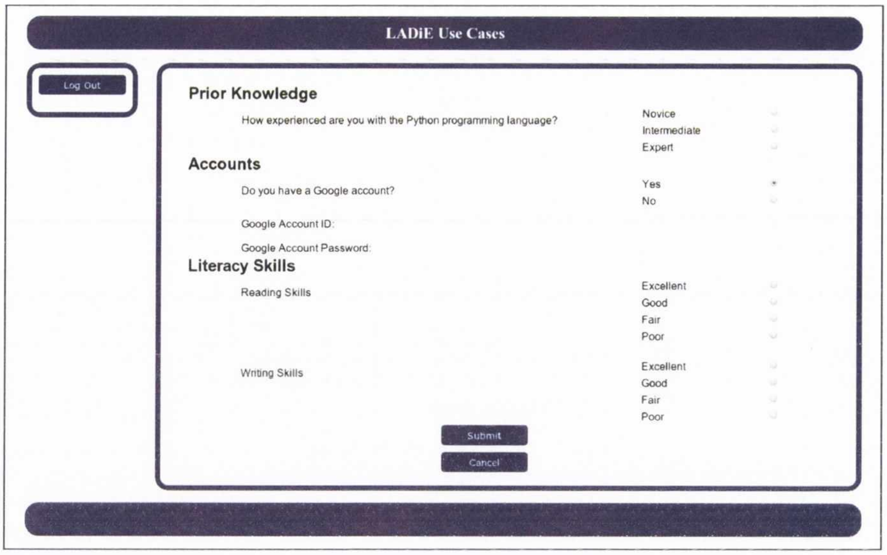


Figure B.5: Screenshot of Login Page

Learner Model Elicitation The webapp provides a Learner Model elicitation mechanism in the form of a simple HTML form that is used to elicit the necessary information from the user in order to populate their Learner Model. The form can also be used to update an existing Learner Model for a returning user.

As shown in figure B.6, this form consists of three questions corresponding to the three key areas of the Learner Model as discussed previously. The learner is first asked to rate their level of skill in the Python programming language. They have three choices, beginner, intermediate or expert. The next question asks the user if they have any literacy issues and finally the user is asked if they have a Google account.

Obviously these questions would be unrealistic in a real world scenario but for the purpose of populating the Learner Model for these use cases they are sufficient.



Prior Knowledge

How experienced are you with the Python programming language? ○ Novice ○ Intermediate ○ Expert

Accounts

Do you have a Google account? ● Yes ○ No

Google Account ID:

Google Account Password:

Literacy Skills

Reading Skills ○ Excellent ○ Good ○ Fair ○ Poor

Writing Skills ○ Excellent ○ Good ○ Fair ○ Poor

Figure B.6: Screenshot of the Learner Model Elicitation Page

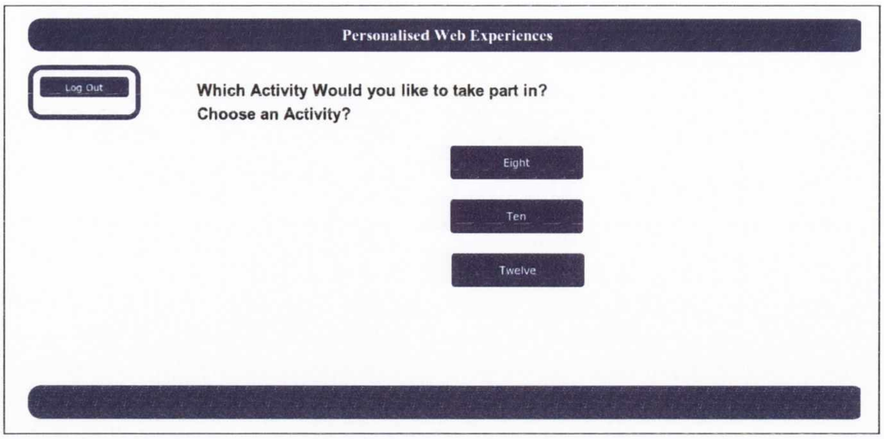
Build The build functionality is invoked when the user first creates their Learner Model or whenever they update their model through the Learner Model Elicitation Page. The aim of this is to create a new PWE for the learner and to deploy that model so that the user can begin to interact with it. The generation of a new PWE requires the web portal to start the execution of the Narrative Model by the Adaptive Engine and for the resulting APCM to be converted to an equivalent personalised WSBPEL process, which is deployed ready for the learner to interact with it.

Activity Selection When a returning user logs in to the webapp they are presented with a page that allows them to select one of the three use cases to take part in, see figure B.7. Similarly, a new user will be presented with this screen after completing the Learner Model elicitation form.

Activity This page provides the interface through which the user interacts with the PWE. As shown in figure B.8, this page provides the user with the portlet through which they can interact with their personalised service composition as part of the PWE.

B.1.0.3 Use Case Scenarios

Scenario One The first scenario implemented was based on the LADiE use case 8 as described in the previous section. For convenience the activity diagram for this use case



Personalised Web Experiences

Log Out

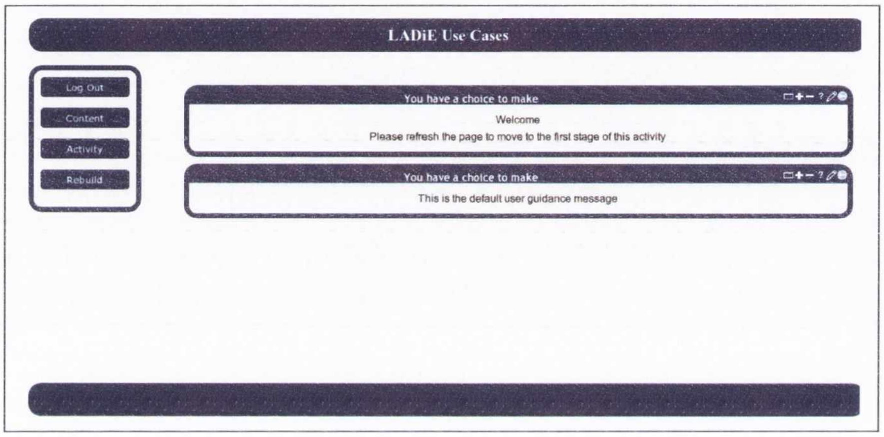
Which Activity Would you like to take part in?
Choose an Activity?

Eight

Ten

Twelve

Figure B.7: Screenshot of Use Case Selection Page



LADiE Use Cases

Log Out

Content

Activity

Rebuild

You have a choice to make

Welcome
Please refresh the page to move to the first stage of this activity

You have a choice to make

This is the default user guidance message

Figure B.8: Screenshot of Learners initial view of activity

has been repeated below, see figure B.9. As illustrated, this use case requires the use of 3 of the 5 supported workflow patterns:

- Sequence
- Parallel Split
- Synchronisation

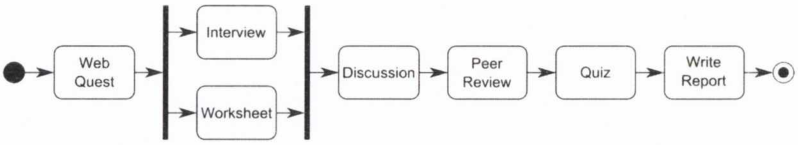


Figure B.9: Activity Diagram for Scenario One

In addition to demonstrating the ability of the system to support these three workflow patterns, the use case also demonstrates the use of the adaptive service selection mechanism in order to adaptively select services that are appropriate based on the literacy skills of the learner as well as the learner's ability to access the internet. The requirement for these two adaptive axes is specified in the extensions to the LADiE use case. In addition, this use case also demonstrates the dynamic composition of services. As shown in the use case activity diagram, the first task specified is a Webquest. There is no single service in the set of services made available to the system that can satisfy the requirements of a Webquest and as such, the system will need to dynamically compose a set of services that can be used to realise the task.

The narrative model used to describe the activity and the adaptive behaviours discussed is shown in figure B.10 The code itself can be considered in two parts, the first is a function, **generateProblem** (see lines 1 to 46), that allows for the generation of a PDDL planning problem without the need to repeatedly specify the static components of the problem. The function takes as a parameter the set of goals that the AI Planner should attempt to satisfy. An important part of the formulation of the planning problem in the context of the Narrative Model is the adaptive manipulation of the objects that are specified to exist in the planning problem. As shown in lines 27-29 and 37-39, the Narrative Model contains rules that adaptively specify the literacy skills of the learner as well as other important aspects of the learner such as their access to the internet. These rules are informed by the Learner Model, which is queried using an XPath expression to retrieve the necessary information from the model, lines 21-25 and 31-35 of the Narrative Model. By controlling the objects that are declared to exist in the planning problem, the planning process itself can be influenced by the Adaptive Engine so that the most appropriate service or composition of services is selected. In addition to the **generateProblem** the first part of the narrative contains a function call to the **generateDomain** AE Custom Function, line 48. This function, described in the implementation chapter **??**, generates the planning domain as mentioned previously.

The second part of the Narrative Model, lines 50 to 64, is focused on creating an Abstract Personalised Composition Model that conforms to the structure of the activity. In this use case there is no adaptive sequencing to be carried out so this is relatively straight forward. As shown in lines 55-64 of the Narrative Model, figure B.10, the APCM is generated through a series of AE Custom Function calls that allow the structure of the model to be manipulated.

```
(defun generateProblem (goal)
  (setq PROBLEM
    "(define (problem test-problem)
        (:domain test)
        (:objects
          ian - username,
          key - password,
          web - searchType,
          video - searchType,
          document - docType,
          presentation - docType,
          SQL - quizTopic,
          python - language,
          synchronous - discussionType,
          asynchronous - discussionType)

        (:init
          (webaccess)"
  )

  (setq LITERACY_XPATH_QUERY
    "//accessibility/language[typename/tyvalue='English']/proficiency[@profmode='Read']/text()")

  (setq LITERACY_QUERY_RESULT
    (first (xpath-query-model LEARNER_MODEL LITERACY_XPATH_QUERY)) )

  (if (string-equal LITERACY_QUERY_RESULT "excellent")
    (setq PROBLEM (concatenate 'string PROBLEM "(isLiterate)"))
  )

  (setq GOOGLE_XPATH_QUERY
    "boolean(//securitykey[typename/tyvalue='GoogleAccount'])" )

  (setq GOOGLE_QUERY_RESULT
    (first (xpath-query-model LEARNER_MODEL GOOGLE_XPATH_QUERY)) )

  (if (string-equal GOOGLE_QUERY_RESULT "true")
    (setq PROBLEM (concatenate 'string PROBLEM "(hasGoogleAccount)"))
  )

  ;; Close off init statement
  (setq PROBLEM (concatenate 'string PROBLEM ")" ))

  ;; Add goals to problem definition and close off all parenthesis
  (concatenate 'string PROBLEM "(:goal " goal "))" )
)

(generateDomain)

;; Create initial composition model structure
(create-model COMPOSITION_MODEL)
(update-model COMPOSITION_MODEL "composition")
(add-attribute COMPOSITION_MODEL "name" "LADiEUseCase")

(update-model COMPOSITION_MODEL "sequence")
  (addService "newService1" (generateProblem "(and (search web)(bookmarks))") "Step 1")
  (update-model COMPOSITION_MODEL "parallel")
    (addService "newService2" (generateProblem "(interview)") "Step 2")
    (addService "newService3" (generateProblem "(worksheet)") "Step 3")
  (cd COMPOSITION_MODEL "..")
  (addService "newService4" (generateProblem "(discussion asynchronous)") "Step 4")
  (addService "newService5" (generateProblem "(quiz SQL)") "Step 5")
  (addService "newService6" (generateProblem "(created document)") "Step 6")
(cd COMPOSITION_MODEL "..")
```

Figure B.10: Narrative Model for Scenario One

In this case, the first thing that is specified is a **sequence** element, line 55, to act as a wrapper for the whole activity. This is added to the APCM using the **update-model** custom function. The next step is to add a service corresponding to the Webquest task as described by the sequence diagram in figure B.9. This is achieved by calling the **add-service** custom function, line 56. This will, as discussed in section **??** of the Implementation Chapter, invoke the AI Planner service and process the response in order to update the APCM. The **add-service** custom function takes as a parameter the planning problem that the AI Planner should attempt to solve in PDDL syntax. To make the generation of this problem easier the function discussed previously is used so that only the planning goals need to be specified. In this instance, the required service(s) need to allow the user to achieve two things, to search an information source and to record interesting search results. To specify these two requirements the individual goals are wrapped in an 'and' construct, which is part of the PDDL syntax.

The next step in the activity is a parallel split, which allows the user to access the services for the interview and worksheet tasks at the same time. As shown in line 57 of the Narrative Model, this is achieved by adding a new 'parallel' element to the APCM, again using the **update-model** custom function. This is added immediately after the previous service so it will follow in sequence from that service as they are both contained in the sequence element specified previously. Within this parallel element, two calls are made to the **add-service** function corresponding to the two services required.

The next line in the narrative, line 60, 'steps out' of the parallel element of the APCM so that subsequent services or control flow constructs are added to the model as part of the 'sequence' wrapper element and not the 'parallel' element. The remaining tasks in the activity are then added to the model in lines 61-63 through further calls to the **add-service** custom function.

Figures B.11 and B.12 show the APCMs generated from the execution of this narrative model for two different learners. These APCMs were generated based on the example Learner Models discussed previously, see figures B.2 and B.3 respectively. As can be seen by comparing the services selected for the worksheet and quiz tasks, lines 25 and 38 of the APCMs, the differences in the two Learner Models result in different services being selected.

Another interesting feature of the generated APCMs is that, as discussed previously,

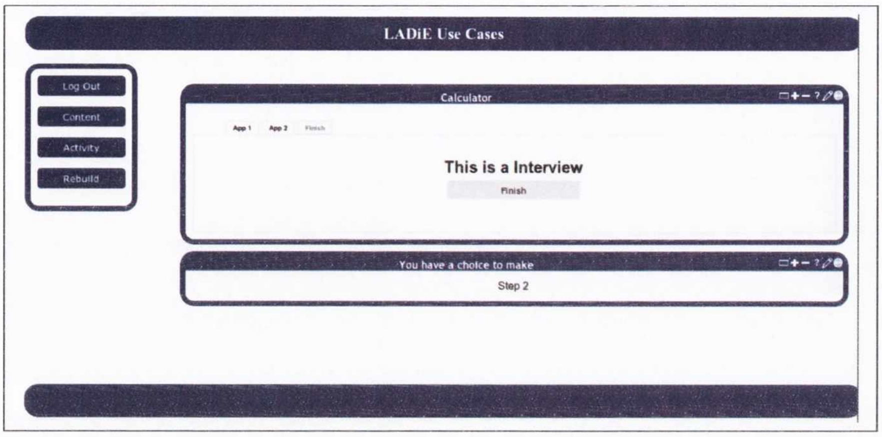
```xml
<?xml version="1.0" encoding="UTF-8"?>
<composition name="LADiEUseCase8">
  <sequence>
    <parallel>
      <service name="GOOGLESEARCH" type="wsrp">
        <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
        <portlet>SearchPortlet.SearchPortlet</portlet>
        <endcondition>yes=Finish</endcondition>
        <guidance>Step 1</guidance>
      </service>
      <service name="GOOGLENOTEBOOK" type="wsrp">
        <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
        <portlet>GoogleNotebookPortlet.GoogleNotebookPortlet</portlet>
        <endcondition>yes=Finish</endcondition>
        <guidance>Step 1</guidance>
      </service>
    </parallel>
    <parallel>
      <service name="INTERVIEW" type="wsrp">
        <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
        <portlet>InterviewPortlet.InterviewPortlet</portlet>
        <endcondition>yes=Finish</endcondition>
        <guidance>Step 2</guidance>
      </service>
      <service name="WORKSHEET" type="wsrp">
        <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
        <portlet>WorksheetPortlet.WorksheetPortlet</portlet>
        <endcondition>yes=Finish</endcondition>
        <guidance>Step 3</guidance>
      </service>
    </parallel>
    <service name="GMAILCLIENT" type="wsrp">
      <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
      <portlet>SimpleEMailPortlet.SimpleEMailPortlet</portlet>
      <endcondition>yes=Finish</endcondition>
      <guidance>Step 4</guidance>
    </service>
    <service name="QUIZ" type="wsrp">
      <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
      <portlet>AudioQuiz.AudioQuiz</portlet>
      <endcondition>quit=Quit</endcondition>
      <guidance>Step 5</guidance>
    </service>
    <service name="GOOGLEDOCUMENTS" type="wsrp">
      <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
      <portlet>GoogleDocsPortlet.GoogleDocsPortlet</portlet>
      <endcondition>yes=Finish</endcondition>
      <guidance>Step 6</guidance>
    </service>
  </sequence>
</composition>
```

Figure B.11: Scenario One APCM for test user 1

```
<composition name="LADiEUseCase8">
  <sequence>
    <parallel>
      <service name="YOUTUBESEARCH" type="wsrp">
        <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
        <portlet>YouTubeSearchPortlet.YouTubeSearchPortlet</portlet>
        <endcondition>yes=Finish</endcondition>
        <guidance>Step 1</guidance>
      </service>
      <service name="SIMPLEBOOKMARKING" type="wsrp">
        <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
        <portlet>bookmarkportlet.BookmarkPortlet</portlet>
        <endcondition>Finish=yes</endcondition>
        <guidance>Step 1</guidance>
      </service>
    </parallel>
    <parallel>
      <service name="INTERVIEW" type="wsrp">
        <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
        <portlet>InterviewPortlet.InterviewPortlet</portlet>
        <endcondition>yes=Finish</endcondition>
        <guidance>Step 2</guidance>
      </service>
      <service name="WORKSHEET" type="wsrp">
        <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
        <portlet>WorksheetPortlet.WorksheetPortlet</portlet>
        <endcondition>yes=Finish</endcondition>
        <guidance>Step 3</guidance>
      </service>
    </parallel>
    <service name="MYEMAILCLIENT" type="wsrp">
      <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
      <portlet>SimpleEMailPortlet.SimpleEMailPortlet</portlet>
      <endcondition>yes=Finish</endcondition>
      <guidance>Step 4</guidance>
    </service>
    <service name="SIMPLEQUIZ" type="wsrp">
      <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
      <portlet>Quiz2.Quiz</portlet>
      <endcondition>quit=Quit</endcondition>
      <guidance>Step 5</guidance>
    </service>
    <service name="GOOGLEPRESENTATION" type="wsrp">
      <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
      <portlet>GooglePresentationPortlet.GooglePresentationPortlet</portlet>
      <endcondition>yes=Finish</endcondition>
      <guidance>Step 6</guidance>
    </service>
  </sequence>
</composition>
```

Figure B.12: Scenario One APCM for test user 2

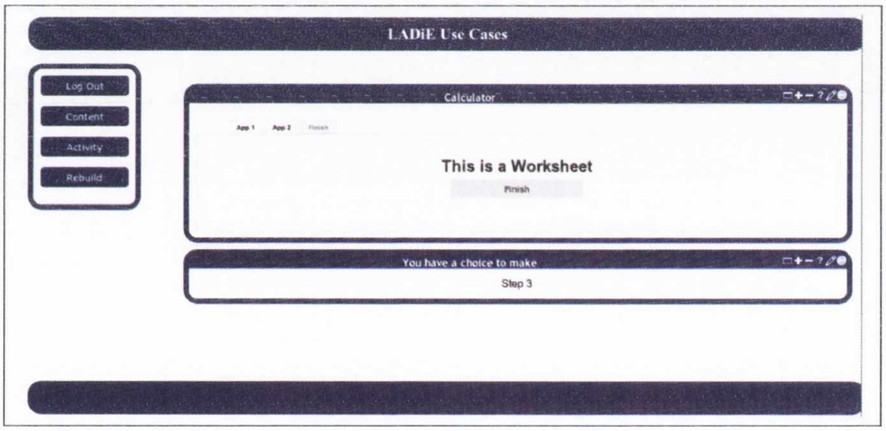
it was necessary for the AI Planner to generate a composition of services to satisfy the requirements of the Webquest task. Lines 4 to 17 of the APCM show that the planner selected two services, a search service and a bookmarking service to satisfy these requirements.



This is a Interview

Step 2

Figure B.13: Screenshot of interactive service to support the first task of Scenario One



This is a Worksheet

Step 3

Figure B.14: Screenshot of interactive service to support the second task of Scenario One

Scenario Two The purpose of this scenario is to demonstrate the use of the Exclusive Choice and Simple Merge workflow patterns in an activity. As shown in the previous analysis of the LADiE use cases, none of the use cases require either of these patterns. In order to demonstrate and evaluate the use of these patterns as part of a realistic activity a LADiE use case was modified to incorporate both an Exclusive Choice and Simple Merge. The activity diagram for this modified use case is shown in figure B.15. As can be seen from the diagram, the modified activity is based on LADiE use case 8, which was implemented previously in its original form. The use case has been modified so that instead of the learner first taking part in the Webquest and then the interview tasks, they are instead given a choice between either activity. For example, the learner can choose to obtain the necessary information by searching the internet or alternatively by interviewing an expert in the subject domain.

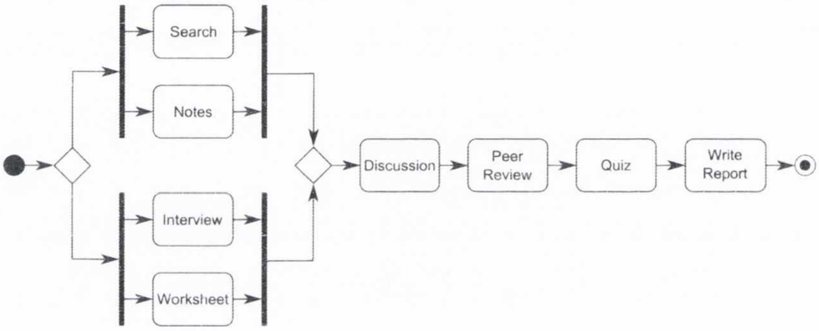


Figure B.15: Activity Diagram for Scenario Two

The corresponding Narrative Model for this use case is provided below, see figure B.16. The significant differences between this model and the Narrative Model from the unmodified version of use case 8 can be seen in lines 29-38 where the two parallel services corresponding to the Webquest and interview tasks of the original use case 8 have been wrapped in a **condition** element. The means that the learner will be given a choice between the two possible paths in the activity, which are represented by the parallel elements in the generated APCM. An example of a APCM generated for this activity is shown in figure B.17.

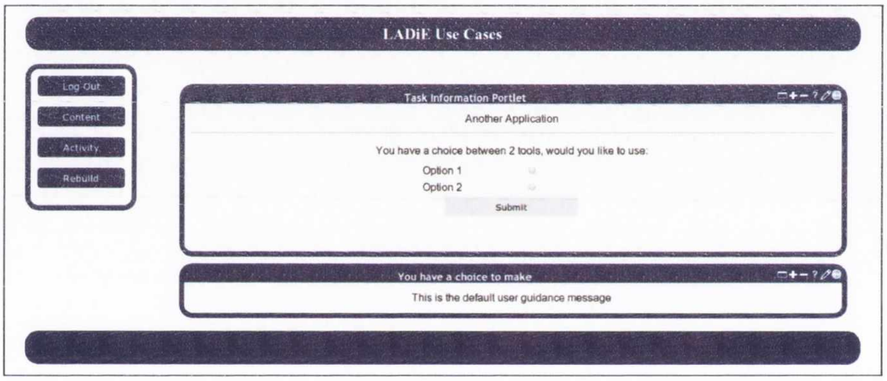
A screenshot of the activity portlet in which the user is presented with the option between the two possible paths is shown in figure B.18. Based on the option selected in this portlet view, the user will follow the corresponding path in the activity.

```
(defun generateProblem (goal)
    (setq START
        "(define (problem test-problem)
                    (:domain test)
                    (:objects
                        ian - username,
                        key - password,
                        web - searchType,
                        video - searchType,
                        document - docType,
                        presentation - docType,
                        SQL - quizTopic,
                        python - language,
                        synchronous - discussionType,
                        asynchronous - discussionType)
                    (:init (webaccess)(hasGoogleAccount))
                    (:goal ")
    (setq END "))")

    (concatenate 'string START goal END)
)

;; Create initial composition model structure
(create-model COMPOSITION_MODEL)
(update-model COMPOSITION_MODEL "composition")
(add-attribute COMPOSITION_MODEL "name" "LADiEUseCase")

(update-model COMPOSITION_MODEL "sequence")
    (update-model COMPOSITION_MODEL "condition")
        (update-model COMPOSITION_MODEL "parallel")
            (addService "newService1" (generateProblem "(bookmarks)") "Step 1")
            (addService "newService2" (generateProblem "(search web)") "Step 2")
        (cd COMPOSITION_MODEL "..")
        (update-model COMPOSITION_MODEL "parallel")
            (addService "newService3" (generateProblem "(interview)") "Step 3")
            (addService "newService4" (generateProblem "(worksheet)") "Step 4")
        (cd COMPOSITION_MODEL "..")
    (cd COMPOSITION_MODEL "..")
    (addService "newService5" (generateProblem "(discussion asynchronous)") "Step 5")
    (addService "newService6" (generateProblem "(quiz SQL)") "Step 6")
    (addService "newService7" (generateProblem "(created document)") "Step 7")
(cd COMPOSITION_MODEL "..")
```

Figure B.18: Narrative Model for Scenario Two

```
<?xml version="1.0" encoding="UTF-8"?>
<composition name="LADiEUseCase8">
  <sequence>
    <condition>
      <parallel>
        <service name="SIMPLEBOOKMARKING" type="wsrp">
          <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
          <portlet>bookmarkportlet.BookmarkPortlet</portlet>
          <endcondition>Finish=yes</endcondition>
          <guidance>Step 1</guidance>
        </service>
        <service name="GOOGLESEARCH" type="wsrp">
          <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer
          </endpoint>
          <portlet>SearchPortlet.SearchPortlet</portlet>
          <endcondition>yes=Finish</endcondition>
          <guidance>Step 2</guidance>
        </service>
      </parallel>
      <parallel>
        <service name="INTERVIEW" type="wsrp">
          <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
          <portlet>InterviewPortlet.InterviewPortlet</portlet>
          <endcondition>yes=Finish</endcondition>
          <guidance>Step 3</guidance>
        </service>
        <service name="WORKSHEET" type="wsrp">
          <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
          <portlet>WorksheetPortlet.WorksheetPortlet</portlet>
          <endcondition>yes=Finish</endcondition>
          <guidance>Step 4</guidance>
        </service>
      </parallel>
    </condition>
    <service name="GMAILCLIENT" type="wsrp">
      <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
      <portlet>SimpleEMailPortlet.SimpleEMailPortlet</portlet>
      <endcondition>yes=Finish</endcondition>
      <guidance>Step 5</guidance>
    </service>
    <service name="SIMPLEQUIZ" type="wsrp">
      <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
      <portlet>Quiz2.Quiz</portlet>
      <endcondition>quit=Quit</endcondition>
      <guidance>Step 6</guidance>
    </service>
    <service name="GOOGLEDOCUMENTS" type="wsrp">
      <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
      <portlet>GoogleDocsPortlet.GoogleDocsPortlet</portlet>
      <endcondition>yes=Finish</endcondition>
      <guidance>Step 7</guidance>
    </service>
  </sequence>
</composition>
```

Figure B.19: Scenario Two APCM for test user 1



Another Application

You have a choice between 2 tools, would you like to use:

Option 1 ○
Option 2 ○

You have a choice to make

This is the default user guidance message

Figure B.20: Screenshot of Scenario Two, User selection of Exclusive Choice branch

Scenario Three The third scenario implemented is designed to demonstrate the use of the adaptive sequencing of services as part of a real world activity. Although the LADiE use cases can easily be modified to make use of adaptive selection of services by taking into account the extensions to each use case there is no such scope for adaptive sequencing within the definitions of the use cases. As such, it was necessary to modify an existing use case so that it incorporates this functionality.

To achieve this, two LADiE use cases have been used as the basis for a single new activity in which the path through the activity is based on the Learner Model. The basis for this activity is LADiE use case 12 in which the learner uses an online programming service to complete a worksheet. During this activity a discussion forum is available to the learner so that they can discuss the activity with their classmates. Although the original LADiE use case was based on the learner working with a SCP[1], it has been modified in this instance so that the SCP environment is replaced with an interpreter for the Python programming language. As specified in the use case, this is a static activity without any adaptive sequencing. To incorporate adaptive sequencing into this activity, the sequencing of task in LADiE use case 12 has been extended to incorporate aspects of use case 16. This results in a more complex activity in which alternative branches exist, which can be adaptively selected by the AE based on the Learner Model. LADiE use case 16 is a project planning activity in which students collaborate on how they will carry out their project and write a report detailing their plans.

By combining these two activities a new adaptive activity has been defined in which inexperienced or intermediate students are presented with a worksheet based activity, as they would in the original definition of LADiE use case 12. In contrast, users that are experienced in the subject matter (ie. the python programming language) are instead provided with a project based activity in which they plan and research a project. In both cases the learners are provided with a discussion service and a programming environment service.

To illustrate these two alternative service compositions, two activity diagrams have been provided, which illustrate the resulting compositions. Figure B.21 is an activity diagram representing the activity for an inexperienced or intermediate level student while figure B.22 represents the activity of an expert level student.

[1] Symbolic Computational Program

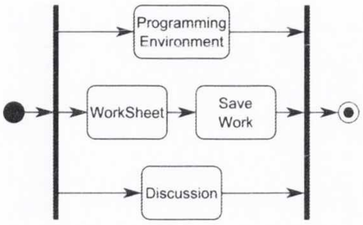


Figure B.19: Activity Diagram for first possible composition from Scenario Three

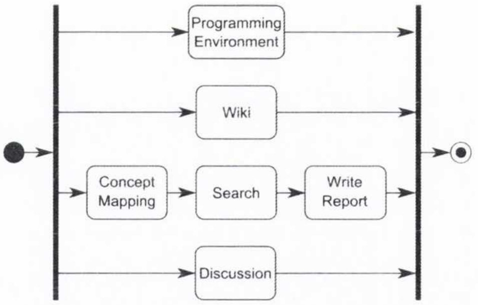


Figure B.20: Activity Diagram for second possible composition from Scenario Three

The Narrative Model for this activity is provided in figure B.21. The important part of this specific model are the adaptation rules that describe the conditions under which the alternative adaptive branches should be taken. Lines 34-35 shows how the appropriate property of the learner, in this case their skill level for the python programming language, is retrieved from the Learner Model using an XPath expression. The information retrieved from the model is then used to inform that conditional branch rule, line 36. Depending on the value of the Python skill property in the Learner Model, one of two different sequences of services will be added to the APSC.

By executing this Narrative Model with two different Learner Models (see figures B.2 and B.3) corresponding to learners with different values for the skill of the learner in Python results in two different APCMs as shown in figure B.22 and B.23

```
(defun generateProblem (goal)
    (setq START
        "(define (problem test-problem)
                   (:domain test)
                   (:objects
                       ian - username,
                       key - password,
                       web - searchType,
                       video - searchType,
                       document - docType,
                       presentation - docType,
                       SQL - quizTopic,
                       python - language,
                       synchronous - discussionType,
                       asynchronous - discussionType)
                   (:init (webaccess))
                   (:goal ")
    (setq END "))")

    (concatenate 'string START goal END)
)

(generateDomain)

;; Create initial composition model structure
(create-model COMPOSITION_MODEL)
(update-model COMPOSITION_MODEL "composition")
(add-attribute COMPOSITION_MODEL "name" "LADiEUseCase12")

(update-model COMPOSITION_MODEL "parallel")
    (addService "newService1" (generateProblem "(and (interpreter python))") "Step 1")
    (addService "newService2" (generateProblem "(and (discussion synchronous))") "Step 2")

    (setq XPATH_QUERY "//qcl[title='python']/level/text/text()")
    (setq QUERY_RESULT (first (xpath-query-model LEARNER_MODEL XPATH_QUERY)))
    (if (string-equal QUERY_RESULT "expert")
        (list
            (addService "newService3" (generateProblem "(and (wiki))") "Step 3")
            (update-model COMPOSITION_MODEL "sequence")
                (addService "newService4" (generateProblem "(and (concept-mapping))") "Step 4")
                (addService "newService5" (generateProblem "(and (search web))") "Step 5")
                (addService "newService6" (generateProblem "(and (write report))") "Step 6")
            (cd COMPOSITION_MODEL "..")
        )
        (list
            (update-model COMPOSITION_MODEL "sequence")
                (addService "newService3" (generateProblem "(and (worksheet))") "Step 3")
                (addService "newService4" (generateProblem "(and (uploadFile))") "Step 4")
            (cd COMPOSITION_MODEL "..")
        )
    )

(cd COMPOSITION_MODEL "..")
```

Figure B.21: Narrative Model for modified Use Case 12

```
<?xml version="1.0" encoding="UTF-8"?>
<composition name="LADiEUseCase12">
  <parallel>
    <service name="PYTHONINTERPRETER" type="wsrp">
      <endpoint>http://kdeg-vm-13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
      <portlet>ProgrammingEnvironmentPortlet.ProgrammingEnvironmentPortlet</portlet>
      <endcondition>Finish=yes</endcondition>
      <guidance>Step 1</guidance>
    </service>
    <service name="SIMPLEIRC" type="wsrp">
      <endpoint>http://kdeg-vm-13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
      <portlet>IRCPortlet.IRCPortlet</portlet>
      <endcondition>Finish=yes</endcondition>
      <guidance>Step 2</guidance>
    </service>
    <service name="WIKI" type="wsrp">
      <endpoint>http://kdeg-vm-13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
      <portlet>WikiPortlet.WikiPortlet</portlet>
      <endcondition>Finish=yes</endcondition>
      <guidance>Step 3</guidance>
    </service>
    <sequence>
      <service name="BUBBLUS" type="wsrp">
        <endpoint>http://kdeg-vm-13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
        <portlet>MindMappingPortlet.MindMapperPortlet</portlet>
        <endcondition>Finish=yes</endcondition>
        <guidance>Step 4</guidance>
      </service>
      <service name="GOOGLESEARCH" type="wsrp">
        <endpoint>http://kdeg-vm-13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
        <portlet>SearchPortlet.SearchPortlet</portlet>
        <endcondition>Finish=yes</endcondition>
        <guidance>Step 5</guidance>
      </service>
      <service name="GOOGLEDOCUMENTS" type="wsrp">
        <endpoint>http://kdeg-vm-13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
        <portlet>GoogleDocsPortlet.GoogleDocsPortlet</portlet>
        <endcondition>Finish=yes</endcondition>
        <guidance>Step 6</guidance>
      </service>
    </sequence>
  </parallel>
</composition>
```

Figure B.24: Scenario Three APCM for test user 1

```
<?xml version="1.0" encoding="UTF-8"?>
<composition name="LADiEUseCase12">
  <parallel>
    <service name="PYTHONINTERPRETER" type="wsrp">
      <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
      <portlet>
        ProgrammingEnvironmentPortlet.ProgrammingEnvironmentPortlet
      </portlet>
      <endcondition>Finish=yes</endcondition>
      <guidance>Step 1</guidance>
    </service>
    <service name="SIMPLEIRC" type="wsrp">
      <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
      <portlet>IRCPortlet.IRCPortlet</portlet>
      <endcondition>Finish=yes</endcondition>
      <guidance>Step 2</guidance>
    </service>
    <sequence>
      <service name="WORKSHEET" type="wsrp">
        <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
        <portlet>WorksheetPortlet.WorksheetPortlet</portlet>
        <endcondition>Finish=yes</endcondition>
        <guidance>Step 3</guidance>
      </service>
      <service name="FTPCLIENT" type="wsrp">
        <endpoint>http://vm13.cs.tcd.ie:8080/producer/wsrp/router/markup/MyProducer</endpoint>
        <portlet>FTPPortlet.FTPPortlet</portlet>
        <endcondition>Finish=yes</endcondition>
        <guidance>Step 4</guidance>
      </service>
    </sequence>
  </parallel>
</composition>
```

Figure B.25: Scenario Three APCM for test user 2

B.2 Narrative Models

```
(setq DOMAIN_MODEL "domainModel")
(setq SERVICE_MODEL_CONNECTION "serviceModelConnection")
(setq PLANNER_WS_URI "http://localhost:8080/planner/Planner3.jws")
(setq PLANNER_OPERATION "plan")
(setq DOMAIN_NAME "test")
(setq COMPOSITION_MODEL "compositionModel")
(setq LEARNER_MODEL "user")

(defun processPlan( actions guidanceMsg )
  (dolist (action actions)
    (setq BASE_QUERY
          (concatenate 'string "/service[general/name='" action "']/technical"))
    (setq SERVICE_TYPE_QUERY (concatenate 'string BASE_QUERY "/type/text()"))
    (setq URI_QUERY (concatenate 'string BASE_QUERY "/getMarkupURI/text()"))
    (setq PORTLET_ID_QUERY (concatenate 'string BASE_QUERY "/portletIdentifier/text()"))
    (setq CONDITION_QUERY (concatenate 'string BASE_QUERY "/exitCondition/text()"))

    (setq TYPE
       (first
          (remote-xpath-query-collection SERVICE_MODEL_CONNECTION SERVICE_TYPE_QUERY)))

    (if (string-equal TYPE "wsrp")
      (list
        (setq URI (first
                     (remote-xpath-query-collection SERVICE_MODEL_CONNECTION URI_QUERY)))

        (setq PORTLET
          (first
            (remote-xpath-query-collection SERVICE_MODEL_CONNECTION PORTLET_ID_QUERY)))

        (setq ENDCONDITION
          (first
            (remote-xpath-query-collection SERVICE_MODEL_CONNECTION CONDITION_QUERY)))

        (update-model COMPOSITION_MODEL "service")
          (add-attribute COMPOSITION_MODEL "name" action)
          (add-attribute COMPOSITION_MODEL "type" TYPE)
          (update-model COMPOSITION_MODEL "endpoint" URI)
          (update-model COMPOSITION_MODEL "portlet" PORTLET)
          (update-model COMPOSITION_MODEL "endcondition" ENDCONDITION)
          (update-model COMPOSITION_MODEL "guidance" guidanceMsg)
        (cd COMPOSITION_MODEL "..")
      )
    )
  )
)
```

```
(defun doSteps ( steps guidanceMsg )
  (if (= (length steps) 0)
    (return-from doSteps 0)
    (list
      (setq STEP (first steps))

      (setq actions
        (xpath-query-model "solutionModel"
          (concatenate ’string "//step[@time=’" STEP "’]/action/name/text()") ))

      (setq TEST (&gt; (length actions) 1))

      (if TEST
        (list
          (update-model COMPOSITION_MODEL "parallel")
          (processPlan actions guidanceMsg)
          (cd COMPOSITION_MODEL "..")
        )
        (list
          (processPlan actions guidanceMsg)
        )
      )

      (doSteps (rest steps) guidanceMsg)
    )
  )
)

(defun generateDomain ()
  ;; Generate problem domain model
  (create-model DOMAIN_MODEL)
  (update-model DOMAIN_MODEL "services")

  (setq modelList (list-models SERVICE_MODEL_CONNECTION))

  (dolist (modelName modelList)
    (load-model SERVICE_MODEL_CONNECTION modelName "tempModel")
    (append-model DOMAIN_MODEL "tempModel")
    (remove-model "tempModel")
  )
)
```

```
(defun addService ( name goal guidanceMsg )

  ;; Call AI planning Web Service
  ;; PDDL-PROBLEM is a list so it should be fine to pass it in without parenthesis
  (setq PLAN (call-web-service   PLANNER_WS_URI
                  PLANNER_WS_URI
                  PLANNER_OPERATION
                    (list DOMAIN_NAME (model-to-string DOMAIN_MODEL) goal)
                   '( "domainName" "domain" "problem")
        )
  )

  (create-model "solutionModel" PLAN)
  (setq STEPS (xpath-query-model "solutionModel" "/solution/step/@time") )
  (doSteps STEPS guidanceMsg)
  (remove-model "solutionModel")
)

(defun generateProblem (goal)
  (setq PROBLEM
    "(define (problem test-problem)
          (:domain test)
          (:objects
            ian - username,
            key - password,
            web - searchType,
            video - searchType,
            document - docType,
            presentation - docType,
            SQL - quizTopic,
            python - language,
            synchronous - discussionType,
            asynchronous - discussionType)

          (:init
            (webaccess)")

          (setq LITERACY_XPATH_QUERY
          "//accessibility/language[typename/tyvalue='English']/proficiency[@profmode='Rea
          (setq LITERACY_QUERY_RESULT
                (first (xpath-query-model LEARNER_MODEL LITERACY_XPATH_QUERY)))
          (if (string-equal LITERACY_QUERY_RESULT "excellent")
            (setq PROBLEM (concatenate 'string PROBLEM "(isLiterate)"))
          )
          (setq GOOGLE_XPATH_QUERY
                "boolean(//securitykey[typename/tyvalue=’GoogleAccount’])")
          (setq GOOGLE_QUERY_RESULT
                (first (xpath-query-model LEARNER_MODEL GOOGLE_XPATH_QUERY)))
          (if (string-equal GOOGLE_QUERY_RESULT "true")
            (setq PROBLEM (concatenate ’string PROBLEM "(hasGoogleAccount)"))
          )

          ;; Close off init statement
          (setq PROBLEM (concatenate ’string PROBLEM ")" ))

          ;; Add goals to problem definition and close off all parenthesis
          (concatenate ’string PROBLEM "(:goal " goal "))" )
)

(generateDomain)

;; Create initial composition model structure
(create-model COMPOSITION_MODEL)
(update-model COMPOSITION_MODEL "composition")
(add-attribute COMPOSITION_MODEL "name" "LADiEUseCase")

(update-model COMPOSITION_MODEL "sequence")
  (setq webQuestGoal (generateProblem "(and (search web)(bookmarks))"))
  (addService "webQuestService" webQuestGoal "Step 1")
  (update-model COMPOSITION_MODEL "parallel")
    (setq interviewGoal (generateProblem "(interview)"))
    (addService "interviewService" interviewGoal "Step 2")
    (setq worksheetGoal (generateProblem "(worksheet)"))
    (addService "worksheetService" worksheetGoal "Step 3")
  (cd COMPOSITION_MODEL "..")
  (setq discussionGoal (generateProblem "(discussion asynchronous)"))
  (addService "discussionService" discussionGoal "Step 4")
  (setq peerReviewGoal (generateProblem "(peerreview)"))
  (addService "peerReviewService" peerReviewGoal "Step 5")
  (setq quizGoal (generateProblem "(quiz SQL)"))
  (addService "quizService" quizGoal "Step 6")
  (setq reportGoal (generateProblem "(created document)"))
  (addService "reportService" reportGoal "Step 7")
(cd COMPOSITION_MODEL "..")
```

B.3 PDDL Domain Definition

```
(define (domain test)
  (:requirements :typing)
  (:types username password quizTopic searchType discussionType docType language)

  (:predicates (isliterate)
      (quiz ?SQL - quizTopic)
      (wiki)
      (webaccess)
      (search ?video - searchType)
      (bookmarks)
      (send-message)
      (discussion ?asynchronous - discussionType)
      (worksheet)
      (mindMap)
      (conceptMap)
      (uploadFile)
      (hasGoogleAccount)
      (created ?document - docType)
      (notes-taken)
      (interview)
      (interpreter ?python - language)
  )

  (:action SIMPLEQUIZ
    :parameters ( ?id - username ?passwd - password ?SQL - quizTopic)
      :precondition (and (isliterate))
    :effect (and (quiz ?SQL))
  )

  (:action WIKI
    :parameters ( ?id - username ?passwd - password)
    :effect (and (wiki))
  )

  (:action YOUTUBESEARCH
    :parameters ( ?id - username ?passwd - password ?video - searchType)
      :precondition (and (webaccess))
    :effect (and (search ?video))
  )

  (:action SIMPLEBOOKMARKING
    :parameters ( ?id - username ?passwd - password)
    :effect (and (bookmarks))
  )
```

```
    (:action MYEMAILCLIENT
            :parameters ( ?id - username ?passwd - password ?asynchronous - discussion
            :effect (and (send-message) (discussion ?asynchronous))
    )

    (:action WORKSHEET
            :parameters ( ?id - username ?passwd - password)
        :precondition (and (isliterate))
            :effect (and (worksheet))
    )

    (:action AUDIOCONF
            :parameters ( ?id - username ?passwd - password ?synchronous - discussionT
        :precondition (and (not (isliterate)))
            :effect (and (discussion ?synchronous))
    )

    (:action AUDIOQUIZ
            :parameters ( ?id - username ?passwd - password ?SQL - quizTopic)
        :precondition (and
         (not (isliterate)))
            :effect (and (quiz ?SQL))
    )

    (:action AUDIOWORKSHEET
            :parameters ( ?id - username ?passwd - password)
        :precondition (and
         (not (isliterate)))
            :effect (and (worksheet))
    )

    (:action BUBBLUS
            :parameters ( ?id - username ?passwd - password)
        :precondition (and (webaccess))
            :effect (and (mindMap) (conceptMap))
    )

    (:action FTPCLIENT
            :parameters ( ?id - username ?passwd - password)
            :effect (and (uploadFile))
    )

    (:action SIMPLEFORUM
            :parameters ( ?id - username ?passwd - password ?asynchronous - discussion
        :precondition (and (isliterate))
            :effect (and (discussion ?asynchronous))
    )
```

```
        (:action GMAILCLIENT
                :parameters ( ?id - username ?passwd - password ?asynchronous - discussionType)
            :precondition (and (hasGoogleAccount))
                :effect (and (send-message) (discussion ?asynchronous))
        )

        (:action GOOGLEDOCUMENTS
                :parameters ( ?id - username ?passwd - password ?document - docType)
            :precondition (and (hasGoogleAccount)(webaccess))
                :effect (and (created ?document))
        )

        (:action GOOGLENOTEBOOK
                :parameters ( ?id - username ?passwd - password)
            :precondition (and (webaccess)(hasGoogleAccount))
                :effect (and (notes-taken) (bookmarks))
        )

        (:action GOOGLEPRESENTATION
                :parameters ( ?id - username ?passwd - password ?presentation - docType)
            :precondition (and (hasGoogleAccount)(webaccess))
                :effect (and (created ?presentation))
        )

        (:action GOOGLESEARCH
                :parameters ( ?id - username ?passwd - password ?web - searchType)
            :precondition (and (webaccess))
                :effect (and (search ?web))
        )

        (:action SIMPLEIRC
                :parameters ( ?id - username ?passwd - password ?synchronous - discussionType)
                :effect (and (discussion ?synchronous))
        )

        (:action INTERVIEW
                :parameters ( ?id - username ?passwd - password)
                :effect (and (interview))
        )

        (:action PYTHONINTERPRETER
                :parameters ( ?id - username ?passwd - password ?python - language)
            :precondition (and (webaccess))
                :effect (and (interpreter ?python))
        )
)
```

Appendix C

Evaluation Experiment Three

C.1 SQL Course Artifacts

C.1.1 Narrative Model

```
<narrative>
    <id>sql narrative</id>
    <type>JESS</type>
    <code>
(deffunction get-multifield-value (?field)
    (bind ?var (implode$ ?field))
    (return (sub-string 2 (- (str-length ?var) 1) ?var))
)

(deffunction candidate-selector (?ccg)
    ;; dumb candidate selector, knows that there is only one candidate so it
    ;; simple retrieves it and returns it to the caller
    (load-model "candidateGroupCollection" ?ccg "ccg")
    (bind ?result (get-multifield-value (search-model "ccg" "pagelet")))
    (remove-model "ccg")
    (return ?result)
)

(deffunction add-subsection (?sectionName ?requirement ?ccg)
    (if (eq (length$ (intersection$ (search-model "learnerModel" "langstring") (create$ ?requirement))) 1)
    then
    ;; Select the correct lo to include in the course
    (bind ?lo (candidate-selector ?ccg))

    ;; load the lo from a repository
    (load-model "learningObjectCollection" ?lo "lo")

    (bind ?list (search-model-for-attributes "lo" "file" "href"))
    (remove-model "lo")

    (update-model ?modelName "subsection")
    (update-model ?modelName "name" ?sectionName)
    (update-model ?modelName "id" ?sectionName)

    (foreach ?var ?list
        (update-model ?modelName "card")
        (update-model ?modelName "cg" ?var)
        (cd ?modelName "..")
    )

    (cd ?modelName "..")

    else
    (return 1)
    )
)

(bind ?modelName (get-multifield-value (search-model "learnerModel" "identifier")))
(create-model ?modelName)

(update-model ?modelName  "course")
;;(cd ?modelName "course")

(update-model ?modelName "name" "SQL Course")

(update-model ?modelName "section")
;;(cd ?modelName "section")
(update-model ?modelName "name" "Database Concepts")

(add-subsection "Introduction" "db.concepts.introduction" "cg01")

(add-subsection "The Relational Model" "db.concepts.relational_model" "cg02")

(add-subsection "Relational DBMS Architecture" "db.concepts.rdbms" "cg03")
```

```
(add-subsection "Structured Query Language" "db.concepts.sql.syntax" "cg04")

(cd ?modelName "..")
(update-model ?modelName "section")
;;(cd ?modelName "section" "1")
(update-model ?modelName "name" "Creating a Database")

(add-subsection "Data Types" "db.tables.data_types" "cg05")

(add-subsection "DDL : Create Statement" "db.tables.create" "cg06")

(add-subsection "Table Constraints" "db.tables.constraints" "cg07")

(add-subsection "DDL : Alter and Drop Statements" "db.tables.modify" "cg08")

(add-subsection "Introduction to Views" "db.views" "cg09")

(add-subsection "Create View Statement" "db.createview" "cg10")

(cd ?modelName "..")
(update-model ?modelName "section")
;;(cd ?modelName "section" "2")
(update-model ?modelName "name" "Populating a Database")

(add-subsection "Introduction to Data Manipulation Language" "db.tables.populate" "cg11")

(add-subsection "Insert Statement" "db.tables.populate.insert" "cg12")

(add-subsection "Update Statement" "db.tables.populate.update" "cg13")

(add-subsection "Delete Statement" "db.tables.populate.delete" "cg14")

(cd ?modelName "..")
(update-model ?modelName "section")
;;(cd ?modelName "section" "3")
(update-model ?modelName "name" "Database Retrieval")

(add-subsection "The Simple Select Statement" "db.retrieval.simple.select" "cg15")

(add-subsection "The Select Statement" "db.retrieval.select" "cg16")

(add-subsection "Expression Operators" "db.retrieval.select.operators" "cg17")

(add-subsection "The Where Clause" "db.retrieval.clauses.where" "cg18")

(add-subsection "Search Conditions" "db.retrieval.clauses.where.search_conditions" "cg19")

(add-subsection "An Introduction to Joins" "db.retrieval.joins" "cg20")

(add-subsection "An Introduction to Simple Joins" "db.retrieval.simple.joins" "cg21")

(add-subsection "Aggregate Functions" "db.retrieval.aggregate_functions" "cg22")

(cd ?modelName "..")
(update-model ?modelName "section")
;;(cd ?modelName "section" "4")
(update-model ?modelName "name" "Database Applications")

(add-subsection "Writing Database Applications using Embedded SQL" "db.applications.embedded_sql" "

(add-subsection "Structure of Embedded SQL Applications" "db.applications.embedded_sql.structure" "

(store-model "coursesCollection" ?modelName ?modelName)</code>
</narrative>
```

Figure C.1: Narrative Model for Personalised SQL Course used in Experiment One

C.1.2 Learner Model

```
<learner>
    <general>
        <identifier>ian</identifier>
    <name>
            <surname>O'Keeffe</surname>
      <forename>Ian</forename>
    </name>
  </general>
  <educational>
        <adaptivity>
            <adaptivitytype name="competencies.required">
        <set type="ALL">
          <candidate>
                        <langstring lang="en">db.concepts.introduction</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.concepts.relational_model</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.concepts.rdbms</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.concepts.sql.syntax</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.tables.constraints</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.tables.data_types</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.tables.create</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.tables.modify</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.views</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.createview</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.tables.populate</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.tables.populate.insert</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.tables.populate.update</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.tables.populate.delete</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.retrieval.simple.select</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.retrieval.select</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.retrieval.select.operators</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.retrieval.clauses.where</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.retrieval.clauses.where.search_conditions</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.retrieval.joins</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.retrieval.simple.joins</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.retrieval.aggregate_functions</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.applications.embedded_sql</langstring>
          </candidate>
          <candidate>
                        <langstring lang="en">db.applications.embedded_sql.structure</langstring>
          </candidate>
        </set>
      </adaptivitytype>
    </adaptivity>
  </educational>
</learner>
```

Figure C.2: Example Learner Model for Personalised SQL Course used in Experiment One

C.1.3 Content Model

```
<pagelet>
    <general>
        <identifier>mod011-002</identifier>
        <title>Definitions</title>
        <keyword>definitions</keyword>
        <keyword>database</keyword>
        <keyword>database management system</keyword>
        <keyword>DBMS</keyword>
        <description>Definitions for database and database management system</description>
        <language>en</language>
    </general>
    <pedagogical>
        <objectivestaught>
            <objective>db.concepts.introduction</objective>
        </objectivestaught>
        <supportedlearningstyle/>
        <semanticdensity/>
        <displayarea/>
    </pedagogical>
    <technical>
        <location>mod011-002.html</location>
        <format/>
        <requirements>none</requirements>
        <size/>
    </technical>
</pagelet>
```

Figure C.3: Example content metadata from the Contnet Model for Personalised SQL Course used in Experiment One

C.1.4 Screenshots

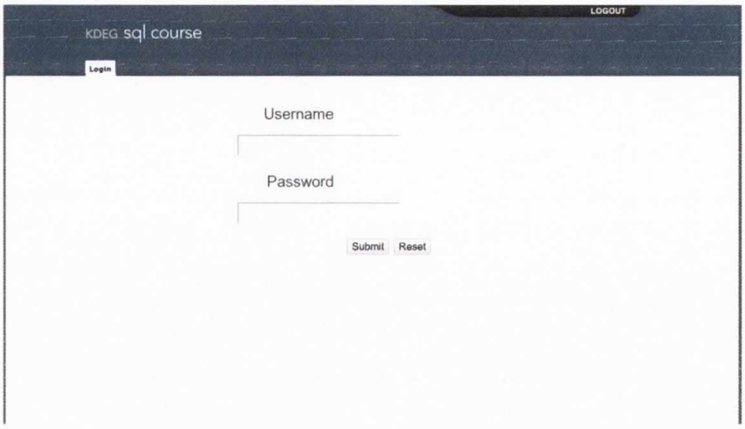
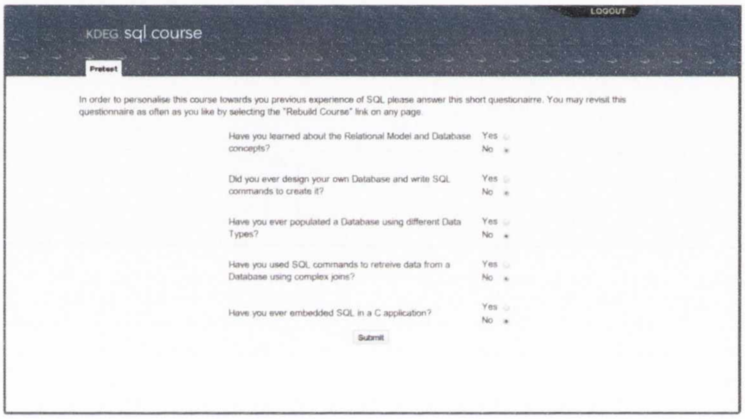


Figure C.4: Screenshot of the Adaptive SQL Course login page



In order to personalise this course towards you previous experience of SQL please answer this short questionairre. You may revisit this questionnaire as often as you like by selecting the "Rebuild Course" link on any page.

Have you learned about the Relational Model and Database concepts? Yes ○ No ●

Did you ever design your own Database and write SQL commands to create it? Yes ○ No ●

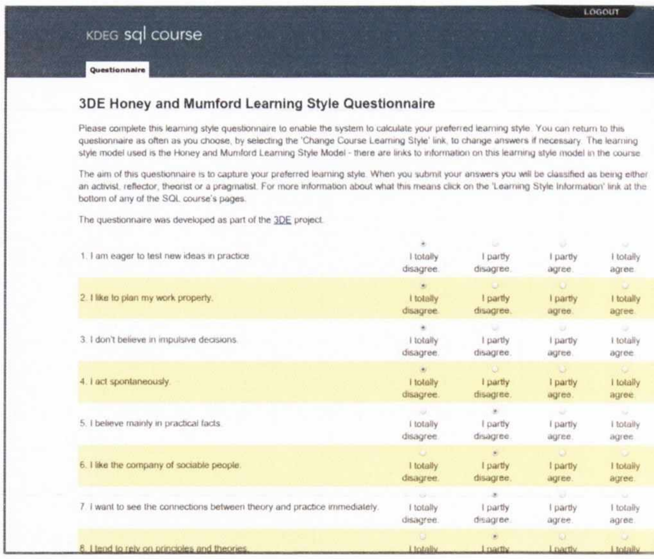
Have you ever populated a Database using different Data Types? Yes ○ No ●

Have you used SQL commands to retreive data from a Database using complex joins? Yes ○ No ●

Have you ever embedded SQL in a C application? Yes ○ No ●

Figure C.5: Screenshot of the Adaptive SQL Course scope questionnaire

Course Learning Style Questionnaire.png



3DE Honey and Mumford Learning Style Questionnaire

Please complete this learning style questionnaire to enable the system to calculate your preferred learning style. You can return to this questionnaire as often as you choose, by selecting the 'Change Course Learning Style' link, to change answers if necessary. The learning style model used is the Honey and Mumford Learning Style Model - there are links to information on this learning style model in the course

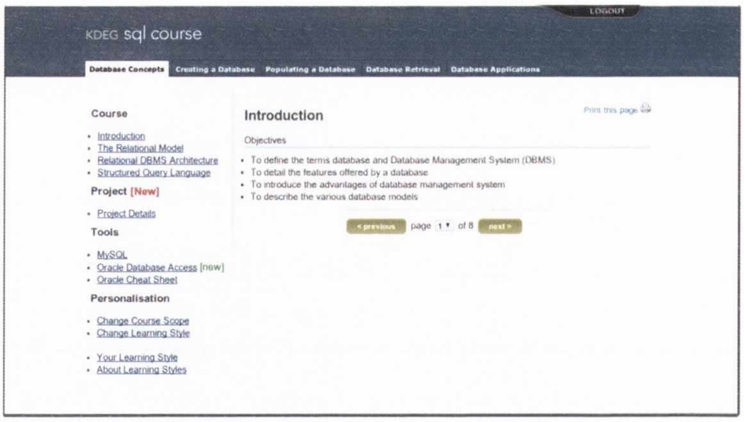
The aim of this questionnaire is to capture your preferred learning style. When you submit your answers you will be classified as being either an activist, reflector, theorist or a pragmatist. For more information about what this means click on the 'Learning Style Information' link at the bottom of any of the SQL course's pages.

The questionnaire was developed as part of the 3DE project.

1. I am eager to test new ideas in practice.	● I totally disagree.	○ I partly disagree.	○ I partly agree.	○ I totally agree.
2. I like to plan my work properly.	● I totally disagree.	○ I partly disagree.	○ I partly agree.	○ I totally agree.
3. I don't believe in impulsive decisions.	● I totally disagree.	○ I partly disagree.	○ I partly agree.	○ I totally agree.
4. I act spontaneously.	● I totally disagree.	○ I partly disagree.	○ I partly agree.	○ I totally agree.
5. I believe mainly in practical facts.	○ I totally disagree.	● I partly disagree.	○ I partly agree.	○ I totally agree.
6. I like the company of sociable people.	○ I totally disagree.	● I partly disagree.	○ I partly agree.	○ I totally agree.
7. I want to see the connections between theory and practice immediately.	○ I totally disagree.	● I partly disagree.	○ I partly agree.	○ I totally agree.
8. I tend to rely on principles and theories.	○ I totally	● I partly	○ I partly	○ I totally

Figure C.6: Screenshot of the Adaptive SQL Course learning style questionnaire

Course content page.png



Course

- Introduction
- The Relational Model
- Relational DBMS Architecture
- Structured Query Language

Project [New]

- Project Details

Tools

- MySQL
- Oracle Database Access [new]
- Oracle Cheat Sheet

Personalisation

- Change Course Scope
- Change Learning Style

- Your Learning Style
- About Learning Styles

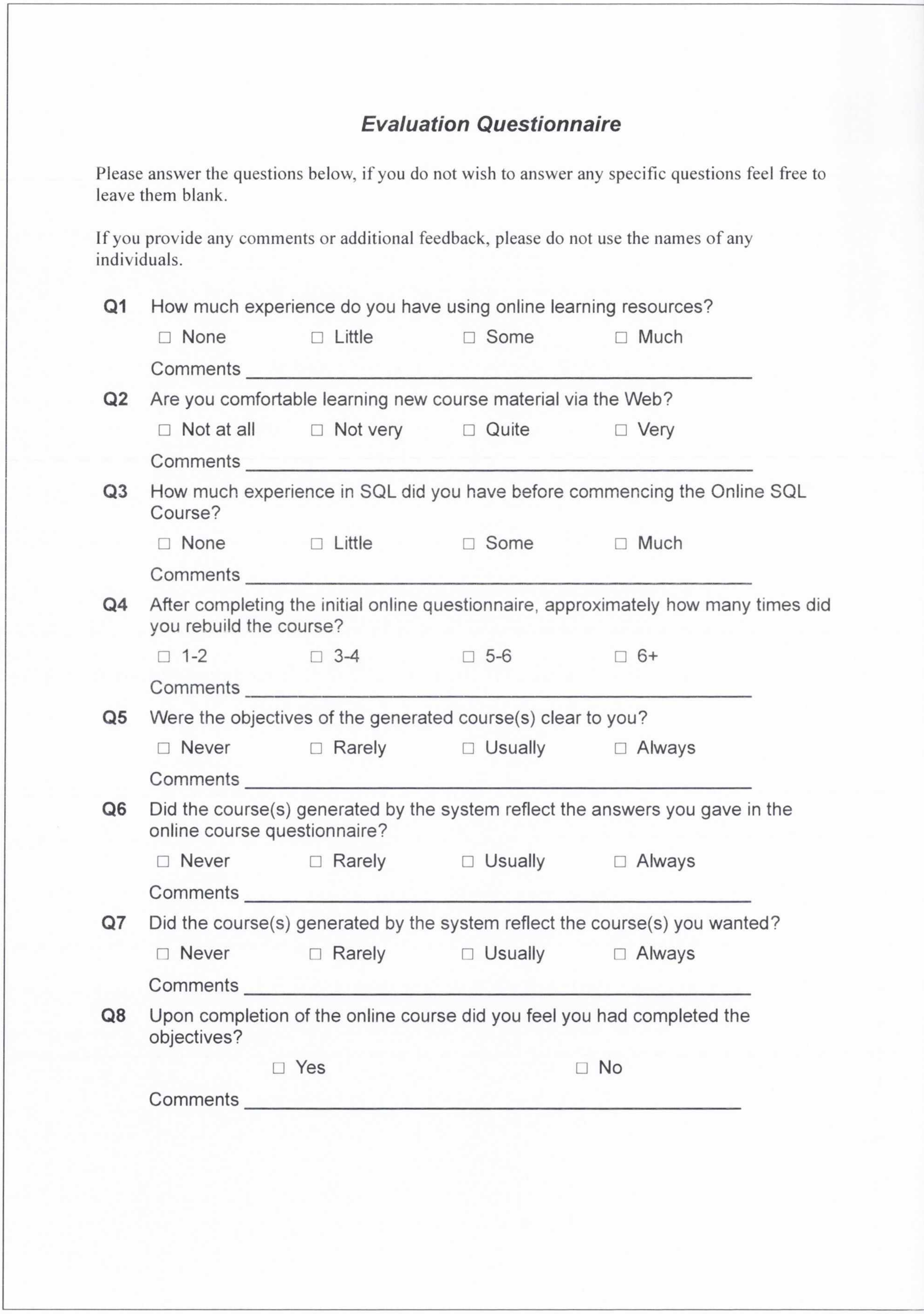
Introduction

Objectives

- To define the terms database and Database Management System (DBMS)
- To detail the features offered by a database
- To introduce the advantages of database management system
- To describe the various database models

Figure C.7: Screenshot of an example content page from the Adaptive SQL Course

C.2 Evaluation Questionnaires



Evaluation Questionnaire

Please answer the questions below, if you do not wish to answer any specific questions feel free to leave them blank.

If you provide any comments or additional feedback, please do not use the names of any individuals.

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

_ 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

_ Populating a Database

_ Database Applications

_ Creating a Database

_ Database Retrieval

Comments

____

Evaluation Questionnaire

Please answer the questions below, if you do not wish to answer any specific questions feel free to leave them blank.

If you provide any comments or additional feedback, please do not use the names of any individuals.

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

_ 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

_ Populating a Database

_ Database Applications

_ Creating a Database

_ Database Retrieval

Comments

____

Q18 Are there any additional features you would like to see in the course?

□ Yes □ No

Comments

____

Q19 Did you experience any technical difficulties with the course?

□ Yes □ No

Comments

____

Q20 Please use the space below to add any additional comments.

____
____
____
____
____
____
____
____
____
____
____
____

Thank you for answering the questionnaire.

System Usability Scale

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	□ 1	□ 2	□ 3	□ 4	□ 5
2. I found the system unnecessarily complex	□ 1	□ 2	□ 3	□ 4	□ 5
3. I thought the system was easy to use	□ 1	□ 2	□ 3	□ 4	□ 5
4. I think that I would need the support of a technical person to be able to use this system	□ 1	□ 2	□ 3	□ 4	□ 5
5. I found the various functions in this system were well integrated	□ 1	□ 2	□ 3	□ 4	□ 5
6. I thought there was too much inconsistency in this system	□ 1	□ 2	□ 3	□ 4	□ 5
7. I would imagine that most people would learn to use this system very quickly	□ 1	□ 2	□ 3	□ 4	□ 5
8. I found the system very cumbersome to use	□ 1	□ 2	□ 3	□ 4	□ 5
9. I felt very confident using the system	□ 1	□ 2	□ 3	□ 4	□ 5
10. I needed to learn a lot of things before I could get going with this system	□ 1	□ 2	□ 3	□ 4	□ 5

C.3 Experimental Data

C.3.1 Q17

Course Section	Rating Scores				
	1	2	3	4	5
Database Concepts	0	1	10	10	5
Creating a Database	2	3	8	7	6
Populating a Database	2	3	7	8	6
Database Retrieval	4	1	6	8	7
Database Applications	2	7	6	6	5

Table C.1: Effectiveness of Subject Matter Representation Results (Q17)

C.4 Correlation Data

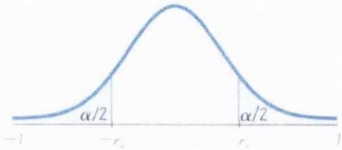
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16
Q1																
Q2	0.52															
Q3	0.13	0.27														
Q4	0.29	0.06	0.06													
Q5	-0.17	0.02	0.10	0.12												
Q6	0.23	0.41	0.31	-0.11	0.32											
Q7	-0.19	0.16	-0.48	-0.15	0.25	0.33										
Q8	-0.09	0.22	0.20	0.11	0.50	0.35	0.25									
Q9																
Q10	0.15	0.23	-0.05	0.19	0.47	0.08	0.02	0.31								
Q11	0.18	-0.15	-0.07	0.24	-0.48	-0.16	-0.44	-0.06		-0.26						
Q12	0.10	-0.08	-0.18	0.22	-0.31	-0.32	-0.22	-0.73		-0.46	0.12					
Q13	0.28	0.28	0.46	-0.02	0.38	0.38	0.31	-0.06		0.19	-0.52	-0.02				
Q14	0.01	0.09	0.20	0.03	0.42	0.42	0.18	0.22		0.38	-0.24	-0.38	0.22			
Q15	-0.10	0.05	0.11	0.26	-0.24	-0.24	0.10	-0.26		-0.29	-0.09	0.62	-0.06	-0.27		
Q16	0.25	-0.07	-0.07	-0.18	0.17	0.17	-0.06	0.06		-0.20	-0.21	0.15	0.19	-0.03	0.10	

Table C.2: Correlation Data

C.4.1 Spearman Correlation Table

Critical Values for the Spearman Rank Correlation Test

n = number of pairs of sample data
Alpha values (α) listed are for 2-tails



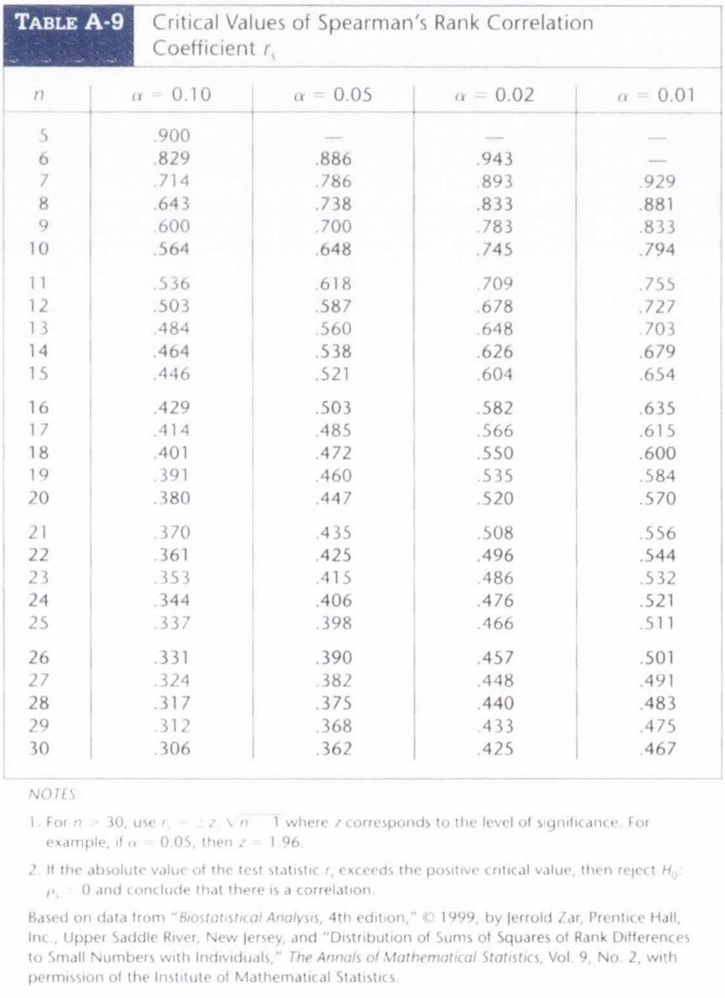


TABLE A-9 Critical Values of Spearman's Rank Correlation Coefficient r_s

n	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.02$	$\alpha = 0.01$
5	.900	—	—	—
6	.829	.886	.943	—
7	.714	.786	.893	.929
8	.643	.738	.833	.881
9	.600	.700	.783	.833
10	.564	.648	.745	.794
11	.536	.618	.709	.755
12	.503	.587	.678	.727
13	.484	.560	.648	.703
14	.464	.538	.626	.679
15	.446	.521	.604	.654
16	.429	.503	.582	.635
17	.414	.485	.566	.615
18	.401	.472	.550	.600
19	.391	.460	.535	.584
20	.380	.447	.520	.570
21	.370	.435	.508	.556
22	.361	.425	.496	.544
23	.353	.415	.486	.532
24	.344	.406	.476	.521
25	.337	.398	.466	.511
26	.331	.390	.457	.501
27	.324	.382	.448	.491
28	.317	.375	.440	.483
29	.312	.368	.433	.475
30	.306	.362	.425	.467

NOTES:

1. For $n > 30$, use $r_s = \pm z / \sqrt{n-1}$ where z corresponds to the level of significance. For example, if $\alpha = 0.05$, then $z = 1.96$.
2. If the absolute value of the test statistic r_s exceeds the positive critical value, then reject H_0: $\rho_s = 0$ and conclude that there is a correlation.

Based on data from "*Biostatistical Analysis*, 4th edition," © 1999, by Jerrold Zar, Prentice Hall, Inc., Upper Saddle River, New Jersey, and "Distribution of Sums of Squares of Rank Differences to Small Numbers with Individuals," *The Annals of Mathematical Statistics*, Vol. 9, No. 2, with permission of the Institute of Mathematical Statistics.

C.4.2 Table of Calculated Values

Appendix D

Evaluation Experiment Four

D.1 JMeter Test Plan

```
<?xml version="1.0" encoding="UTF-8"?>
<jmeterTestPlan version="1.2" properties="2.1">
  <hashTree>
    <TestPlan guiclass="TestPlanGui" testclass="TestPlan" testname="Test Plan" enabled="true">
      <stringProp name="TestPlan.comments"></stringProp>
      <boolProp name="TestPlan.functional_mode">false</boolProp>
      <boolProp name="TestPlan.serialize_threadgroups">false</boolProp>
      <elementProp name="TestPlan.user_defined_variables" elementType="Arguments" guiclass="ArgumentsPanel" testclass="Argu
        <collectionProp name="Arguments.arguments"/>
      </elementProp>
      <stringProp name="TestPlan.user_define_classpath"></stringProp>
    </TestPlan>
    <hashTree>
      <ThreadGroup guiclass="ThreadGroupGui" testclass="ThreadGroup" testname="Thread Group" enabled="true">
        <elementProp name="ThreadGroup.main_controller" elementType="LoopController" guiclass="LoopControlPanel" testclass=
          <boolProp name="LoopController.continue_forever">false</boolProp>
          <stringProp name="LoopController.loops">1</stringProp>
        </elementProp>
        <stringProp name="ThreadGroup.num_threads">10</stringProp>
        <stringProp name="ThreadGroup.ramp_time">1</stringProp>
        <longProp name="ThreadGroup.start_time">1277719589000</longProp>
        <longProp name="ThreadGroup.end_time">1277719589000</longProp>
        <boolProp name="ThreadGroup.scheduler">false</boolProp>
        <stringProp name="ThreadGroup.on_sample_error">continue</stringProp>
        <stringProp name="ThreadGroup.duration"></stringProp>
        <stringProp name="ThreadGroup.delay"></stringProp>
      </ThreadGroup>
      <hashTree>
        <CounterConfig guiclass="CounterConfigGui" testclass="CounterConfig" testname="UserID" enabled="true">
          <stringProp name="CounterConfig.start">1</stringProp>
          <stringProp name="CounterConfig.end"></stringProp>
          <stringProp name="CounterConfig.incr">1</stringProp>
          <stringProp name="CounterConfig.name">UserID</stringProp>
          <stringProp name="CounterConfig.format"></stringProp>
          <boolProp name="CounterConfig.per_user">false</boolProp>
        </CounterConfig>
        <hashTree/>
        <HTTPSampler guiclass="HttpTestSampleGui" testclass="HTTPSampler" testname="/LADiE" enabled="false">
          <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Argume
            <collectionProp name="Arguments.arguments"/>
          </elementProp>
          <stringProp name="HTTPSampler.domain">kdeg-vm-11.cs.tcd.ie</stringProp>
          <stringProp name="HTTPSampler.port">8080</stringProp>
          <stringProp name="HTTPSampler.connect_timeout"></stringProp>
          <stringProp name="HTTPSampler.response_timeout"></stringProp>
          <stringProp name="HTTPSampler.protocol">http</stringProp>
          <stringProp name="HTTPSampler.contentEncoding"></stringProp>
          <stringProp name="HTTPSampler.path">/LADiE</stringProp>
          <stringProp name="HTTPSampler.method">GET</stringProp>
          <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
          <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
          <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
          <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
          <stringProp name="HTTPSampler.FILE_NAME"></stringProp>
          <stringProp name="HTTPSampler.FILE_FIELD"></stringProp>
          <stringProp name="HTTPSampler.mimetype"></stringProp>
          <boolProp name="HTTPSampler.monitor">false</boolProp>
          <stringProp name="HTTPSampler.embedded_url_re"></stringProp>
        </HTTPSampler>
        <hashTree>
          <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header Manager" enabled="false">
            <collectionProp name="HeaderManager.headers">
              <elementProp name="Accept-Language" elementType="Header">
                <stringProp name="Header.name">Accept-Language</stringProp>
                <stringProp name="Header.value">en-us,en;q=0.5</stringProp>
              </elementProp>
      <elementProp name="Accept" elementType="Header">
        <stringProp name="Header.name">Accept</stringProp>
        <stringProp name="Header.value">text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8</stringProp>
      </elementProp>
      <elementProp name="Keep-Alive" elementType="Header">
        <stringProp name="Header.name">Keep-Alive</stringProp>
        <stringProp name="Header.value">115</stringProp>
      </elementProp>
      <elementProp name="User-Agent" elementType="Header">
        <stringProp name="Header.name">User-Agent</stringProp>
        <stringProp name="Header.value">Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.3) Gecko/20100423 Ubuntu/10.04 (luc
      </elementProp>
      <elementProp name="Referer" elementType="Header">
        <stringProp name="Header.name">Referer</stringProp>
        <stringProp name="Header.value">http://kdeg-vm-11.cs.tcd.ie:8080/LADiE/activitySelector.jsp</stringProp>
      </elementProp>
      <elementProp name="Accept-Encoding" elementType="Header">
        <stringProp name="Header.name">Accept-Encoding</stringProp>
        <stringProp name="Header.value">gzip,deflate</stringProp>
      </elementProp>
      <elementProp name="Accept-Charset" elementType="Header">
        <stringProp name="Header.name">Accept-Charset</stringProp>
        <stringProp name="Header.value">ISO-8859-1,utf-8;q=0.7,*;q=0.7</stringProp>
      </elementProp>
    </collectionProp>
  </HeaderManager>
  <hashTree/>
</hashTree>
<HTTPSampler guiclass="HttpTestSampleGui" testclass="HTTPSampler" testname="/LADiE/" enabled="true">
  <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" enable
    <collectionProp name="Arguments.arguments"/>
  </elementProp>
  <stringProp name="HTTPSampler.domain">kdeg-vm-11.cs.tcd.ie</stringProp>
  <stringProp name="HTTPSampler.port">8080</stringProp>
  <stringProp name="HTTPSampler.connect_timeout"></stringProp>
  <stringProp name="HTTPSampler.response_timeout"></stringProp>
  <stringProp name="HTTPSampler.protocol">http</stringProp>
  <stringProp name="HTTPSampler.contentEncoding">UTF-8</stringProp>
  <stringProp name="HTTPSampler.path">/LADiE/</stringProp>
  <stringProp name="HTTPSampler.method">GET</stringProp>
  <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
  <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
  <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
  <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
  <stringProp name="HTTPSampler.FILE_NAME"></stringProp>
  <stringProp name="HTTPSampler.FILE_FIELD"></stringProp>
  <stringProp name="HTTPSampler.mimetype"></stringProp>
  <boolProp name="HTTPSampler.monitor">false</boolProp>
  <stringProp name="HTTPSampler.embedded_url_re"></stringProp>
</HTTPSampler>
<hashTree>
  <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header Manager" enabled="true">
    <collectionProp name="HeaderManager.headers">
      <elementProp name="Accept-Language" elementType="Header">
        <stringProp name="Header.name">Accept-Language</stringProp>
        <stringProp name="Header.value">en-us,en;q=0.5</stringProp>
      </elementProp>
      <elementProp name="Accept" elementType="Header">
        <stringProp name="Header.name">Accept</stringProp>
        <stringProp name="Header.value">text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8</stringProp>
      </elementProp>
      <elementProp name="Keep-Alive" elementType="Header">
        <stringProp name="Header.name">Keep-Alive</stringProp>
        <stringProp name="Header.value">115</stringProp>
      </elementProp>
```

```
              <elementProp name="User-Agent" elementType="Header">
                <stringProp name="Header.name">User-Agent</stringProp>
                <stringProp name="Header.value">Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.3) Gecko/20100423 Ubuntu
              </elementProp>
              <elementProp name="Referer" elementType="Header">
                <stringProp name="Header.name">Referer</stringProp>
                <stringProp name="Header.value">http://kdeg-vm-11.cs.tcd.ie:8080/LADiE/activitySelector.jsp</stringProp>
              </elementProp>
              <elementProp name="Accept-Encoding" elementType="Header">
                <stringProp name="Header.name">Accept-Encoding</stringProp>
                <stringProp name="Header.value">gzip,deflate</stringProp>
              </elementProp>
              <elementProp name="Accept-Charset" elementType="Header">
                <stringProp name="Header.name">Accept-Charset</stringProp>
                <stringProp name="Header.value">ISO-8859-1,utf-8;q=0.7,*;q=0.7</stringProp>
              </elementProp>
            </collectionProp>
          </HeaderManager>
          <hashTree/>
        </hashTree>
        <HTTPSampler guiclass="HttpTestSampleGui" testclass="HTTPSampler" testname="/LADiE/j_security_check" enabled="true"
          <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Argume
            <collectionProp name="Arguments.arguments">
              <elementProp name="j_username" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.name">j_username</stringProp>
                <stringProp name="Argument.value">ladie_user_$UserID</stringProp>
                <stringProp name="Argument.metadata">=</stringProp>
                <boolProp name="HTTPArgument.use_equals">true</boolProp>
              </elementProp>
              <elementProp name="j_password" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.name">j_password</stringProp>
                <stringProp name="Argument.value">ladie_user_$UserID</stringProp>
                <stringProp name="Argument.metadata">=</stringProp>
                <boolProp name="HTTPArgument.use_equals">true</boolProp>
              </elementProp>
              <elementProp name="login" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.name">login</stringProp>
                <stringProp name="Argument.value">Login</stringProp>
                <stringProp name="Argument.metadata">=</stringProp>
                <boolProp name="HTTPArgument.use_equals">true</boolProp>
              </elementProp>
            </collectionProp>
          </elementProp>
          <stringProp name="HTTPSampler.domain">kdeg-vm-11.cs.tcd.ie</stringProp>
          <stringProp name="HTTPSampler.port">8080</stringProp>
          <stringProp name="HTTPSampler.connect_timeout"></stringProp>
          <stringProp name="HTTPSampler.response_timeout"></stringProp>
          <stringProp name="HTTPSampler.protocol">http</stringProp>
          <stringProp name="HTTPSampler.contentEncoding"></stringProp>
          <stringProp name="HTTPSampler.path">/LADiE/j_security_check</stringProp>
          <stringProp name="HTTPSampler.method">POST</stringProp>
          <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
          <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
          <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
          <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
          <stringProp name="HTTPSampler.FILE_NAME"></stringProp>
          <stringProp name="HTTPSampler.FILE_FIELD"></stringProp>
          <stringProp name="HTTPSampler.mimetype"></stringProp>
          <boolProp name="HTTPSampler.monitor">false</boolProp>
          <stringProp name="HTTPSampler.embedded_url_re"></stringProp>
        </HTTPSampler>
        <hashTree>
          <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header Manager" enabled="true">
            <collectionProp name="HeaderManager.headers">
              <elementProp name="Content-Type" elementType="Header">
                <stringProp name="Header.name">Content-Type</stringProp>
                <stringProp name="Header.value">application/x-www-form-urlencoded</stringProp>
              </elementProp>
              <elementProp name="Accept-Language" elementType="Header">
                <stringProp name="Header.name">Accept-Language</stringProp>
                <stringProp name="Header.value">en-us,en;q=0.5</stringProp>
              </elementProp>
              <elementProp name="Accept" elementType="Header">
                <stringProp name="Header.name">Accept</stringProp>
                <stringProp name="Header.value">text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8</stringProp>
              </elementProp>
              <elementProp name="Keep-Alive" elementType="Header">
                <stringProp name="Header.name">Keep-Alive</stringProp>
                <stringProp name="Header.value">115</stringProp>
              </elementProp>
              <elementProp name="User-Agent" elementType="Header">
                <stringProp name="Header.name">User-Agent</stringProp>
                <stringProp name="Header.value">Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.3) Gecko/20100423 Ubuntu/10.04 (luc
              </elementProp>
              <elementProp name="Referer" elementType="Header">
                <stringProp name="Header.name">Referer</stringProp>
                <stringProp name="Header.value">http://kdeg-vm-11.cs.tcd.ie:8080/LADiE/</stringProp>
              </elementProp>
              <elementProp name="Accept-Encoding" elementType="Header">
                <stringProp name="Header.name">Accept-Encoding</stringProp>
                <stringProp name="Header.value">gzip,deflate</stringProp>
              </elementProp>
              <elementProp name="Accept-Charset" elementType="Header">
                <stringProp name="Header.name">Accept-Charset</stringProp>
                <stringProp name="Header.value">ISO-8859-1,utf-8;q=0.7,*;q=0.7</stringProp>
              </elementProp>
            </collectionProp>
          </HeaderManager>
          <hashTree/>
        </hashTree>
        <HTTPSampler guiclass="HttpTestSampleGui" testclass="HTTPSampler" testname="/LADiE/activitySelectionProcessor.jsp" enabled="tr
          <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" enable
            <collectionProp name="Arguments.arguments">
              <elementProp name="activity" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.name">activity</stringProp>
                <stringProp name="Argument.value">1</stringProp>
                <stringProp name="Argument.metadata">=</stringProp>
                <boolProp name="HTTPArgument.use_equals">true</boolProp>
              </elementProp>
            </collectionProp>
          </elementProp>
          <stringProp name="HTTPSampler.domain">kdeg-vm-11.cs.tcd.ie</stringProp>
          <stringProp name="HTTPSampler.port">8080</stringProp>
          <stringProp name="HTTPSampler.connect_timeout"></stringProp>
          <stringProp name="HTTPSampler.response_timeout"></stringProp>
          <stringProp name="HTTPSampler.protocol">http</stringProp>
          <stringProp name="HTTPSampler.contentEncoding"></stringProp>
          <stringProp name="HTTPSampler.path">/LADiE/activitySelectionProcessor.jsp</stringProp>
          <stringProp name="HTTPSampler.method">POST</stringProp>
          <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
          <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
          <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
          <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
          <stringProp name="HTTPSampler.FILE_NAME"></stringProp>
          <stringProp name="HTTPSampler.FILE_FIELD"></stringProp>
          <stringProp name="HTTPSampler.mimetype"></stringProp>
          <boolProp name="HTTPSampler.monitor">false</boolProp>
          <stringProp name="HTTPSampler.embedded_url_re"></stringProp>
        </HTTPSampler>
```

```
<hashTree>
  <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header Manager" enabled="true">
    <collectionProp name="HeaderManager.headers">
      <elementProp name="Content-Type" elementType="Header">
        <stringProp name="Header.name">Content-Type</stringProp>
        <stringProp name="Header.value">application/x-www-form-urlencoded</stringProp>
      </elementProp>
      <elementProp name="Accept-Language" elementType="Header">
        <stringProp name="Header.name">Accept-Language</stringProp>
        <stringProp name="Header.value">en-us,en;q=0.5</stringProp>
      </elementProp>
      <elementProp name="Accept" elementType="Header">
        <stringProp name="Header.name">Accept</stringProp>
        <stringProp name="Header.value">text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8</stringProp
      </elementProp>
      <elementProp name="Keep-Alive" elementType="Header">
        <stringProp name="Header.name">Keep-Alive</stringProp>
        <stringProp name="Header.value">115</stringProp>
      </elementProp>
      <elementProp name="User-Agent" elementType="Header">
        <stringProp name="Header.name">User-Agent</stringProp>
        <stringProp name="Header.value">Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.3) Gecko/20100423 Ubuntu
      </elementProp>
      <elementProp name="Referer" elementType="Header">
        <stringProp name="Header.name">Referer</stringProp>
        <stringProp name="Header.value">http://kdeg-vm-11.cs.tcd.ie:8080/LADiE/activitySelector.jsp</stringProp>
      </elementProp>
      <elementProp name="Accept-Encoding" elementType="Header">
        <stringProp name="Header.name">Accept-Encoding</stringProp>
        <stringProp name="Header.value">gzip,deflate</stringProp>
      </elementProp>
      <elementProp name="Accept-Charset" elementType="Header">
        <stringProp name="Header.name">Accept-Charset</stringProp>
        <stringProp name="Header.value">ISO-8859-1,utf-8;q=0.7,*;q=0.7</stringProp>
      </elementProp>
    </collectionProp>
  </HeaderManager>
  <hashTree/>
</hashTree>
<HTTPSampler guiclass="HttpTestSampleGui" testclass="HTTPSampler" testname="/LADiE/build.jsp" enabled="true">
  <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Argume
    <collectionProp name="Arguments.arguments"/>
  </elementProp>
  <stringProp name="HTTPSampler.domain">kdeg-vm-11.cs.tcd.ie</stringProp>
  <stringProp name="HTTPSampler.port">8080</stringProp>
  <stringProp name="HTTPSampler.connect_timeout"></stringProp>
  <stringProp name="HTTPSampler.response_timeout"></stringProp>
  <stringProp name="HTTPSampler.protocol">http</stringProp>
  <stringProp name="HTTPSampler.contentEncoding"></stringProp>
  <stringProp name="HTTPSampler.path">/LADiE/build.jsp</stringProp>
  <stringProp name="HTTPSampler.method">GET</stringProp>
  <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
  <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
  <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
  <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
  <stringProp name="HTTPSampler.FILE_NAME"></stringProp>
  <stringProp name="HTTPSampler.FILE_FIELD"></stringProp>
  <stringProp name="HTTPSampler.mimetype"></stringProp>
  <boolProp name="HTTPSampler.monitor">false</boolProp>
  <stringProp name="HTTPSampler.embedded_url_re"></stringProp>
</HTTPSampler>
```

```
        <hashTree>
          <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header Manager" enabled="true">
            <collectionProp name="HeaderManager.headers">
              <elementProp name="Accept-Language" elementType="Header">
                <stringProp name="Header.name">Accept-Language</stringProp>
                <stringProp name="Header.value">en-us,en;q=0.5</stringProp>
              </elementProp>
              <elementProp name="Accept" elementType="Header">
                <stringProp name="Header.name">Accept</stringProp>
                <stringProp name="Header.value">text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8</stringProp>
              </elementProp>
              <elementProp name="Keep-Alive" elementType="Header">
                <stringProp name="Header.name">Keep-Alive</stringProp>
                <stringProp name="Header.value">115</stringProp>
              </elementProp>
              <elementProp name="User-Agent" elementType="Header">
                <stringProp name="Header.name">User-Agent</stringProp>
                <stringProp name="Header.value">Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.3) Gecko/20100423 Ubuntu/10.04 (luc
              </elementProp>
              <elementProp name="Accept-Encoding" elementType="Header">
                <stringProp name="Header.name">Accept-Encoding</stringProp>
                <stringProp name="Header.value">gzip,deflate</stringProp>
              </elementProp>
              <elementProp name="Accept-Charset" elementType="Header">
                <stringProp name="Header.name">Accept-Charset</stringProp>
                <stringProp name="Header.value">ISO-8859-1,utf-8;q=0.7,*;q=0.7</stringProp>
              </elementProp>
            </collectionProp>
          </HeaderManager>
          <hashTree/>
        </hashTree>
        <CookieManager guiclass="CookiePanel" testclass="CookieManager" testname="HTTP Cookie Manager" enabled="true">
          <collectionProp name="CookieManager.cookies"/>
          <boolProp name="CookieManager.clearEachIteration">true</boolProp>
        </CookieManager>
        <hashTree/>
        <ResultCollector guiclass="ViewResultsFullVisualizer" testclass="ResultCollector" testname="View Results Tree" enabled="true">
          <boolProp name="ResultCollector.error_logging">false</boolProp>
          <objProp>
            <name>saveConfig</name>
            <value class="SampleSaveConfiguration">
              <time>true</time>
              <latency>true</latency>
              <timestamp>true</timestamp>
              <success>true</success>
              <label>true</label>
              <code>true</code>
              <message>true</message>
              <threadName>true</threadName>
              <dataType>true</dataType>
              <encoding>false</encoding>
              <assertions>true</assertions>
              <subresults>true</subresults>
              <responseData>false</responseData>
              <samplerData>false</samplerData>
              <xml>true</xml>
              <fieldNames>false</fieldNames>
              <responseHeaders>false</responseHeaders>
              <requestHeaders>false</requestHeaders>
              <responseDataOnError>false</responseDataOnError>
              <saveAssertionResultsFailureMessage>false</saveAssertionResultsFailureMessage>
              <assertionsResultsToSave>0</assertionsResultsToSave>
              <bytes>true</bytes>
            </value>
          </objProp>
          <stringProp name="filename"></stringProp>
        </ResultCollector>
        <hashTree/>
        <ResultCollector guiclass="TableVisualizer" testclass="ResultCollector" testname="View Results in Table" enabled="t
          <boolProp name="ResultCollector.error_logging">false</boolProp>
          <objProp>
            <name>saveConfig</name>
            <value class="SampleSaveConfiguration">
              <time>true</time>
              <latency>true</latency>
              <timestamp>true</timestamp>
              <success>true</success>
              <label>true</label>
              <code>true</code>
              <message>true</message>
              <threadName>true</threadName>
              <dataType>true</dataType>
              <encoding>false</encoding>
              <assertions>true</assertions>
              <subresults>true</subresults>
              <responseData>false</responseData>
              <samplerData>false</samplerData>
              <xml>true</xml>
              <fieldNames>false</fieldNames>
              <responseHeaders>false</responseHeaders>
              <requestHeaders>false</requestHeaders>
              <responseDataOnError>false</responseDataOnError>
              <saveAssertionResultsFailureMessage>false</saveAssertionResultsFailureMessage>
              <assertionsResultsToSave>0</assertionsResultsToSave>
              <bytes>true</bytes>
            </value>
          </objProp>
          <stringProp name="filename">/home/okeeffei/Desktop/Test Results/AEBuild1.jmx</stringProp>
        </ResultCollector>
        <hashTree/>
      </hashTree>
    </hashTree>
  </hashTree>
</jmeterTestPlan>
```

D.2 Adaptation Process Benchmarking Raw Data

1	10	20	50		100			
3883	15269	28066	68543	87734	141598	178632	183873	187225
	15876	28817	69765	88028	144102	178696	184606	188063
	15437	29323	70850	88035	147084	179360	183299	186993
	15795	30950	71163	87703	150177	179575	185369	188375
	16692	30390	75202	89365	155282	179305	183769	187568
	17277	32211	75492	89810	156980	180327	185178	187725
	16963	32058	76438	89364	157408	180529	185611	188201
	17837	32394	76238	89551	172549	179731	186010	187304
	18147	32953	77128	89764	174634	180572	185412	188439
	18931	33186	78305	91115	174549	178873	185801	189388
		33458	79768	90649	175332	180467	186405	188612
		33869	80522	90660	176570	181527	185699	188786
		34098	81060	91663	176529	181766	186383	189642
		34195	81643	90943	177805	180805	186246	189236
		34161	82593	91573	176691	180778	185152	191448
		34896	82193	91532	177059	180776	185504	190301
		35240	82620	92167	177170	180489	186342	192572
		35086	83725	93522	178300	181778	185885	191909
		36213	84263	93432	177529	181683	187422	192323
		36347	85710	93281	178099	181332	185910	193570
			85577	93765	177975	180179	186526	195200
			85320	94061	177737	181049	187043	196112
			86483	94077	178640	182377	186648	195802
			86192	94629	178684	181703	187667	195483
			87019	94841	178708	183489	186770	196801

Table D.1

D.3 Adaptation Process Benchmark Server Performance Data

Time (seconds)	User	System	idle	Load Avg. (1m)
1	0.005	0.007	99.964	0.48
2	0	0	100	0.48
3	0	0	100	0.44
4	0	0	100	0.44
5	0	0	100	0.44
6	0	0	100	0.44
7	21.772	16.329	61.013	0.44
8	25.48	24.456	49.296	1.13
9	23.949	35.032	39.108	1.13
10	18.961	45.584	28.182	1.13
11	16.495	39.562	40.077	1.13
12	15.769	41.026	42.436	1.13
13	10.163	53.45	36.01	2.8
14	8.462	48.077	42.179	2.8
15	10.376	53.826	35.668	2.8
16	6.202	66.408	25.323	2.8
17	7.445	45.828	45.956	2.8
18	9.744	50.769	39.103	4.58
19	5.21	48.539	46.252	4.58
20	11.84	40.927	39.254	4.58
21	4.853	52.235	37.42	4.58
22	10.982	58.915	29.845	4.58
23	14.487	38.974	39.359	5.57
24	5.912	54.465	38.868	5.57
25	6.226	42.567	51.207	5.57
26	5.74	56.122	37.628	5.57
27	2.293	57.07	40.637	5.57
28	3.502	91.18	4.41	7.05
29	3.927	93.063	0.785	7.05
30	9.961	56.792	24.709	7.05
31	7.188	43.632	43.127	7.05
32	3.444	70.234	25.338	7.05
33	8.171	66.018	25.422	7.28

Time (seconds)	User	System	idle	Load Avg. (1m)
34	5.634	52.881	41.101	7.28
35	3.275	50.63	45.97	7.28
36	3.634	37.845	54.511	7.28
37	6.122	44.898	47.832	7.28
38	3.904	66.499	29.345	7.9
39	6.769	69.604	23.116	7.9
40	3.046	55.457	32.741	7.9
41	3.596	87.828	5.809	7.9
42	8.184	51.023	39.514	7.9
43	2.915	41.065	54.499	8.15
44	2.757	67.043	29.449	8.15
45	4.611	82.74	12.648	8.15
46	7.692	48.718	43.59	8.15
47	8.279	46.572	44.502	8.15
48	6.801	45.97	47.103	9.02
49	10.89	51.527	36.122	9.02
50	2.046	51.918	37.468	9.02
51	9.724	49.934	36.268	9.02
52	9.231	44.744	44.231	9.02
53	5.784	49.486	44.73	9.02
54	8.428	3.774	84.78	9.02
55	5.882	31.74	62.255	9.02
56	5.89	45.967	48.015	9.02
57	5.35	56.178	38.344	9.02
58	8.312	51.535	39.514	10.7
59	6.692	48.391	42.6	10.7
60	6.08	45.019	40.621	10.7
61	6.536	52.68	40.654	10.7
62	5.964	47.335	44.543	10.7
63	3.133	68.17	28.697	11.2
64	4.768	84.636	10.596	11.2
65	7.314	78.457	14.229	11.2
66	6.795	48.974	43.462	11.2

Time (seconds)	User	System	idle	Load Avg. (1m)
67	7.426	49.168	43.278	11.2
68	7.995	55.701	36.173	12.07
69	8.915	48.062	40.827	12.07
70	6.606	52.332	36.528	12.07
71	5.619	47.51	46.871	12.07
72	9.974	50.656	38.583	12.07
73	3.581	48.849	47.315	12.54
74	10.145	48.09	41.502	12.54
75	9.278	47.036	43.557	12.54
76	10.143	17.295	72.432	12.54
77	2.128	0.501	96.996	12.54
78	0.878	0	99.122	11.54
79	1.129	1.004	90.59	11.54
80	0.879	1.633	87.186	11.54
81	2.005	0.251	97.744	11.54
82	0.999	0	96.879	11.54
83	0.753	0	94.228	10.85
84	1.38	0.125	98.494	10.85
85	1.754	0.125	98.12	10.85
86	1.126	0	98.874	10.85
87	0.751	0	99.249	10.85
88	1.128	0.125	98.622	9.98
89	1.757	1.631	93.099	9.98
90	1.253	1.128	94.987	9.98
91	1.629	0	98.371	9.98
92	1.382	0.251	97.487	9.98
93	0.876	0.125	98.874	9.18

Table D.2

D.4 Runtime Performance Benchmarking Raw Data

1	10	25	50		100				200							
53	55	51	53	58	80	50	1036	2265	40	62	930	2013	2975	3941	4969	5938
	44	43	43	47	64	41	1080	2319	44	59	998	2043	2996	3980	5006	5975
	38	40	59	41	44	43	1131	2371	47	60	1024	2078	3032	4016	5048	6036
	39	43	53	63	47	47	1176	2416	47	59	1044	2115	3054	4061	5085	6048
	38	40	44	47	49	59	1219	2457	45	100	1094	2151	3138	4098	5120	6094
	38	37	84	60	40	100	1261	2502	48	136	1164	2193	3172	4134	5164	6130
	40	40	49	131	44	143	1315	2555	48	174	1150	2228	3200	4175	5202	6166
	39	38	52	222	48	195	1370	2599	51	208	1214	2265	3221	4214	5238	6205
	38	38	60	293	45	236	1469	2650	39	249	1255	2308	3282	4252	5274	6245
	38	43	58	390	55	288	1515	2693	70	283	1296	2347	3291	4303	5317	6286
		44	48	442	51	328	1558	2744	66	330	1329	2391	3343	4330	5352	6334
		41	39	554	45	405	1610	2790	79	366	1377	2427	3399	4381	5391	6371
		40	53	625	51	423	1657	2836	44	406	1412	2464	3428	4462	5431	6407
		71	45	721	47	464	1698	2882	41	445	1458	2505	3463	4486	5471	6443
		38	62	802	50	520	1751	2928	46	486	1495	2540	3504	4566	5507	6481
		37	48	889	45	576	1796	2979	53	518	1537	2581	3543	4574	5545	6545
		38	58	983	71	618	1853	3021	98	555	1575	2618	3582	4622	5584	6562
		38	50	1074	48	662	1892	3071	47	591	1616	2656	3619	4655	5622	6603
		39	44	1146	59	712	1939	3116	42	643	1653	2693	3662	4696	5666	6640
		38	66	1222	41	757	1985	3158	43	692	1700	2732	3701	4726	5707	6678
		40	61	1355	47	806	2029	3202	45	741	1738	2773	3743	4782	5772	6715
		37	77	1351	54	848	2077	3253	40	778	1764	2817	3780	4817	5784	6760
		38	42	1486	45	890	2119	3297	47	816	1802	2862	3819	4854	5843	6798
		71	94	1561	48	941	2164	3337	72	851	1839	2951	3868	4893	5863	6849
		64	46	1654	48	985	2219	3379	50	890	1895	2925	3902	4929	5901	6889

Table D.3: Direct

1	10	25	50		100				200							
94	92	100	111	164	188	336	986	2537	121	1225	1231	2455	2794	3879	4973	5123
	88	111	121	100	159	366	1008	2628	104	1374	1486	1780	2897	4152	4807	5146
	95	99	150	114	180	370	1176	2646	221	1331	1525	2702	2848	4087	4599	5168
	90	130	128	187	236	393	1379	2693	308	1356	2209	2668	3017	3941	5034	5247
	93	130	158	262	246	415	1459	2687	343	1363	1354	2494	3037	4136	5174	5286
	102	116	127	329	132	445	1504	2729	369	1258	1422	2230	3167	4241	5093	5309
	94	124	164	465	259	474	1636	2813	294	1653	1583	1713	3191	4368	4858	5369
	94	132	140	533	146	442	1595	2897	541	1446	1825	2085	3096	4284	5339	5496
	85	135	208	611	183	588	1694	2910	280	1310	1390	2294	3228	4350	4952	5510
	91	118	243	680	194	553	1751	2934	380	1397	1525	2173	3295	4157	5377	5479
		148	210	779	224	534	1951	2990	392	1315	1414	2277	3225	4475	4611	5613
		112	145	854	190	605	1879	3106	182	1340	1541	2489	3462	4519	4922	5679
		105	198	939	213	684	1819	3117	453	1256	1529	2333	3505	4439	4614	5576
		106	184	971	256	628	1897	3223	386	1214	1385	2400	3380	4599	5013	5633
		119	175	1043	220	741	1873	3225	230	1350	1397	2597	3538	4636	5359	5822
		135	109	1091	209	745	1986	3180	354	1416	2084	2642	3614	4721	5011	5706
		141	98	1203	154	723	2026	3336	205	1470	1413	2558	3511	4557	4708	5749
		130	118	1264	122	748	2162	3270	853	1460	2079	2547	3696	4706	4735	6618
		127	115	1407	176	768	2143	3313	410	2117	2203	2526	3746	4749	4863	6666
		133	108	1492	212	792	2196	3329	478	1990	1473	2636	3828	4730	4876	6677
		125	108	1571	208	815	2243	3409	434	1565	2214	2778	3837	4751	4868	6723
		124	138	1651	227	840	2235	3492	208	1470	1468	2930	3959	4814	4998	6736
		118	21	1701	333	864	2354	3627	765	1585	1500	2859	3711	5030	4988	6794
		111	107	1779	282	937	2432	3666	406	1400	1927	2760	3877	4987	5023	6837
		107	172	1905	286	964	2479	3709	351	1678	2518	2851	3836	4717	5075	6866

Table D.4: Proxied via WSBPEL Process

D.5 Runtime Performance Benchmarking Raw Data

1	10	25	50		100				200							
41	47	54	41	61	48	43	1060	2260	50	40	905	1978	2972	3958	4995	5981
	41	48	59	56	49	46	1107	2330	42	53	946	2017	3009	4001	5033	6018
	43	40	48	50	39	75	1158	2357	84	47	982	2055	3048	4037	5077	6064
	43	39	59	40	44	99	1210	2403	43	51	1023	2090	3093	4108	5113	6112
	38	41	49	95	47	72	1259	2446	50	88	1061	2128	3126	4119	5154	6124
	42	45	42	167	49	114	1303	2503	39	121	1103	2165	3169	4153	5189	6168
	39	43	59	238	41	195	1354	2546	42	168	1149	2207	3234	4191	5235	6212
	42	39	54	329	41	212	1401	2604	46	203	1187	2246	3247	4290	5269	6246
	41	40	46	402	44	227	1443	2644	44	239	1227	2284	3286	4329	5314	6284
	41	40	57	477	52	292	1498	2700	50	285	1263	2331	3325	4373	5343	6323
		39	48	528	43	346	1544	2743	45	322	1301	2372	3364	4409	5384	6364
		43	64	579	46	383	1595	2797	43	363	1344	2416	3400	4446	5424	6401
		39	73	650	47	429	1636	2841	44	401	1401	2453	3462	4484	5464	6443
		41	51	743	50	480	1681	2883	44	446	1424	2490	3481	4521	5510	6481
		39	45	835	43	522	1736	2934	49	474	1462	2532	3520	4561	5547	6518
		37	57	908	45	577	1783	2975	42	519	1591	2571	3558	4604	5589	6555
		38	49	1004	48	622	1829	3034	52	560	1628	2613	3599	4645	5625	6592
		37	47	1074	48	679	1872	3078	45	600	1668	2644	3640	4681	5662	6641
		40	55	1148	50	734	1924	3129	46	635	1710	2694	3681	4720	5700	6679
		41	46	1242	41	759	1969	3176	48	678	1748	2733	3717	4761	5747	6721
		37	60	1338	44	820	2015	3232	58	720	1784	2769	3758	4798	5782	6761
		40	104	1412	44	865	2071	3273	45	758	1822	2814	3799	4843	5821	6798
		41	55	1507	50	916	2112	3319	46	798	1885	2857	3837	4878	5861	6834
		41	49	1578	72	958	2167	3375	42	837	1898	2892	3882	4917	5900	6871
		40	45	1669	48	1005	2208	3416	46	869	1937	2931	3918	4953	5943	6915

Table D.5: performBlockingInteraction Direct

1	10	25	50		100				200							
93	86	145	107	220	113	377	860	2375	136	696	2085	1880	3243	3528	4670	5640
	74	100	171	298	130	326	985	2423	149	785	2025	1914	3213	3554	4453	5577
	88	87	113	373	167	340	1011	2468	201	682	1505	1941	3171	4244	4510	5595
	77	204	184	468	148	345	1215	2537	369	729	2047	2265	3647	3606	4755	5603
	81	125	114	484	195	362	1334	2594	299	837	2134	2473	3530	3633	4777	5765
	77	120	102	507	210	386	1416	2616	250	763	2071	2534	3649	3696	4783	5551
	76	108	106	660	189	410	1503	2757	199	794	1525	2319	3368	3725	4968	5646
	75	106	98	750	187	430	1473	2732	241	687	2083	2609	3228	4421	5047	5679
	79	99	121	825	236	458	1523	2850	349	1279	1702	2882	2979	3710	4846	5639
	77	98	106	876	120	471	1639	2841	240	1015	2126	2578	3734	3748	4908	6358
		94	99	950	182	519	1662	2814	346	1288	1955	2911	3651	3802	5105	6474
		101	127	1004	182	535	1685	2881	351	1275	1550	2655	3807	3823	4960	6495
		117	176	1060	143	562	1698	2947	434	681	1614	2252	3802	3881	5098	6511
		86	116	1118	154	516	1719	2931	386	690	1810	2865	3850	3892	4982	6591
		105	93	1199	192	536	1827	3044	376	1902	1754	2778	3182	3957	5125	6611
		106	98	1293	221	676	1808	3056	393	1469	1741	2920	3860	3998	5019	6673
		99	176	1371	202	661	1902	3156	450	1494	1996	2963	3237	4012	5057	6714
		90	127	1420	121	815	2025	3249	451	2078	1665	2416	3922	4058	5171	6727
		101	169	1516	116	655	2090	3208	506	1705	1961	2473	4041	4072	5159	6767
		87	179	1591	116	673	2106	3270	470	2107	1760	3211	3994	4265	5231	6787
		100	102	1663	207	760	2249	3339	654	1935	2132	3043	4051	4291	5449	6881
		78	103	1756	239	842	2200	3418	352	1612	2118	3018	3412	4208	5229	6917
		84	99	1791	266	864	2246	3473	641	1608	1825	3182	3980	4584	5306	6936
		81	90	1843	424	801	2286	3531	392	2134	2254	3132	3469	4381	5567	6977
		78	148	1956	315	864	2374	3570	690	2006	2104	3130	3530	5029	5679	7000

Table D.6: performBlockingInteraction Proxied via WSBPEL