Terms and Conditions of Use of Digitised Theses from Trinity College Library Dublin

Copyright statement

All material supplied by Trinity College Library is protected by copyright (under the Copyright and Related Rights Act, 2000 as amended) and other relevant Intellectual Property Rights. By accessing and using a Digitised Thesis from Trinity College Library you acknowledge that all Intellectual Property Rights in any Works supplied are the sole and exclusive property of the copyright and/or other IPR holder. Specific copyright holders may not be explicitly identified. Use of materials from other sources within a thesis should not be construed as a claim over them.

A non-exclusive, non-transferable licence is hereby granted to those using or reproducing, in whole or in part, the material for valid purposes, providing the copyright owners are acknowledged using the normal conventions. Where specific permission to use material is required, this is identified and such permission must be sought from the copyright holder or agency cited.

Liability statement

By using a Digitised Thesis, I accept that Trinity College Dublin bears no legal responsibility for the accuracy, legality or comprehensiveness of materials contained within the thesis, and that Trinity College Dublin accepts no liability for indirect, consequential, or incidental, damages or losses arising from use of the thesis for whatever reason. Information located in a thesis may be subject to specific use constraints, details of which may not be explicitly described. It is the responsibility of potential and actual users to be aware of such constraints and to abide by them. By making use of material from a digitised thesis, you accept these copyright and disclaimer provisions. Where it is brought to the attention of Trinity College Library that there may be a breach of copyright or other restraint, it is the policy to withdraw or take down access to a thesis while the issue is being resolved.

Access Agreement

By using a Digitised Thesis from Trinity College Library you are bound by the following Terms & Conditions. Please read them carefully.

I have read and I understand the following statement: All material supplied via a Digitised Thesis from Trinity College Library is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of a thesis is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form providing the copyright owners are acknowledged using the normal conventions. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.
Personalised eLearning Development Environments

A thesis submitted to the
University of Dublin, Trinity College
for the degree of
Doctor in Philosophy

Declan Dagger
Knowledge and Data Engineering Group,
Department of Computer Science,
Trinity College,
Dublin

Submitted June 2006
Declaration

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

Declan Dagger
June 2006
Permission to lend or copy

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

[Signature]
Declan Dagger
June 2006
ACKNOWLEDGEMENTS

During the writing of this thesis I have been influenced by many people. Their influence, guidance and support have made it a joyous adventure and a rewarding experience. I would like to now acknowledge the important contributions that these people have made over the past years which have allowed me to get to where I am today.

Firstly, I would like to thank my supervisor, Vincent Wade, whose vast experience and active support has made this work possible.

As importantly I would like to thank my family. My parents, Margaret and Paul, for their unconditional support and guidance through the past 25 years of my life and the great example which they have set and which I strive to live up to. They have provided me the tools I need to survive in this world. I would also like to thank my brother, Martin, for his support in the past and my uncle, Vincent, for his insight and influence during my educational ventures. Here I would like thank my girlfriend, Claire, who provided support and encouragement over the years and who endured my long, and sometimes boring, speeches about my research.

I would also like to extend my appreciation to the members of the Knowledge and Data Engineering Group (KDEG) and the Centre for Learning Technology (CLT) in Trinity College, Dublin. In particular, I would like to thank Owen Conlan whose friendship and insightful contributions have had a significant impact on this work.

Finally, I would like to express my sincere gratitude to Intel Ireland for funding the last four years of research which has lead to the realisation of this thesis. More specifically, I would like to express my gratitude to Josh Painter and Jim Kelly, who supported this process in their respective mentoring roles.

I sincerely thank you all.

Life grants nothing to us mortals without hard work. – Horace (65 BC – 8 BC)
ABSTRACT

Personalised eLearning is seen as a key element for next generation educational programmes [DFES, 2005]. The goal of personalised eLearning is to support eLearning content, activities and collaboration, adapted to the specific needs and influenced by specific preferences of the learner and built on sound pedagogic strategies. It seeks to maximise the potential of each learner by providing individually personalised learning experiences. More specifically, it offers the vision of dynamically composed courses which are tailored to an individual’s specific needs, experience, prior knowledge, computing environment, connectivity and communication preferences. Personalised eLearning is an enabling technology that allows learners of varying degrees of experience, knowledge and capabilities to access advanced learning opportunities. However some of the major challenges to the mainstream adoption of personalised eLearning are the complexity of design and the time involved in composing the adaptive learning experience. The key goal in personalised eLearning development is to support the teacher in composing adaptive and non-adaptive eLearning experiences by reusing knowledge pertaining to best practice in eLearning, pedagogical strategies and instructional design techniques. This thesis proposes an innovative approach to supporting the teacher in developing pedagogically-driven personalised eLearning experiences. The thesis describes the requirements, both educational and technical, for personalised eLearning development environments. Based on these requirements, the thesis proposes a flexible and extensible architecture for personalised eLearning development environments. Presented in this work are the theoretical models, the design and the implementation of personalised eLearning development environments. To evaluate and validate this research, a detailed analysis of the different aspects of this research is presented, outlining and addressing course developer satisfaction, pedagogical scaffolding, ease of reuse and usability issues.
# TABLE OF CONTENTS

[ACKNOWLEDGEMENTS](#) .................................................................................................................................................. iii

[ABSTRACT](#) ............................................................................................................................................................ iv

[TABLE OF CONTENTS](#) ............................................................................................................................................... v

[TABLE OF FIGURES](#) ................................................................................................................................................... xii

[ABBREVIATIONS](#) ...................................................................................................................................................... xvi

1 Introduction ................................................................................................................................................................. 1

1.1 Motivation ............................................................................................................................................................... 1

1.2 Research Question .................................................................................................................................................. 3

1.3 Goals and Objectives .......................................................................................................................................... 4

1.4 Contribution to State of the Art .......................................................................................................................... 5

1.5 Overview of Thesis .............................................................................................................................................. 6

2 State of the Art of Adaptive Hypermedia Systems, Adaptive Educational Hypermedia
   Systems and Intelligent Tutoring Systems ................................................................................................................. 9

2.1 Introduction ............................................................................................................................................................ 9

2.2 Hypermedia ........................................................................................................................................................... 10

2.3 Axes of Personalisation ....................................................................................................................................... 13

2.3.1 Commentary on Axes of Personalisation ....................................................................................................... 16

2.4 Techniques of Adaptivity ..................................................................................................................................... 17

2.4.1 Commentary on Techniques of Adaptivity ..................................................................................................... 21

2.5 Standards and Technologies which support eLearning and Adaptivity ................................................................. 22

2.5.1 Enabling Technologies .................................................................................................................................... 22

2.5.2 Conforming Standards ................................................................................................................................... 23

2.5.3 Analysis ............................................................................................................................................................ 24

2.6 Analysis of Personalisation Architectures .......................................................................................................... 25

2.6.1 AHAM, LAOS and LAG .................................................................................................................................. 27

2.6.1.1 Analysis .......................................................................................................................................................... 31

2.6.2 IMS LD ............................................................................................................................................................... 32

2.6.2.1 Analysis .......................................................................................................................................................... 35

2.6.3 Multi Model Metadata Driven Approach to Adaptivity ..................................................................................... 35

2.6.3.1 Analysis .......................................................................................................................................................... 38
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.6.4 Comparisons</td>
<td>38</td>
</tr>
<tr>
<td>2.7 Conclusions</td>
<td>40</td>
</tr>
<tr>
<td>3 State of the Art of Educational Development, Course Construction Methodologies and Authoring Tools for Developing Personalised eLearning</td>
<td>42</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>42</td>
</tr>
<tr>
<td>3.2 Key Challenges in Composing Education</td>
<td>42</td>
</tr>
<tr>
<td>3.3 Educational Development</td>
<td>43</td>
</tr>
<tr>
<td>3.4 Course Construction Methodology</td>
<td>45</td>
</tr>
<tr>
<td>3.5 Survey of Authoring Systems for Personalised and Personalisable eLearning</td>
<td>48</td>
</tr>
<tr>
<td>3.5.1 Authoring Intelligent Tutoring Systems (ITS)</td>
<td>49</td>
</tr>
<tr>
<td>3.5.1.1 Introduction</td>
<td>49</td>
</tr>
<tr>
<td>3.5.1.2 REDEEM</td>
<td>50</td>
</tr>
<tr>
<td>REDEEM: Goals and Objectives</td>
<td>51</td>
</tr>
<tr>
<td>REDEEM: Architectural Influences</td>
<td>51</td>
</tr>
<tr>
<td>REDEEM: Course Construction Methodology</td>
<td>51</td>
</tr>
<tr>
<td>3.5.1.2.1 Building a Personalised Course using REDEEM</td>
<td>52</td>
</tr>
<tr>
<td>Describing Course Material</td>
<td>52</td>
</tr>
<tr>
<td>Developing Teaching Strategies</td>
<td>54</td>
</tr>
<tr>
<td>Categorising Students</td>
<td>54</td>
</tr>
<tr>
<td>Strategy Refinement</td>
<td>55</td>
</tr>
<tr>
<td>3.5.1.3 Analysis</td>
<td>57</td>
</tr>
<tr>
<td>3.5.2 Authoring IMS Learning Design</td>
<td>59</td>
</tr>
<tr>
<td>3.5.2.1 Introduction</td>
<td>59</td>
</tr>
<tr>
<td>3.5.2.2 ASK-LDT</td>
<td>59</td>
</tr>
<tr>
<td>ASK-LDT: Goals and Objectives</td>
<td>60</td>
</tr>
<tr>
<td>ASK-LDT: Architectural Influences</td>
<td>60</td>
</tr>
<tr>
<td>ASK-LDT: Course Construction Methodology</td>
<td>61</td>
</tr>
<tr>
<td>3.5.2.2.1 Building a Personalised Course using ASK-LDT</td>
<td>62</td>
</tr>
<tr>
<td>Definition of Pedagogical Elements</td>
<td>62</td>
</tr>
<tr>
<td>Definition of Environments</td>
<td>63</td>
</tr>
<tr>
<td>Learning Scenario Design</td>
<td>64</td>
</tr>
<tr>
<td>Adding Adaptivity to a Learning Scenario</td>
<td>64</td>
</tr>
<tr>
<td>Statistical Analysis of Learning Scenario</td>
<td>65</td>
</tr>
<tr>
<td>Wrapping Learning Scenario in a Content Package</td>
<td>66</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.5.2.3 Analysis</td>
<td>67</td>
</tr>
<tr>
<td>3.5.3 Authoring for Adaptive Educational Hypermedia Systems (AEHS)</td>
<td>69</td>
</tr>
<tr>
<td>3.5.3.1 Introduction</td>
<td>69</td>
</tr>
<tr>
<td>3.5.3.2 MOT</td>
<td>69</td>
</tr>
<tr>
<td>MOT: Goals and Objectives</td>
<td>69</td>
</tr>
<tr>
<td>MOT: Architectural Influences</td>
<td>70</td>
</tr>
<tr>
<td>MOT: Course Construction Methodology</td>
<td>70</td>
</tr>
<tr>
<td>3.5.3.2.1 Building a Personalised Course using MOT</td>
<td>71</td>
</tr>
<tr>
<td>Create Concept Map</td>
<td>71</td>
</tr>
<tr>
<td>Convert Concept Map to Lesson</td>
<td>75</td>
</tr>
<tr>
<td>Creating an Adaptive Strategy using MOT</td>
<td>77</td>
</tr>
<tr>
<td>3.5.3.3 Analysis</td>
<td>78</td>
</tr>
<tr>
<td>3.6 Conclusions</td>
<td>80</td>
</tr>
<tr>
<td>4 Design</td>
<td>83</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>83</td>
</tr>
<tr>
<td>4.2 Influences from the State of the Art</td>
<td>85</td>
</tr>
<tr>
<td>4.2.1 Educational Development and Course Construction Methodologies</td>
<td>86</td>
</tr>
<tr>
<td>4.2.2 Authoring of Personalised eLearning</td>
<td>87</td>
</tr>
<tr>
<td>4.3 Adaptive Course Construction Methodology (ACCM)</td>
<td>88</td>
</tr>
<tr>
<td>4.4 Personalised eLearning Development Environments (PEDE)</td>
<td>90</td>
</tr>
<tr>
<td>4.4.1 Educational Requirements</td>
<td>91</td>
</tr>
<tr>
<td>4.4.2 Technical Requirements</td>
<td>93</td>
</tr>
<tr>
<td>4.4.3 Usability Guidelines</td>
<td>93</td>
</tr>
<tr>
<td>4.4.4 PEDE Architecture</td>
<td>95</td>
</tr>
<tr>
<td>4.5 Key Design Elements of Personalised eLearning</td>
<td>98</td>
</tr>
<tr>
<td>4.6 Modelling the Disparate Design Elements of Personalised eLearning</td>
<td>101</td>
</tr>
<tr>
<td>4.6.1 Narrative Structures (Pedagogical Models)</td>
<td>101</td>
</tr>
<tr>
<td>4.6.2 Activity Models</td>
<td>105</td>
</tr>
<tr>
<td>4.6.3 Subject Matter Concept Space (Subject Area Models)</td>
<td>109</td>
</tr>
<tr>
<td>4.6.4 Narrative Attributes (Personalisation Axes Models)</td>
<td>111</td>
</tr>
<tr>
<td>4.6.5 Candidate Learning Resource Models</td>
<td>115</td>
</tr>
<tr>
<td>4.6.6 PEDE Narrative (Personalised eLearning Design Models)</td>
<td>117</td>
</tr>
<tr>
<td>4.7 Influences of Standards and Technologies on PEDE</td>
<td>117</td>
</tr>
<tr>
<td>4.7.1 Representing Narrative Structures (Pedagogy)</td>
<td>118</td>
</tr>
</tbody>
</table>
Pedagogies (Narrative Structures) ................................................................. 305
Activities ........................................................................................................ 306
Subject Areas ................................................................................................. 306
Personalisation Axes (Narrative Attributes) .................................................. 307
Learning Resources ....................................................................................... 307
Learners (LIP) ............................................................................................... 308
Personalised eLearning Designs .................................................................... 309
APPENDIX VII – Phase 1 Trial and Evaluation ............................................... 310
TABLE OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Techniques of Adaptive Hypermedia</td>
<td>18</td>
</tr>
<tr>
<td>2-2</td>
<td>AHAM extension of Dexter's Hypertext Reference Model, extract from [De Bra et al., 1999]</td>
<td>28</td>
</tr>
<tr>
<td>2-3</td>
<td>5 Layers of the LAOS Model, extract from [Cristea &amp; Mooij, 2003c]</td>
<td>29</td>
</tr>
<tr>
<td>2-4</td>
<td>LAG extension of the AM in LAOS, extract from [Cristea et al. 2003c]</td>
<td>31</td>
</tr>
<tr>
<td>2-5</td>
<td>Logical Architecture of IMS LD, extract from [IMS LD BPG]</td>
<td>33</td>
</tr>
<tr>
<td>2-6</td>
<td>the Multi-Model Metadata-Driven Approach</td>
<td>37</td>
</tr>
<tr>
<td>3-1</td>
<td>a Simple Course Construction Methodology</td>
<td>46</td>
</tr>
<tr>
<td>3-2</td>
<td>a Course Construction Methodology</td>
<td>47</td>
</tr>
<tr>
<td>3-3</td>
<td>REDEEM ITS Course Construction Methodology [Ainsworth et al., 2003]</td>
<td>52</td>
</tr>
<tr>
<td>3-4</td>
<td>Describing Course Material with REDEEM</td>
<td>53</td>
</tr>
<tr>
<td>3-5</td>
<td>Defining a Teaching Strategy with REDEEM</td>
<td>54</td>
</tr>
<tr>
<td>3-6</td>
<td>Categorising Learners with REDEEM</td>
<td>55</td>
</tr>
<tr>
<td>3-7</td>
<td>Refining Teaching Strategy with REDEEM by matching learner categories with teaching strategies</td>
<td>56</td>
</tr>
<tr>
<td>3-8</td>
<td>Refining Teaching Strategy with REDEEM by indicating teaching styles</td>
<td>57</td>
</tr>
<tr>
<td>3-10</td>
<td>ASK-LDT Course Construction Methodology [Karampiperis &amp; Sampson, 2004]</td>
<td>62</td>
</tr>
<tr>
<td>3-11</td>
<td>Defining Activity Types using ASK-LDT</td>
<td>63</td>
</tr>
<tr>
<td>3-12</td>
<td>Defining Environments using ASK-LDT</td>
<td>64</td>
</tr>
<tr>
<td>3-13</td>
<td>designing Learning Scenarios using ASK-LDT</td>
<td>65</td>
</tr>
<tr>
<td>3-14</td>
<td>analysing a Learning Scenario using ASK-LDT</td>
<td>66</td>
</tr>
<tr>
<td>3-15</td>
<td>creating a Content Package using ASK-LDT</td>
<td>67</td>
</tr>
<tr>
<td>3-16</td>
<td>MOT Course Construction Methodology based on LAOS</td>
<td>71</td>
</tr>
<tr>
<td>3-17</td>
<td>Existing Concepts Maps in MOT</td>
<td>72</td>
</tr>
<tr>
<td>3-18</td>
<td>Creating a Concept Map using MOT</td>
<td>73</td>
</tr>
<tr>
<td>3-19</td>
<td>Defining Concept Attributes with MOT</td>
<td>74</td>
</tr>
<tr>
<td>3-20</td>
<td>Electromagnetism Concept Map created using MOT</td>
<td>75</td>
</tr>
<tr>
<td>3-21</td>
<td>Defining the Logical Flow of a Lesson with MOT</td>
<td>76</td>
</tr>
<tr>
<td>3-22</td>
<td>Adding weights to Sub Lessons with MOT</td>
<td>77</td>
</tr>
<tr>
<td>3-23</td>
<td>Creating an Adaptive Strategy with MOT</td>
<td>78</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>ACCM</td>
<td>Adaptive Course Construction Methodology</td>
<td></td>
</tr>
<tr>
<td>ACCT</td>
<td>Adaptive Course Construction Toolkit</td>
<td></td>
</tr>
<tr>
<td>AH</td>
<td>Adaptive Hypermedia</td>
<td></td>
</tr>
<tr>
<td>AHS</td>
<td>Adaptive Hypermedia Systems</td>
<td></td>
</tr>
<tr>
<td>APeLS</td>
<td>Adaptive Personalised eLearning Service</td>
<td></td>
</tr>
<tr>
<td>CBT</td>
<td>Computer Based Training</td>
<td></td>
</tr>
<tr>
<td>CP</td>
<td>Course Package</td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td>Dublin Core</td>
<td></td>
</tr>
<tr>
<td>DOM</td>
<td>Document Object Model</td>
<td></td>
</tr>
<tr>
<td>DTD</td>
<td>Document Type Definition</td>
<td></td>
</tr>
<tr>
<td>gAPeLS</td>
<td>Generic Adaptive Personalised eLearning Service</td>
<td></td>
</tr>
<tr>
<td>HCI</td>
<td>Human Computer Interaction</td>
<td></td>
</tr>
<tr>
<td>ITS</td>
<td>Intelligent Tutoring System</td>
<td></td>
</tr>
<tr>
<td>JDOM</td>
<td>Java Document Object Model</td>
<td></td>
</tr>
<tr>
<td>JESS</td>
<td>Java Expert System Shell</td>
<td></td>
</tr>
<tr>
<td>JSP</td>
<td>Java Server Pages</td>
<td></td>
</tr>
<tr>
<td>LIP</td>
<td>Learner Information Packaging</td>
<td></td>
</tr>
<tr>
<td>LD</td>
<td>Learning Design</td>
<td></td>
</tr>
<tr>
<td>LLL</td>
<td>Lifelong Learning</td>
<td></td>
</tr>
<tr>
<td>LMS</td>
<td>Learning Management System</td>
<td></td>
</tr>
<tr>
<td>LO</td>
<td>Learning Object</td>
<td></td>
</tr>
<tr>
<td>LOM</td>
<td>Learning Object Metadata</td>
<td></td>
</tr>
<tr>
<td>PEDE</td>
<td>Personalised eLearning Development Environment</td>
<td></td>
</tr>
<tr>
<td>SCORM</td>
<td>Sharable Content Object Reference Model</td>
<td></td>
</tr>
<tr>
<td>SS</td>
<td>Simple Sequencing</td>
<td></td>
</tr>
<tr>
<td>VARK</td>
<td>Visual, Auditory, Read/Write, Kinesthetic</td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
<td></td>
</tr>
<tr>
<td>XSLT</td>
<td>Extensible Stylesheet Language Transformation</td>
<td></td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Motivation

Adaptive, personalised eLearning offers an important alternative to the traditional ‘one size fits all’ [Laurillard, 1993] approach of online learning [Brusilovsky, 2001] [Brusilovsky, 1998] [Eklund & Brusilovsky, 1998]. More specifically it offers the potential of uniquely addressing the specific learning goals [Kaplan et al., 1993], prior knowledge [Milosavljevic, 1997] and context of a learner [Sheinberg, 2001] to improve their satisfaction and motivation, thus creating a more engaging experience. Personalised eLearning is seen as a key element for next generation eLearning systems [DFES, 2005] [Brusilovsky, 2004].

The core goal of personalised eLearning\(^{1}\) is to support eLearning content, activities and collaboration, adapted to the specific needs and influenced by specific preferences of the learner and built on sound pedagogic strategies [Wade & Power, 1998a] [Wade & Power, 1998b]. In an eLearning experience, for example, personalisation could involve the selection of the most appropriate learning resources based on the learner’s preferred learning style (more pragmatic learners could receive more examples or more interactive content) or the selection of the most appropriate subject concepts based on the learner’s prior knowledge of the specific subject area. In achieving this goal, personalised eLearning can offer many tangible benefits to the entire educational process such as teacher and learner empowerment [Conlan et al., 2004] [Bajraktarevic et al., 2003], educational community collaboration and tailored eLearning delivered “just in time” and “just for you”.

However there are several issues which restrict the mainstream adoption of personalised eLearning. These issues relate to the technical coordination of building instances of such systems, the complexities of developing and composing personalised eLearning offerings and the time expenditure in testing and appraising the produced personalised eLearning design. This thesis focuses on the tools to develop personalised courses and therefore describes the requirements for such tools throughout the writing. Developing such personalisable experiences has typically been a

\(^{1}\) Personalisation involves the process of being adaptive towards a user’s characteristics and preferences. Personalised eLearning generally refers to personalisation towards a learner’s objectives, prior knowledge and learning preferences.
very complex, time consuming and expensive task [De Bra et al., 2003] [Eklund & Brusilovsky, 1999]. Current personalised eLearning development typically involves multi-skilled development teams of technologists, instructional developers, subject matter experts, and requires considerable effort in the integration of the adaptive techniques, pedagogy and curriculum. Personalised eLearning development (as describes in this thesis) differs from Personalised Learning Environments (PLE) [PLE] in that they focus on specific educational experiences and parts of a learner’s learning history. In comparison PLE seek to provide the learners with the ability to personalise their learning “path” through life, by assembling and organising sections of courses (from various vendors such as secondary, tertiary and corporate sectors) in a central location towards their specific life goals. “In this way a large amount of Virtual Learning Environment functionality is placed with the learner either as a desktop application or an independently hosted portal. Institutions would still provide content via repositories, undertake assessment and so on, but learners would interact with these using their personal systems (Personal Learning Environment), comprising their preferred tools and ways of working” [PLE].

Successful personalisable courses, developed using intelligent tutoring technology and adaptive hypermedia technology, tend to have been developed as "once off" offerings or developed as research vehicles. The personalisation development tools that exist today do not provide sufficient support, pedagogically or technically, to the experienced and inexperienced personalised eLearning developers. This support should allow the developers to rapidly compose and execute effective personalised eLearning designs. Pedagogically, this describes support for design diversity, i.e. supporting the design of different educational strategies and models such as the ability to design case studies and to identify which aspects of an eLearning experience should be personalisable. From a technical perspective, this describes support for model development, model reuse, sequencing specification, identification of appropriate learning content and application of personalisation.

---

4 Intelligent Tutoring represents the roles of the teacher through Artificial Intelligence agents. By tracking learner performance, the agents can automatically change to suit the learner.
5 Adaptive Hypermedia represents intelligence in various models, external to the software which reconciles this intelligence and is therefore more flexible than Intelligent Tutoring.
The lack of technical support typically originates with the core design or composition methodology that the development environment is built upon. These development environments are driven by the underlying system design methodology, whether this is an educational standard such as IMS Learning Design [IMS LD] or an architecturally specific adaptive framework such as AHAM [De Bra et al., 2002]. The absence of pedagogical support exists because of the technical focus of these existing development environments [Cristea & de Mooij, 2003a] [Cristea de Mooij, 2003b] [Cristea & de Mooij, 2003c] [Cristea, 2004] [Brusilovsky, 2003], the lack of flexibility in realising the multiple diverse information sources which influence personalised eLearning [Brusilovsky, 2003] and because best practice in educational development is not the key driving force of the personalised eLearning composition process [Brusilovsky, 2003] [Sampson et al., 2005] [Cristea de Mooij, 2003a].

Intelligent Tutoring Systems (ITS) and Adaptive Hypermedia Systems (AHS) have failed to be adopted as a mainstream approach to personalised eLearning in higher education, further education, secondary/tertiary education or corporate training. This, mainly, is due to the inflexibilities and the inherent costs which are emerging from this trend in the lack of support for the teacher.

The key motivations of this research include the lack of educational focus, the technical prerequisite requirements imposed on course developers and the inherent architectural inflexibilities. This range of issues can be directly addressed by identifying and representing the core fundamental requirements of personalised eLearning development environments, personalised eLearning services and the disparate design elements involved with the personalised eLearning development process. The motivation behind this research is to address the core issues of personalised eLearning development in a bid to make the power of personalised eLearning accessible and usable by the educational community.

1.2 Research Question

The research question posed in this thesis is to identify and investigate the key extensible design elements, models and influences required for the composition of personalised eLearning experiences and to develop an integrated, model driven environment which facilitates rapid personalised eLearning development by educators or teachers (non-technical). This research is
focused on the design of a composition tool to support personalised eLearning course development by non-technical course developers. The personalised eLearning development environment should support the course designer, equally both technical and non-technical, in producing pedagogically-driven, activity-oriented personalised eLearning experiences while supporting an extensible range of diverse educational principles and information models.

The argument is that personalisation of educational experiences involves various information sources such as pedagogic and instructional design, activities, subject matter areas, personalisation axes and learning resources. By modelling these various information sources, a support framework for personalised eLearning developers can be provided through the personalised eLearning development environment. The meta-semantics of the model structures involve directed graphs, property lists, control flow sequence models and entity models. In this way the abstract process of model reconciliation is illustrated in an Adaptive Course Construction Methodology (ACCM) and the various information sources become models in the personalised eLearning development environment. This research is exploring the possibilities of building personalised eLearning development environments which maximise flexibility and extensibility of the disparate design elements of personalised eLearning while providing both pedagogical and technical support to the teacher when incorporating them to produce personalised eLearning designs.

1.3 Goals and Objectives

In answering the proposed research question, this thesis illustrated and evaluated a pedagogically-oriented support framework for the composition of personalised eLearning experiences. The goals of which are:

1. To identify and specify requirements for the elements and influences of personalised eLearning experiences
2. To specify an integration process for elements and influences of personalised course composition
3. To develop a suite of tools which support crucial aspects of the personalised eLearning development process
4. To enhance the efficiency of the specified integration process
5. To evaluate the process and suite of tools to support teachers in developing personalised eLearning

To achieve these goals there are certain core objectives that must first be reached. One objective is to identify the current trends in personalised eLearning, Adaptive Hypermedia, Adaptive Educational Hypermedia Systems and authoring tools for adaptive educational hypermedia through a state of the art review and appraisal. Another key objective is to identify the disparate design elements, information sources and influences of personalised eLearning development. A crucial objective following on from this is to specify an extensible set of models to represent these core entities such as pedagogical and educational strategies, learning activities, subject matter knowledge, personalisation axes and personalisable learning designs. Once these models have been formulated, it is the objective of this research to extend existing and proven methodologies for curriculum design to establish an adaptive course construction methodology to capitalise on the flexibility of these separated knowledge models. Based on this methodology, an integral objective is to create a supporting framework, personalised eLearning development environment specifications and a suite of composition tools to allow experienced and, more importantly, inexperienced educators to rapidly compose and appraise personalised eLearning designs. In order to successfully address the goals and objectives set forth in this thesis, an in depth evaluation of the both the process of integration and the support given to teachers, through the suite of tools, is performed.

1.4 Contribution to State of the Art

Several notable contributions to the state of the art of personalised eLearning, which are illustrated throughout this thesis, are attributed to this work. The thesis proposed a novel approach to integrating the disparate design elements and influences required for a personalised eLearning experience. In this approach, the core elements of personalised eLearning, such as pedagogy, activities, subject matter knowledge, personalisation axes and learning resources, were identified, specified and modelled to maximise their reuse, increase the efficiency and effectiveness of the personalised eLearning composition process and provide a base support framework for personalised eLearning development environments.

5
This applied research proposes the modelling of pedagogic knowledge (Narrative Structures) [Dagger et al., 2004], in terms of its activities, concepts, attributes and workflow. This modelled pedagogy helps to produce informative, effective and useful pedagogical models which form the foundations of pedagogically correct personalised eLearning designs. It also proposes the modelling of personalisation axes (Narrative Attributes) [Dagger et al., 2004], in terms of intelligent logic, usability descriptions and usage guidelines, to produce effective, reusable and accessible sources of personalisation support. This thesis presents an Adaptive Course Composition Methodology, which defines a cyclical process for designing personalised eLearning offerings based on the support structure of the knowledge models representing the disparate design elements. As an integral component of this research, a personalised eLearning development environment was prototyped, designed and developed, providing a suite of tools to support the teacher in composing personalised educational experiences. This personalised eLearning development environment is built in reference to the Adaptive Course Composition Methodology and utilised the model-driven support framework offered by the disparate design elements of personalised eLearning. The results of which facilitated the evaluation metrics for this applied research.

1.5 Overview of Thesis

This thesis proposes an innovative approach to capturing and representing the disparate design elements, models and influences involved in the composition of personalised eLearning experiences. Also described is a novel methodology for using the captured and represented knowledge in composing personalised eLearning designs. To facilitate a core objective of this applied research, an insight into the design, prototype and implementation of a personalised eLearning development environment that supports the course developer, both pedagogically and technically, in composing personalised eLearning experiences is provided. Presented through this work are the theoretical knowledge models, the Adaptive Course Composition Methodology, the personalised eLearning development environment and the evaluation and analysis of the research carried out as part of this applied research.

Chapter two begins with a state of the art review and appraisal of current Adaptive Hypermedia Systems (AHS), Intelligent Tutoring Systems (ITS) and Adaptive Educational Hypermedia Systems (AEHS). This is achieved by firstly, introducing and describing adaptivity and
personalisation and how this "technology" is being applied to the field of education. Secondly, the various standards and technologies which support adaptivity and enable personalisation are examined. Following this is a review of the prominent architectural designs for achieving personalisation which establishes the technical context for reviewing the personalisation authoring systems.

Chapter three establishes the educational context for this research. This includes an insight into educational development and how course construction methodologies use constructive alignment to ensure that developed educational experiences are pedagogically founded. Based on these educational development requirements, a review of the state of the art of authoring tools for personalised and personalisable eLearning composition is provided. The aim of which is to provide a balanced view of how the existing authoring tools are progressively evolving.

Chapter four illustrates the design and architectural vision of this research and formulates the base requirements for personalised eLearning development environments. Based on influences from chapters two and three, an Adaptive Course Construction Methodology (ACCM) was developed which demonstrates the fundamental design principles for composing pedagogically-driven, activity-oriented personalised eLearning. This provides the educational foundation for the architectural design of personalised eLearning development environments. Based on this, the disparate design elements of personalised eLearning composition are detailed and the process of modelling each element is described.

Chapter five illustrates the implementation of a personalised eLearning development environment named the Adaptive Course Construction Toolkit (ACCT). This includes a description of the architecture of the ACCT and the different models which it uses. Based on the design specifications in chapter four, the components of the ACCT, based on processes of the ACCM, are described. To remain faithful to the review process of chapter three, the steps involved in composing a personalised eLearning experience with the ACCT are illustrated.

Chapter six presents the results of the trial and evaluation of the applied research carried out through this thesis. To align with the stated research question, see section 1.2, the ACCT was
trialled with real teachers and instructional developers. The analysis of which, focuses on several aspects of this research, including course developer satisfaction, scaffolding for pedagogical diversity, usability of PEDE and usability of the ACCT.

Finally, the thesis concludes with a description of the objectives and achievements of this research, a summary of key contributions to the state of the art of personalised eLearning development which are attributed to this research and a discussion of the pertinent future work to continue and grow this research.
2 State of the Art of Adaptive Hypermedia Systems, Adaptive Educational Hypermedia Systems and Intelligent Tutoring Systems

2.1 Introduction

Dynamically assembling multimedia offerings at run-time to suit the needs and situation of a user, referred to as Adaptive Hypermedia (AH) [Brusilovsky, 1996], presents a considerable research challenge. Such offerings may be tailored to the user’s background, goals and objectives, current interests and tasks, as well as to their contextual surroundings, including device properties, network latency and ambient conditions [Brusilovsky, 1996] [Albert & Lukas, 1999] [Fox et al, 1996] [Mohan et al, 1999]. The key facilitator in achieving such tailoring is the description and execution of appropriate adaptive strategies. Traditionally, adaptive strategies have been embedded in either the media they operate across or in the adaptive engine that performs the adaptation [De Bra et al., 1999]. This situation is far from ideal since the intelligence contained within the adaptive strategy cannot be easily extrapolated and used (reused) outside of its original operating context due to the implicit technical co-dependence between information and adaptive strategy [Brusilovsky et al., 1996b].

The potential value-add from adaptive hypermedia is very real. Adaptive hypermedia can facilitate personalisation on many levels such as the adaptation of video media to presentation preferences, adapting hypermedia to the user’s context or adapting educational experiences to the prior knowledge and preferences of the learner. This is quite evident in the areas of personalised news [Kalfoglou et al., 2001], personalised video [Dogon et al., 2002] [Millard et al., 2003] and personalised eLearning [Brusilovsky, 2004] [De Bra et al., 2003].

Adaptivity is a broad term which is used to describe dynamic system changes based on certain influencing entities. Personalisation typically refers to the adaptivity of a system with a focus on certain characteristics of the end user and their environment. In the context of this thesis, although the phrases "Adaptive Hypermedia", "Adaptive Hypermedia Systems", "Adaptivity" and
“Personalisation” are frequently interchanged, except where explicitly stated, the focus is on personalisation.

The objective of this chapter is to provide the reader with a detailed insight into the current trends of adaptation and personalisation systems. It provides a discussion of the various axes and techniques of personalisation in order to establish the potential benefits that personalisation systems can offer over traditional hypermedia systems. The chapter then provides a detailed description of the various standards and technologies which can support and enable eLearning and personalisation. This acts as an introduction to and foundation for the analysis of the past and current personalisation systems and architectures. Based on this analysis, the chapter concludes by discussing the elements of an adaptive hypermedia architecture which an author must be aware of in order to create personalised eLearning.

2.2 Hypermedia

The word hypermedia implies a combination of “Hyperspace” and “Multimedia”. Hyperspace prescribes an existence in more than three dimensions. Using a hyperlinked structure provides the user with the ability to “jump” across links to reach resources at the far reaches of hyperspace. Multimedia refers to applications that combine text, graphics, full-motion video, and sound in an integrated package.

Provisioning support for the vast diversity in possible navigational paths in a Hypermedia system is not a trivial task [De Bra, 1998]. The user may suffer from “cognitive overload” as they attempt to comprehend the expanse of unstructured information. Similarly they may become disoriented within the system and not know where the current node is positioned in relation to the node which contains the information they require. This phenomenon, coined ‘Lost in Hyperspace’ [Laurillard, 1993], is frequent in situations where users are exposed to large stores of seemingly unorganised information, for example the World Wide Web. Even experienced users become spatially confused in systems whose information structure is random and incoherent. This non-linear representation of the information domain is quite similar in respect to “hypertext”, originally conceived by Vannevar Bush in the mid 1940’s [Bush, 1945] and later coined by Ted Nelson in 1969 [Nelson, 1981].
Adaptive Hypermedia (AH)

Adaptive Media (AM) involves the application of adaptivity to the general area of media such as email, commentary and video. Adaptive Hypermedia (AH) is the specific application of adaptivity to web based navigational structures and content. Adaptive Hypermedia Systems (AHS) have become an extensive and innovative research area in the past decade. They have been applied in areas such as personalised eLearning, virtual interactive personalised museum tours and personalised news. Traditionally, the architecture of an AHS consists of three core input sources, the hyperspace i.e. the set of hyperlinked resources, the strategic rules of how to adapt and a description or profile of the user. These input sources are then reconciled by an Adaptive Engine (AE) [De Bra et al., 1999], based on the strategic rules, to produce a personalised output. The AE is typically a modification of a rules-based logic engine.

One of the primary goals of AHS is to alleviate the cognitive and spatial overload that may be experienced by users of these hypermedia systems. AH attempts to do this by adapting the structure and appearance of the system, individually and dynamically, for each user. The system collates information about each user in a unique user profile and uses this modelled information to make assumptions about how best to change the system to benefit an individual user. The system may infer user objectives, based on their interactions, to provide a tailored information space. These interactions can also help in specifying relevant paths to the information required [De La Passardiere & Dufresne, 1992].

Adaptive systems often infer the requirements of a user and then automatically modify the system accordingly. Based on this automation, a key controversial problem is introduced, namely the sufficient balance of control. This control relates to the user, the system and the user’s awareness of the system made changes i.e. the transparency of the adaptivity. The correctness of the automatic assumptions made by the system cannot be guaranteed. Hence the user must be empowered within the adaptation process in order to take responsibility for the adaptivity of the system.

The system adaptivity may be hidden entirely from the user so that the user is unaware of changes made by the system on their behalf [Brusilovsky, 1995] [Murray et al., 1998]. This has been the approach of many agent based and intelligent systems in the past. Alternatively the adaptivity may
be negotiated with the user, allowing the user to accept or reject modifications suggested by the system. The modifications may be visible to the user but the user may not be able to change them.

Ideally, users should have a certain level of control over the adaptivity but should not have to control it on a continuous basis [Espinoza & Hook, 1995]. System designers must attempt to strike a balance between the control granted to the user and the usability of the system. It is imperative that users should not be surprised, disoriented or displeased by the changes made by the system [De La Passardiere & Dufresne, 1992] and that they feel empowered by the system. When the usability of the interface is in conflict with the potential effectiveness of the system, the designer must attempt to provide adequate balance.

Adaptive Educational Hypermedia (AEH)

Adaptive Educational Hypermedia (AEH) is the specific application of the architectures, the methods and the techniques of AH to eLearning and e-education. There are subtle differences here in that the end user, the person who should be empowered by such systems, is the learner. The type of "adaptivity" that is used in such systems should be based on the application of a certain type of strategic knowledge, namely educational strategy or pedagogy. The profile of the learner then influences this adaptivity process through its characteristics and context, facilitating adaptivity based on various axes of personalisation, as later described. This learner-centric application of adaptivity to education is also known as personalised eLearning. In the context of this thesis, the terms AEH and personalised eLearning are frequently interchanged and should be taken to be semantically similar.

Personalised eLearning is seen as a key element of next generation educational programmes [Brusilovsky, 2004]. It seeks to maximise the potential of each learner by providing individually personalised learning experiences. More specifically, it offers the vision of dynamically composed courses which are tailored to an individual's specific needs, experience, prior knowledge, computing environment, connectivity and communication preferences. Personalised eLearning is an enabling technology that allows learners of varying degrees of experience, knowledge and capabilities to access advanced learning opportunities. The key goal of personalised eLearning is to provide eLearning content, activities and collaboration, adapted to the specific needs and
influenced by the specific preferences of the learner, based on sound pedagogic strategies [Wade & Power, 1998a] [Dagger et al., 2005].

2.3 Axes of Personalisation

Personalisation can occur based on many different user properties such as prior knowledge and competence levels, prerequisites and context. Prior knowledge and competence adaptivity is adapting to the learner’s prior knowledge (conceptions and misconceptions) and competence (proven knowledge) of the information domain being presented. Prerequisite adaptivity is adapting to the currently required prerequisites of the learner such as pre-session defined learning outcomes. Context adaptivity is adapting to the preferences of the learner and semantics of the learner’s current situation.

Personalisation involves the tailoring of information, interface, communication and structuring towards the specific requirements, characteristics, preferences and context of the end user. Personalisation can utilise the techniques of adaptivity, presented later in section 2.4, to achieve its goals of customising and tailoring experiences toward the end users. For example, in the area of personalised news filtering (Adaptive Media), the structure of the information (Adaptive Navigation) and the information itself (Adaptive Presentation) can be personalised towards the preferences of the user. This could, for example, achieve personalisation of news articles based on specific topics, brands, people or context.

The goal of the following sections is to provide an insight into the various axes of personalisation which exist, namely, prior knowledge and competence, aims and goals, preferences, history, cultural background, communication style and needs, cognitive and learning style, device and latency.

Prior Knowledge and Competence

Psychological models of prior knowledge and competence show that novice learners and expert learners have quite different ways of acquiring knowledge. This includes not only different explanations of content such as more directives for the novice learner and the use of more freedom towards the expert learner, but also different navigation approaches such as breadth-first or depth-
first navigation. Depending on their prior knowledge or competence level, the learning content made accessible to the learner is determined by applying meta-information about prerequisite relationships between the concepts of the subject domain and the prior knowledge of the current learner. Systems providing such adaptivity based on the theory of knowledge spaces [Doignon & Falmange, 1999] are the ALEKS system developed by Falmagne and the RATH [Hockemeyer et al., 1998] system developed at the University of Graz (UoG).

Aims and Goals
Learners and/or teachers may differ in their conceptions about the aims and goals of a learning experience, i.e. the learning outcomes. A system could adapt by directing learners towards those concepts they (or the teacher) have specified as a crucial element of a specific outcome [Kaplan et al., 1993]. In educational scenarios where a learner is pursuing qualifications, their goals may be to concentrate on learning the essential concepts to pass an examination, whereas their teacher’s goals may be that they want the learners to become more proficient at solving certain problem types. These contrasting goals, although different in educational focus, are collectively important in achieving personalised eLearning.

Preferences
Basing personalisation on learner preferences is another classical approach from the field of Human Computer Interaction (HCI). The systems interface is adapted to the learner’s preferences, generally determined through options or preferences menus [Helander et al., 1997]. Learner preferences may be used to give the learner a greater sense of familiarity and comfort with the rendering interface.

History
History trails, footprints which are made by the system, landmarks which are made by the learner and progression cues may be personalised by the system based on the learner’s history. Access to a learner’s history can provide a valuable source of information, regarding such things as accreditations to date, performances to date and past misconceptions, towards which personalisation can be based. This approach has been realised in the AHA! system by De Bra and Calvi [De Bra & Calvi, 1998].
Cultural Background
Personalisation towards cultural background is a more classical approach to adaptivity facilitating, for example, native language, familiar measures and weights, or specific ways of writing things (e.g. colloquial expressions). In a teaching context, this may also be extended to cover other local references, e.g. by naming well-known brands, persons, or incidents. This has partly been realised in the ALEKS system [Doignon & Falmange, 1999]. However, deeper cultural adaptivity is currently a focus for ongoing research [BRICKS].

Communication Style and Needs
Learners may differ in their communication style and needs, for example, they may have a preference for clear directives versus a broader freedom of choice. This topic also includes special communication needs, for example, in the case of handicapped learners who may need special input devices with different facilities, or who may be restricted in the selection of output devices. One example of catering for this special type of adaptivity can be found in the AVANTI system developed by Kobsa’s group [Brusilovsky & De Bra, 1998].

Cognitive and Learning Style
Learning and cognitive styles can also provide the basis for personalisation. These are, at least in their realisation, closely related to the former point of communication styles. Learners differ in their preferred way of “learning” experience and cognitive processing. Examples for considering different cognitive styles are visual, textual, or auditory presentation of information [VARK]. Different learning styles include the presentation of examples, presentation of theoretical knowledge, and practical exercises [Felder & Silverman, 1988] [Honey & Mumford, 1992]. An example of a tutoring system adapting according to learning styles is the CAMELEON system by Laroussi and Benahmed [Ottmann & Tomek, 1998].

Device
Device adaptivity is the ability to adapt information to the characteristics of a specific device. A feature of today’s lifestyle has been the increased mobility of learning situations [Trifonova & Ronchetti, 2003]. Learning takes place in flexibly timetabled sessions at work, at home and when
commuting [Taylor & Evans, 2005] [Lundin & Nulden, 2003] [Lundin & Magnusson, 2003]. With the revolution of mobile devices the learner’s demand for “learning on the go” must be supported [Goh & Kinshuk, 2004]. One of the key challenges of this mobile learning environment, often referred to as mLearning, is to deliver the appropriate learning resource to the learner. This can be achieved by adapting the content to the preferences of the learner, the characteristics of the current device [Binstock, 2003] and the appropriately selected pedagogy.

Latency
Latency adaptivity involves adapting the delivery of information to the characteristics of the underlying communication medium. In an interactive environment, this can be quite crucial since latency refers to the period of time taken for a finite piece of information, i.e. a data packet, to travel to its destination. In a low latent and intermittent environment, such as an occasionally connected computing (OCC) environment [Fineberg, 2003], the delay between user prompt and user response can be quite significant which can lead to slow-functioning systems and user dissatisfaction. Adapting to this contextual information would provide the ability to react to potential delays, allowing the information delivery mechanism to run smoothly.

2.3.1 Commentary on Axes of Personalisation
Many of the axes of personalisation have been identified and described above. However, other alternative axes of personalisation exist, for example, context, i.e. knowledge which, though not defined a-priori, provide useful additional input to the core models of adaptation [O’Connor, 2004]. A key point here is that these axes are closely tied to the objectives of the overall gain which the user is trying to achieve. Another important issue is that personalisation does not need to be directly related to, or specifically influenced by, pedagogical strategy to be useful or effective within a personalised eLearning experience. For example, adapting based on context or environmental properties does not have any direct link with an educational objective although it could be useful within an eLearning scenario.

On the other hand, the information required for personalisation is, typically, not uniquely tied to specific axes; there is no one to one mapping between an information source and a personalisation axis. For example, information about the conceptual representations of a subject area can facilitate
multiple axes of personalisation based on such things as prior knowledge and learning outcomes. Similarly, information about the physical properties of a learning resource can facilitate personalisation based on cultural background, communication style and needs and learning/cognitive style. Likewise, information about the environment of a learning experience can facilitate personalisation based on device and latency. Just focusing on one axis of personalisation would not necessarily inform the information models to be supported in eLearning. Similarly, focusing on one entity, e.g. user model or subject model, is not enough to perform personalisation on even a single axis. So there is an abstract separation of the axis of personalisation and the information model(s) which it needs to operate.

However, it is sometimes the case were you may not know which axes of personalisation are required until runtime. This would typically happen in scenarios where context adaptivity or ad-hoc adaptivity is required. Therefore the course developer must be able to define the personalisation axes supported, at the very least, at design time.

### 2.4 Techniques of Adaptivity

The main motivation for this section is to illustrate the programmatic process involved with initialising the various techniques of adaptivity. This provides a technical background to adaptivity which benefits the analysis of personalisation authoring systems, as detailed in chapter three. The motivations from this respect are that if no authoring tools exist, then these techniques would have to be hand written. Also, since the majority of authoring systems require a level of "programming" in order to facilitate the addition of adaptivity, it must first be understood.

The techniques of adaptivity are used to physically alter either the structure of a hyperspace environment or the leaf nodes of a hyperspace link. The techniques represent the manipulation mechanisms for achieving this hyperspace adaptivity. They are the finest grain of adaptivity, in that they provide the low-level assembly language mechanisms for producing adaptivity.
Figure 2-1 illustrates the different techniques of adaptivity [Brusilovsky, 1998] which fall under the broad coining of Adaptive Presentation (AP) and Adaptive Navigation (AN). AP techniques can be applied to multimedia and text as illustrated in Figure 2-1. In this regard, it can provide adaptivity of natural language and canned text. For example, based on different cultural backgrounds, an adaptive natural language processor can produce different dialects [Sincaton, 1992] from the same content base for the learner. In reference to canned text, adaptivity can facilitate the inserting/removing of particular text segments, alerting learners to specific sections of text, stretchtext as described below and the sorting of text based on such things as topic. AN techniques on the other hand perform adaptivity of the navigational space such as providing guidance to a learner through “next” and “previous” buttons; such as sorting/hiding/annotating as described below; and adapting map navigational look and feel.

**Adaptive Presentation (AP)**

AP is a mechanism by which the contents of a rich media resource can be altered or modified to satisfy such things as the end user’s preferences, information goals and context. The techniques of
this methodology allow the user to interact with the system to modify the rich media being produced for them. An example of such a system in the domain of personalised eLearning, is APeLS [Conlan & Wade, 2004], which provides guidance and mechanisms which empower learners in controlling the personalisation effects and the pace of the learning offering. Some of the core mechanisms for achieving adaptive presentation are outlined below, namely conditional text, stretch text and frame-based adaptivity.

• **Conditional Inclusion**

Conditional inclusion prescribes the inclusion or exclusion of information, processes or services based on rules decision logic. In its finest state, a conditional statement is an *if-then-else* rule containing an antecedent and a consequent. The antecedent represents the condition being evaluated (preceded by “if”) and the consequent represents the action to take based on the negative evaluation of the condition (preceded by “else”). This method, when applied to fine grained assets, would provide the ability to conditionally include or exclude, with reference to the user model, certain assets in order to personalise the output for the user. For example, the learner may experience a different rendering of the complete “page”, i.e. the same subject concept but displayed within a different pedagogical context.

The condition can be described with a simple Boolean value (true or false). True indicates a readiness to be included, false indicates a readiness to be excluded. For a multi-model metadata-driven system [Conlan et. al, 2002] the learner’s characteristics may be reconciled with subject matter concepts, pedagogical design principals, workflow characteristics, activity structures, context sources and candidate content groups.

• **Stretch Text**

Stretch text is a method whereby sections of a rich media resource can be expanded or contracted based on the user’s competencies with that topic and/or their pre-stated information goals. This method makes use of “hot words” which are “links” that when clicked on replace the “hot word” with a description or example of what the “hot word” is hiding. An example of such a system is MetaDoc [Boyle & Encarnacion, 1994]. MetaDoc uses a method of displaying
all content with the “hot word”s collapsed (hidden). With respect to the user’s knowledge and previous accomplishments it activates or de-activates the “hot word”s.

- **Frame-Based**
  Frame-based Adaptive Presentation is the most powerful of all content adaptation techniques. All concept information is displayed in a single frame, a navigation frame. The slots of the navigation frame can contain various descriptions of the concepts to be taught, links to other frames, examples etc. A rules-based system then decides on what should be displayed in the navigation frame for each user based on his or her user model. Two such systems are Hypadapter [Hohl et al., 1996], which uses a “presentation priority” scheme to decide who should see what, and EPIAIM [De Rosis et al., 1992] uses a method of ordering each scheme into a subset of slots.

**Adaptive Navigation (AN)**
AN is the most commonly used form of adaptivity due to its effectiveness and relative ease of application. The goal of AN is to adapt the navigational structure of a hyperspace to the end user’s preferences, characteristics and context. The techniques of this methodology allow the user to interact with the system and control certain aspects of the navigational structure of the system such as hiding hyperlinks, sorting hyperlinks and annotation hyperlinks. Systems like APeLS (Adaptive Personalised eLearning Services) [Conlan & Wade, 2004] use the semantics of proven strategies, e.g. pedagogy, to influence the effect of this adaptivity technique. Some of the core mechanisms for achieving adaptive navigation are outlined below, namely link sorting, link hiding and link annotating.

- **Sorting**
  The sorting of links into an ordered structure with respect to the user is the most common form of AN. This technique is mainly used in the field of IR (Information Retrieval) where the links are ordered with respect to the current content page. HYPERFLEX [Kaplan et al., 1993], in addition to the above, maintains a list of possible search goals. If the user selects one of these goals then sorting takes into account the relevance of the displayed links to the current goal. Another example of this sort of adaptivity is used in Amazon (Online Bookstore) where an
ACF (Automated Collaborative Filtering) system is used to group people with a common interest together and display a list of books that people with similar common interests have previously purchased.

- **Hiding**
  The hiding technique involves the application of pedagogical rules to decide on the concepts and content that should be shown at a particular point in hyperspace and which concepts and content should remain hidden. This technique accounts for the type of concept, its links and state of the user model. Applying these rules can lead to a system that changes a “hot word” to normal text to preserve consistency. Examples of systems that employ this technique are HyperTutor [Perez et al., 1995] and SYPROS [Harrer & Herzog, 1999].

- **Annotation**
  The annotation based technique involves the construction of a conceptual network to represent the backbone of the hyperspace in relation to the pedagogical domain being modelled. The nodes representing concepts are connected using different types of relationships, i.e. “is-a”, “part-of”, and “prerequisites”. These concept nodes are linked to the actual content pages that form the concept. An example of such a system is the OPAL [Conlan et al., 2002] system.

### 2.4.1 Commentary on Techniques of Adaptivity

The techniques of adaptivity describe the programmatic processes of achieving personalisation based on the identified information sources of the axes of personalisation. It is rarely the case in personalisation, however, that just one technique is applied in isolation. Typically, adaptive techniques are used in conjunction with each other and complement each other. For example, the adaptive effects of several implementations involve modifying the structure or sequence of how learning content is spatially organised while at the same time altering the look and feel of the learning content which is presented. In the past, Adaptive Navigation has been the most widely used form of adaptivity in personalised eLearning, as typically it is very appealing to eLearning as it can personalise the workflow or the learning process.
Adaptive Presentation, on the other hand, is a complex and potentially problematic area of adaptivity when not only dealing with open corpus content but also in-house content. Arbitrarily different content formats, different content structures and different visual representations (look and feel) presents a very challenging obstacle to overcome. With techniques such as inserting, removing and altering, the process of rearranging content can cause semantic drift, can break pedagogical or instructional principles and can create various representation problems. Although advances in language-based adaptivity, such as those emerging from the field of Natural Language Processing, are beginning to address these challenges, it is still a novel area of research and much work must still be done.

### 2.5 Standards and Technologies which support eLearning and Adaptivity

In order to discuss personalisation systems and architectures there are several existing supporting standards which need to be considered. These include standards for content representation, service discovery and knowledge modelling. Similarly, there are several candidate technologies which can support the architectures of adaptivity and eLearning such as web services and Artificial Intelligence (AI). Based on these candidate standards and technologies, this section aims to provide a level of detail which compliments the following section of this chapter, Analysis of Personalisation Architectures.

#### 2.5.1 Enabling Technologies

Firstly we can discuss the various technologies which support architectures for adaptivity and eLearning. Several Artificial Intelligence (AI) techniques exist such as Neural Networks, Expert Systems, KNN Classifiers, and so on [Russell & Norvig, 1995]. These can all make intelligent decisions based on the inputs with which they are supplied. Typically, AHS architectures are based on Expert Systems which consist of collections of “knowledge” and an inference engine capable of reasoning across this “knowledge” [SEMS]. This architectural approach suits AHSs since they typically reason across user and domain information based on some strategic guidance rules, as previously described.
These strategic rules or “Rules Logic” of adaptivity can be represented in a declarative language such as RuleML [RuleML] or in a procedural language such as BPML [BPML] and can be executed in many different reconciliation engines such as JESS [JESS] and YASU [YASU]. Java Expert System Shell (JESS), for example, is a java based expert system which interprets rules written in a LISP [LISP] like language. Rule Mark-up Language (RuleML) on the other hand is an XML based rules language and can be inferred in the Mandarax [Mandarax] engine. Business Process Modelling Language (BPML) as the name indicates provides a language for modelling different business processes. These languages however share something in common; they all interact with and infer across sets of information models. Therefore each of these approaches and languages for representing “strategy” could potentially be used in an adaptive system to represent adaptive properties.

With the emergence of the semantic web, service-oriented architectures are becoming a popular enabling technology for knowledge-driven systems. Similarly, adaptive systems are beginning to realise the potential of Service-Oriented Architectures (SOA) [Erl, 2004]. Services, regardless of the internal workings, syntax and representation, must expose themselves and their functionality in a specific way. The process model of web services indicates an Inputs, Outputs, Preconditions and Effects (IOPE) architecture. This means that a service must take some input and produce some output. The services execution chain is informed by any defined preconditions and the effects inform the working environment of the changes which result from the services execution. Based on this, services and their process model could be seen as an enabling technology for personalisation axes and activities in adaptive eLearning architectures.

2.5.2 Conforming Standards

Secondly we examine the various standards which can support adaptivity and eLearning. Several standards for marking up content are available namely, IEEE Learning Object Metadata (LOM) [IEEE LOM], IMS Learning Resource Metadata (LRM) [IMS LRM] and Dublin Core [DC]. ADL’s Sharable Content Object Reference Model (SCORM) [SCORM] provides an information model and an XML binding for combining aspects of the previously mentioned standards and other content packaging standards such as IMS Content Packaging (CP) [IMS Content Package]. It supports the metadata description of content through a possible ten different categories, namely
LOM, General, Lifecycle, Meta Metadata, Technical, Educational, Rights, Relation, Annotation and Classification. These categories can provide an expert system with an abundance of information regarding taxonomy and vocabulary of content which the executing rules can access and infer upon. Therefore each of these content standards could be used to support the rules execution in an adaptive system.

As previously mentioned, services must expose themselves in a specific way. Access to a service is facilitated through a service interface. There are several standards which facilitate the description of this interface, namely, Web Ontology Language for Services (OWL-S) and Web Service Description Language (WSDL). The information in these interfaces can support adaptivity by providing an information source for an expert system to reason across.

The various other sources of information, such as Subject Area description, which can support and facilitate adaptivity, must be modelled so that the expert systems can reason and infer across them [Henze & Nejdl, 2002]. Several standards for modelling this information exist, such as Unified Modelling Language (UML) [UML], Entity Relationship Modelling (ERM) [ERM] and Web Ontology Language (OWL) [OWL]. For example, OWL facilitates modelling at a conceptual level in that every entity of the modelled world is represented as a concept. Concepts are then related to other concepts using semantic relationships. In this way, a detailed model of an information space can be made.

2.5.3 Analysis

In order for adaptivity and eLearning architectures to be successful, there are several enabling technologies and conforming standards which the designer must be aware of, as described in the preceding sections. Some of the main criticisms about the existing standards which can support adaptivity and eLearning are the complexities of implementation. For example, although SCORM provides an excellent facility for marking up learning content, there are ten possible categories in which to describe a piece of content. If a content body consists of one hundred pieces of learning content then potentially there are one thousand descriptive categories to implement. This is extremely time consuming and contextually sensitive. Similarly, OWL provides a rich descriptive language for modelling information, however, it is very tedious and time consuming to implement.
This indicates that the designer must be aware of both the pros and cons of a standard before choosing to implement it. Similarly, the enabling technologies can enforce restrictions and specific design decisions. For example, JESS would be more powerful and extendable than RuleML. “Jess is [the] rule engine for real programmers. Jess is all about integration with other systems: it's easy to embed, it's easy to extend, it's easy to call out to Java from the rule language, and licensed users get the source code, so you can modify anything in the engine if you want.” [Friedman-Hill, 2003]

The architectures of these systems and the authoring tools, where present, for building adaptivity for such systems are heavily influenced by the chosen technologies and standards. It is therefore a key requirement for adaptivity and eLearning architectures to be described and built with flexibility in mind, often utilising internal canonical forms which are standards-independent but standards-informed facilitating translation if required. This flexibility, if implemented effectively, can support and facilitate interoperability.

### 2.6 Analysis of Personalisation Architectures

The following presents an analysis of the key personalisation architectures. The reason for this analysis is that the current state of the art of authoring tools for personalisation are heavily influenced by the architectures, i.e. the enabling technologies and chosen standards, of the delivery systems. Therefore an analysis of the existing personalisation architectures provides a solid foundation for the authoring systems review in chapter three.

#### First Generation ITS and AHS

Personalisation can support the user in many different ways as illustrated in the preceding sections. Typically, the mechanisms which provide this support depend on both the architectural design and the implementation methods. The fundamental architectures for personalisation originate from Intelligent Tutoring Systems (ITS) and Adaptive Hypermedia Systems (AHS). The base requirement for a personalisation system is that it has some modelled representation of the User (user model) and some intelligence which describes how to apply this user information to adapt the systems output. Typically this user model provides information to the core system about the characteristics of the user which the system can inadvertently utilise to perform some kind of personalisation for that user. The representation of the intelligence (referred to in this thesis as
"adaptive strategy" or "narrative"), however, differs quite dramatically depending on which base architecture is adopted.

Typically, ITS represent their architectural components as a series of "agents", computer programmes [Rao & Georgeff, 1995] based on Artificial Intelligence (AI) techniques, which are responsible for carrying out some specific task, for example, adapting structure based on the specific user or monitoring user performance [Baylor, 2000]. The core issue with the ITS approach is that it tries to represent the different roles such as, coordinator and facilitator, and the different components such as, user monitor, as agents [Kendall, 1998]. This approach disenfranchises the facilitator, for example the teacher, and quite frequently attempts to replace the facilitator completely. Another key issue is that the strategy for adapting is typically represented in an agent or series of agents which greatly restricts the flexibility of the system since the strategy cannot be extrapolated from the system and reused or repurposed elsewhere [Mengelle & Frasson, 1996]. This approach, despite being present for many decades, has failed to be adopted as a de facto approach since implementations are so specific that they restrict any notion of reusability or interoperability. Examples of such systems are DOCENT [Winne, 1991], IDE [Russell et al., 1988], ISD Expert [Merrill, 1987] and Expert CML [Jones & Wipond, 1991].

In comparison to ITS, AHS represent the architectural components as information models which embody data sources such as the content, and embedded rules of how to adapt the content, and the user. An Adaptive Engine (AE) is then responsible for reconciling the adaptive strategy, typically defined in the content’s rules, and the user model to produce a personalised output. Although this solution offers many potential benefits over traditional ITS, it still promotes inflexibility. The intelligence has been moved from a software agent to the content. This is still not ideal since the content becomes restricted to the domain in which the intelligent rules exist. Therefore, the co-dependence between intelligence and content restrict the reusability of both the content and the intelligence of how to adapt the content. Examples of such systems are AHA! [De Bra et al., 1999] and InterBook [Brusilovsky et al., 1996].
Second Generation AHS

The second generation of personalisation architectures and systems are taking a service oriented approach [Brusilovsky et al., 2004a] [Brusilovsky et al, 2005]. These Adaptive Hypermedia Services are splitting the traditional monolithic approach to personalisation into suites of services. Services such as content searching services, presentation services, collaboration services and monitoring services are being created in order to promote reuse of system components [Türker et al, 2005] [Brady et al, 2005] [Brady et al, 2005b]. For example, a customer may prefer the assessment features of Blackboard [Blackboard] and the learner management tools of WebCT [WebCT]. If these separate elements of the systems were available as services the customer may assemble the LMS that best suits their needs. An additional benefit of this framework is the possibility to include personalised services at different points in the Portal, which all of the services are accessible through. For example, the Portal may include not only personalised eLearning services, but also personalised learner management and course management tools. So in this approach, services can be joined to provide tailored delivery platforms for personalisation [Brusilovsky et al, 2005]. Pioneering examples of such systems are KnowledgeTree [Brusilovsky, 2004a], KnowledgeSea [Brusilovsky & Rizzo, 2004] and APeLS [Conlan & Wade, 2004].

Outlined in the following sections are three of the key architectures for personalisation. Firstly, the AHAM (Adaptive Hypermedia Authoring Model) [De Bra et al., 1999], LAOS (Layered approach to AHAM) [Cristea & Mooij, 2003] and LAG (Layers of Adaptivity Granularity) [Cristea & Verschoor, 2004] architectures from the Technical University of Eindhoven (TU/e). Secondly, the standards based architecture for personalisation, namely IMS LD [IMS LD]. Finally, an abstraction based approach to AHS, namely the architecture of the Multi Model Metadata Driven Approach to Adaptivity [Conlan et al, 2002] from Trinity College Dublin. This provides the technical background and understanding needed for the review of authoring systems presented in chapter three.

2.6.1 AHAM, LAOS and LAG

The Adaptive Hypermedia Application Model (AHAM) [De Bra et al, 1999], a “generalised” informal model for applying adaptive hypermedia, was developed at the Technical University of Eindhoven (TU/e) in the Netherlands. It is based on the Dexter hypertext reference model [Halasz
& Schwartz, 1990], consisting of storage, runtime and within-component layers. AHAM, as illustrated in Figure 2-2, extends the storage layer of the Dexter model to include a user model and a teaching model while maintaining access to the within-component layer through the anchoring layer. The user model stores how the user is related to the domain model. The teaching model, which consists of pedagogical rules, utilises the user model information to adapt the domain model and produce presentation specifications for that user so the run-time layer can deliver.

![Diagram](image)

Figure 2-2, AHAM extension of Dexter's Hypertext Reference Model, extract from [De Bra et al., 1999]

LAOS (Layered WWW AHS Authoring Model and their corresponding Algebraic Operators) [Cristea & Mooij, 2003c] is a 5 layer adaptive hypermedia authoring model which extends AHAM. LAOS consists of a domain model, a goals and constraints model, a user model, an adaptation model and a presentation model as illustrated in Figure 2-3.

The domain model (DM) is used to specify a hierarchical conceptual domain and associate learning resources with concepts. The DM attempts to categorise granularity by distinguishing between a concept, a page and a fragment. All other models use filtered versions of the DM as illustrated in Figure 2-3.
The goals and constraints model (GM) supports the definition of "relationships" between the concepts of the DM. The supported relationship types are logical, i.e. "and" and "or". In the GM it is possible to say, for example, that concept A and concept B have an "and" relationship. This relationship information effects, for example, the goals of adaptivity and the constraints of the user model.

The user model (UM) is then an overlay of the information stored in the goals and constraints model. The UM consists of three components:

1. a concept map of user variables and their values
2. a history concept map, i.e. an overlay of the GM omitting concepts not yet seen
3. a future concept map, i.e. an overlay of the GM omitting concepts already seen

The adaptation model (AM) specifies a series of condition/action rules which are based on chosen adaptivity and the filtered UM. These condition/action rules, "pedagogical rules" [De Bra et al, 1999], influence the instantiation of and the continued maintenance of the UM. A further refinement of the AM is found in the Layers of Adaptation Granularity (LAG) model, explained later in this section.

The presentation model (PM) represents the physical properties and the delivery environment. This can then be used to translate an adapted output to a required format.

LAG is a 3-layer adaptation model consisting of low-level direct adaptation techniques, a medium-level adaptation language, and high-level adaptation strategies, as illustrated in Figure 2-4.

The direct adaptation techniques support the adaptive presentation and the adaptive navigation techniques as described earlier in this chapter. This type of adaptivity is defined as a: \{DM, UM, AM, PM\}, based on the domain model, user model, adaptation model and presentation model. This can be further broken down into subsets such as a: \{update, generate\} where update can work on \{DM, UM, PM\} and generate can work on \{PM\}. 

30
The Adaptation language facilitates the grouping of the low-level rules into medium-level rules and provides an “assembly language for adaptivity” [De Bra et al., 2003b]. Rules such as “if/then/else”, “while/do”, “for/do”, “break”, “generalise” and “specialise” are supported in this language. The “generalise” and “specialise” rules are based on Goldstein’s theory of knowledge evolution during knowledge acquisition [Howell & Cooke, 1989].

The adaptation strategies are based on abstract, concrete, active and reflective cognitive theories. It provides a high level representation of the adaptation language and the adaptation techniques, as previously described.

2.6.1.1 Analysis

AHAM addresses some of the concerns about the Dexter hypertext reference model regarding its user-independence. It proposes an extension to the storage layer, providing an overlay mechanism for specifying a user schema based on the domain of the hypertext. The teaching model then overlays “pedagogical” rules to adapt to the learner. LAOS then extends AHAM to provide a layered framework for building adaptive hypermedia and incorporates different layers of adaptation granularity through LAG.
However the waterfall architecture of LAOS implies that the domain model must exist before a goals and constraints model can be specified; the GM model must exist for a user model to be specified and so on down through the layers; creating a tightly coupled AH authoring framework and course composition methodology.

The Key models of this approach are domain, goals and constraints, user, adaptation and presentation. However, there is no explicit facilitation for information sources which exist outside of the five pre-existing layers. If an external source of information becomes a valid input, this must be placed in one of the existing layers, for example context information would be represented in the presentation layer, so too would device information. In the LAOS approach, the "adaptivity" references instances of the information stored in the user model and the goals and constraints model, allowing it to make logical decisions based on this information. However, since the goals and constraints model directly inherits from the domain model, which contains explicit references to content or services, the potential reuse of the conceptual descriptions of the subject area is limited due to its strict co-dependence with explicit content. This restricts the potential reuse of every layer which appears below the domain model layer in LAOS. Therefore, this architectural approach promotes the explicit creation of once off adaptive hypermedia experiences.

2.6.2 IMS LD

IMS Learning Design (LD) [IMS LD] is an eLearning specification published by IMS Global Consortium [IMS GC] which utilises several existing IMS standards. Its main goal is "the development of a framework that supports pedagogical diversity and innovation, while promoting the exchange and interoperability of e-learning materials". It is a revision and extension of the Open University of Netherlands, Educational Mark-up Language (EML) [EML]. IMS LD uses a theatre metaphor to represent education, i.e. methods, plays and acts. The core element of IMS LD is a method. For each method element there is a nested structure of play, act and role-parts, as illustrated in Figure 2-5. Like in a theatre, a play can contain many different acts. The contained acts are run in sequence; each act being triggered by finishing the preceding act. The play is then complete when all acts have been completed. The synchronisation points which illustrate transitions between acts are used by participating roles. Role parts are then used to link the method
with components; which can be activity structures, individual activities (both learning and support) and environments. This assigns an activity to a role for a specific act.

Regardless of the pedagogy involved, in practice, every learning design consists of a method prescribing various activities for different learner and teacher roles in a certain order. Each activity refers to a collection of specific objects and services, called the "environment", required to perform the given activity, as illustrated in Figure 2-5. IMS LD can be implemented on three different levels, Level A, Level B and Level C.

"Learning Design Level A contains all the core vocabulary needed to support pedagogical diversity". Learning Design Level B adds Properties and Conditions to level A, which enables personalisation and more elaborate sequencing [IMS SS] and interactions based on learner portfolios [ePortfolios]. It can be used to direct the learning activities as well as record outcomes. The separation of Properties and Conditions into a separate Schema also enables it to be used
independently from the rest of the Learning Design Specification, typically as an enhancement to IMS Simple Sequencing [IMS SS].

Learning Design Level C adds Notification to level B, which, although a fairly small addition to the specification, adds significantly to the capability, but potentially also to the implementation task where something similar is not already in place.

IMS Learning Design is preferably integrated into an IMS Content Package to create a so called 'Unit of Learning'. The IMS Simple Sequencing Specification can be used to (a) sequence the resources within a learning-object and (b) sequence the different learning-objects and services within an environment. The Simple Sequencing elements can be namespaced into the 'any' place holders of the elements learning-object and environment. These place holders are specified in the binding of IMS LD. Placeholders for meta-data are on various structures within the IMS Learning Design. IMS/LOM Meta-Data can be included at these places.

IMS LD can specify sets of user values which personalisation can occur on. This is achieved through global and local properties which can define a specific value from a learner model which the learning design can be adapted towards. Several mechanisms exist for referencing these user values such as independently defined learner profiling services or IMS QTI applications.

Learning Objectives and Prerequisites can refer to resources that are defined according to this specification. This is seen as a further refinement when needed. Also supported are simple resources (e.g., textual descriptions) of the learning objectives through the standard 'item' mechanism as can be found in IMS Content Packaging. The structure of IMS Learning Design properties can be mapped fully to the IMS LIP. IMS Enterprise can be used for mapping learners and support staff to roles when instantiating a learning design. With the IMS Learning Design Specification it is possible to include SCORM content. It would be necessary to have its type set and the runtime system would have to be able to deliver and manage the SCORM content.
2.6.2.1 Analysis

IMS LD is a fully standardised approach to implementing personalisation. It utilises a range of existing IMS standards such as Content Packaging, LIP, ePortfolios and Simple Sequencing. It takes a community based approach to education as it uses a series of activities, comprising of different roles within the activities, and some sequencing information describing how these activities can be grouped and chained.

The key models of IMS LD are the activity models and the sequencing models. The sequencing of activities and content is achieved through the Simple Sequencing specification. The values of the user can be accessed through local and global property place holders within the UoL but this referencing does not specify the various sources which can provide this information such as existing models, learner profiling services and provides no mechanisms for retrieving said information.

The IMS LD approach uses a level of abstraction and provides access to content/services through an “environment”. Therefore an environment can be included in a number of different activities while the content/services of the environment are not directly referenced. It can also represent and reference various other information sources through a combination of its “environment” element and its use of local and global properties.

2.6.3 Multi Model Metadata Driven Approach to Adaptivity

The Multi-Model, Metadata Driven approach to Adaptive Hypermedia [Conlan et al., 2002] was pioneered at Trinity College Dublin. The approach taken here extended certain models of existing architectures of AH and added new models where needed. The core methodology of this architecture is abstraction. Separating the concerns of adaptivity and eLearning would make reuse outside of original development context easier [Dagger et al, 2003]. The separation was applied to content/rules and content/domain. In this way the rules for manipulating and adapting the content were no longer embedded in the content itself, making it easier to reuse the content and the rules outside of their initial developed context. Similarly, the content used to teach a domain were no longer embedded in the conceptual description of the domain, making it easier to share knowledge about a subject area without having to distribute a set of content which can be used to teach it.
The Multi Model Metadata Driven Approach to Adaptive Hypermedia describes three core models or information sources, namely, the content model, the learner model and the narrative model. The content model is the metadata describing an actual learning resource. The content model should be able to describe the aspects of a learning resource that make it suitable for a particular learning need or technical requirement. This content model should be generic, scalable and extensible, thus facilitating its use in the description of both the fine-grained pieces of learning content and complete adaptive learning systems (and anything in between). The learner model is an overlay of the sections of the domain which are being taught adaptively as specified in the narrative. The narrative model describes the sequencing rules and metadata, developed by domain and pedagogical experts, which govern the range and scope of personalised outputs that the adaptive engine can produce for the learner. The narrative enables the course author(s) to separate the rules which govern how the personalised course is generated from the content that is included in that course.

Another key architectural component of this approach is candidacy [Dagger et al., 2003]. Candidacy is an abstraction mechanism which facilitates the grouping of models based on a “like” principle and supports the separation of concerns. For example, multiple content resources that teach similar concepts but maybe in slightly different ways could be seen as candidates for teaching that concept. The architecture for candidacy supports this through a Candidate Grouping mechanism, allowing those “like” resources to be logically grouped. This principal can also be applied to various different models of this architecture, such as strategic rules and pedagogic strategies.
The multi-model approach, as illustrated in Figure 2-6, proposes that the adaptive engine is responsible for executing the narrative in order to reconcile the various input models in order to produce a personalised output. The execution of the narrative rules by the rules engine, initiates the candidate selection process which is powered by the narrative and influenced by the properties of the modelled learner. This process can then produce a series of personalised outputs.

The multi-model approach, as illustrated in Figure 2-6, is not limited to the three models of learner, content and narrative as previously described. Various other modelled information sources can be used within the decision making process of the narrative. The narrative can reconcile information such as learner characteristics, learning environment, domain descriptions, candidate learning resources, learning resource models and any other modelled information source. For example, it may be desirable to represent the device which the learner is using or the environment they are currently learning in as separate models. When the narrative is being interpreted the input from the device model and environment model may be used to select appropriate content for that device and
environment. For example, if the device display is small you may only want the content that “fits” to be delivered or if the environment is noisy, auditory content may not be appropriate. Both of these example models provide input into the narrative execution process and may be comprised of metadata only.

2.6.3.1 Analysis

The multi-model metadata driven approach to adaptivity represents intelligence in a “narrative” which can reason across and access all models of the architecture. The narrative separates the rules from the concepts, contents and services which the rules can execute across. This separation is seen as an enabling mechanism for reusing the pedagogical rules and intelligence of narrative outside of their original operating context. Another key factor of this approach is candidacy. The multi-model architecture uses abstraction through candidacy, as previously described, which can be applied to all information models. This provides a mechanism for the narrative to access the various information models such as domain, content, environment and so on and make decisions based on values in the learner model.

The key models of this approach are limitless. Currently, they are narrative, content and learner models. But this architecture provisions for multiple influencing models, so in essence, if it can be modelled then it can be reasoned across.

Sequencing occurs at the conceptual level based on the decisions of the narrative and the selected appropriate candidate resources. Selection can occur at a conceptual level through candidacy or at a specific content model level. The user model is typically an overlay of the “adaptable” aspects of the personalised output.

2.6.4 Comparisons

The analyses in the preceding sections provided a detailed description of the various different architectural approaches to personalisation. More specifically it described the AHAM, LAOS and LAG architectures for adaptive hypermedia; the IMS LD standards-based architecture for personalised eLearning; and finally the Multi-Model Metadata Driven architecture for Adaptive Hypermedia. Based on this architectural analysis, several comparisons can be made in reference to
the architectures use and representation of strategic intelligence, content and services and facilitation of multiple information sources.

Each of the described architectures represents strategy in subtly different ways even though at a basic level they all represent strategy in rules logic. The LAOS approach uses different layers of adaptivity granularity to describe strategy. The strategy in this case, as previously described, is tightly coupled with the domain of content and the goals and constraints on that domain. So although complex rules can be defined with this approach, the tight coupling means that without the content the rules have no meaning. With IMS LD, strategic rules are represented in simple sequencing which is, as the name states, simple. This facilitates the description of basic Boolean rules around content and activities. However, the sequencing does not refer to explicit pieces of content but to “environments” and “activities” which makes it possible to reuse IMS LD strategy across different content bases. The multi-model approach on the other hand supports the description of complex rule sets while at the same time maintaining independence from specific content and services through its candidacy architecture.

Content and services are tightly coupled to the domain of LAOS which creates co-dependence between concept descriptions and the content and services which can be used to teach those concepts. On the other hand, IMS LD uses an “environment” to store reference to content and services and then allows an activity to reference one or more environments. This means that the “environment” can change without having to alter the pedagogical structure. Similarly, the multi-model approach uses candidacy to group similar content and services into candidate groups. The narrative can then be used to pedagogically sequence candidate groups. Therefore, as the content base grows and changes the overall pedagogical structure of the course can remain. These factors also influence the reusability of pedagogy since tight dependence restricts reusability whereas loose abstraction facilitates reusability.

LAOS implicitly supports the influence of different information sources which are not explicitly outlined in its different layers. If a new information source or model must be added to the framework, then its values need to be mapped into one of the existing layers. For example, information about a learner’s learning style or a learner’s learning environment might be stored in
the presentation model, allowing these information sources to be reasoned upon. In contrast, IMS LD uses a combination of “environment”, local and global properties to accommodate multiple information sources. The multi-model approach, however, as the name states provisions for multiple, diverse information sources in its base architectural design. As long as the narrative can access the information model it can reason across it.

As this section illustrates, although the majority of architectural approaches appear to be similar there are subtle differences which attribute to their originality. These differences, although they appear minor, can greatly influence the difficulty and complexity of authoring personalisation for such architectures.

2.7 Conclusions

The motivations of this chapter were three fold; to identify the essential elements of personalisation; to highlight the difficulties and complexities in composing personalisation; and to illustrate the diverse approaches to implement AHS.

The core essential elements of personalisation are; a body of hyperlinked documents, resources and services; rules and intelligence which describe how to adaptively present and structure these documents, resources and services; and a modelled representation of user properties. Although these elements can appear in many permutations and implementations, their inclusion in a personalisation system is essential for functional success. Therefore, in its most basic form, any authoring tool for building personalised eLearning must be capable of building sets of executable rules based on information about the concept or content domain and information about learner properties.

The second motivational aspect of this review involves highlighting the difficulty and complexity of authoring the required “models” for the various different personalisation architectures. As illustrated in this chapter, there are many diverse ways of achieving adaptivity. The “rules” which produce adaptivity can be represented in such forms as IMS Simple Sequencing (IMS LD approach), assembly language for adaptivity (LAOS approach) and narrative (Multi-Model approach). However, although these approaches are different, they each share a common trait;
authoring the required rules to execute in such architectures is not only complex and time consuming, they also require some level of programming skill from the author. Typically, the creation of a personalised course requires input from various experts such as subject domain experts, adaptive systems engineers, instructional designers and pedagogical theorists. This not only increases the cost of creating a personalised course, it also increases the time to create, test and deploy it which does not support a rapid prototyping approach to systems development.

The results of this analysis illustrates that in order for a developer to build a personalised output, i.e. personalised eLearning, they must be aware of the core properties of personalisation. They must understand how “adaptive strategies” work. That is an understanding of the diverse information sources which can affect the results of personalisation. They must understand the “granularity of adaptivity”. This implies an understanding of the different levels of adaptivity that can be used. Finally they must understand how to use content and services appropriately. Knowledge of these core elements can enable the development of successful personalisation.
3 State of the Art of Educational Development, Course Construction Methodologies and Authoring Tools for Developing Personalised eLearning

3.1 Introduction

We can see, from chapter two of this thesis, that adaptivity has been a focus for improving the relevancy of content and enhancing the user experience by giving the user an element of control. However, it is vitally important that personalisation systems also reflect educational concerns.

This chapter explores the educational underpinnings for personalised eLearning and the methodologies for educational development. However a full treatise of the diverse educational/pedagogic theories and design models is beyond the scope of this thesis. Rather we provide a comprehensive review of the existing state of the art of personalised eLearning authoring tools. Each system is analysed based on their goals and objectives, their architectural influences and their process of creating a personalised course.

3.2 Key Challenges in Composing Education

Constructivism [Fosnot, 1996] [Duffy & Cunningham, 1996] sees learning as a dynamic process in which learners construct new ideas or concepts based on their existing knowledge and their current educational context. It implies that learners do not passively absorb information but construct it through mental construction. Based on this premise, instructional and pedagogical novelty is required to address the issues of more traditional and static education. This novelty should facilitate the dynamism of learning. For example, based on specified learning outcomes, we may wish to enable the learner to simply recall or comprehend certain information or we may want them to be able to analyse problems or synthesize new solutions. This indicates that a wide range of possible strategies would exist to cover all possible levels of cognition, e.g. Blooms’ taxonomy, see APPENDIX I – Psychology and Cognition. Extensive work has been carried out in the area of mapping pedagogical theory to educational practice to support better pedagogically-informed eLearning design [Mayes & de Freitas, 2004]. Some examples of this include the associationist/empiricist approach where learning is viewed as activity [Putnam, 1995] [Skinner,
1975] [Fodor & Pylyshyn, 1988], the cognitive approach where learning is viewed as achieving understanding [Simon, 1979] and the situative approach where learning is seen as social practice [Barab & Duffy, 2000]. The work carried out in mapping pedagogical theory to educational practice describes scenarios where eLearning does not require new pedagogies but requires a refinement of existing theories to accommodate and utilise the functional novelty that the “e” brings to eLearning. It is beyond the scope of this thesis to delve into any further detail on specific educational theories and practices, although detailed definitions of these terms can be found in the thesis glossary and a number of published books including [Reigeluth, 1999] [Tennyson & Dijkstra, 1997] [Trifonas, 2003].

Currently one of the key challenges of education is to accommodate the diverse characteristics and requirements of the learner. Therefore educational policy makers are suggesting a shift from static education to personalisation [Laurillard, 1993] [DFES, 2005]. More activity focused pedagogies are beginning to emerge, establishing the foundations for “learner-centric” educational environments. This leads to the modification of educational roles. Learners become responsible for their learning and are more actively engaged. The role of the teacher transforms from disseminator of information to facilitator of knowledge acquisition, providing a support framework for the educational process [Carey, 1993]. The sharing of initiative, control and responsibility transports educational theories into the 21st century facilitating contextualised, “anytime, anywhere” learning [Horton, 2000] [Dagger, 2003].

3.3 Educational Development

Education involves the activities of instructing, teaching, facilitating and supporting in knowledge transfer/acquisition and learning. One can divide education up into two broad sectors, namely, formal and informal [Cook & Smith, 2004]. Formal education describes the hierarchically structured school system that flows from primary level to tertiary level. It also includes the organised programs created in business for technical and professional training. Informal learning describes a lifelong process whereby individuals acquire attitudes, values, skills and knowledge from daily experience. This includes influences and resources in their environment, from family and neighbours, from work and play, from the market place, the library and the mass media [Conner, 2004]. Educational development typically involves the detailed process of identifying
base requirements and forecasting further demands on learners at all levels. In the case of formal education, this process can be captured in an educational/training development plan or syllabus.

An educational development plan may be used to describe such things as educational focus, educational prerequisites, performance objectives, accreditation criteria, evaluation mechanisms and conceptual scoping [EDP]. They can be used in formal education as a communication mechanism and contract which shifts responsibility, initiative and control to learners, accommodating learner empowerment. An educational development plan can be justified from administrative, accreditation, educator and learner perspectives that benefit from documented descriptions, milestones and records of the plans performance. It also helps to develop a base educational standard which can greatly enhance economic and social growth.

Quality assurance in an educational development plan can insure continuous monitoring and improvement of the success and effectiveness of the plan [El-Khawas et al., 1998]. The quality assurance metrics can inform the changes and modifications made to the educational plan. An iterative cycle can be applied to eliciting critical feedback from the university or corporation (schools/faculties/departments) and previous review processes. For example, a course/training portfolio could contain information about the subject outlines, learner statistics, achievements and previous changes made.

Educational development is not a static process; it evolves and grows as demands and expectations change, although at a somewhat slower pace. Based on this, traditional educational development plans need modernisation to accommodate for the fast growing technological and social economies [Willis, 1994]. The growing demands for distance education are beginning to be satisfied and the technological advancements of the past are providing more efficient and effective communication mediums to achieve the technical goals of distant education [Moore & Kearsley, 1996].

As educational development methodologies evolve, however, the need to include all learning experiences (formal, informal and social), coined “lifelong learning” [Longworth & Davies, 1996] [Candy, 1991], must be realised. These diverse inputs can impact on all levels of an instantiated
curriculum and novel mechanisms for capturing and reasoning across this vast array of information must be developed.

3.4 Course Construction Methodology

The development of any course typically follows a syllabus authoring process. A key aspect of any development process is curriculum alignment. This involves aligning Learning Goals, Leaning Objectives (Learning Outcomes) with the Assessment techniques by which those outcomes are to be assessed [Mayes & de Freitas, 2004]. The curriculum then aligns the subject matter appropriate for the course with the expressed goals, objectives and assessment. Finally teaching or instructional strategies appropriate for the aligned curriculum and an evaluation strategy to ensure continuous course/syllabus improvement are designed. The development process for an aligned curriculum is iterative, as illustrated in Figure 3-1, meaning that typically there is refinement of the goals, objectives, assessment, instructional strategy, subject matter and evaluation so as to ensure a consistent, yet deliverable course.

Course construction methodologies unlike instructional design approaches, however, are seldom documented. They are typically habitual processes used by educational development practitioners. This lack of documentation makes it difficult if not impossible to share this knowledge, expertise and best practice. Typically a course construction methodology consists of six high level phases, namely analysis, planning, designing, developing, implementing and evaluating. The analysis phase involves the identification of problems and the proposal of solutions based on pre-evaluations, pre-assessments and requirements gathering actions. The planning phase involves the definition of scope, instructional methods and techniques, objectives and constraints. Designing prototype technology-based instructional methods involves actions such as storyboarding, creation of flowcharts and entity relationship diagrams. The development phase involves the identification and selection of appropriate learning material and services required. Reuse of existing material and services could help to reduce workload and increase efficiency of this development phase. Implementing means instantiating and launching the developed course to its target audience. Through constant and persistent evaluation cycles, the course effectiveness can be monitored and increased. Based on this cyclical approach, iterations of the construction process are effectively supported.
Course construction methodologies must be driven by educational development plans and curriculum alignment requirements. These driving factors, as illustrated in the preceding sections, have a great impact on the success and effect of educational offerings. Based on this, a simple course construction methodology, as illustrated in Figure 3-1, must tightly couple the identification of learning outcomes, the specification of evaluation metrics and the assembly of pedagogical and activity sequencing. Therefore, this describes situations where the learning outcomes are deliberately expressed in a form that is assessable.

Figure 3-2, illustrates a course construction methodology, designed in Trinity College Dublin, which was developed in parallel with curriculum alignment and educational development requirements. Typically, the educational developer identifies the required aims and learning outcomes for the course. The types of learning outcomes typically dictate the selection and sequencing of learning methods and activities respectively, hence directly affecting the chosen pedagogical approach. Learning outcome types indicate the different methods of knowledge acquisition, for example, to build something or to analyse something. To contextualise the educational design, descriptions of the subject area are used. This contextualisation is then instantiated with the selection of appropriate learning content and services. Then as part of the quality assurance process, assessment metrics are identified in order to assess learner performance.
against learning outcomes. Finally, the evaluation of the course construction phases provides informative feedback and analysis of the effectiveness of the course design and provides a loop back into the course construction process to edit and modify different elements of the course.

Figure 3-2, a Course Construction Methodology

This methodology is quite similar to the IEEE reference guide for instructional design and development [IEEE RGIDD]. Here they describe instructional design as “the process through which an educator determines the best teaching methods for specific learners in a specific context, attempting to obtain a specific goal”. The steps in this methodology involve; needs assessment; analysis of learner types; identification of learning objectives; selection of instructional strategy; development of learning materials; and specification of evaluation metrics.

The most difficult and time consuming phases of implementing this course construction methodology is the identification, selection and customisation of the learning methods and activity sequencing; the representation and description of the subject area domain; and the selection of appropriate learning content and services. Based on these complexities, the traditional course
construction methodology must be extended to allow for the sharing of this complex information and processes across the development of multiple courses.

3.5 Survey of Authoring Systems for Personalised and Personalisable eLearning

The primary goal of this survey is to provide an analysis of the diverse systematic and educational approaches to building personalised and personalisable eLearning. In doing so it illustrates the significant influences of each system based on the underlying system architectures, as described in chapter two of this thesis, and their use of educational development and course construction methodologies. Based on this analysis, the core requirements for Personalised eLearning Development Environments (PEDE), as described in this thesis, were gathered and outlined.

The growing recognition of the potential of eLearning is producing a vast number of diverse authoring tools for creating different flavours of “eLearning”, from content assembly applications to learning activity creation applications. For example, in the area of learning activity creation there are a number of recognised systems such as the Learning Activity Management System (LAMS) [Dalziel, 2003], MOT + [Paquette et al. 1999], CopperAuthor [CopperAuthor], and the Learning Design Toolkit [Fill et al., 2004] from the Dialog Plus [Freeston & Davies, 2004] project. However, the systems chosen for review in this chapter reflect an ability to create “personalised” or “personalisable” eLearning and accurately represent the various approaches to personalised eLearning authoring based on the architectural views described in section 2.6.

The following sections contain an introduction to each system being appraised, an outline of the goals and objectives of each system, the architectural influences of each system, the implied course construction methodology and the process of creating a personalised course using the system. Each system is analysed based on two fundamental levels of support for the course developer, namely educational support and technical support (usability).

Educational support is crucial for personalised eLearning development environments to be successful. Through this support, scaffolding for diverse pedagogic approaches and instructional methods must be provisioned. There needs to be a focus on the non-technical user, i.e. view the
teacher as the course developer. This involves various levels of technical transparency and innovation when designing PEDE. There also needs to be support for the underlying design lifecycle, i.e. the educational development and course construction methodology. To create a successful PEDE there needs to be inherent support for diverse aspects of personalisation in order to support and accommodate diversity of student.

For comparative purposes, a number of technical issues are also addressed. These look at influences such as who the end user of the system will be and the systems flexibility towards existing and emerging standards. We must also identify issues relating to complexity, i.e. the usability of system and its Human-Computer Interaction (HCI) properties. We also need to identify the complexities of the course development process. A key feature of a successful PEDE is the ability it affords in rapidly creating, modifying and managing the design process.

The final analysis section describes the fundamental similarities and differences between the appraised systems and that of Personalised eLearning Development Environments as proposed in this thesis.

3.5.1 Authoring Intelligent Tutoring Systems (ITS)

3.5.1.1 Introduction

Intelligent Tutoring Systems (ITS) are usually computer-based (agent-based) instructional systems which incorporate techniques and methods from artificial intelligence, as described in chapter two. Typically they embody two main characteristics, namely instructional content which specifies what to teach and an educational strategy which illustrates how it should teach [Wenger, 1997] [Ohlsson, 1987]. This framework describes ITS as both the educational provider and the educational facilitator. Based on this framework, authoring ITS can be a complex and time consuming process since the core elements of the ITS, namely the strategy and the content, are created for a specific purpose [Brusilovsky et al., 1996].

ITS authoring tools have been broken down into seven main categories [Murray, 1999], namely; curriculum sequencing and planning such as DOCENT [Winne, 1991], IDE [Russell et al., 1988], ISD Expert [Merrill, 1987] and Expert CML [Jones & Wipond, 1991]; tutoring strategies such as
Eon [Murray 1998], GTE [Van Marcke, 1998] and REDEEM [Major et al., 1997]; device simulation and equipment training such as DIAG [Towne, 1997], RIDES [Munro et al., 1997], SIMQUEST [Van Joolingen et al., 1996] and XAIDA [Redfield, 1996]; domain expert systems such as Demonstr8 [Anderson & Skwarecki, 1986], D3 Trainer [Reinhardt et al., 1995] and Training Express [Clancey & Joerger, 1988]; multiple knowledge types such as CREAM-Tools [Nkambou et al., 1996], DNA [Shute, 1998], ID-Expert [Merrill, 1998], IRIS [Arruarte et al., 1997], XAIDA; special purpose such as IDLE-Tool/IMAP [Jona & Kass, 1997], LAT [Sparks et al., 1999]; and adaptive hypermedia such as CALAT [Kiyama et al., 1997], GETMAS [Wong & Chan, 1997], InterBook [Brusilovsky et al., 1996], MetaLinks [Murray et al., 1998].

The REDEEM (Reusable Educational Design Environment and Engineering Methodology) system, which is reviewed here, was chosen based on its approach of supporting the course developer in creating pedagogy around existing domains of learning content.

3.5.1.2 REDEEM

The problems faced by the application of ITS in the real world is one of the main driving forces behind the development of authoring environments for the construction of ITS. The reuse of learning resources across multiple ITS is limited and sometimes impossible. Similarly, the instructional strategies developed by the educationalist were even less reusable due to domain, content and system restrictions. For successful development of ITS, the system requires some mapping between the high level pedagogy described by the author and the low level intelligence to be interpreted by the ITS during delivery. REDEEM focuses on authoring pedagogy rather that on helping instructors create domain material. It offers a compromise between the static but simple Computer Based Training (CBT) and the dynamic albeit complex ITS.

One of the issues that occur here is how to sequence the instruction within the learning domain. The “one size fits all” [Laurillard, 1993] paradigm does not work for environments broadcasting to many and differing modalities. REDEEM addresses this by allowing the author to develop multiple routes through the course to offer some pedagogical diversity [Ainsworth & Flemming, 2005] [Ainsworth & Grimshaw, 2004]. While providing this flexibility, REDEEM monitors the relevance of the various routes through the learning material by allowing educators to specify the pre-
requisite knowledge necessary for particular sections. REDEEM also facilitates the transfer of some control over sequencing to the learner.

There is significant evidence that different types of learner benefit from different pedagogical strategies [Cronbach & Snow 1977; Kyllonen & Shute 1989]. Consequently, authoring must support the development of different and diverse pedagogical strategies. REDEEM based ITS can routinely apply a variety of teaching styles during a single lesson. Learner control can be configured using REDEEM's strategy authoring tools. One issue raised is that learners with a lower aptitude should be given less control over how they learn. Difficulty levels can be assigned to each of the learning resources and the delivery system can decide on which to choose. In REDEEM, teachers are allowed to configure different teaching strategies directly and describe, using a simple refinement tool, when they would change some aspect of their strategy in response to a learner's performance. The resulting ITS can adapt both the form of and the content of the instantiated educational strategy.

**REDEEM: Goals and Objectives**

The core goal of the REDEEM system was to provide opportunities for authors to reflect upon their professional practice [Ainsworth et al., 2003] by providing usable tools for building online pedagogy for existing learning content. The REDEEM system should not support the explicit creation of domain material. A goal was to support the importing of existing material as a domain model. The educator can then overlay their teaching practices on top of this domain model to produce a personalisable learning experience.

**REDEEM: Architectural Influences**

Based on the section *analysis of personalisation architectures* in chapter two, it can be noted that architectural influences of REDEEM extended from traditional ITS architectures. In this way, REDEEM supports the course developers in defining parameters such as, domain model, learning content and specific rules which describe how the agents of the delivery shell behave.

**REDEEM: Course Construction Methodology**

The REDEEM course construction methodology is illustrated in Figure 3-3. It describes the process of creating a personalisable course as defining *what* we want to teach, next it describes the audience
by identifying *who* we are teaching to and finally it described the teaching strategies of *how* it is be taught. The *who* and the *what* describes the content body which the students learn from. The *who* and the *how* describes how that specific type of student learns. Based on this, a description of an individually tailored learning experience is constructed.

![Diagram](image)

Figure 3-3, REDEEM ITS Course Construction Methodology [Ainsworth et al., 2003]

### 3.5.1.2.1 Building a Personalised Course using REDEEM

The suite of REDEEM authoring tools reflect the course construction methodology, as illustrated in Figure 3-3. It supports the development of specific teaching strategies, selection of specific content and the specification of different learner types. The course developer can then associate the different teaching strategies and content with different types of learner.

**Describing Course Material**

This is the most time consuming stage of the authoring process using REDEEM. For each domain page the teaching material is classified upon a number of dimensional ratings, whether it is familiar, new or introductory to the student and so on, as illustrated in Figure 3-4. This rating is carried out on a slider scale. The teachers can also associate non-computer based tasks previously created to the domain pages.
Although there is an interactive problem solving environment inherent to the content used, the teacher can also prescribe appropriate questions from a question template to which they can give up to five hints. The shell monitors the interactions and informs the learner model as they proceed. Pages can be combined into sections, can be grouped by relevance and can reference relationships between one another. All this combines to form a semantic domain network interpreted by the shell to produce the desired output.
Developing Teaching Strategies

REDEEM provides an interface for developing multiple simple teaching strategies. It again uses a slider-based approach whereby the teacher can embody various teaching strategies by moving the sliders on the different scales to produce the desired effect, as illustrated in Figure 3-5.

![Defining your teaching strategies](image)

**Figure 3-5, Defining a Teaching Strategy with REDEEM**

For example, selecting high teacher choice ensures that the shell chooses what it considers to be the next appropriate page for the learner and selecting high student choice leaves the selection process entirely up to the learner. Another example would be the teacher selecting whether the tests are interspersed, meaning that questions will follow every section, every page or at the end of the course. The different teaching styles envisaged to be included may be problem-based learning, case-based learning etc. and depending on the strategy and learner preferences the shell could selectively choose the content to display.

**Categorising Students**

The granularity of the categories is completely controlled by the author. An author-defined category can either describe a group of learners or an individual learner, as illustrated in Figure 3-6. They can be construed by the category for an entire session or can be updated through the learner's
performance where the shell changes category over time as the learner advances. REDEEM uses this information to modify its instructional behaviour.

![Defining your student categories](image)

Figure 3-6, Categorising Learners with REDEEM

**Strategy Refinement**

Particular teaching strategies can be assigned to each of the student categories so if a learner changes category the shell can change its teaching strategy for that learner, as illustrated in Figure 3-7.
In addition to this the teacher can describe the circumstances under which they would prefer to see the shell’s teaching behaviour change. This works by eliciting information from the teacher regarding the circumstances under which particular aspects of the current strategy might change, as illustrated in Figure 3-8. The teacher can also indicate to the shell to increase the level of learner control as they improve their knowledge of the subject matter.
3.5.1.3 Analysis

The REDEEM authoring tools provide course developers with the ability to create specific ITS which run on the REDEEM shell environment. They support the creation of simple teaching strategies, learner categorisation and association of teaching strategy with learning content. REDEEM was analysed based on the core requirements for any personalised eLearning development environment, namely the educational support offered to the course developer, see section 3.5.

REDEEM provides scaffolding for pedagogical diversity based on a fixed continuum of pedagogic properties. This allows the course developer to define strategies which provide, for example, more teacher choice or more student choice respectively. However, the fixed vocabulary of pedagogic properties limits the definition of strategy to the scope of those properties. Therefore in REDEEM, pedagogic diversity exists in a restricted space.

Another level of pedagogic scaffolding involves metrics for supporting pedagogical reuse and repurposing. REDEEM does not support the import and customisation of existing pedagogical
strategies which potentially increases the development burden on the teacher and limits peer collaborative development potential.

REDEEM provides a successful implementation of the underlying design lifecycle from which it is based, see Figure 3-3. It has been designed with the teacher in mind and appears technically transparent. The course construction methodology which REDEEM is based on, however, does not reflect the modern requirements of educational development or constructive alignment. Although it identifies what will be taught, how it will be taught and to whom it will be taught, REDEEM's course construction methodology does not support explicit alignment of assessment metrics with the stated learning outcomes and the teaching strategies defined.

REDEEM achieves adaptivity by choosing an appropriate path through the learning material based on the category of learner. In essence, REDEEM adaptively selects a fixed path based on the learner type. This form of adaptivity is flexible but not extensible. It provides canned personalisation whereby each learner, associated with a learner category, can experience a different pedagogical strategy. However, different pedagogical scenarios may require different types of personalisation to be applied, see section 2.3. This static approach does not support the diverse aspects of personalisation which supports and accommodates student diversity.

For the purpose of technical comparison, certain design aspects of REDEEM which effect its reusability are now discussed. REDEEM provides a form-based graphical interface allowing the course developer to “configure” the underlying Intelligent Tutoring System. This type of interface design proves quite difficult in visualising the complete composition process. REDEEM does not provide sufficient capabilities for conforming to existing or emerging interoperability standards as described in section 2.5.2. REDEEM operates at a proprietary level. Because of this closed design, interoperability between REDEEM and other personalisation systems would be difficult. The rapid prototyping of personalised eLearning is not supported in REDEEM. Only fully initialised ITS can be produced by REDEEM. There is no intermediate form which can be executed through the REDEEM shell to validate the skeleton of the ITS.
3.5.2 Authoring IMS Learning Design

3.5.2.1 Introduction

Authoring for personalisation systems typically involves the creation of system specific models for adaptivity, user and subject domain. These standards-independent information models tend to be influenced heavily by the architectural components of the delivery system which is responsible for reconciling the information models to produce some adaptive affects. These influences consist of the architectural vision of that system, including its limitations, and the modelling mechanisms for capturing this knowledge. A certain subset of these influences is originating from the standards communities. IEEE LOM and IMS LRM are published standards which can be used to describe learning objects and learning resources respectively. However these standards representations are focused on learning content, learning objects and learning resources. They do not, however, provide any support for representing instantiated educational strategy.

IMS LD, as described in chapter two of this thesis, provides a standards-based approach for representing personalisable eLearning with a pedagogic focus. The following section analyses the ASK-LDT [Karampiperis & Sampson, 2004] authoring tool for creating IMS LD “Units of Learning” (UoL).

3.5.2.2 ASK-LDT

One of the core obstacles to the application of the IMS LD standard is the lack of authoring support for developers. Form editors such as CopperAuthor [CopperAuthor] provide syntax support for the development of LD. They provide, as their type suggests, forms for creating the required elements of the IMS LD specification. However, they are focused at the level of course developer as technical developer, whereby core structural knowledge of the underlying standard is a key prerequisite requirement for developing personalisable eLearning with them. They provide no pedagogical design support to the course developers, i.e. the teacher. The graphical tools for developing IMS LD such as MOT+ [Paquette, 1999] and ASK-LDT support the course developer at a much higher cognitive level than the form editors. They provide drag and drop interfaces for creating IMS LD based on activities, environments and roles and are much easier to use than the first generation form editors.
ASK-LDT is a graphical authoring tool for building learning scenarios or learning activities based on IMS LD Level B standard specification, as described in chapter two. It provides support for defining activity types, environments, learning scenarios and for creating content packages based on the IMS Content Packaging standard. ASK-LDT, however, takes a restricted scope of the IMS LD standard as it enforces certain criteria on the learning scenario, i.e. a learning scenario in ASK-LDT can contain only one method with one play with multiple acts.

**ASK-LDT: Goals and Objectives**

The core goal of the ASK-LDT authoring environment is to support the creation of standards-conformant learning scenarios consisting of activities, both learning activities and supporting activities, environments and roles. A related goal of this is to provide a graphical authoring environment for composing a learning scenario, allowing the course developer to define activity types, environments, learning scenarios and content packages.

**ASK-LDT: Architectural Influences**

Based on the section analysis of personalisation architectures in chapter two of this thesis, it can be noted that architectural influences of ASK-LDT are based on a subset of the IMS LD standard, i.e. level B. Through using the ASK-LDT authoring environment a Sharable Content Object Reference Model (SCORM) content package is created which can then be played in an IMS-LD player such as CopperCore [CopperCore] or RELOAD [RELOAD], as illustrated in Figure 3-9.
Figure 3-9, ASK-LDT Architectural Diagram [Karampiperis & Sampson, 2004]

**ASK-LDT: Course Construction Methodology**

The ASK-LDT course construction methodology is illustrated in Figure 3-10. It describes the steps in creating personalisable eLearning using ASK-LDT. Firstly, the course developer must define the different *activity types* that the learning design supports. Secondly, the course developer defines the different *learning environments*, as described in chapter two of this thesis. The course developer then defines their *learning scenario* based on the previously defined *activity types* and *learning environments*. The course developer can then *analyse the learning design* which provides statistical information based on activity types and learning environments present. Finally, the course developer *populates* the learning design with content and packages up the learning design.
3.5.2.2.1 Building a Personalised Course using ASK-LDT

The ASK-LDT authoring tool reflects the course construction methodology, as illustrated in Figure 3-10. It supports the definition of activity types, the definition of learning environments, the design of learning scenarios, the analysis of learning scenarios and the population of the learning scenario with learning content [Karampiperis et al., 2004].

Definition of Pedagogical Elements

The fundamental element of a Unit of Learning is the activity. The ASK-LDT environment provides mechanisms for defining activity types. Each activity has an id, a name, a description and a type (learner or staff), as illustrated in Figure 3-11. Using the ASK-LDT environment, the course developer can choose the graphical notation for each activity and its colour scheme. These pedagogical elements, i.e. learning activities, can then be used during the formation of a learning scenario.
Figure 3-11, Defining Activity Types using ASK-LDT

Definition of Environments

An environment, as defined in the IMS LD standard, is a container element for encapsulating and sequencing learning objects and services. Learning objects can contain internal references to other IMS schema such as IMS QTI [IMS QTI] or references to external resources. The ASK-LDT environment provides the course developer with tools to create an environment with an id, a name, a description and a number of LOs, as illustrated in Figure 3-12.
Learning Scenario Design

When the course developer has defined the activity types and the environments, they are then ready to design their learning scenario. The ASK-LDT environment, as illustrated in Figure 3-13, provides the course developer with tools to graphically create their learning scenario based on their previously defined activity types and environments. To add an activity to the scenario, the course developer must click on the “activity” button and then click on the workspace to add the activity. Then they must choose the activity type from a drop down menu on the right hand side panel. A simple “link” allows the course developer to add sequence information to the learning scenario.

Adding Adaptivity to a Learning Scenario

To add adaptive sequencing to the scenario, the course developer can use the “rollup” button which allows them to define the step back or step forward metrics for the scenario. Figure 3-13, illustrates a learning scenario where a pre-test activity is used to adaptively select which version of the introduction the learner should see. A further test activity, will decide which lessons the learner should see. The lesson section on the left would be more focused for beginner levels and the lesson section on the right more focused for advanced levels for example.
Figure 3-13, designing Learning Scenarios using ASK-LDT

Statistical Analysis of Learning Scenario

When designing a learning scenario, the ASK-LDT authoring environment supports the statistical analysis of the scenario based on activity types and environments, as illustrated in Figure 3-14. This pie chart indicates the total percentages of each activity type in the learning scenario. In the case of a complex learning scenario the designer can use this analysis to verify the balance of activity types and environments present.
Wrapping Learning Scenario in a Content Package

The ASK-LDT authoring environment supports the exportation of the learning scenario as an IMS Content Package, as illustrated in Figure 3-15. This allows the course developer to represent the instantiated learning scenario and make it available in a standards-conformant way which can be exported and played on IMS compliant learning design players.
3.5.2.3 Analysis

ASK-LDT authoring environment provides a graphical interface for creating "learning scenarios" as Units of Learning in the IMS Learning Design (LD) standard. It supports the definition of activity types, environments, units of learning and IMS Content Packages. ASK-LDT was analysed based on the core requirements for any personalised eLearning development environment, namely the educational support offered to the course developer, see section 3.5.

ASK-LDT provides scaffolding for pedagogical diversity based on a descriptive set of activity types. The course developer can use the base set of activity types, default in ASK-LDT, and also create their own activity types in order to provide the definition of the pedagogical elements which the learning scenario is defined in terms of. This is a flexible approach and supports the definition of diverse pedagogic scenarios.
Another level of pedagogic scaffolding involves metrics for supporting pedagogical reuse and repurposing. ASK-LDT supports importing and customising existing pedagogic strategies, through its Course Package functionality, which increases the potential for peer collaboration in the design process.

ASK-LDT provides a successful implementation of the underlying design lifecycle from which it is based, see Figure 3-10. However, although ASK-LDT is less complex than its form-based counterparts, it still has a core focus on the course developer being a technical engineer due to its tight coupling with the underlying standard and its lack of abstraction during crucial parts of the design process; see section on Wrapping Learning Scenario in a Content Package. ASK-LDT does not reflect the modern requirements of educational development or constructive alignment. Although it identifies the activity types, the learning environments, the learning scenarios and the learning content which the “Unit of Learning” uses, its course construction methodology does not support explicit alignment of assessment metrics with the stated learning outcomes and the teaching strategies defined.

ASK-LDT achieves adaptivity by using “roll-ups”, whereby a condition can be tested and a resulting action taken. However, the course developer is responsible for explicitly identifying the “source” of the information being tested in the condition. This approach requires a high level of technical input from the course developer and supports the definition of limited functionality such as described in the IMS Simple Sequencing specification. A key point here is that global adaptation strategies, i.e. strategies applied across complete “Units of Learning”, cannot be defined and are not supported by either the ASK-LDT authoring environment or the IMS LD standard.

For the purpose of technical comparison, certain design aspects of ASK-LDT which affect its reusability are now discussed. ASK-LDT amalgamates elements of a strictly form-based graphical interface with more functional drag and drop capabilities. The drag and drop interface provides a visual environment for creating learning scenarios based on the previously defined activity types. The learning scenario design interface is intuitive and easy to use, however the form-based interfaces are highly interdependent on the syntax and semantics of the IMS LD standard. ASK-LDT is based on a modified view of the IMS LD standard and incorporates the standards flexibility.
inherent to IMS LD, such as its links to IMS SS, IMS LIP, IMS QTI, and so on. Rapid prototyping of personalised eLearning can be achieved by playing an IMS LD package, created using ASK-LDT, on an IMS LD Player such as CopperCore or RELOAD.

3.5.3 Authoring for Adaptive Educational Hypermedia Systems (AEHS)

3.5.3.1 Introduction

One of the major criticisms of AEHS is the complex and time consuming nature of building such a system [De Bra et al., 2000] [Brusilovsky et al., 1998] [Forrester, 2000]. This complexity typically leads to AEHS developed to serve specific needs, inevitably reducing their cost-effectiveness and limiting their flexibility and durability. This situation is far from ideal. However, several advances have been made to reduce the complexities of authoring such systems. Research which investigates authoring and design patterns looks to aid in the development of authoring environments which can support the course developers and decrease development costs [Brown et al., 2004]. This broad research topic aims at identifying the repetitive design and authoring steps used by real developers of adaptive educational hypermedia. This novel research direction has lead to the formulation and development of several systems for authoring adaptive educational hypermedia systems.

The following section reviews the My Online Teacher (MOT) system, developed at the Technical University of Eindhoven TU/e. Its focus is in the area of authoring adaptive and adaptable educational hypermedia.

3.5.3.2 MOT

My Online Teacher (MOT) was created at the Technical University of Eindhoven (TU/e) to produce adaptive hypermedia offerings for the AHA! system. MOT provides tools for the course developer to compose lessons and the adaptivity specifications for the content that the lessons will use to teach a specific topic to a discrete set of learners.

*MOT: Goals and Objectives*

The high level goals of MOT are to build an application with specific tools for manipulating concept maps, for constructing lessons based on a concept map and a method for calculating correspondence weights between concept attributes. This leads to a system which can reuse model
parts (concepts, links, attributes and values) and a system which can apply adaptive hypermedia adaptation rules to selected learning content.

**MOT: Architectural Influences**

Based on the section *analysis of personalisation architectures* in chapter two of this thesis, it is noted that the architectural influences of MOT extended from the AHAM, LAOS and LAG frameworks. MOT provides a web-based interface to create the necessary resources for adaptive hypermedia. The implementation of MOT was based on My English Teacher (MyET) [MyET] [Cristea et al. 2000] and on the extended LAOS model for authoring adaptation [Cristea & Aroyo, 2002]. The implementation of MOT contains three core layers; a *conceptual hierarchical layer* (of atomic and composite concepts, built of a number of attributes); a *lesson layer*, dealing with alternative presentation of contents at attribute level or above; and a third layer of *student adaptation and presentation adaptation*. MOT, however, only supports the Domain Model layer and Goals and Constraints Model layer of AHAM. [Cristea & Mooij, 2003a]

**MOT: Course Construction Methodology**

The MOT course construction methodology is illustrated in Figure 3-16 and describes the steps involved in creating a personalisable eLearning offering based on the LAOS model. Firstly the course developer must define, through the domain model, the concepts that are available in the course. Secondly, the course developer must define the relationships of the concepts in a goals and constraints model. Then an overlay of this is used to define the learner model characteristics. Penultimately, you define the rules which work with the learner model characteristics. Finally, the course developer defines the presentation format.
3.5.3.2.1 Building a Personalised Course using MOT

MOT is a web-based application which, based on the course construction methodology described in the preceding section, allows a course developer to perform the following operations

- Select/Create Concept Maps and/or Lessons
- View Concept Attributes
- Convert Concept Maps to Lessons
- Import other Concept Maps as Sub Concept Maps

Create Concept Map

The Concept Map forms the basis for all adaptive eLearning experiences produced using MOT. It encapsulates the conceptual information which represents a subject domain. It represents the foundation for the lesson layer which allows the lesson developer to create goals and constraints for the concept area. MOT provides access to previously created Concept Maps, as illustrated in Figure 3-17. A user can select and view any Concept Map which exists but can only edit those which they have initially created.
To create a Concept Map using MOT you simply name the new concept map and then add concepts and sub-concepts to the map, as illustrated in Figure 3-18. When creating a Concept Map, the developer can insert links to other concept maps, copy concepts from other concept maps or create a new independent concept map.
To build the concept map, the developer can simply start adding concepts into the root. For each concept the developer can define its dependent attribute set. These attributes include title, keywords, pattern, and so on as illustrated in Figure 3-19. The attributes can then be populated with appropriate information.
Figure 3-20 illustrates a partial Concept Map for the area of electromagnetism. It highlights the conceptual hierarchy of the subject area and the associated attributes of each concept in the hierarchy.
Convert Concept Map to Lesson

To create a custom Lesson using MOT, the developer firstly describes the Concept Map, as outlined in the preceding section. This Concept Map can then be converted to form the basis for a Lesson (Goals and Constraints). The lesson sequence, inherits directly from the hierarchical formation of the concept map, as illustrated in Figure 3-21. MOT provides the ability to change the ordering of the sub lessons by using the logical operators AND/OR. It also allows the developer to import pre-existing lessons.
The developer can also add weights and labels to the lessons and sub lessons, as illustrated in Figure 3-22. These weights can be used during the logical processing of the lesson structure.
Creating an Adaptive Strategy using MOT

Adaptive strategy in MOT refers to the process of adapting content and structure of a lesson to the characteristics defined in the user model. This implementation of the LAG adaptive language can create instantiated representations of specific learning styles such as field dependent learners [Silverman, 1987] for specific concept maps. Figure 3-23 illustrates the MOT environment for creating adaptive strategies. The condition-action rules specify the attribute of the rule (e.g. a concept from the Concept Map or an element from the User model), the logical operator and the value to be tested against. This rule evaluation only supports numerical values. There is no way to compare attribute values based on semantic reasoning.
3.5.3.3 Analysis

MOT is an application for authoring adaptive educational hypermedia systems based on the AHAM, LAOS and LAG models. It provides an interface for the educational developer to create Concept Maps to represent specific subject areas. It provides the ability to create lessons from these concept maps and then create the rules of an adaptive strategy for the lesson. MOT was analysed based on the core requirements for any personalised eLearning development environment, namely the educational support offered to the course developer, see section 3.5.

MOT provides scaffolding for pedagogical diversity based on a series of lessons. The course developer can define multiple lessons (foundations of the Goals and Constraints model) based on a single concept map. The hierarchy of the concept map provides the basis for the lesson structure. The course developer can then modify each lesson by adding weights and labels to concepts or concept groups within the lesson. This provides a flexible framework for extending conceptual representations of subject areas to form sequencing suggestions for diverse pedagogies. However, there is no internal support offered by MOT for the course developer in creating pedagogic diversity and no representation of Activity, a key element of constructive pedagogies.
Another level of pedagogic scaffolding involves metrics for supporting pedagogical reuse and repurposing. MOT supports the reuse and repurposing of existing concept maps and lesson structures by taking a community pooling approach to these resources. Since MOT is web based it stores all information models online so each user of the system can view and import existing resources, increasing the potential for peer collaboration in the design process.

MOT provides a successful implementation of the underlying design lifecycle from which it is based, see Figure 3-16. However, this design lifecycle, i.e. LAOS and LAG, is focused on the engineering of Adaptive Hypermedia Systems in general. The core focus of MOT, based on this, is not the course developer as a teacher but the course developer as an adaptive systems engineer. This technically oriented course construction methodology does not necessarily facilitate the modern requirements of educational development or constructive alignment. Although it allows the description of hierarchical concepts, the structuring of goals and constraints around those concepts and the creation of adaptive strategies for the resulting lessons, its course construction methodology does not support explicit alignment of assessment metrics with the stated learning outcomes and the teaching strategies defined.

MOT achieves adaptivity by allowing the course developer to write explicit condition-action rules around the modified concepts of a lesson and the overlaid properties of the user model. It provides an environment which facilitates the development of complex and powerful adaptive affects, see Figure 3-23. However, as can also be seen in Figure 3-23, the development of those rules requires a high level of technical input and a fundamental understanding of rules logic, a skill which the average course developer cannot be expected to have. A key point here is that similar to ASK-LDT, MOT does not support the definition of global strategies, i.e. adaptive strategies across complete lessons, although the LAG model does provision for it.

For the purpose of technical comparison, certain design aspects of MOT which affect its usability are now discussed. MOT is a web-based application which amalgamates elements of a form-based interface and elements of tree based visualisation. The interface provides mechanisms for creating, editing and selecting concept maps and lessons and provides an interface for creating rule sets. However, a key usability issue here is that tree based visualisations cannot sufficiently represent the
logical structuring of an ontology, in this case a concept map. MOT provides support for certain standards such as SCORM and LOM which can increase the potential for interoperability. MOT has also been used to convert lessons to WHURLE [Cristea et al. 2005] which demonstrates basic interoperability.

3.6 Conclusions

The state of the art in authoring of personalisation for education falls under three main categories, namely ITS Authoring, AEHS Authoring and IMS LD Authoring. Each of these areas contains significant benefits and also significant restrictions. Each of the systems has similar goals and objectives, but attempts to achieve these in diverse ways, as illustrated in the preceding sections. However, they have one fundamental thing in common. They have not been adopted as de facto authoring architectures or applications for producing personalised eLearning.

Each of the reviewed systems provides pedagogic scaffolding on some level. REDEEM provides scaffolding based on pedagogic terms and strategy definition, ASK-LDT provides scaffolding based on activity structuring and learning scenario creation and MOT provides scaffolding based conceptual sequencing. These different levels of scaffolding provide the course developer with some of the basic tools for creating “pedagogy”, however, their course construction methodologies do not provide holistic educational development and constructive alignment support to the course developer, i.e. the non-technical teacher. They typically provide interfaces to create the basic units of the underlying systems architectures. In this way, these systems are restricted by the limitations of the base architectural requirements of the underlying delivery mechanisms. For ITS, this typically points to the system specific implementations of the original system model consisting of user, teacher and content. For IMS LD, this indicates the co-dependence of the system with the underlying standard schema. For AEHS, this illustrates a dependence on the 5 layered model of LAOS. These restrictions, however, not only limit the flexibility of the systems, they also increase the complexities of the compositional process.

From this state of the art evaluation it is evident that for the successful adoption of personalised eLearning as an every day educational tool, significant changes must occur. Although personalisation can greatly benefit the learner, the educator, the content provider and the service
provider, several core issues have been identified. The development of successful personalised education typically requires expert input from the fields of adaptive systems engineering, pedagogy creation, subject area design, instructional development, curriculum alignment and learning content development. This team of experts can significantly increase the cost and time consumption of personalised eLearning development. By taking a knowledge and data engineering approach to personalised eLearning, several provisions for the resolution of these issues can be made. Information modelling and data mining techniques can provide a solid foundation for solving these issues. For example, Information modelling can extrapolate knowledge about the disparate design elements of personalised eLearning, namely, educational best practice, pedagogical approaches, instructional design principles, subject area descriptions, activity structures, personalisation mechanisms and learning resources. Data mining can extract personalised eLearning design patterns in order to formalise personalised eLearning compositional strategies and functional usability requirements.

Personalised eLearning Development Environments (PEDE), as described in this thesis, utilise the techniques and approaches of knowledge and data engineering to provide a pedagogically solid framework for composing personalised eLearning. This is a necessary requirement since the tools of today provide little or no support to the teacher when building personalised educational offerings. The ITS authoring tools allow the course developer to build specific instances of ITS based on many different paradigms. The IMS LD authoring tools allow the course developer to create instantiated Units of Learning. The AEHS authoring tools allow the course developer to create the basic model inputs for adaptive hypermedia systems where the user is the learner.

Apart from some of the technical issues identified within the reviewed systems, a fundamental requirement which will determine the success of personalised eLearning going forward is the support offered to the non-technical course developer, i.e. the teacher. This support must be based on eLearning best practices, educational composition methodologies, pedagogical strategies, instructional design principles and proven axes of personalisation. Personalised eLearning Development Environments which are built upon tested and proven course composition methodologies will ensure that pedagogy and associated support is the core driving factors in the development of personalised eLearning. Chapter 4 discusses the Adaptive Course Construction
Methodology (ACCM), which extends a proven course construction methodology by accommodating representations of the disparate design elements of personalised eLearning. Chapter 4 describes and illustrates the architectural design of Personalised eLearning Development Environments (PEDE) based on the processes of the Adaptive Course Construction Methodology (ACCM).
4 Design

4.1 Introduction

Although personalised eLearning can offer many tangible benefits to the entire educational process such as teacher and learner empowerment [Conlan et al. 2004] [Bajraktarevic et al. 2003], educational community collaboration and tailored eLearning delivered “just in time” and “just for you”, several issues restrict its mainstream appeal. These issues relate to the technical coordination of building instances of such systems, the complexities of developing and composing personalised eLearning offerings and time utilization in testing and appraising the produced personalised eLearning experience. This range of issues can be addressed by identifying and representing the core fundamental requirements of personalised eLearning development environments, personalised eLearning services and the disparate design elements involved with the personalised eLearning development process.

In order for current and future generations of personalised eLearning to improve educational effectiveness and efficiency, there are fundamental requirements that must be realised during their design, based on the discussions in chapter two and chapter three of this thesis. Firstly, the disparate design elements and information sources used to create personalisable eLearning must be identified, separated and modelled. Secondly, the personalised eLearning development environments must be technically transparent, standards independent and based on proven educational development methodologies. And finally the eLearning environment must be flexible in order to be durable, maintainable and extensible.

The types of information sources include pedagogic and instructional design, activities, subject areas, personalisation axes and learning resources as illustrated in Figure 4-1. When these information sources are modelled, detailed descriptions about what/when/where and how the information can and should be used is elicited. The models provide guidelines and scenario-based use cases to support the teacher in building constructive and effective eLearning experiences. Although several standards such as IMS Learning Design [IMS LD] attempt to model the
pedagogical process they do not inherently prescribe guidelines for how to use the pedagogy or provide scenarios for their use.

The usability of personalised eLearning development environments is typically designed towards a specific end delivery system or a specific set of standards [Sampson et al., 2005] as illustrated in chapter two and chapter three of this thesis. This inevitably leads to development environments that are either too system specific and technical to use or are too restricted by the constraints of the specific set of standards. Although some of these development environments allow the user to create rich adaptive hypermedia systems and eLearning offerings their appeal to the teaching community is limited since they are generally too complex (both technically and graphically), they do not provide a substantial range of pedagogical support to the course developer and they are too restrictive. Systems like My Online Teacher (MOT), ASK-LDT and REDEEM, as discussed in chapter three, assist in developing adaptive, adaptable and pedagogical online courseware, respectively, but are still too technically complex and do not offer enough support to the course developer. In comparison, the Adaptive Course Construction Toolkit (ACCT) [Dagger et al., 2004], an implementation of personalised eLearning development environments, presented as part of the contribution of this thesis, is designed and implemented to provide pedagogy, activity, subject matter, personalisation and learning resource based support to the course developer in addressing some of the key barriers to the mainstream adoption of personalised eLearning.

Learning environments have traditionally been developed as large monolithic systems. However, this has led to rigidity and high cost. Newer environments are attempting to be realised via Service Oriented Architectures. This is reflected in eLearning, for example, by the JISC suggested eLearning Framework [Wilson et al., 2004] and its underlying services. Emergent eLearning environments are moving towards service oriented architectures therefore we must be aware of this when authoring personalised eLearning to be offered as a service.
Figure 4-1 shows the teachers (in the role of course developers) interacting (A) with Personalised eLearning Development Environments to create adaptive course offerings, which may then be personalised towards their learners’ needs. This interaction involves access to a number of models used to support the authoring process. The primary of which is the pedagogical support model, which actively engages the teachers in the creation of pedagogically sound courses. Additional models, namely the subject area knowledge, personalisation axes, activities and learning resources, are also used in the creation of personalised course offerings. These offerings are transformed and loaded into a suitable Personalised Educational Management Service from where the learner’s may engage the personalised eLearning (B). The role of the teachers in this process is to support the learner in a mentoring/tutoring capacity during the learning process (C).

4.2 Influences from the State of the Art

Several factors from the state of the art analyses of chapters 2 and 3 have influenced the applied research of this thesis. These influences can be examined based on how they affect the core
elements of the personalised eLearning development environment being designed. The main elements of this system ensure-

- The course developer's support, both educationally and technically
- The identification, separation and representation of the disparate design elements of personalised eLearning
- The creation of pedagogically driven personalised eLearning designs
- The usability and flexibility of the personalised eLearning development environment

The specific implications of these influences on the design of the personalised eLearning development environments are discussed under the following headings; Educational Development and Course Construction Methodologies; and Authoring of Personalised eLearning.

4.2.1 Educational Development and Course Construction Methodologies

During the analysis of educational development and course construction methodologies in chapter three, several fundamental observations were made concerning the need for modernisation of educational development and the process oriented nature of course construction methodologies. Current educational development focuses at a "one size fits all" level, be it a certain grade of education or a certain skill set of employment. This group paradigm fails in realising the individuality of both learners and educators. By understanding constructivism [Bruner, 1960], provisions for the uniqueness of both educators and learners can be made. Educators need to be empowered to effectively and efficiently create pedagogically driven eLearning offerings while maintaining a mentoring/tutoring role in the realisation of this eLearning. Learners must be empowered to control the pace, mode and behaviour of the learning experience based on such things as their prior knowledge, objectives, aims and context. This balance of control is essential in producing successful personalised and personalisable eLearning.

Also discussed in chapter three was the process oriented nature of course construction methodologies for traditional educational development. The development steps of a proven course construction methodology can then form the base requirements for any personalised eLearning development environment regardless of underlying technologies and standards. So for example, a
tool which allows the course developer to; specify aims and learning outcomes; describe the subject area; define learning methods and activity sequences based on the defined aims, learning outcomes and subject area descriptions; associate personalisation axes with the learning methods and activities; select learning content and services; and test and deploy personalised eLearning designs on multiple diverse platforms can then fully support personalised educational development with curriculum alignment. This adaptive course construction methodology is described in section 4.3.

Based on these influences, personalised eLearning development environments must utilise the research and innovation in educational development and course construction methodologies to be accepted and used as tools of day to day education.

4.2.2 Authoring of Personalised eLearning

Based on the state of the art analysis in chapter two and chapter three, several influences to the design of this research can be attributed. These influences have one driving and substantial factor, the support offered to the course developer when constructing a personalised offering. The support influences are broken down to a finer level consisting of educational support and technical support.

The educational support provided to the course developer must be transparent. The course developer must feel educationally empowered, i.e. they must not feel that the system is dictating how something should be done but that the system is supporting them by providing best practice information and usage guidelines throughout all stages of course development. Within the course construction process, the most time consuming and difficult stages should receive the most detailed support by the personalised eLearning development environment. This indicates that the base architectural influence of the systems must be an educational development paradigm, something which the educators can relate their own teaching processes to. The resultant system must provide scaffolding for pedagogical diversity, i.e. be able to facilitate the support of multiple diverse pedagogical approaches. The system must also facilitate the support of subject area and activity creation/representation while supporting the discovery of appropriate learning content.

The technical support should provide a functional and usable graphical interface to allow the teacher to create and prototype personalised eLearning designs. This means that the system should
be able to both, utilise existing and produce new, design elements of personalised eLearning. The internal representation of these disparate design elements of personalised eLearning should be extensible and standards independent so as to maintain the true scope of personalised eLearning. This internal format, which can be standards-informed, can then be transformed into the specific required outputs for the delivery system of choice. The designed system must also actively support and facilitate reusability, not only at the content level but also at the strategic level, conceptual level, semantic level and knowledge level. This total reusability is a key enabling factor for true interoperability. The system must provide mechanisms for rapid prototyping of personalised eLearning designs allowing the course developer to affectively test their design in real-time, thus reducing overall development time and costs. This rapid prototyping actively supports the iterative process of a course construction methodology. For a designed system to be affective it must be usable. It must utilise visualisation and abstraction at all levels of the personalised eLearning design process to alleviate the cognitive and spatial overload sometimes experienced with less visual environments. Research in the area of adaptive educational hypermedia authoring patterns [Brown et al, 2005] could help to shape the usability of a personalised eLearning development environment by identifying patterns in the authoring process and customising the interface towards the patterns of the specific course developers. Utilising this valuable information could make it possible to personalise the PEDE to the individual course developer. A novel idea of dynamic personalised interface generation could create multiple diverse research streams.

Based on these influences, personalised eLearning development environments must be constructed on both an educational support and a technical support framework in order to be successfully adopted in general education.

4.3 Adaptive Course Construction Methodology (ACCM)

The adaptive course construction methodology, as illustrated in Figure 4-2, was designed to enhance traditional course construction methodologies with explicit support for; various pedagogical theories and instructional design principles; subject and service ontologies; personalisation axes and adaptivity techniques; and user modelling mechanisms. More specifically, the methodology should (i) facilitate the specification of different types of adaptivity to be embedded in the design e.g. adaptivity based on prior knowledge, context, etc., (ii) facilitate the
reuse and modification of one or more instructional designs, (iii) facilitate the identification of subject concepts and various services and (iv) encourage the reuse of content assets or model elements. The adaptive course construction methodology should support the course developer in identifying what parts of the course need to be adapted, and what criteria should be used for this adaptivity. For example, the course composer should be able to specify that the entire course be adaptable based on the learners' prior knowledge, but that specific activities (e.g. a discussion) should be based on the learners' preferred communication or collaboration style.

The modelling of pedagogical theory and instructional design principles provides valuable information to the course developer when choosing the appropriate learning methods and activity sequences to be used in their course. The subject area modelling provides information regarding the semantics (for example, conceptual granularity, inherent semantic flow, vocabulary and taxonomy) and the syntax (for example, metadata schema and format) of the subject domain of the course. This information, both semantic and syntactic, can provide the base requirements for generating learning content and service discovery queries [Lawless et. al, 2005]. The modelling of personalisation axes makes it feasible and affordable to use and reuse the mechanisms and techniques of adaptivity and personalisation to support the course developer. Through the techniques of user modelling, their characteristics, preferences, ePortfolio and context can be gathered and utilised in the course construction process to provide reasoning information to the personalisation axes to make decisions upon.
Based on the goals and objectives of this thesis and the influences from the state of art, several levels of requirements for successful personalised eLearning development environments have been identified. An architecture for PEDE, based on the adaptive course construction methodology and an extension to the multi model metadata driven approach to adaptive hypermedia [Conlan et al, 2002a], has also been defined.

An active design strategy throughout this thesis has been the consultation with real course developers on a number of issues from functionality to usability to expectations. Through formal and informal discussions with pedagogic designers, subject matter experts, instructional developers and content developers from both the Centre for Learning Technologies (CLT) at Trinity College Dublin and the IT Innovation Centre Skool team at Intel Ireland, a series of prototype components and models were designed and developed. This iterative design process was used to further refine the requirements for PEDE and is described in section 6.2.
The following sections describe the educational requirements, the technical requirements and the usability requirements for PEDE. Educational requirements refer to the support offered by the PEDE during the adaptive course composition process in the form of pedagogical support, activity support, personalisation support, subject matter support, learning resource support and rapid prototyping support. The technical requirements refer to the ability to import/export, create/customise and deploy/test the disparate design elements of personalised eLearning. Usability requirements illustrate the usability design guidelines and criteria for the development of successful and usable systems. The PEDE architecture section (4.4.4) describes and illustrates both the abstract and the technical architectures for PEDE based on the outlined requirements.

4.4.1 Educational Requirements

This section identifies key requirements for an educational course developer and identifies the failings of existing systems to address these issues. As one of the main critiques of existing adaptive educational hypermedia authoring and personalised eLearning development systems is their lack of support for the non-technical course developer, a core goal of this research is to illustrate the effectiveness of separating the disparate design elements of personalised eLearning and using this separated information to provide a federated educational support environment for designing personalised eLearning. This federated educational support must enable pedagogy construction, activity composition, personalisation identification and description, subject matter representation, learning resource search and selection and rapid prototyping of personalised eLearning designs.

The course developer must be pedagogically supported by the PEDE. This form of support must aid the development of pedagogically-driven personalised eLearning by describing, to the course developer, the process of creating effective eLearning pedagogy. This process and associated workflow can then be employed by the course developer when they are creating their own eLearning (both personalised and non-personalised) offerings. It provides the guidelines and usage scenarios for instantiating the pedagogical approach.

Based on the active categories of Bloom's taxonomy, pedagogies that are rich in activities provide a more engaging experience for the learner. This active approach to pedagogy must be supported by
the PEDE. The course developer must be supported with information, design guidelines and usage scenarios whether they are creating the own activities or customising existing activities.

During the creation of personalised eLearning a key source of information is the semantic knowledge about the subject area. The course developer, therefore, must be supported to both create effective subject matter concept space models and also to be able use, reuse and repurpose existing representations of subject matter areas. In doing so, the subject matter concept space must remain content independent (although candidacy can be used) to be truly reusable, durable and extensible. The main reasons for this is that subject concepts change less over time compared with content, which changes frequently and is updated regularly. Logical representations of subject area can then be used throughout different institutes, universities, corporations and countries making it a valuable and accessible source of information for providing support to the course developer within PEDE.

PEDE should facilitate the creation of both adaptive and non-adaptive courses. By supporting the course developer to understand the process of personalisation and simplifying the application of adaptivity to an eLearning design, they can more efficiently and effectively create personalisable eLearning designs. The support offered must be based on educationally sound personalisation axes, for example, prior knowledge, learning styles, learning outcomes and so on.

In order to create instantiated personalised eLearning designs, the course developer must be able to search for and select appropriate candidate resources. The PEDE must support the course developer through this task by providing facilities to search multiple local and remote repositories and select appropriate candidate resources for use within their eLearning design. The PEDE can aid the candidate discovery process by leveraging the existing knowledge in the personalised eLearning design to describe the “blue prints” of the candidate requirements [Dagger et al, 2003]. Through this process the PEDE should be able to dynamically generate semantic queries for harvesting candidate resources in order to support the learning resource selection process of the course developer [Lawless et. al, 2005].
In order to provide the course developer with the ability to create effective personalised eLearning designs, PEDE must support a rapid prototyping approach. This allows the course developer to iteratively test the semantics of the eLearning design in real time. This approach also supports the adaptive course construction methodology, as described in chapter three of this thesis.

The federated support environment offered by the PEDE must be educationally focused, more so than technically focused like the systems and diverse approaches to adaptive hypermedia authoring and personalisation authoring as demonstrated in the systems review of chapter two.

4.4.2 Technical Requirements

Based on the educational requirements and the compositional influences defined earlier, several core technical requirements for effective PEDE have been identified. For a PEDE to be useful, it must support the course developer with the ability to import and export all of the key elements of personalised eLearning, from pedagogy to activities to subject matter concept spaces to personalisation axes. The PEDE must also support the course developer with tools to create and customise the different design elements. As part of the rapid prototyping process, effective PEDE must also support the deployment and testing of personalised eLearning designs through remote and local services.

PEDE must have the ability to interact with local and remote services and also local and remote databases in order to access and share the essential knowledge contained in the different elements.

4.4.3 Usability Guidelines

A key criticism of current adaptive educational hypermedia authoring systems and personalisation construction systems is their usability. Although the systems are effective at producing a desired output [Conlan & Wade, 2004], the steep learning curve involved with understanding the technical and interface components can be restrictive. In order to create an effective and usable system there are three core principals of usability which must be acknowledged and understood [Shneiderman, 1998]. These guidelines have been applied to a rapid prototyping approach which led to the incremental refinement of a number of usability aspects of the PEDE, a process which is described in the evaluation chapter section 6.2.
Firstly, we must recognise the diversity of our user base [Shneiderman, 1998] [Hansen, 1971]. Although it is impossible to design a single system which meets every users complete requirements, identifying and characterising a user base can significantly reduce re-design time. Several ways of achieving this involve user profiling and analysis of authoring patterns. Also the accurate identification of tasks to be performed by the different users and the frequencies of those tasks can significantly reduce design time. In the case of PEDE, the user base in the non-technical course developer, i.e. the teacher.

Secondly, we should employ the “eight golden rules of interface design” [Shneiderman, 1998], namely:

- Consistency
  - Maintain reliable action sequences, terminology, colours, fonts and so on.
- Shortcuts
  - Reduce the number of interactions and increase the pace by using special keys, hidden commands, macros and so on.
- Informative Feedback
  - For every action there should be an informative response from the system which is subtle for minor actions and substantial for major actions.
- Design Dialogues to yield Closure
  - Using different levels of informative feedback during action sequences to give the user a feeling of accomplishment.
- Error Prevention and Simple Error Handling
  - Design the system to limit the potential for user mistakes and if such mistakes occur the system should recognise this and offer informative and constructive feedback for recovering.
- Easy Reversal of Actions
  - Where possible, provide functionality so that users can undo/redo actions automatically.
- User Empowerment
  - Where all levels of system user, from novice to advanced, feel as though they are in total control over the operation of the system instead of just guiding the system to
perform tasks. This entails a delicate balance of manual and automatic action sequences.

- Reduce Short Term Memory Load
  - Provide action sequences with no more than seven plus or minus two actions. Ideally, an action sequence should consist of no more than four actions in order to keep the user focused.

And finally, "prevention is better than cure"; prevent errors from happening with intuitive design [Shneiderman, 1998]. A significant loss of productivity can be attributed to user error [Brown & Gould, 1987]. People of all skill levels make mistakes. Therefore system developers should strive to design intuitive and useful systems which actively prevent user error. Three techniques can reduce errors by ensuring complete and correct actions:

- Correct Matching Pairs
  - Provides a syntax checking mechanism for determining correctness

- Complete Sequences
  - Provide a collection of automatic and manual action sequences, allowing the user to create action sequences based on atomic and composition actions through macros and programmability.

- Correct Commands
  - Provide canned commands (macros) with usage instructions to prevent their misuse.

The implementation of these guidelines in PEDE can lead to the creation of affective, efficient and user friendly systems.

4.4.4 PEDE Architecture

An architecture for personalised eLearning development environments (PEDE) has been designed which proposes course developer empowerment through an adaptive course composition support framework. This framework places the course developer in the central role and supports them in the process of creating personalised eLearning designs with information, guidelines and instantiations of the disparate design elements of personalised eLearning such as pedagogy (Narrative
Structures, activities, subject area (Subject Matter Concept Space), personalisation axes (Narrative Attributes), learning resources and various other sources of information, as illustrated in Figure 4-3. By providing sufficient tools for accessing, composing and customising these different elements, a technologically-sound support environment can be provided.

The technical architecture, illustrated in Figure 4-4, describes personalised eLearning development environments which provide access to the key elements of personalised eLearning via service interfaces and supports the course developer with a set of tools to interact with, customise and create each of the different key elements of personalised eLearning design. It also indicates the ability to publish personalised eLearning designs as "course packages". As depicted in Figure 4-4, PEDE should have the ability to rapidly prototype, deploy and test personalised eLearning designs, through web service interfaces, in associated personalised eLearning services.

An example scenario based on Figure 4-4 could be; the course developer uses the PEDE tools to identify and customise an existing pedagogical model (Narrative Structure) to form the base of their personalised eLearning design. This Narrative Structure, based on the WebQuest pedagogical theory, describes a sequence of conceptual activities. The course developer further customises their design by selecting and customising an existing Activity from the remote collection of predefined activity structures. The next step the course developer takes is to define the subject area for their eLearning design, say they are teaching “Electricity and Electromagnetism”. The course developer now uses the PEDE tools to create a new Subject Matter Concept Space for the subject area. The course developer can then use these “subject” descriptions to contextualise their eLearning design. The next step the course developer takes is to search for and select appropriate learning resources to help teach their design, a step which is fully supported by the PEDE tools. Penultimately, the course developer identifies and selects, from the remote collection of Narrative Attributes, the types of personalisation they want to apply to their design. The PEDE then supports the application of this personalisation to their design. Finally, the course developer tests their personalised eLearning design through a number of existing personalised eLearning services, functionality which the PEDE fully supports.
The following sections describe in detail the key design elements to be supported by personalised eLearning development environments, how these diverse information sources can be modelled and the possible standards that can be used to mark them up in.

4.5 Key Design Elements of Personalised eLearning

As previously described, there are a number of key design elements of personalised eLearning development, namely, pedagogies, activities, subject areas, personalisation axes, learning resources and learner profiles.

Narrative Structures

A pedagogic strategy usually consists of an arranged sequence of conceptual tasks and activities that need to be performed. This workflow is usually accompanied with best practice principals and use case guidelines to illustrate the maximum potential benefit offered by the strategy. Through a pedagogical modelling mechanism we can create an accessible and flexible instance of the pedagogy, called a Narrative Structure model. The model contains descriptive and usage
information for each of the high level concepts/activities of the pedagogical strategy and suggests a possible sequencing of these abstract elements. Typically, pedagogical strategies can be represented as a series of high-level descriptive concepts representing learning activities to be undertaken. This is further explained in APPENDIX VI - Key Design Elements of Personalised eLearning.

Activities
Activities typically consist of some form of task(s), associated tools which could be used to perform the task(s), a description of roles within the task and appropriate learning content. Activities require some intelligent sequencing of operations. This sequencing describes the workflow between sub-activities. These elements and attributes of activities can be modelled to produce an accessible and reusable form of the knowledge incorporated in that activity, called the Activity model.

Subject Matter Concept Space
One of the key sources of information in any educational experience is knowledge about the subject area. This information contains conceptual and relational descriptions about the subject area. It provides scope to the personalised eLearning experience, it provides knowledge to the personalisation axes and it supports the course developer when designing personalised eLearning offerings. However, an influential factor in the successful representation of subject area is the use of commonly understood vocabularies and semantics to describe the domain. There are certain standards which actively promote the use of common vocabularies in describing a subject area. For example, the Library of Congress Subject Headings (LCSH) 28 [LCSH] provides an up to date and comprehensive list of subject headings which could provide a solid and distinct taxonomy for describing subject area information.

Narrative Attributes
A Narrative Attribute is used to describe a specific axis of personalisation and an associated collection of adaptive techniques. It also identifies the different information sources which it operates across, such as learner and subject area. The adaptive techniques are typically in the form of “selectors”. A selector, as the name states, performs a selection operation based on a number of different inputs, outputs, preconditions and effects.
Learning Resources

The information stored in a learning resource’s metadata can be used by personalised eLearning development environments to identify similarities between it and the information stored in the personalised eLearning designs. For example, if the metadata for a piece of content describes it as being: i) pedagogically neutral, ii) a certain technology type such as Flash or HTML, iii) a certain modality type such as visual or aural, and iv) not copyright protected; then it may be offered as appropriate candidate content for a personalised course using a WebQuest pedagogy, designed for diverse VARK [VARK] learners in a university curriculum. So the metadata of a learning resource can support the process of selecting appropriate content. The inverse of this is also true, meaning that a personalised eLearning design provides the blue prints for the content types which it requires to fulfil its purpose. In this way, it can provide the semantic information required in order to source appropriate learning content.

Learner Profiles

A personalised eLearning design provides a rich source of information regarding the types of personalisation required. Through analysing the applied personalisation in a personalised eLearning design, the base set of learner requirements can be gathered. For example, in a personalised eLearning design teaching electromagnetism through a WebQuest pedagogical theory might apply personalisation based on prior knowledge to the “Introduction” section. Now let’s assume that this section contains three concepts, “conduction of electricity”, “magnetism” and “circuits”. This information now describes a scenario where the engaging learner requires knowledge pertaining to the “prerequisite” requirements of these concepts, based on their specification in the subject area description, in order for them to be appropriate. Inferring across this scenario, we can produce requirements for a subset of learner information based on the “concepts” that are being personalised in the personalised eLearning design. A mechanism for storing this learner information is ePortfolios [ePortfolio].
Personalised eLearning Designs (PEDE Narrative)

When the course developer uses the PEDE to build an educational experience, the result is a "personalised eLearning design" [iii]. The personalised eLearning design contains information about the course developers chosen and customised pedagogy, activity types, subject area, personalisation axes, selected candidate learning resources and any other information sources used. It is therefore an organised collection of the key elements of personalised eLearning, called a PEDE Narrative.

4.6 Modelling the Disparate Design Elements of Personalised eLearning

A core goal of this research based on the research question of chapter one, "to identify and investigate the key extensible design elements, models and influences required for the composition of personalised eLearning experiences and to develop an integrated, model driven environment which facilitates rapid personalised eLearning development by educators or teachers (non-technical)", is to provide mechanisms for capturing, modelling and representing the disparate design elements of personalised eLearning. The following sections provide detailed descriptions of the capturing and modelling process applied to the disparate design elements of personalised eLearning. They also illustrate and describe the schema used to represent this captured and modelled information.

4.6.1 Narrative Structures (Pedagogical Models)

Pedagogy is a fundamental information source used in building successful educational offerings. It provides the structure and the guidance to support educational development. Therefore it is essential that this vital information be captured and used in meaningful ways, such as reusing, repurposing and general sharing. The capturing of this information is made possible by appropriate modelling mechanisms.

Figure 4-5 illustrates the logical flow of information within a process for modelling pedagogy. Typically the high level and abstract design theory and principles provide the theoretical information for the particular pedagogy [Mayes & de Freitas, 2004]. This descriptive detail

[iii] By Personalised eLearning Design we actually mean the physical model which contains the description of the course developer's composition.
pertaining to the psychological process instilled in the pedagogical approach is then used to create guidelines and best practice information for using the pedagogical approach. This step identifies a number of concepts and activities to be performed and the types of scenario to which the pedagogical strategy could be applied. Following this, the strategical sequencing of the concepts and activities is described. This strategical sequencing provides potential course developers with solid and detailed information on how best to sequence the concepts and activities in order to achieve the pedagogical goals of the strategy.

![Logical Narrative Structure (Pedagogical) Modelling Process](image)

In order to successfully represent pedagogical strategies, a model schema was designed to capture the essential information, as illustrated in Figure 4-6. The core element of this schema is the “concept”. More specifically, the “concept” is of “type” “narrative”. A Narrative Structure is therefore a collection of “narrative” concepts. The “narrative” concepts could represent a number of pedagogical elements such as an activity or an event. Take for example the WebQuest pedagogical approach. When this is represented as a Narrative Structure, each of the WebQuest’s sub elements (concepts), such as “introduction”, “task” or “process”, would be represented as “narrative”
The "relationships" element of "narrative" concepts facilitates the description of how one concept relates to other concepts within the Narrative Structure and supports the definition of sequencing requirements, i.e. the workflow of the pedagogy. This can be used to create hierarchical relationships, ontological relationships, graphical relationships and entity relationships. This provides the basic support for the semantic sequencing of the pedagogical elements and can be utilised by the pedagogy's use case descriptions. To clarify this, take the WebQuest pedagogy for example. The flow between the "introduction", "task" and "process" "narrative" "concepts" can be described using the "relationships" element, provisioning for sequence information. Therefore, a Narrative Structure represents both the various elements and activities of a pedagogical approach and the workflow and sequencing possibilities of various elements, in a structured way.

The "id" attribute of a "narrative" "concept" is a globally distinctive identifier used to classify the concept. The approach of using a globally distinct identifier facilitates the customisation of the narrative structure. It also supports the blending of pedagogies. For example, if a blended Narrative Structure was created to represent the overall approach of a case-based pedagogy with elements of this making use of the WebQuest pedagogy, then the "introduction" concepts of both pedagogies, although different, can be distinguished and understood uniquely.

The "concept" element has several child elements, namely:
- "name"
- "description"
- "relationships"
- "candidates"

Each "concept" element can contain multiple child "concept" elements facilitating the construction of diverse data structures such as trees, graphs and lists.
The “name” element represents the generic name of the pedagogical element being modelled, for example Didactic Introduction, WebQuest Introduction or Case-Based Problem.

The “description” element facilitates the ability to provide a high level description of what the pedagogical element is and what is tries to achieve. It forms the basic support framework available to the course developer and is also used to provide the sample usage guidelines for both the pedagogy and the elements of the pedagogy.

The “candidates” element, as illustrated in Figure 4-6, facilitates the support of the architecture for candidacy [Dagger et al, 2003] allowing candidate concepts, resources, activities, etc. to be associated with the core concept. Candidacy provides a level of abstraction which allows the
personalised eLearning design to remain content independent while at the same time supporting the instantiation of the pedagogy. This approach also supports the diverse reuse of narrative structures in that the personalised eLearning designs can not only be shared with the base “narrative” and the instantiated “subjectmatter” concepts but also with reference to any of the potential candidates that may have been associated. Modelling pedagogy in the way described in this thesis clearly supports its reuse on three distinct levels.

4.6.2 Activity Models

Activities typically consist of some form of task(s), associated tools which could be used to perform the task(s), a description of roles within the task and appropriate learning content. The purpose of activities originates from an identified need to address the higher order cognitive skills of learning such as application, analysis, synthesis and evaluation, as described in Bloom’s taxonomy, see APPENDIX I – Psychology and Cognition. Figure 4-7 illustrates the different categories of activities such as high-level composite activities, low-level composite activities and atomic activities which in some way address this active approach to knowledge acquisition and retention.

![Figure 4-7, Logical Learning Activity Modelling Process](image-url)
Atomic activities describe the simplest form of activity, for example, email, discussion, chat and instant messaging. For each of these atomic activities there are associated roles, conditions and communication tools and mediums available such as web forums for chat and email clients for email and so on. On their own they provide little support for educational process except for casual and social learning. Low-level composite activities take these simple atomic activities and wrap them with “light” strategies. For example, as part of the “Posting Assignments” low-level composite activity, the learner may be asked to “please use either the submission form or your favourite email client to post your assignments”. In this case the activity relates to posting an assignment and the strategy relates to using the appropriate submission medium as part of an exercise. High-level composite activities build upon the initial layers of atomic and low-level composite activities by allowing larger and more detailed strategies to be applied to each layer. For example, as part of the “Peer Review” high-level composite activity, the learner may be asked to perform the peer review process, as illustrated in Figure 4-8, of

1. Submission
2. Technical Corrections
3. Open Discussion
4. Final Response
5. Submit Revised Work
6. Peer Review Completion
7. Publication and Conclusion

The workflow of this activity would suggest that the learner must first submit a piece of work to be reviewed. Following on from this, the role of reviewer is to perform some technical corrections. When the technical corrections are completed and the author notified, a discussion session is carried out. When the final response from all reviewers has been explained, the author must submit their final version. Once submitted the appropriate completion process is enabled and the submitted piece is then published. At each of the steps in the workflow, a series of low-level composite and atomic activities are employed to fulfil the task. In this way, modelling of activities as described in this thesis uses a layered approach to representing and aggregating the elements of the activity.
The research carried out in this thesis has created a schema for representing modelled learning activities. As illustrated in Figure 4-9, the core element of this schema is the "activities" element. This provides a storage mechanism which can be used to represent atomic, low-level composite and high-level composite activities.

The sole child of this element is the "activity" element which is then used to actually model an atomic activity. This means that the "activities" element can have one or many "activity" children. The "type" attribute is used to define the type of activity, be it a learning activity or a support activity.

The "activity" element contains several child elements, namely:

- "name"
- "aggregation_type"
- "description"
- "concept"
The “name” element is used to store the logical name of activity being modelled. The “aggregation_type” element describes whether the activity is a composite or an atomic activity. The “description” element allows the learning activity developer to provide a detailed description of the activity and also provide some usage and potential sequencing information. This provides the basic support provision for the course developer with respect to learning activities. The “concept” element is identical to that used to model a Narrative Structure, i.e. it is of type “narrative”. It provides the course developer with the ability to add pedagogical information to the activity. It also provides another level of support to course developers by making this pedagogical information explicit. The “communications” element is used to describe the service requirements of this activity, for example, if the activity requires an email service, this is where that service requirement description would be placed.

```
<xsd:element name="activities">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="activity" minOccurs="1" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1" />
            <xsd:element name="aggregation_type" type="xsd:string" minOccurs="1" maxOccurs="1" />
            <xsd:element name="description" type="xsd:string" minOccurs="1" maxOccurs="1" />
            <xsd:element name="concept" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="communications" minOccurs="1" maxOccurs="1" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

![Figure 4-9, XML Schema for Learning Activities](image)

Since activities can be composite or atomic, the “activity” child element supports the nesting of activities, hence the ability to create low-level and high-level composite activities.
4.6.3 Subject Matter Concept Space (Subject Area Models)

The information regarding subject area is a key source in creating both personalised and non-personalised eLearning designs. By representing the knowledge of a subject area in model form, we can then reason about it and use it in a systematic way. As illustrated in Figure 4-10, the concept space consists of a collection of concepts and associated relationships which define and describe their co-dependence. The different relationship types can produce various information representations. In Figure 4-10, the “Root Concept” “CONTAINS” both “Concept A” and “Concept B”. Both “Concept A” and “Concept B” then contain a number of concepts, respectively, and so on. Therefore, the “CONTAINS” relationship is used to describe a hierarchical structuring of concepts. Also note that Figure 4-10 has a number of “PREREQUISITE” relationships. These are used to describe the competence dependencies between concepts. For example, concept “M” is a “PREREQUISITE” of concept “N”, therefore this describes a situation where the knowledge represented by concept “M” is a prerequisite requirement for the knowledge represented by concept “N”.

![Figure 4-10, Logical Concept Space Modelling Process](image-url)
One of the key issues here, as with all attempts at knowledge representation, is the applied granularity. Granularly coarse concepts are both difficult to extract and reuse and provide limited reasoning capabilities with regard to associated relationships. On the other hand, granularly fine concepts are easier to extract and reuse and provide greater reasoning capabilities. However, they are typically more time consuming to create.

In order to model knowledge representation scenarios, as depicted in Figure 4-10, a schema for representing a subject matter concept space was developed, as illustrated in Figure 4-11. This schema is similar to that used to represent Narrative Structures in that the root element is a "concept". The distinction between the two, however, is made during initiation; meaning that all concepts used in a subject matter concept space are "subjectmatter" type concepts. This distinction is facilitated through the concepts "type" attribute. The semantic relationships of a subject area, as illustrate in Figure 4-10, can then be modelled through use of the "relationships" element which supports the description of multiple directional and diverse relationships.
4.6.4 Narrative Attributes (Personalisation Axes Models)

Personalisation is a strategically driven process of reasoning about various information sources such as subject area, learning goals, teaching objectives and learning environment to adapt an educational experience. Figure 4-12 illustrates the logical flow of information within a process for modelling axes of personalisation (Narrative Attributes). In its simplest form, it takes several information sources as inputs, makes some decisions based on those inputs and the strategy which is driving it and then produces some useful and meaningful outputs. This technical view of personalisation axes is quite similar to the applied process model of web services, i.e. consisting of Inputs, Outputs, Preconditions and Effects (IOPE). Applied to personalisation axes; the inputs represent any information which can be utilised in the reasoning process (subject matter
information, learner characteristics and so on); the outputs represent the personalised output of the axes which can be for example, filtered input information or newly constructed information (learning activity structures, personalised hyperspace and so on); the preconditions are used to describe the requirements that must be met prior to the execution and completion of personalisation axes (learner’s competence level, educator’s scoping parameters, and so on); and the effects describe the changes made to the environment by executing the personalisation axes (increase in learner’s knowledge level, modification of hyperspace, and so on). In some cases, however, both the preconditions and inputs and the effects and outputs are almost identical, so they do not need to be represented individually.

In order to successfully represent personalisation axes (Narrative Attributes), a model schema was developed as illustrated in Figure 4-13. The root element of a Narrative Attribute is the “narrative_attribute”. The “narrative_attribute” contains several child elements, namely:

- “name”
- “type”
The "name" element facilitates the logical naming of the Narrative Attribute. The "type" element is used to describe the personalisation type represented by the Narrative Attribute. For example:

- "adaptiveaxes.actor.knowledgespaces"

indicates that this Narrative Attribute uses actor-focused (learner) adaptivity based on the theory of knowledge spaces [Doignon & Falmagne, 1999] learning style.
The “description” element forms the basic unit of course developer support by providing a mechanism for describing what the Narrative Attribute is and what it is trying to achieve.
The "parameters" element supports the specification of required inputs for the Narrative Attribute. For example, a Narrative Attribute for personalisation based on Prior Knowledge requires two inputs to function,

- the competencies required to proceed to the associated concept. These can be gathered from the subject matter concept space by reasoning across concept relationships.
- the competencies learned so far. These can be gathered from the learner's profile based on what the learner has learned so far.

This is used to specify the "inputs" as described by the IOPE process model of web services.

The "relationships" element is identical to that used in the all other schemas, which supports the creation and definition of complex relationships between the various Narrative Attributes.

The "candidates" element of a Narrative Attribute is used to specify the location of the candidate selectors which represent the candidate Adaptive Techniques for the required personalisation axis. Candidate selectors can differ, for example, in return types, i.e. one selector can return a list format whereas a different selector may return a model format. This information can then be used during the reconciliation of the Narrative Attribute for a specific personalised eLearning design.

4.6.5 Candidate Learning Resource Models

Candidacy [Dagger et al, 2003] as applied to learning resources facilitates the logical grouping of "similar" learning resources based on various different categories such as learning styles similarities, prerequisite similarities or technical similarities. From an authoring perspective, candidacy can support the course developer in both sourcing and selecting the most appropriate learning resources to satisfy their personalised eLearning design. These abstraction mechanisms, as illustrated in Figure 4-14, which are employed by candidacy also allow the course author to design the course in a more structured way without necessarily being concerned with the individual pieces of content that are used to realise their personalised eLearning design.
In Figure 4-14, the candidate resource group has two candidates, A and B. Learning resources in the same candidate content group are equivalent on some axis, usually the concept they teach. Conceptually, candidate groups can be formed for any set of the key elements of personalised eLearning, such as pedagogy, activities, subject area, personalisation axes and so on. For example, we could have candidate narrative structures for diverse pedagogical approaches, candidate narrative attributes which could perform the same type of adaptivity but produce different outputs such as a list or a model and candidate activities which perform similar tasks using different activity types such as submission using email or web form.

```xml
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="candidate_resource.xsd" xmlns="candidate_resource.xsd" elementFormDefault="qualified">
  <xsd:element name="candidate_resource">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="metadata_location" type="xsd:string" minOccurs="1" />
        <xsd:element name="candidate_location" type="xsd:string" minOccurs="1" />
        <xsd:element name="candidate_title" type="xsd:string" minOccurs="1" />
        <xsd:element name="candidate_description" type="xsd:string" minOccurs="1" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Figure 4-15, Candidate Resource Schema
Based on this, a schema for representing candidate resources was defined, as illustrated in Figure 4-15. The core element of this schema is the “candidate_resource” which contains several elements as follows:

- “metadata_location”
- “candidate_location”
- “candidate_title”
- “candidate_description”

The “metadata_location” element is used to point to the location at which the metadata can be accessed from. This can be anything from a database location to a service interface.

The “candidate_location” element indicates the physical location of the candidate. This location may be local (stored in a file system) or remote (resource repository or service interface).

The “candidate_title” indicates the original name of the candidate resource. The “candidate_description” is used to describe what this candidate resource is.

4.6.6 PEDE Narrative (Personalised eLearning Design Models)

PEDE narrative is the embodiment of the course developer’s use of chosen pedagogy(s), defined activity types, specified subject areas, associated personalisation axes, selected candidate learning resources and any other information sources used to produce personalised eLearning. Therefore it is an amalgamation or collection of all previously described models.

4.7 Influences of Standards and Technologies on PEDE

Several possible standards and technologies exist which could be used to represent the key elements of personalised eLearning development. Although PEDE should be aware of these various standards and technologies, their internal representations should not be limited by the capabilities of the current standards and technologies and should exist independently from them. Therefore, the PEDE internally represents the key elements of personalised eLearning development in a standards-independent XML format. However, the internally represented format can be translated into some
of the various standards and technologies which exist. The following sections outline and discuss those candidate standards and technologies in relation to the key elements of personalised eLearning.

4.7.1 Representing Narrative Structures (Pedagogy)

There are several candidate metadata standards for representing pedagogy (in that they can represent some form of sequencing) such as IMS LD (Learning Design) [IMS LD], IMS SS (Simple Sequencing) [IMS SS], ebXML (eBusiness Extensible Mark-up Language) [ebXML], BPML (Business Process Modelling Language) [BPML] and so on. Systems such as ASK-LDT which are built primarily on these standards are technically complex and are not educationally driven. However, to ignore these standards would also be incorrect. Being “standards compliant” can lead to greater public recognition and can support accessibility since communities are speaking the same “language”. In situations where communities differ, syntax mapping and transformation mechanisms can bridge the gaps. Following on from this, the approach taken in this thesis supports a standards-independent internal representation for the key elements of personalised eLearning development. This standards-independent representation can then be transformed into a number of diverse standards, such as IMS LD and IMS SS. However, during this transformation process, certain capabilities of the personalised eLearning design are lost. This occurs because we are mapping a powerful representation to a restrictive and specifically-purposed standard.

Using a standards-independent internal representation for pedagogy is a more flexible and durable approach since it does not need to be updated as the version of the standards change and provides a richer mechanism for representing diverse pedagogical approaches. The internal form which is used is standard XML. By identifying mappings between these descriptions we can then translate the internal form into the desired output as described in section 5.3.7.

4.7.2 Representing Subject Matter Concept Space

Several candidates for representing a subject matter concept space also exist, namely the Web Ontology Language (OWL) and Entity Relationship Modelling. Although these standards could sufficiently support the representation of a subject matter concept space they are very complex and time consuming to implement. The SMCS is a light weight ontology with a specific purpose. This standards independent form can be transformed into the existing standards and vice versa, meaning
the existing standards can be transformed into an SMCS. In this case however, certain meaning may be lost in the translation to the SMCS.

A standards-independent internal representation of a subject area provides a more flexible, durable and extensible approach since it is not directly linked to any existing standards which are subject to change and which can restrict the potential information that can be represented. The internal form which is used is standard XML.

4.7.3 Representing Narrative Attributes (Personalisation Axes)

Several candidates exist for representing personalisation axes, namely Web Ontology Language for Services (OWL-S), Web Service Description Language (WSDL), Rule Mark-up Language (RuleML) and Adaptive Hypermedia Languages. As previously described, Narrative Attributes share similarities with the process model of web services in that they have inputs, outputs, preconditions and effects. However, this approach would only provision for the description of the Narrative Attribute and the ability to make it accessible. The internal workings of Narrative Attributes use rules logic to perform some reconciliation on the inputs to produce the required outputs. This rules logic currently takes the form of Java Expert System Shell (JESS) [JESS]. However, multiple different languages as described in section 2.5 could be used to perform the same tasks.

Although this thesis proposes JESS to represent the processing of a Narrative Attribute (i.e. its candidate selectors), a standards-independent internal representation of Narrative Attributes provides a flexible, durable and extensible approach since its independence from existing standards and languages facilitates a lossless representation of personalisation, i.e. the metadata of personalisation axes.

4.7.4 Representing Learning Resource Candidates

Several candidates exist for representing learning resources such as IMS Learning Resource Metadata (LRM), IEEE Learning Object Metadata (LOM), SCORM and Dublin Core. These standards are focused on either representing the atomic assets of a learning resource or the learning resource itself. However, as previously described, we are not describing individual resources; we are representing candidate resources which existing standards do not cater for.
Based on this, a standards-independent internal representation for learning resource candidates provides a flexible, durable and extensible approach since its independence from existing learning resource standards and languages facilitates a lossless representation of learning resource candidacy.

4.7.5 Representing Course Packages

There is only one existing candidate for representing course packages, namely IMS Content Packaging (CP). This has also been implemented in other standards such as IMS LD and reference models such as SCORM. However, this standard is for representing learning content and not the information models which describe personalised eLearning, i.e. the key elements of personalised eLearning development.

Based on this fact a standards-independent internal representation of a course package was chosen. This provides flexibility in representing and packaging the key elements of personalised eLearning in a manner which ensures that the detail of a pedagogically-driven eLearning experience is not lost.

4.7.6 Representing PEDE Narrative (Personalised eLearning Designs)

Existing candidates for representing personalised eLearning designs are IMS Learning Design (LD) and IMS Simple Sequencing (SS). However, as with Narrative Structures, these candidate standards fall short of the requirements for representing personalised eLearning designs.

Therefore a standards-independent internal representation of personalised eLearning designs was chosen which provides a flexible, durable and extensible approach since its independence from existing standards facilitates a lossless representation of personalised eLearning.

4.8 Summary

Chapter 4 discussed the design of personalised eLearning development environments based on the key elements of personalised eLearning development and the Adaptive Course Construction Methodology. It has demonstrated that the key elements of personalised eLearning development can be modelled to both maximise the support for developing personalised eLearning and maximise
the reuse of the vital information contained in those key elements. This chapter also described the various factors which influence both the capturing of and the modelling of the different design elements. For all models involved, issues such as semantic granularity and technical granularity were detailed and discussed. Introduced here were the requirements for personalised eLearning development environments such as educational, technical and usability requirements. Also introduced here was an architecture for personalised eLearning development environments. This chapter identified and described the potential candidate standards and representations for the key elements and discusses them individually. The chapter finished by addressing the various standards and technologies for potential use in personalised eLearning development environments.
5 Implementation

5.1 Introduction

Chapter four described the design of personalised eLearning development environments (PEDE) based on the representation of the disparate design elements of personalised eLearning and the Adaptive Course Construction Methodology (ACCM). Chapter 5 now discusses the Adaptive Course Construction Toolkit (ACCT), an implementation of personalised eLearning development environments (PEDE) which provides a suite of tools to support non-technical and technical course developers in composing personalised eLearning designs.

This chapter describes the implementation process through a series of sections focusing on the architecture of the ACCT, the models of the ACCT and the components of the ACCT. To echo the course development process of the state of the art of authoring tools for adaptivity, this chapter describes a series of steps towards creating, testing and deploying a personalised eLearning design with the ACCT. Described herein are the various levels of reusability supported by the ACCT, i.e. reusability of all the disparate design elements of personalised eLearning. This chapter also describes the process of outputting personalised eLearning created using the ACCT for different personalisation architectures such as IMS LD.

5.2 Architecture of the ACCT

The core components of the Adaptive Course Construction Methodology should be implemented by a Personalised eLearning Development Environment to support educational requirements. Based on the complexity and importance of producing pedagogically sound eLearning, this implementation should at least provide the ability to represent the subject area, select appropriate pedagogies and activity sequences, select appropriate content and services, associate adaptivity with conceptual course structures, test and deploy personalised eLearning designs and provide mechanisms for modelling learners.

The architecture of the Adaptive Course Construction Toolkit (ACCT), as illustrated in Figure 5-1, facilitates and supports the access, the composition and the customisation of the disparate design
elements of personalised eLearning. In addition, provisions for accessing, composing and customising other information sources are made. Since PEDE and the ACCT extend the research of the multi-model approach to adaptive hypermedia, they must facilitate the support of multiple and diverse information sources. Basically, *if it can be modelled it can be used*. The ACCT achieves this by basing its information structure on XML; remaining standards-independent and standards-informed by internally representing the information models in a standards-independent form; and by using a componentised interface design.

Based on topological advancements in web technology, access to the various information sources is facilitated through web service architectures while at the same time supporting the use of locally stored information. In this way the models can be stored locally in a file-system or database or stored in a remote repository.

The basic transaction unit of the ACCT is XML. It therefore provides access mechanisms for connecting to local and remote XML data stores and for retrieving XML from them. Access is provided through SOAP/XMLRPC connections to the XML DataBase (XMLDB) [XML DB] where the local XMLDB Application Program Interface (API) can be used to perform a range on actions on the stored data based on XPath, XQuery and native commands.

The ACCT uses a virtual workspace for storing “in-progress” models. This, as the name states, provides local temporary storage for models which the course developer is currently designing, as illustrated in Figure 5-1.
The architecture of the ACCT illustrates the ability to publish personalised eLearning designs in a number of ways. Firstly, the customised models which the course developer has created can be exported as a course package. The manifest of this course package describes the contents, which is typically the PEDE narrative, the SMCS and customised relationship definitions, search histories, the narrative rules and any other customised or created models. Secondly, the ACCT supports the publication of personalised eLearning designs to a generic version of the Adaptive Personalised eLearning Service (gAPeLS). The ACCT also supports the publication of personalised eLearning designs to existing eLearning standards such as IMS LD and IMS SS. The publication processes of the ACCT are detailed later in this chapter in sections entitled Model Translator/Interpreter and Publishing the Personalised eLearning Service.

5.3 Components of the ACCT

The ACCT is implemented on a component-based architecture. Several components have been created as part of the ACCT, namely, the course package, the workspace, the import/export facility, the subject matter concept space builder, the narrative builder, the repository search, the model
translator/interpreter and the personalised eLearning design publisher. These components, mainly implemented in Javax Swing, are now described in detail.

### 5.3.1 Course Package

An ACCT course package is used to store and move personalised eLearning designs and their associated models. It is a collection of all the models used by the course developer during the process of developing their personalisable course. Each course package contains a “manifest” file which describes the contents of the package. Figure 5-2 illustrates a sample course package manifest. Here we can see that the package contains references to many models, the models which the course developer has used to create the personalised eLearning design, namely the concept space, the PEDE narrative, the customised relationship definitions, the learning resource search results, and so on. This and other sample course packages can be found in APPENDIX II - Models, Schema and Transforms.

```
<manifest>
  <file type="narrative_concept_space" uri="acct_distribution_course_skool_narrative.xml"/>
  <file type="narrative_rules" uri="acct_distribution_course_skool_narrative_rules.xml"/>
  <file type="subject_matter_concept_space" uri="acct_distribution_course_skool_smcs.xml"/>
  <file type="subject_matter_concept_space_graph" uri="acct_distribution_course_skool_smcs_graph.xca"/>
  <file type="template_concept_space" uri="org/oat/tools/util/tmpmodels/cs_template.xml"/>
  <file type="specific_narrative_attributes" uri="org/oat/tools/util/tmpmodels/narrative_attributes.xml"/>
  <file type="specific_narrative_concepts" uri="org/oat/tools/util/tmpmodels/narrative_structures.xml"/>
  <file type="casestudy_sample_narrative" uri="org/oat/tools/util/tmpmodels/default_custom_narrative.xml"/>
  <file type="last_search_results" uri="acct_distribution_course_skool_last_search_results.xml"/>
  <file type="sql_function_definitions" uri="org/oat/tools/util/tmpmodels/acct_sql_function_definitions.xquery"/>
  <file type="sql_find_name_function_definitions" uri="org/oat/tools/util/tmpmodels/find_name_function_definitions.xquery"/>
  <file type="relationship_definitions" uri="org/oat/tools/util/tmpmodels/acct_relationships.xml"/>
  <file type="learning_activities" uri="org/oat/tools/util/tmpmodels/learning_activities.xml"/>
  <file type="smcs_to_svg" uri="org/oat/tools/util/tmpmodels/smc-to-svg.xsl"/>
  <file type="narrative_to_rules" uri="org/oat/tools/util/tmpmodels/narrative_to_rules.xsl"/>
  <file type="search_history" uri="acct_distribution_course_skool_search_history.xml"/>
</manifest>
```

**Figure 5-2, ACCT Course Package Manifest**

### 5.3.2 Workspace

The ACCT uses a virtual file workspace for all “in-progress” models, i.e. any model that the course developer is currently working on. The workspace is initiated through the manifest file stored in an ACCT course package. The manifest file is parsed using JDOM [JDOM] and all required models are loaded into the ACCT virtual workspace. The workspace has open private access, i.e. can only be accessed by the ACCT, however all components of the ACCT have access to the files stored in the workspace. There is a simple versioning mechanism in place whereby an “in-progress” model can only be asynchronously accessed by one component at a time. The Course Package can then be updated based on the information stored in ACCT workspace.
5.3.3 Importing/Exporting Information Models

The ACCT supports the import and export of many of its models. This facilitates, for example, collaborative composition of personalised eLearning designs by allowing several different course developers to share their designs. The import/export functionality of the ACCT is facilitated by both the previously described workspace and course package and supports the disparate design elements of personalised eLearning. To import a model, the course developer can simply select an existing course package and import the appropriate model from its manifest. This functionality actively promotes knowledge reuse.

Take for example a situation where two different educators are teaching in the same subject area, a frequent occurrence in formal education. In this case the subject area may be defined, based on learning objectives, at a departmental level. The department may also supply a collection of resources which can be used at the individual educators’ discretion. This information can then be stored in an incomplete course package. The first course developer “imports” the subject matter concept space, from the provided base course package, into their workspace. They can now use the supplied subject area descriptions to start building their own personalisable course. Concurrently a second course developer can “import” the same subject area descriptions into their workspace and commence the construction of their personalisable course. Both personalised eLearning designs are based in the same subject area but may differ in pedagogical approach in accordance with the individual educators’ design preferences. Similarly, different course developers can use this import/export functionality to share whole or part eLearning designs.

5.3.4 Subject Matter Concept Space Builder

The ACCT Subject Matter Concept Space builder component provides the course developer with a suite of tools for graphically creating and customising an ontological representation of subject area. This component allows the subject expert to create and describe a subject area in “subjectmatter” concepts. It allows the creator to specify relationships between concepts in the concept space based on a selection of predefined relationship types. The course developer is supported in customising the concept space through creating new custom relationship definitions. Figure 5-3 illustrates the ACCT Subject Matter Concept Space Builder component with a sample SMCS based in the area of Electricity and Electromagnetism. Here we can see the graphical nature of the SMCS builder. In
comparison to the reviewed systems in chapter three, the “graph” authoring tool of the ACCT for representing a concept domain model is unique. Other state of the art authoring tools use lists and form editors to create their concept spaces. The graph authoring tool of the SMCS builder component is implemented using the JGraph [JGraph] API. This API, as the name states, supports the construction of “graphs” using java which fits with the ACCT’s component base of Java Swing.

![ACCT Subject Matter Concept Space Builder Component](image)

In the left pane of this component a list of all concepts can be seen, as illustrated in Figure 5-3. In the right pane is the graphical layout of these concepts. Each concept is represented by an “ellipse”, the text display of which is the name of the concept. In this case a hierarchical layout is used. The arrows between the concepts signify directional relationships.

For example, the concept “Electricity and Electromagnetism” is related to several sub concepts such as “Magnetism” and “Conduction of Electricity”. The relationships type is “contains”. The “contains” relationship type is used to describe a hierarchy. Also illustrated in Figure 5-3 are the graph tools. These represent available graph functionality such as save, zoom-in, zoom-out, grouping, undo, redo and exportation to SVG. The “save” button allows the course developer to
save iterative developments of the SMCS to the running workspace. They can shrink and grow the
graph based on visual preferences using the zoom functionality. Concepts and relationships can be
grouped in order to move or reorganise them. The usability requirements for PEDE, specified in
chapter three and chapter four, indicate that course developers must be in control of the
composition process. To support this, the ACCT provides undo and redo functionality as part of
the SMCS builder component. In a bid to actively support reuse and collaboration, the course
developer can export the visual representation of the SMCS from the SMCS Builder as SVG. This
process is detailed later in this chapter in a section entitled Model Translator/Interpreter.

The graphical SMCS, as illustrated in Figure 5-3, is saved in the ACCT workspace in two ways.
Firstly, the reusable and meaningful concepts and relationships as stored in a light weight canonical
ontology as seen in APPENDIX II - Models, Schema and Transforms. Secondly, the layout of the
SMCS is stored in an acctgraph file. This file contains only the concept’s location and dimensions
within the graph as seen in APPENDIX II - Models, Schema and Transforms. When the graphical
SMCS is loaded, the layout file is parsed in conjunction with the flat SMCS file. This parsing
process takes each concept from the flat SMCS and places it in its appropriate location with the
appropriate dimensions as described in the layout file. Once all concepts are added to the graphical
SMCS, a second iteration of parsing occurs whereby each concept’s relationships, as described in
the flat SMCS, are added to the graphical SMCS. This process allows the course developer to not
only save the semantics of their SMCS but also their visual arrangement of the concepts which
make up the SMCS.

5.3.5 Narrative Builder
The ACCT Narrative builder component provides the course developer with multiple layers of
composition support through a suit of tools for graphically composing personalised eLearning
designs. The course developer can create custom pedagogical strategies using the supplied
Narrative Structures and Activity models. They can then contextualise their design to a subject area
by adding subject matter concepts from their previously created SMCS. In order to make a non-
adaptive eLearning design adaptive, the course developer can associate the provided Narrative
Attributes with their design to provide a personalisation mechanism for their course. The course
developer can then instantiate their personalised eLearning design by searching for and selecting
appropriate candidate learning resources through the repository search service. This functionality is provided through a state of the art drag and drop interface. This allows the course developer to customise their design by dragging elements of the different models into the narrative builder and dropping them into their design.

Figure 5-4 illustrates the Narrative Builder component of the ACCT. In the left pane we can see that the course developer has a canvas for designing their course. This canvas can be filled by dropping in and customising elements of the supplied models from the palette in the right pane. For example, this particular design describes a WebQuest approach to teaching concepts in the domain of Electricity and Electromagnetism. As part of the “introduction” section, the course developer has identified that personalisation based on prior knowledge should be applied to four concepts, namely, Electricity and Electromagnetism, Conduction of Electricity, Circuits and Magnetism. This indicates that the associated “subjectmatter” concepts will be delivered, during execution of this course, based on the prior knowledge of the engaging learner. This and other personalised eLearning designs can be found in APPENDIX II – Models, Schema and Transforms.
As part of the usability requirements of chapter three and chapter four, the ACCT must support the
correction of all user actions. This is achieved in the Narrative Builder component by the “undo”
“redo” buttons which allow the course developer to rollback any changes made. Figure 5-5
illustrates the popup menu available in the canvas of the Narrative Builder. This allows the course
developer to “save” the current development of their design in the ACCT workspace. They can edit
a “narrative” concept, functionality which supports the creation of customised pedagogical
strategies. The course developer can delete any of the elements present in the canvas such as
Narrative Structures, Activities, SMCS concepts and Narrative Attributes. They can also change the
order of the pedagogical elements by using the “move up” and “move down” mechanism. The
course developer can view the properties of any of the concepts on their canvas. Finally, the course
developer can view the candidate learning resources which have been associated with the
“subjectmatter” concepts. The reason for hiding learning resources from the canvas view is because
the Narrative Builder focuses the course developer on creating pedagogically-sound personalised
eLearning designs, i.e. focusing on pedagogy more so than content.

![Figure 5-5, ACCT Narrative Builder Popup Menu](image)

The Narrative Builder component is, like the SMCS Builder, written in Java Swing. Another
similar aspect is the fact that the Narrative Builder translates the graphical narrative into a PEDE
narrative model which is then stored in the ACCT workspace. This translation process parses the
graphical narrative and transforms it through the PEDE Narrative schema to form a PEDE
Narrative. The PEDE narrative can be saved in three different ways using the ACCT. Firstly, a
pedagogy-only version can be saved whereby only “narrative” type concepts are saved. Secondly, a
contextualised version can be saved whereby both “narrative” and “subjectmatter” concepts are
saved, providing scaffolding for pedagogical diversity. Finally, an instantiated version can be saved
which stores all concepts and learning resources.
5.3.6 Resource Search

The ACCT Resource Search component provides a utility for course developers to search across multiple remote or local, secured or unsecured XML databases for appropriate learning resource, as illustrated in Figure 5-6. This component is supplemental to the Narrative Builder component described in the preceding section. The course developer can specify the location of the repository, the port which it accepts connections on and the collection in which to search. The course developer can specify a username and password for secure connections or leave it blank for unsecure connections. The resource search can work across a number of XML databases, namely Xindice [Xindice] and eXist [eXist]. The course developer can search for learning resources based on three different criteria. Firstly, they can search based on the contextual prior usage information of the learning resources, i.e. educational situations where these learning resources have been used to teach before. This information, which is stored in MetaSCO, can provide community based support for sourcing appropriate learning resources. However, this functionality has not been fully implemented yet. Secondly, the course developer can search a collection based on the learning resource metadata present. This mechanism directly searches across the existing metadata descriptions of a learning resource. The course developer can specify a number of “keywords” to search for in the metadata and can specify the number of returns using the slider as illustrated in Figure 5-6.

![Search a Learning Resource Repository](image)

Figure 5-6, ACCT Resource Search Component
Another mechanism available in this component is the ability to “open” an XML database collection. This simply opens the collection, scans the metadata present and creates a candidate resource list which the ACCT Narrative Builder component can then access.

The Search Resource component is implemented in Java Swing and uses various XML-related API’s such as those native to Xindice and eXist and other utilities such as JDom.

5.3.7 Model Translator/Interpreter

The ACCT is built on a model driven architecture. It therefore needs to translate and interpret many different modelled information sources so that the ACCT components can access, use, customise and create each of the models. There are multiple roles which this ACCT component can play. Firstly, it must be able to translate the XML modelled disparate design elements of personalisation into a canonical form that is understood by the ACCT java-based components and vice versa. These sources include the SMCS, the Activities, the Narrative Structures, the Narrative Attributes and the learning resources. Secondly, this component must support the translation of the models used and developed in the ACCT into a form which can be interpreted and executed by gAPeLS. Finally, the translator/interpreter component must be able to interact with different standards schemas such as IMS LD.

The schema of the SMCS, as described in chapter four, can be mapped directly into a collection of “cells” and “edges” which the ACCT SMCS builder component can interpret and render. As previously described, the flat SMCS and its associated layout file are translated to produce the graphical SMCS, i.e. SMCS concepts map to graph “cells” and relationships between concepts and map to graph “edges” which are used to connect “cells” together. This translation process produces a graphical SMCS which can be seen in Figure 5-3.

Another method for translating the graphical SMCS into a distributable standards format involves the conversion of the SMCS into Structured Vector Graphics (SVG). This process use XSLT [XSLT] and XPath [XPath] to transform the SMCS. The steps involved are similar to those used to transform the SMCS into the graphical SMCS except the primitives used in the transform are SVG. This transform can be seen in APPENDIX II – Models, Schema and Transforms.
One of the key functional requirements of the ACCT is to provide an easy to use graphical mechanism for composing personalised eLearning designs based on the disparate design elements of personalised eLearning. This graphical PEDE narrative, however, cannot directly run in gAPeLS since this service is based on the JESS rules engine. The model translator/interpreter component provides a mechanism for converting a PEDE narrative into the rules primitives that the gAPeLS service can interpret. This occurs by transforming the tree-like structure of the PEDE narrative into explicit JESS commands. The actual XSLT used to transform the PEDE narrative into JESS rules can be found in APPENDIX II – Models, Schema and Transforms.

One of the key levels of support provided by the ACCT allows the course developer to rapidly prototype, test and deploy a personalised eLearning design in gAPeLS. As part of this deploying process, the ACCT translates and exports a course package to the gAPeLS service, including the narrative rules as described in this section. The model translator/interpreter component of the ACCT is available to the gAPeLS service which allows this service to auto-generate a learner modelling instrument based on the PEDE Narrative. This process which extracts information about the types of personalisation present in a narrative, based on its Narrative Attribute associations, is detailed later in this chapter in a section entitled Building an Adaptive Course with the ACCT. The actual XSLT transform which automatically builds the learner modelling instrument can be found in APPENDIX II – Models, Schema and Transforms.

The models used in the ACCT directly inherit from the canonical models described in chapter four. However, to demonstrate the flexibility and power of the ACCT as a personalised eLearning development environment, a mechanism to translate the dynamic adaptivity of a PEDE narrative into canned adaptivity in an IMS LD was developed. This translation process maps the schema of the PEDE narrative to the schema of IMS LD which provides a translation channel between the two forms. The actual XSLT used to perform this translation can be found in APPENDIX II – Models, Schema and Transforms.
5.3.8 ACCT Generic Adaptive Personalised eLearning Service (gAPeLS)

As part of the rapid prototyping process supported by the ACCT, a generic version of APeLS was formed, namely gAPeLS. This generic personalised eLearning service supports the ability of a course developer to rapidly create, test and deploy a personalised eLearning design in a web environment. gAPeLS supports the presence of a single learner, i.e. the test learner. The course developer can then use the learner modelling instrument to engage the personalisable course on all possible levels, allowing them to rapidly test the semantics and pedagogy of the course against real “possible” learning scenarios. The use of this service is described later in this chapter in a section entitled *Building an Adaptive Course with the ACCT*.

5.4 Building an Adaptive Course with the ACCT

As part of the state of the art reviews in chapter three, several systems were evaluated and compared. During this evaluation process, the steps involved in creating and testing an adaptive course using each reviewed system was detailed. Based on this review process, the following sections describe the steps involved with creating and testing a personalisable course using the ACCT suit of components. More specifically, it describes the process of creating and loading a course package and workspace, importing models from an existing course package, representing a subject matter concept space, composing a personalised eLearning design and publishing a personalised eLearning design.

5.4.1 Creating/Loading a Course Package and Workspace

The first step in creating a personalised eLearning design using the ACCT involves either the creation of a new course package and workspace or the loading of an existing course package and workspace. As described previously in this chapter in a section entitled *Components of the ACCT*, the ACCT facilitates the creation and loading of a course package and workspace. Figure 5-7 illustrates the process of creating a course package and workspace using the ACCT.

When the course developer clicks on “File >> Create New Course Package” the associated dialogues of this component popup, as illustrated in Figure 5-7. Here the course developer can specify the location in which the course package is saved and the name of the package. They must specify the name of their custom narrative model and the name of their custom SMCS. When the
course developer clicks on the “Create Course Package” button, the specified course package is created and the ACCT virtual workspace is loaded.

![Figure 5-7, Creating a Course Package and Workspace with the ACCT](image)

When the workspace is initialised, the course developer can use the other ACCT components to start creating their personalised eLearning design.

In a similar fashion, the course developer can load an existing course package into the ACCT, as illustrated in Figure 5-8. In this case the course developer clicks on “File >> Load Course Package” and selects the location of the package they want to load using the components associated dialogues. Once the appropriate package has been selected, the course developer then clicks the “Load” button which loads the course package and initialises the ACCT workspace.
5.4.2 Importing Models from an Existing Course Package

As a mechanism to promote the reuse of the diverse and valuable assets of the ACCT, i.e. its information models, the ACCT allows the course developer to import different models from existing course packages. This allows course developers to share and collaborate on personalised eLearning designs by sharing the models of their designs. Figure 5-9 illustrates the steps involved with importing a model from an existing course package.

When the course developer clicks on “File >> Import Models”, the associated dialogues of this component popup, as illustrated in Figure 5-9. Here the course developer can specify the location of the existing course package and choose the model type that they wish to import from the package, i.e. either the SMCS or the PEDE Narrative. When the course developer clicks on the “Load” button, the chosen model is then parsed into the existing course package and workspace of the developer.
5.4.3 Representing the Subject Matter Concept Space

One of the most time consuming steps in the creation of personalised eLearning, both traditionally (chapter two) and through authoring environments (chapter three), was the specification of and description of the subject area. The ACCT addresses this by providing a component for graphically authoring a subject matter concept space, as described earlier in this chapter. When the course package and workspace of the ACCT have been initialised, the course developer can then use the SMCS builder component to create and customise the SMCS. They can either customise an existing SMCS, by using the import functionality as previously described, or create a new SMCS.

To modify an existing SMCS, the course developer can use the provided edit functionality, as illustrated by the popup menu of Figure 5-10. They can edit concepts and relationships, delete concepts and relationships and view the properties of a concept.
In order to add a new concept to the SMCS, the course developer can simply click on the “add” icon, as illustrated in Figure 5-11. This adds a “subjectmatter” concept to the graph and pops up a dialogue allowing the course developer to give the concept a unique name and describe what it is. This new concept “Electrode” can then be placed anywhere in the graph, simply by dragging it from its old location and dropping it in its new location.

![Figure 5-11, Adding a Concept to an SMCS](image)

The course developer can now create relationships between the new concept and the existing concepts in the SMCS. To create a relationship between two concepts, the course developer simply clicks in the centre of a concept, holds down the mouse button, drags the mouse over to the concept they would like it to relate to and releases the mouse button.
Once this action has been performed the dialogue, as illustrated in Figure 5-12, pops up which allows the course developer to define the relationship type and describe what the relationship means. If the relationship types provided as default by the ACCT are insufficient, the course developer is free to create their own custom relationship types. To do this, they must click on the "new" button which opens a dialogue allowing the developer to specify the custom relationship type, as illustrated in Figure 5-13.
Here, the course developer can specify the relationship type and define the relationship cardinality. They can also describe exactly what this custom relationship is.

When the course developer has finished creating, editing or customising the SMCS they can simply save it to the ACCT workspace and course package by clicking on the “floppy disk” icon in the SMCS builder’s toolbar. Note that the SMCS must be saved in order to make it available to the Narrative Builder component when composing the personalised eLearning design. The SMCS Builder component supports the course developer in creating a graphical representation of a subject area which they can use to contextualise a personalised eLearning design.

5.4.4 Composing the Personalised eLearning Design

The research question posed in the introduction chapter of this thesis “to identify and investigate the key extensible design elements, models and influences required for the composition of personalised eLearning experiences and to develop an integrated, model driven environment which facilitates rapid personalised eLearning development by educators or teachers (non-technical)”. As an implementation of a personalised eLearning development environment, the ACCT must provide the course developer with a suite of tools to integrate the diverse information models, as previously described, into a personalised eLearning design.

The Narrative Builder component, as previously described in this chapter, provides the tools and utilities for composing custom pedagogical strategies from Narrative Structures and Activities. It supports the contextualisation of this custom strategy by facilitating the insertion of SMCS concepts into the strategy. The course developer can then add personalisation axes to the design through Narrative Attributes. The developer is also supported with access to appropriate learning resource repositories. It also provides the deploy functionality of the ACCT.

5.4.4.1 Defining Pedagogical Strategy

For example, the course developer decides to apply a WebQuest pedagogical approach to their design. They simply drag the “WebQuest” model from the Narrative Structures palette and drop it onto their canvas, as illustrated in Figure 5-14. The course developer can view the “narrative”
concept's description from this model, to provide support guidelines and usage descriptions on how to build a WebQuest. This is achieved through the Narrative Builder’s popup menu. However, the course developer is free to ignore the supplied Narrative Structures and create their own pedagogical strategy or to customise an existing Narrative Structure.

![Image of Narrative Builder](image.png)

Figure 5-14, adding Narrative Structures to PEDE Narrative

This now provides the pedagogical framework for building a personalised eLearning design.

5.4.4.2 Refining Strategy with Activities

The next step for the course developer might involve the extension of the pedagogical framework using Activities. Again, to add an Activity to the eLearning design, the course developer simply drags the activity from the palette of tools in the right pane and drops it into the appropriate "narrative" concept, as illustrated in Figure 5-15.

141
This now provides a more active pedagogical framework for building personalised eLearning.

### 5.4.4.3 Contextualising Strategy using SMCS

The next step the course developer may perform is to contextualise their custom strategy. This can be achieved by accessing and using “subjectmatter” concepts from the previously designed SMCS in their design. To add “subjectmatter” concepts to their design the course developer can simply drag the concepts from the SMCS viewer of the Narrative Builder component and drop them into the appropriate “narrative” concepts of their custom strategy, as illustrated in Figure 5-16.
These set of actions performed by the course developer lead to the contextualisation of the eLearning design by associating it with a specified subject domain.

5.4.4.4 Adding Adaptivity

The next step which the course developer may take involves adding adaptivity to the eLearning design. The review of the state of the art of authoring tools for personalised eLearning showed that all current systems use low-level rules definitions based on either the content or explicit values from the learner model. Although MOT’s base architecture proposes three different layers of adaptivity granularity, only the finest layer consisting of the assembly language for adaptivity is present. In comparison, the ACCT supports the course developer in associating personalisation axes, similar to adaptive strategies of LAG, with their eLearning design to make it personalisable. The process occurs at an abstract level so the course developer does not need to hand write explicit rules of how learners interact with explicit concepts. Also, the association of personalisation occurs at the concept level and not the content level as with the reviewed systems in chapter three.
To make a non-adaptive course adaptive, the course developer must associate Narrative Attributes with their eLearning design. To add a Narrative Attribute, the course developer can simply drag the narrative attribute from the palette provided and drop it onto the appropriate “narrative” or “subjectmatter” concept depending on which type of personalisation axes they choose, as illustrated in Figure 5-17.

Figure 5-17, adding a Narrative Attribute to the PEDE Narrative

These set of actions performed by the course developer converts a non-adaptive course into an adaptive course.

5.4.4.5 Search for and Selecting Candidate Resources

The next step for the course developer to perform is to instantiate the PEDE narrative by searching for and selecting appropriate candidate resources using the Resource Search component of the ACCT.

As previously described in this chapter, the Resource Search component provides the course developer with access to local and remote learning resource repositories. As previously illustrated
in Figure 5-6, the course developer can define the location, port, collection, username and password for accessing a repositories collection. Once these details have been added, they can choose to either search based on keyword or simply “open” a collection of resources. Either way, this component returns a list of candidate resources which are available to the Narrative Builder component through the palette of tools. The course developer can then select the appropriate candidate resources and drop them into the appropriate “subjectmatter” concepts, as illustrated in Figure 5-18. By using the popup menu of the Narrative Builder component, the course developer can “view the candidate learning resources” of any “subjectmatter” concept.

The course developer can now “save” their personalised eLearning design using the popup menu of the Narrative Builder component.

The Narrative Builder component of the ACCT supports the Adaptive Course Construction Methodology as describe in chapter four by supporting and facilitating the creation of personalised eLearning designs from the disparate design elements of personalised eLearning. The following
section describes how to publish and test a personalised eLearning design using the ACCT and gAPeLS respectively.

5.4.5 Publishing the Personalised eLearning Design

One of the key functional aspects of the ACCT is its support of a rapid prototyping process. The course developer can build a personalised eLearning design using the ACCT suit of tools, as described in the preceding sections, and then publish it through the ACCT to gAPeLS.

The publication process flows from the ACCT to gAPeLS. The initial step in publishing a personalised eLearning design sees the course developer using the main menu of the ACCT and selecting “Tools >> Publish Adaptive Course”, as illustrated in Figure 5-19. This opens a dialogue where the course developer can specify the location of the gAPeLS service, i.e. “naxalite.cs.tcd.ie” as illustrated in Figure 5-19. To determine that this service is running at the specified location, the course developer can click the “Test Connection” button which establishes a test connection to the service. If the service is running, a dialogue displays a successful connection message to the course developer. When the course developer clicks on the “Publish Course” button, as illustrated in Figure 5-19, the publication process begins.

![Figure 5-19, Publishing an Adaptive Course with the ACCT](image-url)
The next step in the publication process involves the transfer of the required models, i.e. the course package, to the gAPeLS service. This is achieved by copying the course package to the XML database of the gAPeLS service. Since gAPeLS uses the XML database eXist as its information store, the ACCT can use eXist's API and XML-RPC to connect to and transfer the appropriate models to gAPeLS. This step of the publication process also uses the model translator component of the ACCT to translate the PEDE narrative to the rules narrative as previously described in this chapter. When the models have been transferred, the ACCT configures the course developer's web browser to point at the gAPeLS service at the specified location. The gAPeLS service now becomes active.

When the gAPeLS service is running, the process cycle illustrated in Figure 5-20 becomes active. Note that gAPeLS supports the rapid prototyping of personalised eLearning designs, it therefore provisions for a single instance of the learner model. At the index page the course developer can either choose to execute the adaptive course against a previously defined learner model or build a new learner model. If the course developer decides to execute the adaptive course, gAPeLS checks that there is a valid learner model first. If there is no valid learner model, the course developer is directed to the learner modelling instrument, otherwise gAPeLS checks for a valid personalised course model.
Figure 5-20, gAPeLS Execution Process
In order to dynamically build a learner model, gAPeLS must provide an appropriate learner modelling instrument. This is achieved, as previously mentioned, by reasoning across the PEDE narrative to produce the modelling instrument using XSL. More specifically, it processes the PEDE Narrative selecting all present Narrative Attributes. For each Narrative Attribute, its parameters “type” is checked. A “unit” “type” indicates that the modelling instrument used should appear in isolation, i.e. a one off, fixed questionnaire which can be used to capture learner characteristics. An example of such a modelling instrument is a learning styles inventory. Regardless of educational context, the modelling instrument is the same; therefore this “type” can exist in isolation. An “all” “type”, however, indicates that the modelling instrument must be specifically engineered to capture learner characteristics pertinent to the current eLearning context. An example of such a modelling instrument is prior knowledge capturing. In this case, the learner information being modelled is completely dependent on the subject area being engaged. Currently, the type of personalisation being supported is based on prior knowledge. Therefore, the reasoning process dynamically builds the modelling instrument based on the description of the associated Narrative Attribute and the “subjectmatter” concepts to which the Narrative Attribute has been applied.

The course developer can now use this modelling instrument to generate an instance of a learner model to execute in the adaptive course. This populated learner model contains information, such as learner’s prior knowledge of a subject area and learner’s learning styles, which is used by the Narrative Attributes to perform personalisation.

The next step involves the execution of the narrative against the available information sources. In the example personalised eLearning design, illustrated in this chapter, the introduction section of a WebQuest on Electricity and Electromagnetism is to be personalised based on prior knowledge. The auto-generated questionnaire reflects this personalisation request and provides an instantiated learner model with information about the concepts being introduced. When the narrative executes, the information in the learner model, i.e. competencies.taught is compared to the information about the concepts in the SMCS, i.e. competencies.required. If these two sets of information intersect, then it can be said that the current learner has at least some of the required competencies to engage the concept being taught. If there is no intersection then the learner does not have the required competencies to view the current concept.
This provides the course developer with a mechanism for testing the validity of the personalised eLearning design in real time. To make adjustments to this design the course developer can simply edit the design in the ACCT and re-publish. This supports a rapid prototyping mechanism as proposed in the Adaptive Course Construction Methodology, described in chapter four.

5.5 Implementation of Models

Based on the architectural design of the Adaptive Course Construction Toolkit (ACCT), there are several core models that are used such as narrative structures, activity, subject matter concept space, narrative attributes, selectors, learner, teacher, learning resources and PEDE narrative. These models are detailed in the following sections.

5.5.1 Narrative Structures

Narrative Structures should provide both pedagogical support and a pedagogical framework for building personalised eLearning. As part of the work in this thesis, several narrative structures have been created to illustrate the WebQuest, Didactic and Case Study pedagogical approaches. For example, the WebQuest Narrative Structure, as illustrated in Figure 5-21, describes a six step creation process consisting of introducing the core objectives, describing the tasks to be performed, illustrating the process or steps involved in the tasks, listing available resources, concluding the quest and encouraging the expansion of knowledge and finally describing the evaluation process and metrics.

In chapter four, a schema for representing pedagogy in Narrative Structures was proposed. Figure 5-21, illustrates the representation of the WebQuest pedagogy as a Narrative Structure. The container element for the Narrative Structure is a “concept” of type “model” which simply illustrates that the conceptual structure represented is a model. The “name” of the model reflects that of the modelled pedagogy, i.e. WebQuest. The “description” element can give an overall description of what the modelled pedagogy is. The “relationships” element facilitates the description and representation of workflow across the “narrative” concepts of the Narrative Structure. The “model” type concept then contains several child “narrative” type concepts. Each of the concepts, similar to the “model” concept, contain the name of the sub-concept of the Narrative
Structure, the “description” element which provides the usage and guideline information for the course developer, and so on as described for the “model” concept. The Narrative Structures for this and the other pedagogical models can be found in APPENDIX II – Models, Schema and Transforms.

Narrative Structures form the foundations of pedagogical support offered by the ACCT and are used to provide guidance to the course developer. However, there are several scenarios where making this pedagogical structuring explicit is appropriate. For example, when sharing a personalised eLearning design, the course developer may want to share the structural information making this pedagogical structuring explicit is appropriate. For example, when sharing a

Figure 5-21, Implemented Narrative Structure for WebQuest Pedagogy
making them feel more in control. This could also be a personalisable feature whereby the pedagogical information of a course could be switched on and off adaptively, based on certain characteristics of the learner such as learning styles.

The ACCT supports the import, export, creation and customisation of Narrative Structures through the Narrative Builder component. This is described in the section entitled Components of the ACCT.

5.5.2 Activity

Activity model should provide a framework for representing atomic activities, low-level composite activities and high-level composite activities. In doing so, they must be able to represent workflow across the different types of activities. They must also support the description of pedagogical information for describing this workflow. The activities must be able to describe various candidate communication services which the different activity types can access and use.

Figure 5-22 illustrates an implementation of the Activity schema for representing a “Peer Review” learning activity. The “learning_activity” element represents the container for the activity and is used to describe the granularity of the activity. In this example, the model is of a “composite” activity called “Peer Review” which consists of several child “activities”. Each child “activity” contains a “narrative” concept which can be used to provide pedagogical guidance to the course developer on how to use the activity. As previously noted, the “narrative” concepts “relationships” can be used to define a workflow over the activity structures. The “submission” low-level composite activity, for example, contains two candidate communication services, one for email and one for posting. These candidates can be reasoned across when adaptively rendering the learning activity structure. An example of this would be if the current learning environment exists in a low-bandwidth network and the learner is being asked to submit work in excess of 200 megabytes of information, email would not be an appropriate option since most mail servers typically place restrictions on mail size, i.e. no greater than 2 megabytes. This and another instantiated Activity models can be found in APPENDIX II – Models, Schema and Transforms.
As described in the previous section on Narrative Structures and based on the design of activity models in chapter four, it can be noted that activity models such as that illustrated in Figure 5-22 provide another level of pedagogical support for the course developer. This support and guidelines for usage is contained in the activity descriptions, associated “narrative” concept descriptions and the pedagogical workflow described in the “relationships” of such.
The ACCT supports the use of Activity models in the creation of personalised eLearning designs through the Narrative Builder component. This component is described in the section entitled Components of the ACCT.

5.5.3 Subject Matter Concept Space

A Subject Matter Concept Space should provide a framework for representing the knowledge about a subject area. This means that the conceptual knowledge about subject concepts and their relationships with other concepts in the area can be consolidated in ontological and accessible form.

Figure 5-23 illustrates an implementation of the SMCS schema for representing a subject matter concept space on “Electricity and Electromagnetism”. The root element “smcs” signifies that this model is a subject matter concept space. The “smcs” consists of several child “concept” elements. Note that each of these “concept” elements is of type “subjectmatter”. The “name” signifies the name of the concept. The “description” is used to describe what exactly the concept is. The “relationships” element here differs slightly in purpose to that of a Narrative Structure as it is not used to describe workflow as it is in Narrative Structures and Activities but instead it is used to describe the semantic relationships of the concepts of the SMCS, much like an ontology. From this it can be noted that the “Electricity and Electromagnetism” concept is related to four other concepts within this SMCS, identified by the concept id in the “relationship” elements “xlink:href” attribute.
Electricity and Electromagnetism

This concept covers such things as Electricity and Electromagnetism.

Figure 5-23, Implemented SMCS for Electricity and Electromagnetism Subject Area
The SMCS as illustrated in Figure 5-23 is a conceptual representation of a subject area. It can be noted that it is not contextualised by content references. This content independence supports the reuse of subject area information across many different potential content bodies which can be used to teach it, as described in chapter four. This forms another key level of support for the course developer in that the SMCS model can not only provide relevant subject information but can also be used and reused across many different operational contexts.

The ACCT supports the import, export, creation and customisation of Subject Matter Concept Space models through the Subject Matter Concept Space Builder component. It also supports the instantiation of personalised eLearning designs in the Narrative Builder Component. The use of the SMCS model in these different components is detailed in the section entitled Components of the ACCT.

5.5.4 Narrative Attributes

Narrative Attributes should provide the ability to personalise an eLearning design based on pedagogically sound metrics of adaptivity. This means that they must be able to reason across multiple information sources and make personalisation decisions based on this information. The output of a Narrative Attribute must be formulated so as to correspond to the requirements of a personalised eLearning design. For example, a Narrative Attribute for prior knowledge makes decisions based on similarities in information between the SMCS and learner model. In this sense, the Narrative Attribute can be represented as:

\[ NA_{PK} \equiv M_{SMCS} \cap M_L, \]

where \( M_{SMCS} \) represents the information stored in the SMCS model and \( M_L \) represents the information stored in the learner model. The decision is then made based on the intersection \( \cap \) of the information in the SMCS and the learner model. For example, if the prerequisite knowledge for learning Concept C is both Concept A and Concept B. If the learner has prior knowledge of Concept A only, then there is an intersection between the prerequisites of the concept being taught and the prior knowledge of the learner, i.e. Concept A is the intersection between concepts required and concepts learned. The Narrative Attribute can then make decisions based on this information.
A Narrative Attribute for adaptivity based on learning styles, such as Kolb [Kolb, 1984], would make decisions based on the similarities of the learning style characteristics of the learner and the learning styles related metadata of the candidate content groups. Such a Narrative Attribute can be represented as:

\[ NA_{LS} = M_L \cap M_{CCG}, \]

where \( M_L \) represents the information stored in the learner model and \( M_{CCG} \) represents the information stored in candidate content group models. The decision is then made based on the similarities, or intersection \( \cap \) of the two information sources.

Another example of a Narrative Attribute for adaptivity based on a learner’s environment would make decisions based on similarities between the characteristics of the learner, the context related information of the candidate learning resources metadata and the information available about the current learning environment. This Narrative Attribute would be represented as:

\[ NA_{CT} = M_L \cap M_E \cap M_{CLR}, \]

where \( M_L \) represents the information stored in the learner model, \( M_E \) represents modelled information about the learner’s current environment and \( M_{CLR} \) represents the information stored in candidate learning resource model. In this case, a decision would be made based on the similarities between the modelled information about the environment and the candidate learning resources, and then the modelled information about the learner. This can be achieved by filtering the learning resources based on their similarities to the current learning environment and then reconciling that with the learner information, so only candidate learning resources that satisfy the environmental requirements are compared to the learner information.

Figure 5-24 illustrates an implementation of the Narrative Attribute schema in a Narrative Attribute for personalisation based on prior knowledge. The “name” signifies that the name of the Narrative Attribute is “Prior Knowledge” and its “type” is “adaptiveaxes.actor.knowledgespaces” based on
an external taxonomy of personalisation axes found at the location specified by the “ref” attribute. The “description” can provide usage and guideline information for the Narrative Attribute. The “parameters” of the Narrative Attribute are classified by the “param” element and describe the inputs of the Narrative Attribute. In this case, “param” one refers to “competencies.required”. This information can be found in the SMCS model for the subject area where prerequisite relationships can be used to describe the required competencies for a concept. “param” two is “competencies.learned”. This information can be found in the learner model based on the prior knowledge of the learner in this subject area. The output type of this Narrative Attribute is defined by the chosen candidate selector. The preconditions of this Narrative Attribute are that both the learner model and SMCS exist. The effects of this Narrative Attribute require that the learner model be updated with the appropriate competencies of the concept being tested. These IOPE are similar to that used to describe any web service. This and other Narrative Attributes can be found in APPENDIX II – Models, Schema and Transforms.

```xml
<narrative_attribute id="00102">
  <name lang="en">Prior Knowledge</name>
  <type lang="en">adaptiveaxes.actor.priorknowledge</type>
  <description lang="en">This can be used to request that actor-based adaptation be applied to a NarrativeConcept. The type of adaptability relates to the Theory of Knowledge Spaces. This ....</description>
  <parameters set="all">
    - <param type="competencies.required">
        <restrictions type="string" use="required" />
        <description>This is a comma separated list of concepts that the learner is required to have been exposed to prior to viewing this concept</description>
    </param>
    - <param type="competencies.taught">
        <restrictions type="string" use="required" />
        <description>This is a comma separated list of concepts that the learner has already learned</description>
    </param>
    - <param type="competencies.taught">
        <restrictions type="string" use="required" />
        <description>This is a comma separated list of concepts that the learner will experience through viewing this concept</description>
    </param>
  </parameters>
  <relationships>
    - <relationship xlink:href="#001B">
        <relationship_type xlink:href="#ISA" />
        <relationship_cardinality xlink:href="#1-1" />
        <relationship_description lang="en" />
      </relationship>
  </relationships>
  <candidates>
    - <selector href="http://134.226.38.81:8080/exist/servlet/db/ae/acct_developer_support/narrative_attributes/selectors/prior_knowledge" />
  </candidates>
</narrative_attribute>
```

Figure 5-24, An implementation of a Narrative Attribute for Prior Knowledge Personalisation

As illustrated in chapter four, all elements of personalisation can utilise the Architecture for Candidacy. A Narrative Attribute employs candidacy for its selectors. Different candidate selectors
can, for example, achieve the same goals from the same input types but output their results in different formats. For example, one candidate selector may output in model form and another may output in list form. Both sets of outputs contain the same information but they are structured differently. The implementation of selectors is described in the following section.

The ACCT supports the association of Narrative Attributes with Narrative Concepts and Activities in the Narrative Builder component. The use of Narrative Attributes in this component is detailed later in this chapter in a section entitled Components of the ACCT.

5.5.5 Selectors

As described in the preceding section, a Narrative Attribute can reference multiple candidate selectors. A selector is the rules representation of a Narrative Attribute. So in theory, Narrative Attributes provide the metadata descriptions, usage scenarios and accessibility information for executing selectors.

Figure 5-25 illustrates an implementation of a selector based on prior knowledge personalisation. The “id” identifies the selector as being a Prior Knowledge Selector. This selector is written in the language JESS, as illustrated by its “type” element. The “call_name” describes the functional call which must be used to invoke this rule, i.e. add-prior-knowledge. Inside the “code” element is the actual JESS rules for performing adaptivity based on prior knowledge. Within the body of the selector, you can see the function which gets the competencies.required from the SMCS based on the current concept being tested, identified by ?conceptId. Following this is the function definition for gathering the competencies.learned from the current learner model. Following this is the functional comparison of the competencies.required list and the competencies.learned list. It can be noted here that the intersection of the two lists forms the list of candidates that can be returned by the selector; in this case the candidates are concepts from the SMCS. This is then followed by a rule for determining the output type of this selector. If the ?returnType is a “model”, then this selector builds a model of the chosen candidates and return this, otherwise it returns a raw list of candidates.
The reason for these return types becomes apparent when you consider the role of sub-selectors in a chained adaptation process. If two selectors are nested then the output of the inner selector would become the input of the outer selector. For example, in this case the inner selector may return a "model" and the outer selector may return a "list". This and other selectors can be found in APPENDIX II - Models, Schema and Transforms.

5.5.6 Learner

The implementation of the Learner model in the ACCT occurs at an abstract level. This means the information requirements for the learner model are contained in the personalised eLearning design. The personalised eLearning design can be reasoned across to produce the base schema for a learner model. An implemented version of this schema based on IMS LIP is illustrated in Figure 5-26. This is possible since the eLearning design captures the information of the course, i.e. the pedagogy, the subject area, the activities, and so on as previously described. However the key information for building a learner model schema is extracted from the Narrative Attributes of the eLearning design. For example, as part of a simple WebQuest’s introduction section, the course developer has indicated that this should be adapted based on prior knowledge, i.e. associated a Narrative Attribute.
with it. Based on the implementation of a prior knowledge Narrative Attribute, described earlier in this chapter, we know the required IOPE. Now, for this personalisable course to work we need to know the prerequisite knowledge of the learner with respect to the concepts be taught in the introduction section. We, therefore, know that the learner model must, in its most basic form, contain information about the learners' prerequisite knowledge of section concepts. So from this it is understood that learner model schemas, although basic, can be automatically produced based on the information stored within a personalised eLearning design. Both this process and the process of automatically generating modelling mechanisms for capturing learner information are detailed later in this chapter in a section entitled Publishing the Personalised eLearning Design.
Figure 5-26, an Implemented Learner Model
5.5.7 Learning Resource Candidates

Based on the design of learning resource candidate models in chapter four and the existing standards for representing "learning objects", an implementation of learning resource models should facilitate the architecture for candidacy. This means that they must provide suitable mechanisms for grouping "like" learning resources and locating selected learning resources.

Detailed in chapter four is a schema for representing modelled learning resource candidates. Figure 5-27 illustrates an implementation of this schema for candidate learning resources. As part of a concept’s "candidates" element, there are three candidate resources in this example. For each resource, there is a reference to the location of its metadata, a reference to the physical content, its title and a description. These “candidate” descriptions are automatically generated when a candidate learning resource is selected and placed in the personalised eLearning design, by referencing the resources metadata. This and other examples of candidate resources can be found in APPENDIX II - Models, Schema and Transforms.

```xml
<candidate_resource>
  <metadata_location>http://nosalite.cs.tcd.ie:8000/exist/xmlid/db/skool/content/USMLCD1.xml</metadata_location>
  <candidate_location>http://nosalite.cs.tcd.ie:8000/skool/shockwave/USMLC/USMLCD1.swf</candidate_location>
  <candidate_title>Introduction to Magnets</candidate_title>
  <candidate_description/>
</candidate_resource>

<candidate_resource>
  <metadata_location>http://nosalite.cs.tcd.ie:8000/exist/xmlid/db/skool/content/USMLCD3.xml</metadata_location>
  <candidate_title>Magnetics</candidate_title>
  <candidate_description/>
</candidate_resource>
</candidates>
```

Figure 5-27, an Implementation of the Candidate Learning Resource Schema

5.5.8 PEDE Narrative

A Personalised eLearning Development Environment (PEDE) Narrative is the embodiment of a custom educational strategy based on the disparate design elements of personalised eLearning. This indicates that an implementation of PEDE Narrative should support and facilitate the assembly of the diverse information sources used in personalised eLearning. In essence, it should facilitate the assembly of models such as Narrative Structure, Activities, SMCS, Narrative Attributes and candidate learning resources. It can therefore be represented mathematically as:
where \( M_{NS} \) is the selected or developed Narrative Structures, \( M_A \) is the selected Activity models, \( M_{SMCS} \) represents selected subsections from the concept space, \( M_{NA} \) represents the selected personalisation axes and \( M_{CLR} \) represents the selected candidate learning resources. Several examples of a PEDE Narrative can be found in APPENDIX II - Models, Schema and Transforms.

### 5.6 Summary

As described in section 4.8, an iterative design process was used for PEDE. Similarly, a rapid prototyping approach was used to develop the models and components of the ACCT. In parallel a series of mini-evaluations lead to the further refinement of the prototype components and models. Certain results from these iterative steps became apparent as we trialled the prototype components. For example, earlier versions of the ACCT used a form-like approach to defining the concept space. Through refinement, this then moved towards a more graphical representation in a tree form. Based on the visual limitations of trees to represent non-hierarchical information, a move to the current graph visualisation was taken (see section 6.2). It has since been identified that as the information space being modelled becomes larger (approx. 100 concepts or more), the functionality of a graph needs to change to support such things as encapsulation to aid visualisation and usability.

The goal of this chapter was to describe the implementation of a specific personalised eLearning development environment, namely the Adaptive Course Construction Toolkit (ACCT). This chapter provided a detailed insight into the architecture of the ACCT which is based on the architecture of PEDE as described in chapter four. Also described here is the implementation of the diverse models of personalised eLearning development. The components of the ACCT were described and illustrated in this chapter. To highlight the influence of the course construction methodology on PEDE and the ACCT, the process of building an adaptive course using the ACCT was then described and illustrated.
6 Evaluation

6.1 Introduction

The research question posed in this thesis was to identify and investigate the key extensible design elements, models and influences required for the composition of personalised eLearning experiences and to develop an integrated, model driven environment which facilitates rapid personalised eLearning development by educators or teachers (non-technical). Based on this question a number of key goals were described, namely;

1. to identify and specify requirements for the elements and influences of personalised eLearning experiences,
2. to specify an integration process for elements and influences of personalised course composition,
3. to develop a suite of tools which support crucial aspects of the personalised eLearning development process,
4. to enhance the efficiency of the specified integration process,
5. to evaluate the process and suite of tools to support teachers in developing personalised eLearning.

The proposed integration process is made possible by the separation of the diverse influences and elements of personalised eLearning, the requirements of which were described in chapter 4. The described separation also provides the foundations for reusing all levels of educational knowledge. Through the integration process, based on sound educational development theories and practices, the Personalised eLearning Development Environments can provide an educationally-driven support framework for constructing personalised eLearning.

In order to evaluate this research a number of tracks were taken. Firstly, the development of the ACCT followed a rapid prototyping approach. Through a series of development iterations, initial prototypes for several of the core components for a PEDE were created. The goal of this chapter is to outline the initial requirements identified during these design iterations. Following this is a description of the context of the evaluation approach for the final system. Sections 6.4 through 6.6
describe the evaluation results of the final ACCT system. The chapter then concludes with an overall analysis of the evaluation of this research.

6.2 Requirements Identified During Initial Iterative Design Process

Requirements for these development iterations were refined based on a series of discussions with educational experts and course developers who were consulted from the outset of this research. Expertise from Instructional Designers, Pedagogic Experts and Content Developers from Trinity College’s Centre for Academic Practice and Student Learning/Centre for Learning Technologies, in conjunction with Content Developers from Intel Ireland and Instructional Designers from SKOOOL were used to design and modify the educational models defined in this research. Based on these consultations a number of model refinements were made. For example, when designing the Narrative Structures model, a clear requirement, elicited through discussion and discovery, was to support pedagogic diversity. As described in chapter 3, diversity of pedagogy was a fundamental element of educational design. Therefore the designed model and instantiated examples (Case Study, WebQuest and Didactic), as described in chapters 4 and 5, must be descriptive enough to represent process, workflow and guidelines and also flexible enough to promote diversity. The iterative refinements facilitated the definition of more sophisticated workflow, i.e. it was now possible to define workflow across a number of pedagogic models. These diversity metrics were also applied to the other models of this research.

Also during the development lifecycle a series of mini evaluations, focusing on user testing and acceptance testing, were performed to address both the support offered by the prototype components to the course developer and the usability characteristics of the prototype components (see
APPENDIX VII – Phase 1 Trial and Evaluation for questionnaire used). The mini evaluations involved various fields of expertise such as adaptive systems engineers, semantic web engineers, instructional developers and pedagogues. These iterative evaluations lead to modification of and refinement of the ACCT to its current state, prototype version 0.6.

A number of key issues were addressed during the iterative design, development and evaluation phase of this research. Since the core components of the ACCT should support the rapid development of the information models required for Personalised eLearning, a key step in the evolution of the ACCT to version 0.6 was to enhance and improve those components. This led to iterative refinement of the core components of the ACCT, namely the SMCS Builder, the Narrative Builder and Learning Content Search component.

**Subject Matter Concept Space Builder**

The first SMCS builder was form based, like the interface of the MOT systems (reviewed in chapter three). The form based editor allowed you to define the concepts (much like the current version without the relationship flexibility) but visually represented those concepts as a flat list.

However, it was identified through mini-evaluation (see APPENDIX VII – Phase 1 Trial and Evaluation for questionnaire used) and numerous discussions, that the inability to visualise the concepts of a subject area and the semantics of how those concepts related to other concepts was a serious issue. Following on from this, a later version of the ACCT used a tree-based representation for the SMCS. The information was still entered through a form but now the user could begin to visualise the semantic relationships of the concepts in the SMCS.

However, the visual limitations of representing ontological form in a strictly hierarchical form presented a cognitive problem to the course developers and a technical problem when representing multi-directional relationships. These issues were then addressed by using a graph editor for the SMCS Builder in ACCT version 0.6. This flexible representation allowed the course developer to represent and visually understand the relationships between the concepts in a subject area.

**Narrative Builder**
The Narrative Builder of the ACCT was also improved during this iterative design, development and evaluation phase of the research. The initial versions of the Narrative Builder did not support drag and drop. Models or parts of a model were selected from the "palette" of tools and then a transfer button allowed the course developer to copy to model to their narrative workspace. However, drag and drop functionality was later added based on a requirement to replicate the commonly accepted functionality of file transfer programmes; the course developers expected to be able to drag models or parts of a model from the "palette" of tools and drop them onto their current workspace. This also allowed other subtle refinements of the functionality of the Narrative Builder whereby course developers could rearrange the models of their Personalised eLearning Design simply by dragging and dropping within the Narrative workspace.

Learning Resource Search and Selection

The search functionality of the early ACCT prototypes was quite limited and very slow. This would have to be addressed going forward. The learning resources accessible to the ACCT were made available through the Xindice XML database. This database did not support local XQuery so a work-around was to copy the metadata of the learning resources available into working memory, query them using XQuery and build a temporary model to store the candidate resources. This process was slow and very inefficient concerning memory usage. An alternative solution was needed. This was provided in the form of the eXist XML database which at the time natively supported XQuery. When the search component was modified to connect to and execute across this new database, the querying speed and local memory efficiency dramatically improved. A large remote repository of learning content (360 files) could now be queried in seconds not minutes.

These iterative refinements have helped to identify and address a number of issues, the satisfaction of which, have led to current version of the ACCT version 0.6.

6.3 Context of Final System Evaluation

The evaluation presented here sought to address a number of aspects of this research through a series of trials. Firstly, since one of the principles of eLearning development is to empower teachers in creating personalised eLearning, the thesis examined the level of course developer satisfaction. This involved establishing the experience levels of the trial participants and identifying the clarity
of the design process (as envisioned in the ACCM and as implemented in the ACCT). Secondly, the thesis examined the potential for pedagogic diversity within this applied research. This involved examining the flexibility and extensibility of Personalised eLearning Development Environments (PEDE) and analysing the ease of use of the modelled disparate design elements of personalised eLearning development. Finally the thesis examined the usability of the interface of the ACCT. This involved the application and analysis of proven Human Computer Interaction (HCI) design principles.

Presented here are the results of the most recent evaluation of the ACCT which focused on the non-technical course developer. This trial and evaluation was organised as a one day workshop and consisted of four cohort participants. The workshop involved the three previously mentioned evaluation categories, i.e. level of course developer satisfaction, potential for pedagogic diversity and usability of the interface of the ACCT. This low number of trial participants could be argued as being not statistically significant regarding course developer satisfaction. However, coupled with iterative development strategy of this research, what it does provide are indications of course developer use and satisfaction due to the demographics of the participants, i.e. non-technical course developers. To provide a more statistically significant evaluation, further trials need to be carried out, see 7.4 on Future Work.

The following sections describe in detail the results of the aforementioned evaluation metrics as applied to this research. The chapter then concludes with an overall analysis of the impact and significance of this evaluation.

6.4 Course Developer Satisfaction

The main reason for evaluating user (course developer) satisfaction is that this thesis proposed the development of a suite of tools capable of being used (easily) by non-technical course developers. Firstly, we must establish the motivation and context of this phase of evaluation by examining the evaluation strategy and the background of the participants. We must then focus on issues relating to the course developer, the ACCM and the ACCT. This helps in evaluating the ACCT’s implementation of the ACCM, the course developer’s ability to perform the series of phases within
the ACCM, the course developers understanding of the personalisation process and their overall satisfaction with the ACCT.

6.4.1 Strategy of Evaluation

A series of personalised eLearning development workshops were used to evaluate this research from a course developer's perspective. The format of the workshop consisted of;

1. presentations on personalised eLearning and this research,
2. demonstration of existing personalised eLearning courses,
3. demonstration of the functionality of the ACCT,
4. getting the participants to perform a series of pre-defined tasks using the ACCT,
5. answering the questionnaire (APPENDIX IV – Trial Questionnaire),
6. having an open discussion with participants regarding this research and the workshop.

The interactive part of the workshop consisted of the participants carrying out a prescribed series of tasks with the ACCT such as creating a course package, importing and customising existing personalised eLearning models, building a customised activity-oriented pedagogical strategy, searching for and selecting learning resources and publishing and testing their adaptive personalised eLearning course. These tasks would help demonstrate the ACCT's use of the ACCM as an integration process for developing personalised eLearning. The tasks also provide an insight into how the separation of the diverse models not only supports the integration process but also provides the support framework for the course developer using the ACCT.

6.4.2 Participant's Background

The choice of participant for the series of trials was based on the high level goals and objectives in section 1.3. Since one of these stated goals was to evaluate the process and suite of tools to support teachers in developing personalised eLearning, the chosen trial population should reflect a non-technical user base. These non-technical and educationally focused participants formed the basis of the evaluation analysis presented here.

As illustrated in Figure 6-1, 2 of the trial participants, prior to this trial, had never created an eLearning course. 1 participant had previously created a single eLearning course. The final participant had created between two and five eLearning courses in the past. This fact was quite interesting, based on discussions with the trial participants, as 2 of them were mainly of subject
matter experts. These participants, although having never directly created an eLearning course, had been involved with the specification of and classification of subject areas from which eLearning courses had been created. The remaining 2 participants, i.e. "the eLearning developers", only 1 had direct expertise in all disciplines involved in creating eLearning namely, IT literate, subject knowledge and instructional experience. This indicates an explicit separation of roles within eLearning development whereby the subject experts define the scope of the subject area, the pedagogical experts design a custom educational design based on the subject area descriptions and then the instructional developers create and assemble appropriate learning material to effectively teach based on this.

One of the benefits of the ACCT approach is that it supports non-technical course developers, i.e. real teachers, in creating eLearning. This means that subject experts are supported in creating appropriate pedagogy and pedagogy experts are supported in representing subject areas, supporting multiple roles in a single environment.

As illustrated in Figure 6-2, all 4 trial participants had no previous experience of using a course composition tool to create eLearning, not surprising since 2 of those participants had never created eLearning at all. This provides an interesting insight into the mindset of the participants during this trial. Although they had no prior experience using a course composition tool, through discussions it
was identified that those who had previously created eLearning had indeed used a course composition process, however this process did not consist of an associated composition tool. This indicates that "the eLearning developers" already had a pre-existing course construction methodology whether it was well documented or not. This point is further explored later in this evaluation when discussing the course developer's satisfaction with the ACCM.

![Bar Chart]

**Figure 6-2, Question 2: How many times have you used a course composition tool?**

As illustrated in Figure 6-3, all 4 trial participants had no prior knowledge of Adaptive Hypermedia. This is another interesting point since the participants have no knowledge or experience of what Adaptive Hypermedia actually is, indicating that although they are IT literate they are far from being adaptive system engineers. This helps to further indicate the technical transparency of the ACCT which is described later in this analysis.
As illustrated in Figure 6-4, 3 of the trial participants had no previous knowledge of personalised eLearning. The remaining participant had limited prior knowledge of personalised eLearning. This is quite interesting as 1 participant, prior to the trial, distinguished knowledge about AH from knowledge about personalised eLearning. Although the trial participants were not adaptive systems engineers, through discussion it was indicated that they knew that personalised eLearning involved the tailoring of educational experiences to individual learners however they were unaware that personalised eLearning used techniques and processes from AH to achieve its personalisation. This point also helps to emphasize the ACCT’s use of personalisation abstraction, i.e. hiding the complexities from the course developer, as is described later in this evaluation.
Based on these analysis results, it is evident that the trial participants were excellent candidates to evaluate this research. They were IT literate but were not proficient in the programming skills required to produce Adaptive Hypermedia. They were educationalists but had little or no experience of using technology to develop online eLearning. They had proficient knowledge about the processes of creating educational experiences but had no experience with adaptivity or personalisation.

6.4.3 Course Developer, Adaptive Course Construction Methodology (ACCM) and the ACCT

The components of the ACCT are based on the core processes of the ACCM. It empowers the course developer in creating pedagogically-driven personalised eLearning in a technically transparent manner. Therefore to analyse the course developer’s satisfaction with the support offered by the ACCT in performing the core elements of the ACCM, a series of questions were formulated. These questions focused on a number of key areas, namely; the clarity of the design process inherent in the Adaptive Course Construction Methodology (ACCM); the clarity of the design process in the ACCT, i.e. the ACCT’s implementation of the ACCM; and the course
developer's satisfaction with the ACCT. These question categories helped to identify the benefits and concerns of the ACCM and the ACCT in regard to course developer satisfaction.

6.4.3.1 Clarity of Design Process (ACCM)

Firstly we examined both the clarity of the ACCM and the influences of the ACCM on the ACCT. In order to achieve this, a series of questions were proposed, the results of which are illustrated in Figure 6-5 and Figure 6-6. These questions identified the participant’s satisfaction and understanding of the processes of the ACCM and the ACCT’s implementation of the core processes of the ACCM, respectively. Firstly, all participants had a clear understanding of the different processes of the ACCM. Through discussion it was identified that this clear understanding arose from a familiarity with the process involved (“the eLearning developers”) and from the clear separation and representation of the core components of this process (all participants). The question regarding the ACCT’s implementation of the core elements of the ACCM reflects this understanding and satisfaction. All participants found the ACCT’s implementation of the core elements of the ACCM to be clear and supportive. This is an interesting point since the “non eLearning developers” could identify the correlation between the ACCM and the ACCT’s support framework for building personalised eLearning.

Figure 6-5, Question 23: The Adaptive Course Construction Methodology is...
One of the key challenges in facilitating the widespread adoption of personalised eLearning involves harbouring an understanding of the value-add of personalisation for the educational experience with the course developers. In order to fully appreciate this, one must understand the personalisation process. This helps to describe the different information sources which can affect personalisation and adaptivity and also how this information is then used to achieve personalisation and adaptivity. Narrative Attributes, as previously described, are a key component in facilitating this understanding as described in chapters four and five.

Figure 6-7 illustrates the participants’ response to a question regarding the role of the SMCS in the personalisation process based on a prior knowledge Narrative Attribute. 1 participant found the role of the SMCS in this personalisation process to be very clear. A further 2 participants found it clear with the remaining participant finding it confusing. This is an interesting spread of responses based on the open questions asked in regard to the user’s understanding of the roles of the SMCS in the personalisation process.

One participant’s comment in relation to the role of the SMCS was “Links topics, sub-topics by relationship / learning order / pre-requisites etc -not unlike an eLearning version of Microsoft
Another participant’s comment in relation to the role of the SMCS was “The subject matter concept space is used to graphically show how elements of a personalised lesson work. It is very useful in showing how the lesson works and how it can be personalised”. This indicates an understanding of how the concept-relationship structure of the SMCS is used by the Narrative Attribute to make decisions based on specified prerequisites of concepts in the SMCS and the concepts previously learned.

Figure 6-7, Question 19: The role of the Subject Matter Concept Space in the process of generating a personalised course was...

A key factor in authoring personalised eLearning is the ability to easily add adaptivity to an eLearning design through Narrative Attributes, thus facilitating the description of personalisable characteristics. Figure 6-8 illustrates the user’s understanding of the role played by Narrative Attributes in the personalisation process. 1 participant found the role of the Narrative Attributes in this personalisation process to be very clear. A further 2 participants found it clear with the remaining participant finding it confusing. This is in an interesting point since one participant’s comment in relation to Narrative Attributes was that they “show specific learner traits e.g. prior knowledge”. Another participant’s comment in relation to Narrative Attributes was “The narrative attributes allow you to personalise the overall structure of the lesson to suit the learning style of the learner for example”. These statements indicate a basic understanding of the role of Narrative
Attributes in the personalisation process. Based on further discussion with the trial participants it was identified that the confusion here also originated from the weak graphical representation of the SMCS in the Narrative Builder component of the ACCT. This is something that will be addressed going forward.

![Bar chart showing the clarity of design process](chart.png)

**Figure 6-8, Question 20: The role of Narrative Attributes in the process of generating a personalised course was...**

### 6.4.3.2 Clarity of Design Process (ACCT)

Secondly, the processes of the ACCT were examined. These core processes include: identifying, specifying and sequencing the learning methods and activities; modelling the subject domain; adding adaptivity to this design; and searching for and selecting appropriate candidate learning resources.

As illustrated in Figure 6-9, 1 of the trial participants found the process of designing an adaptive eLearning experience, based on the ACCM, using the ACCT very intuitive. A further 2 participants found the process intuitive with the remaining 1 identifying the adaptive course creation process, using the ACCT, as complex. This point is quite interesting based on discussions with the participants. Of the 3 participants who responded positively, 2 were “the eLearning developers” who were familiar with the process involved with creating eLearning and felt that the suite of ACCT tools was sufficiently intuitive to create an eLearning course. The remaining participant felt
that the ACCT’s suite of tools made the creation of eLearning a complex task. These figures illustrate the intuitiveness of creating customised pedagogy, using the ACCT, based on a foundation of Narrative Structures and Activities. It also depicts the ease at which the course developer can “add” adaptive properties to their eLearning design to create personalisable eLearning designs through the Narrative Builder component.

Two of the most complex and time consuming steps of creating a personalised eLearning course involve the identification, creation and representation of the subject area and the location and selection of appropriate candidate learning resources. The former provides the foundation for building contextualised personalised eLearning, so the course developer must be sufficiently supported in representing a subject area. The latter provides the ability to create instantiated personalised eLearning, so the course developer must have the ability to search locally and remotely for resources. Figure 6-10 illustrates the course developers understanding of representing a subject area using the ACCT. 2 of the trial participants found the process of representing a subject area very intuitive with the remaining 2 finding it intuitive. These positive responses indicate a high satisfaction rate from the participants regarding the SMCS builder component and the ability it provides in representing a subject area. The SMCS builder was one of the most popular components of the ACCT as one participant describes: “My favourite aspect of the tool was the
graphical representation of the subject matter concept space and the learning resources area. I found these very clear”.

When creating an instantiated contextualised personalised eLearning design, a key component is the identification and selection of appropriate learning resources. This is typically a time consuming and complex process. Based on the support offered by the ACCT in this regards, a series of questions were proposed. Figure 6-11, illustrates the response to those questions in which 3 of the participants found the learning resource identification and selection process of the ACCT to be very easy to understand and follow. The remaining 1 participant found this process easy to understand and follow. This is an interesting fact as the trial participants positively commented on their ability to locate appropriate candidate resources and, through the drag and drop interface of the Narrative Builder Component, select appropriate candidates for their personalised eLearning design. This gave the participants an overall feeling of control.
Without the ACCT and its generic personalised eLearning service (gAPeLS), the process of publishing an adaptive course was a time consuming and complex task involving the creation of web application as a front end for the PLS and the porting of all involved models into the memory space of the PLS. However, as described in chapter five, the ACCT alleviates these burdens by
providing a publication service facilitating a rapid prototyping approach to adaptive course creation. Figure 6-13 illustrates the participant’s responses to a question regarding the ease of publishing an adaptive course using the ACCT. 3 of the participants found the ACCT’s publication process very easy. The remaining 1 participant found it easy. One participant commented that adaptive course publication process was the easiest part of using the ACCT.

![Figure 6-13, Question 18: Publishing your adaptive course using the ACCT was...](image)

### 6.4.3.3 Course Developer’s Comments on the ACCT

The overall comments of the course developer were identified through a series of proposed questions which asked the following; in your opinion, what was the easiest part of adaptive course composition using the ACCT (Question 26)?; in your opinion, what was the most difficult part of adaptive course composition using the ACCT (Question 27)?; in your opinion, what was your favourite aspects of the ACCT interface (Question 28)?; and in your opinion, what was your least favourite aspects of the ACCT interface (Question 29)? The overall comments of the course developer were quite positive. Some participants found the publication of their personalised eLearning design easy while others found the selection of learning resources the easiest process, commenting: “The easiest part of course composition using the ACCT was the selection of learning resources. This was very clear and required very little training or help”. Other comments regarding favourite aspects of the ACCT include; “The visual layout of the screens”; and “My favourite...
aspect of the tool was the graphical representation of the subject matter concept space and the learning resources area. I found these very clear”.

On the contrary, the main negative responses regarding course developer comments reflect the complexities of the personalised eLearning and adaptive hypermedia jargon which was used. Even though a “jargon buster” session outlined and described all the “jargon”, it was still the main obstacle in using the ACCT and the main contributor to negative feedback.

6.4.4 Findings and Considerations

There are a number of key findings that arose from the evaluation of course developer satisfaction. The most important of these findings where:

- indications from the trial were that the course developers felt familiar with and comfortable with the processes of the Adaptive Course Construction Methodology
- these results have indicated that the trial participants felt empowered and supported by the ACCT’s implementation of the ACCM, however they felt that the technical language of the ACCT and personalised eLearning in general presented an significant obstacle to user acceptance
- indications from the trial were that the trial participants could understand the process of personalisation and how this can be designed using the ACCT, however they felt that the visualisation of this process in the ACCT was not sufficient
- the results presented here indicate that the trial participants felt satisfied with the level of support, both pedagogical and technical, offered by the ACCT during the design of a personalisable course, however they again felt that the support would have had a greater impact if it relied less on the technical language of AH

Due to the small number of trial participants, these findings can only be trusted as indications of course developer satisfaction as further evaluations consisting of larger participant numbers need to carried out.

Some general comments were as follows: “Overall I think the ACCT tool is excellent and the workshops were well organised and interesting. However, time was needed to digest the vast
amount of information and jargon." "The workshop was very productive and I enjoyed it. I feel I could have got more out of the learning experience if the tool had relied less upon the jargon of personalised e-learning. It was too difficult to learn all of this new terminology and to learn about the tool." "Overall it was very interesting and insightful into a completely new area." These comments indicate a general acceptance of the ACCT and its core functionality although the obstacle of "jargon" needs to be further addressed.

6.5 Scaffolding for Pedagogical Diversity

One of the core goals of the Adaptive Course Construction Methodology (ACCM) was to enhance the functionality of proven educational development processes using knowledge and data engineering techniques. This approach would improve flexibility and extensibility of the ACCM and would increase the potential for pedagogic diversity by founding an authoring environment on it. An integral part of pedagogic diversity is educational development support. This provides scaffolding for the teachers to create pedagogically effective learning experiences by promoting educational theory and instructional best practice. The following sections describe the evaluation of the flexibility and extensibility of PEDE and the separation and ease of reuse of the modelled disparate design elements of personalised eLearning development.

6.5.1 Flexibility and Extensibility of Personalised eLearning Development Environments (PEDE)

The design of personalised eLearning development environments is based on a flexible componentised architectural vision as described in the PEDE Architecture section in chapter 4. Each component is based on a specific element of the Adaptive Course Construction Methodology (ACCM) with an extension for multiple other possible information sources. The componentised architecture provides flexibility and extensibility to PEDE. This flexibility is then realised through the Architecture of the ACCT.

In comparison to the state of the art of personalised eLearning authoring, several benefits can be attributed to the PEDE approach. Since the approach described here extends that of the multi-model metadata driven approach to adaptive hypermedia [Conlan, 2005], there is an inherent flexibility to interact with and interpret multiple different modelled sources. In comparison, the five-layer
approach of LAOS identifies a fixed set of information sources which can be used to author personalisation. To be extensible, the LAOS approach must either extend the use of an existing layer or create a new individual layer. By extending an existing layer, you potentially recreate some of the traditional AH issues of mixing the concerns of different elements, for example, the original issue of embedding adaptive rules logic in content. On the other hand introducing another layer could potentially break the process flow of the LAOS model. So from this we can identify an inherent inflexibility in the LAOS approach to AH authoring. However, some of the issues of the PEDE approach arise from the potential for using unlimited modelled information sources; at what stage could the process become incomprehensible to the course developer due to cognitive overload. These issues need to be further addressed. REDEEM on the other hand allows the course developer to specify teaching and learner properties through fixed vocabularies. These approaches restrict both the flexibility and the extensibility of those authoring environments. In this case the addition of multiple information sources would require redesign of both the methodology and the authoring system.

The componentised approach of the ACCT also supports flexibility and extensibility. For example, the Narrative Builder component provides access to the various modelled information sources through its “palette of models”. This provides an easy to use approach for course developers to take existing models and customise and extend them for their purposes. If they so choose, the course developer can place their customised model back into the “palette of models” so that they can use it in future designs, supporting the extensibility and customisation of the ACCT’s support framework. If another modelled source becomes applicable to a specific personalised eLearning design, then this too can be added into the “palette of models” in this component, providing an extensible framework for adaptive course composition. This is also true of the SMCS Builder. This component supports the course developer in importing and customising SMCS models which allows the course developer to share their subject area descriptions with other course developers, creating potential for a community of practice.
6.5.2 Ease of Reuse of Key Elements of Personalised eLearning Development

As part of the driving research question involves "a set of extensible design elements, models and influences involved in the production of personalised eLearning experiences", separation and modelling are key components in achieving the goals and objectives of this research. However, this separation also provides the foundations for reusing the knowledge and information contained in the disparate design elements of personalised eLearning.

The ACCT actively supports reuse in many different ways and there are several different granularities of pedagogical reuse supported by the ACCT. Firstly the ACCT can support pedagogy-only reuse whereby the "narrative" concepts of a personalised eLearning design, i.e. the customised pedagogical structure, are reused. This is achieved by extracting only the "narrative" concepts from the PEDE narrative. Secondly, the ACCT supports contextualised pedagogy reuse where both the "narrative" and the "subjectmatter" concepts are reused. For example, a personalised eLearning design which teaches a Mechanics course based on the WebQuest pedagogy can be reused so that descriptive information such as, an "WebQuest Introduction" to "Newton's Third Law", can be extracted. Thirdly, the ACCT supports instantiated pedagogy reuse where the entire PEDE narrative is reused. The ACCT also supports the reuse of the SMCS in several ways. Firstly the SMCS can be exported and reused as part of an ACCT course. Secondly, it can be exported and reused as a graphical ontology by transforming it into SVG using the Model Translator/Interpreter component of the ACCT. Finally, the ACCT supports the reuse of the various other information sources involved in the production of personalised eLearning such as activity models and candidate learning resources. The following sections provide an analysis of the ease of reuse of the modelled disparate design elements of personalised eLearning.

Ease of Reuse of Narrative Structures

Narrative Structures form the pedagogical foundations of a personalised eLearning design. It therefore is essential that the reuse of this valuable and expensive source of information be facilitated and supported. The design and implementation of Narrative Structures impacts on their potential for reuse since their internal representation remains standards-independent. In the reviewed authoring environments in chapter three, pedagogy, and its many diverse forms, is not
explicitly used to support course development. REDEEM allows the manipulation of certain educational properties, e.g. learner characteristics and teaching strategies, but does not provide or support adaptivity of scaffolded pedagogy or instructional designs. The ACCT supports the reuse of Narrative Structures, i.e. the description and identification of the pedagogy, the concepts of the pedagogy and the workflow and the guidelines of the pedagogy. This is made possible by the ACCT’s ability to import and export elements of a personalised eLearning design.

As illustrated in Figure 6-14, the course developers, in general, found it quite easy to import models from an existing course package into the ACCT. Then using the Narrative Builder component of the ACCT, the course developer could edit and customise the pedagogical structure which had been imported. This ability shows that the ACCT can support reuse of pedagogical strategy in an easy and intuitive fashion. The facilitation of this reuse originates from the structural design of Narrative Structures (section 4.6.1) and its composite implementation (section 5.5.1).

Ease of Reuse of Activities

The design and implementation of Activity models aims to facilitate Activity reuse. Like Narrative Structures, reuse can occur at many different levels which include the description and concepts of the activity, the workflow and the roles of the activity and the guidelines and communication tools.
of the activity. The ACCT currently supports reuse of the activity at only the activity level, i.e. the activity must be reused as a whole. However, this restriction is implementation specific since the design of Activity models supports reuse at all levels of granularity, just as in the various other models of personalised eLearning development. As illustrated in Figure 6-9, the majority of trial participants found the process of building a PEDE narrative, including the selection of Activities, to be intuitive and easy. The facilitation of this reuse originates from the structural design of Activities (section 4.6.2) and their composite implementation (section 5.5.2).

Ease of Reuse of Subject Matter Concept Spaces

Reuse of a subject matter concept space can occur at various levels of granularity, as previously described. A SMCS can be reused in its entirety, meaning all concepts and defined relationships can be reused. This is the most common form of SMCS reuse. However, reuse of diverse ontological forms of SMCS is possible. For example, by reusing the “contains” related concepts of an SMCS we can share and reuse the hierarchical information about a specific domain of knowledge. Similarly, filtering the ontology by relationship type can produce subsets of the original ontology which can also be reused. The ACCT also supports reusing the SMCS in graphical form as previously mentioned. The ACCT can export the SMCS as an SVG file. This SVG file contains all the information of the ontology but represents the typically flat XML in graphical form with can be viewed through any SVG viewer. This allows course developers to share and collaborate on defining the scope of knowledge, even without having access to the ACCT.

As illustrated in Figure 6-10, the majority of course developers found the graphical representation of the SMCS, as provided by the ACCT SMCS Builder component, very intuitive to edit and customise. This indicates the ease of reuse of existing SMCS by course developers using the ACCT.

Ease of Reuse of Narrative Attributes

Narrative Attributes provide the vital component for the development of personalised eLearning designs. Therefore, reusing them was a key goal of the research described in this thesis. Reuse of Narrative Attributes includes the concepts, the descriptions, the workflow, the guidelines and the
selectors of the narrative attribute. The ACCT supports this reuse in the Narrative Builder component where the course developer can drag and drop axes of personalisation into their eLearning design to request adaptive features. As illustrated in Figure 6-9, the majority of trial participants found the process of building a personalised eLearning design, including the selection and association of personalisation axes, to be intuitive and easy. The facilitation of this reuse originates from the structural design of Narrative Attributes (section 4.6.4) and their composite implementation (section 5.5.4). The understanding of this modelled structure can also be identified in Figure 6-8.

**Ease of Reuse of Resource Candidates**

Although the modelling of resource candidates is not specifically attributed to this research, the facilitation of their reuse increases the potential for producing affective educational experiences and can significantly reduce the time and complexity in producing the experience. Reuse of candidate resources is supported by the ACCT through its Resource Search component. The course developer can search for existing candidate resources and, through the drag and drop interface of the ACCT, select appropriate candidates for their personalised eLearning design. As illustrated in Figure 6-9, the majority of trial participants found the process of building a personalised eLearning design, including the selection appropriate resource candidates, to be intuitive and easy; "The easiest part of course composition using the ACCT was the selection of learning resources. This was very clear and required very little training or help". The facilitation of this reuse originates from the structural design of resource candidate groups (section 4.6.5) and their composite implementation (section 5.5.7).

**Ease of Reuse of PEDE Narrative**

Reuse of the PEDE Narrative is based on the reuse aspects of the above mentioned models of personalised eLearning development. However, the reuse of custom personalised eLearning designs can occur at many different levels of granularity as previously mentioned. By reusing PEDE Narrative at a pedagogical level, the course developer can share and reuse customised pedagogical frameworks which can then, for example, be used by the ACCT to support a customised pedagogical approach. This approach typically reuses "narrative" concept types and defined activity structuring. Another granular level of reuse is contextualised pedagogy. This involves all
elements of "pedagogical" level reuse but also includes the "subjectmatter" concepts of the PEDE Narrative thus providing domain context specific information. The ACCT also supports reuse of instantiated PEDE narrative. The "instantiated" version includes all elements of "contextualised" PEDE narrative but also includes abstractions to the appropriately selected candidate resources. These different levels of reuse, supported by the ACCT, provide a solid support framework and scaffolding for diverse pedagogical reuse. This level of pedagogical reuse is not available in any of the state of the art personalised eLearning environments.

6.5.3 Findings and Considerations

Based on the evaluation of this research's scaffolding for pedagogical diversity, a number of interesting findings were identified. The most significant of these findings where

- indications from the trial were that the course developer felt both empowered and supported through the ease of reuse of disparate design elements of personalised eLearning development and the technical transparency which the ACCT champions.
- indications from the trial were that the flexibility and extensibility of the architecture of PEDE afforded the development of componentised ACCT to empower and support the teachers, however further evaluation is required to identify at what point the level of extensibility becomes a cognitive obstacle to the course developer

Although the size of the evaluation is small, the results are very positive which would lead to a strong indication that the ACCT promotes the ease of reuse of the key elements of personalised eLearning development. This is reflected by the highly positive responses of the trial participants; in the majority of questions, three out of the four participants answered either positively or very positively. Another point here though, is that the ACCT has only been used to create personalised eLearning in a limited content space. The flexibility of the ACCT to interact with multiple other content areas must be addressed.

6.6 Usability

The technical evaluation of this research aimed to identify the usability issues with the Adaptive Course Construction Toolkit (ACCT) in contrast to the current state of the art of systems for authoring personalisation. The metrics for this usability evaluation were based on the eight golden
rules of interface design [Shneiderman, 1998]. A series of questions, based on a modification of the Questionnaire for User Interaction Satisfaction (QUIS) [Chin et al., 1988], were formulated in order to illicit this information from the various trial participants. These questions focused on the terminology used, the use of informative feedback, error prevention and error correction and the general usability of the system. Questions 30 (a)(b) addressed the consistency of the terminology used by the ACCT, see APPENDIX IV – Trial Questionnaire. As illustrated in Figure 6-15, the terminology used by the ACCT, based on this evaluation, was very consistent. In contrast, however, when discussing this with the trial participants they raised an important issue. Although the terminology used by the ACCT was consistent, in their opinion the terminology was consistently confusing. As part of each of the workshops carried out, a jargon buster presentation had been given. Even so, the participants felt that the terminology used by the ACCT, and personalised eLearning in general, was confusing since they could not relate to it. For example, some participants called the SMCS a concept graph in their educational development experience. These semantics created an obstacle to user acceptance which will need to be addressed going forward.

![Figure 6-15, Questions 30(a)(b): Consistency of the Terminology used by the ACCT](image)

The ACCT’s use of informative feedback was also analysed. This examined the instructions offered by the ACCT in order to complete certain actions; the system feedback and closure when
performing an action or series of actions; and the use of state-oriented visual cues such as cursor changes and mouse over tool tips. As illustrated in Figure 6-16, the user’s response to those questions, based on this evaluation, regarding informative feedback proved to be very positive.

Another key requirement for a usable system is to provide error correction mechanisms and where possible, try to prevent errors from happening. The error correction mechanisms can either be automatically performed by the system when it is a system specific error or through the use of undo and redo functionality for user errors. The ACCT’s use of error prevention and error correction were evaluated through a series of questions which focused on the frequency of system errors, the predictability of performing actions and the ability to undo and redo action sequences with the ACCT. Questions 34 (a)(b)(c)(d) evaluated the ACCT’s error prevention capabilities and questions 35 (a)(b)(c) evaluated the ACCT’s error correction capabilities, see APPENDIX IV – Trial Questionnaire. As illustrated in Figure 6-17, Figure 6-18, Figure 6-19, Figure 6-20, Figure 6-21, Figure 6-22, Figure 6-23, Figure 6-24 and Figure 6-25, the trial participants felt that both the error prevention mechanisms of the ACCT and also the error correction capabilities of the ACCT were appropriate and easy, respectively.
Figure 6-17, Question 34: The use of error prevention and error handling was...

Figure 6-18, Question 34(a): Operations were...

Figure 6-19, Question 34(b): System failures occurred...

Figure 6-20, Question 34(c): System warns you about potential problems...

Figure 6-21, Question 34(d): Instructions for correcting errors were...
Based on these results, the general usability of the ACCT, from a Human-Computer Interaction (HCI) perspective, is quite proficient. This indicates that this implementation of PEDE in the form of the ACCT is user friendly and affective.

### 6.6.1 Findings and Considerations

Based on the usability evaluations of the ACCT, a number of interesting findings were identified. These include,

- Indications from the trial were that the general usability of the ACCT was highly rated, indicating the appropriate use of error correction mechanisms, error prevention mechanisms and use of informative feedback.

- However, the course developers felt that the terminology (technical language), although consistent, was too confusing and presented the key obstacle to effectively and efficiently using the ACCT.
These findings and considerations provide a positive indication of the usability characteristics of the ACCT. Even though the number of participants, for the trial presented in this thesis, was only four, from a pure usability perspective, the findings represent a positive indication of the usability of this research. This, coupled with the series of previously performed mini-evaluations, involving user testing and acceptance testing, adds to the significance of these findings.

6.7 Overall Analysis

The purpose of this section is to provide an insight into the significance of this evaluation and provide an unbiased and balanced general analysis. The evaluation chapter strove to address the goals and objectives of this research as formed by the driving research question. These being: to identify and specify requirements for the elements and influences of personalised eLearning experiences; to specify an integration process for elements and influences of personalised course composition; to develop a suite of tools which support crucial aspects of the personalised eLearning development process; to enhance the efficiency of the specified integration process; to evaluate the process and suite of tools to support teachers in developing personalised eLearning.

Several areas of this research were evaluated, namely; the course developer’s satisfaction with the design process; the scaffolding for pedagogical diversity offered by the PEDE architecture and the ACCT; and the usability of the ACCT.

As previously described, a number of iterative mini evaluations where carried out based on prototyped components which helped to capture some of the course development issues and also a significant number of usability related issues. However, one may argue the significance of the findings presented in this thesis based on the number of trial participants and their demographics. There were four users in total. Although this is not statistically significant, it can provide indications to the possible success of this research. They were all associated with the same company, Intel Ireland. They all had similar educational backgrounds, developing IT Education for secondary school levels. They all had similar vested interests in this research. Based on these critical points, a further series of trials have been planned to strengthen and support the thesis findings. These planned trials examine a larger and wider population of teacher with a broader educational background and a higher level of expectation, as addressed in chapter seven. Since the
submission of this thesis, another phase of evaluation has been carried out on 17 4th year ICT
students in Trinity College, Dublin. A tutorial will also be presented, based on this research, at the
4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems

A key factor in the success of research is peer recognition. Based on the work described in this
thesis, a total of thirteen peer reviewed international conference, workshop and journal papers can
be attributed. Several awards have also been won such as best paper award for the last two years at
the International Workshop on Authoring Adaptive and Adaptable Educational Hypermedia
(A3EH), the international forum for authoring educational adaptive hypermedia, short listed for best
paper award at E-Learn 2003, World Conference on E-Learning in Corporate, Government,
Healthcare and Higher Education; short listed for best paper award at EdTech 2005, the Sixth
Annual Irish Educational Technology Users’ Conference; and best presentation at the First Intel
7 Conclusions

7.1 Introduction

This thesis presented the research into Personalised eLearning Development Environments. This innovative approach proposed the development of a framework and set of tools to empower teachers in creating pedagogically rich personalised eLearning experiences. The facilitation of which, was realised through the extension of proven educational development processes and a specification for personalised eLearning development environments.

The purpose of this chapter is to discuss the goals and objectives of this thesis and to illustrate how they have been achieved. It also identifies the contributions of this research to the state of the art of Adaptive Educational Hypermedia Systems and more specifically authoring environments for building personalised and personalisable educational hypermedia. Finally, it concludes with a discussion of the possible future work in extending the research of this thesis.

7.2 Objectives and Achievements

The research question posed in this thesis was to identify and investigate the key extensible design elements, models and influences required for the composition of personalised eLearning experiences and to develop an integrated, model driven environment which facilitates rapid personalised eLearning development by educators or teachers (non-technical).

In addressing the goal of “identifying and specifying requirements for the elements and influences of personalised eLearning experiences”, an integral part of this research involved the identification of and specification of model schema to represent the diverse influences of personalised eLearning development, such as pedagogy and instructional design, activities and communications, subject areas, personalisation axes and candidate resources. The achievement of this objective has been documented by the design and implementation (sections 4.5 and 5.5) of model schema to represent the key elements of personalised eLearning, thus providing the model framework which supports the application of this research.
In addressing the goal of “specifying an integration process for elements and influences of personalised course composition”, the Adaptive Course Construction Methodology was formulated. The ACCM provided the base educational framework for this research. It described a number of key processes involved in the creation of any educationally sound learning experience, such as the constructive alignment of learning outcomes with assessment metrics and teaching strategies. This educational framework was extended and redesigned (section 4.3) to incorporate all aspects of the model framework in order to provide a solid pedagogical development environment based on separated knowledge models. The achievement of this goal is documented in the evaluation of this research (6.4.3).

In addressing the goals of; “developing a suite of tools which support crucial aspects of the personalised eLearning development process”; “to enhance the efficiency of the specified integration process”; and “to evaluate the process and suite of tools to support teachers in developing personalised eLearning”, the specification for a technical framework was designed to utilise the processes and pedagogic scaffolding of the educational framework while using graphical abstractions to provide a level of technical transparency to the model framework. This was formulated to provide the requirements for Personalised eLearning Development Environments (PEDE). The implementation of PEDE saw the development of a personalised eLearning development environment named the Adaptive Course Construction Toolkit (ACCT). The ACCT provides a suite of tools to support non-technical course developers in creating personalised eLearning designs. It also enhances the efficiency of authoring personalised eLearning by reducing the time and complexity of the composition process. The achievement of these goals is documented in the evaluation of this research (6.4.3).

### 7.3 Contribution to the State of the Art

Personalised eLearning Development Environments, as an approach to authoring personalised eLearning, is the primary contribution to the State of the Art made by this thesis and the work described therein. This approach is significantly different to that used by current personalised eLearning authoring tools.
Firstly, the integration process, which the personalised eLearning development environment was based on, enhances modern educational development practices for personalised eLearning. This is a novel approach which no existing authoring environment for personalised eLearning development, based on the state of the art review, has taken. The enhancement provides better pedagogical as well as user support while still providing both an easy way of integrating the key elements of personalised eLearning development and mechanisms for the rapid prototyping and testing of complete personalised eLearning designs.

Secondly, the ACCT has demonstrated that you can offer quite sophisticated levels of personalisation authoring to a non-technical course developer. The ACCT provides intuitive, user friendly and technically transparent interfaces and components which support the course developer through the process of personalised eLearning composition. Finally, the ACCT supports the rapid prototyping and testing of complete personalised eLearning designs which no existing system supports.

Based on these significant differences, the key contributions of this research to the state of the art is the innovative integration of pedagogic and instructional approaches, adaptive properties, services and subject area designs in one authoring system for personalised eLearning which, based on user evaluations, has demonstrated its suitability for non-technical course developers. The ACCT was the first Personalised eLearning Development Environment to offer real pedagogic design support while providing flexible scaffolding for pedagogic diversity. The support for pedagogic design is based on the knowledge framework, more specifically the modelled pedagogic strategies in the form of Narrative Structures. Unlike any existing approach to personalised eLearning development, the ACCT provides guidance and workflow recommendations to the non-technical course developer based on the modelled pedagogy. The architecture of the ACCT supports multiple information models, therefore a range of diverse pedagogic strategies can be supported; if it can be modelled then it can be used. Another key contribution is the ACCT’s use of personalisation services in the form of Narrative Attributes. Existing environments require a level of technical input from the course developer which they typically do not possess, i.e. rules programming. The ACCT on the other hand uses personalisation services, based on axes of personalisation, to achieve adaptivity. In this way, the course developer simply makes an association of a personalisation axis
with sections of their eLearning design. The transformation of the course developers personalised eLearning design automatically generates the required rules based on this design. In this way, the technical difficulties of generating rules for adaptive properties become transparent to the course developer. Another key contribution from this research to the state of the art is the ability to graphically design and represent subject area descriptions and then use these descriptions in the generation of personalised eLearning. To the author's knowledge, prior to the development of the ACCT no authoring tool for personalised eLearning supported the course developer with intuitive graphical tools to create "ontologies" of subject area. This innovation significantly reduces the complexities and time constraints in producing flexible personalised educational experiences.

The affects of this research have also been witnessed by its direct contribution to over thirteen international journals and conferences. The most significant of which are –


7.4 Future Work

There are a number of areas in which the work described in this thesis may be taken forward. Firstly, since this research focuses on empowering course developers in effectively creating personalised eLearning experiences, an extended validation of this applied research would be a definite benefit. This would look at diversifying the demographics of the trial participants and conducting an ongoing series of trials. Secondly, the PEDE implementation in the form of the ACCT will strive to incorporate all aspects of the Adaptive Course Construction Methodology, including the currently absent processes of defining learning outcomes and specifying assessment metrics based on these learning outcomes. Another area of potential, going forward, involves the enhancement of PEDE delivery platforms. This includes specifications for implementing PEDE in different environments such as a web based version of the ACCT or an eclipse plugin of the ACCT. This work will also look at the potential benefits of “adapting” the ACCT’s interface and processes to the preferences and requirements of the course developer. Recently, the identification of adaptive hypermedia authoring patterns have been examined and are becoming a prominent research area in authoring tools for composing adaptive hypermedia. This area could be potentially beneficial to the research detailed in this thesis by providing validation mechanisms for the processes of the ACCM as implemented in the ACCT.

7.4.1 Extended Validation of Applied Research

As described in the evaluation chapter of this thesis, although several key areas were evaluated, a number of enhancements to the evaluation strategy of this research can be achieved. These relate to
such things as the number of trial participants, the satisfaction of the course developer, the supported scaffolding for pedagogic diversity and the usability characteristics of the ACCT.

In relation to the number of trial participants, several issues will be addressed. Firstly, using a larger number of trial participants for future evaluations is proposed. It is hoped that this will increase the statistical significance of evaluation results. Secondly, it is proposed that we perform an iterative series of small trials to address the diverse participant backgrounds towards which this research is focused. This involves a range of participants from non-technically minded course developers to technically minded course developers and adaptive systems engineers. These trials will focus on the core aspects of this research, from course developer satisfaction to scaffolding for pedagogic diversity to usability.

In relation to course developer satisfaction, it is proposed that this research will examine a broad range of course developers based on diverse experience levels and user expectations. In doing so it proposes that we address the processes of the ACCM in finer detail, i.e. examine each of the ACCM processes in isolation and in conjunction with the other processes of the ACCM. This will further help to analyse the acceptance levels of the ACCM as an unbiased and acceptable course construction methodology. There will also be a tighter focus on addressing the processes of the ACCM as implemented through the ACCT. An area which was not examined initially is the satisfaction of the course developer's expectations with the outputs of the ACCT. This will evaluate the personalised eLearning experience produced by its publication through gAPeLS and the course developer's expectations, an important issue to be addressed in driving this research forward.

Further evaluation of the scaffolding for pedagogic diversity supported by this research will be carried out. This will involve further examination of the flexibility and extensibility of the PEDE architecture and the extension of the number of pedagogies supported by the ACCT. A finer grained analysis of the "ease of reuse" of the modelled diverse elements and influences of personalised eLearning development will be carried out. This will detail all aspects of the models, the architecture and ACCT implementation which both facilitate and more importantly impede the ease of reusing these vital knowledge resources. A key component of this involves an in-depth analysis of the pedagogical support framework provided through this applied research. This
includes such things as the effectiveness of the guidelines and workflow descriptions in supporting the design of pedagogically driven personalised eLearning experiences. It will also further address the satisfaction of the course developer with this support framework.

Since a key objective of the research described in this thesis is to make personalised eLearning development environments usable by the non-technical course developer, an iterative process of evaluating the usability of the ACCT must be established in order to progressively move this research forward. This will be achieved through a number of small usability trials focused on identifying the usability issues and designing innovative solutions for the future, inevitably providing the requirements for the next generation of PEDE.

7.4.2 Complete Implementation of the ACCM

Currently, the ACCT implements the most time consuming and technically complex processes of adaptive course composition, namely the pedagogical design and activity sequencing, the design and modelling of subject area, the selection of appropriate resources and services, the association of personalisation axes and the rapid prototyping of personalised eLearning designs. However, the process of specifying learning outcomes is currently not explicitly supported by the ACCT. In moving this research forward, a key goal will be to address and realise all processes of the ACCM and provide supporting components for carrying out each process using the ACCT. One could achieve the integration of learning outcomes and assessment metrics into current PEDE implementations by: i) allowing the course developer to define a set of learning outcomes for the personalised eLearning design; ii) associating the learning outcomes with activities or concepts in the design; iii) realising the generation of assessment metrics based on “assessment templates”, the pre-defined learning outcomes and the activities and concepts of the personalised eLearning design.

7.4.3 Platform Enhancements

The current implementation of the ACCT is entirely java based. The graphical components are realised through various API’s written in Javax SWING. One approach to maximising the potential user base for the ACCT is to enhance the diversity of supported delivery platform. This will include the specifications for a web based implementation of the ACCT, an Eclipse [IBM Eclipse] plugin of the ACCT and an XML based specification of the ACCT. The XML based version of the ACCT will potentially be the first “adaptable” personalised eLearning development environment. The idea
of which is to adapt the “usability” characteristics of the ACCT’s interface (visualisations, action sequences, informative feedback messages and error correction mechanisms) and adapt the levels of support offered to the course developer (non-technical developer vs. technical developer) in a similar fashion to the current processes of adapting educational experiences to the learners.
BIBLIOGRAPHY


[Blackboard] available online at http://www.blackboard.com


[BPML] available online at http://www.bpmi.org/BPML.htm


[BRICKS] Building Resources for Integrated Cultural Knowledge Services (BRICKS), European IST FP6 Integrated Project IST-2002-2.3.1.12 Technology-enhanced learning and access to cultural heritage, information available online at http://www.brickscommunity.org/prj


[Connexions] Sharing Knowledge and Building Communities. Available online at http://cnx.rice.edu/


[CopperAuthor] CopperAuthor, an editor for IMS Learning Design, Available at http://sourceforge.net/projects/copperauthor/

[CopperCore] available online at http://coppercore.org/


212


[Design Principles for Educational Software] available online at http://www.design-principles.org


[Dublin Core] The Dublin Core Metadata Initiative is an open forum engaged in the development of interoperable online metadata standards that support a broad range of purposes and business models, available online at http://dublincore.org/


[IBM Eclipse] IBM Eclipse project is “an open source community whose projects are focused on providing an extensible development platform and application frameworks for building software” available online at http://www.eclipse.org/

[EdNA] Education Network Australia Online. Available at http://www.edna.edu.au


[EML] available online at http://eml.ou.nl/eml-ou-nl.htm

[EPD] Educational Development Plan, description available online at http://www.careerprep.org/edps.htm

[ePortfolio] a tool to aid lifelong learning, available online at http://www imsglobal.org/ep/


[eXist] an XML Database, available online at http://www.exist.org


219


[IMS Content Package] Available online http://www.imsglobal.org/content/packaging/

[IMS GC] available online at http://www.imsglobal.org


[IMS LD BPG] IMS LD Best Practice Guide available online at http://www.imsglobal.org/learningdesign/ldv1p0/imsld_bestv1p0.html

[IMS SS] a mechanism for providing simple sequencing of content objects available online at http://www.imsglobal.org/simiplesequencing/

[IMS QTI] available online at http://www.imsglobal.org/question/index.cfm

[JDOM] Java API for interacting with and manipulating DOM structures in XML, available online at http://www.jdom.org
[JGraph] Java API for developing graphs, available online at http://www.jgraph.org

[JESS] available online at http://herzberg.ca.sandia.gov/jess/


[LCSH] Library of Congress Subject Heading, a taxonomy of subject headings, available online at http://www.itsmarc.eom/crs/shed0014.htm
[LISP] LISP Processing Language, information available online at http://www.lisp.org


[Lydia] LydiaLearn, provides global access to eLearning resources. Available online at http://www.lydialearn.com/


[Metasaur] University of Sydney’s Metasaur Project. Demonstration available online at www.it.usyd.edu.au/~alum/demos/metasaurhci/


[MyET] My English Teacher (MyET),
http://wwwis.win.tue.nl/~alex/MyEnglishTeacher/TeachersSite/index.html


[OWL] Web Ontology Language for the semantic web available online at http://www.w3.org/TR/owl-ref/


226


[PLE] Personalised Learning Environment, a project funded by JISC, information available online at, http://www.cetis.ac.uk/members/ple/resources/ple_summary


[RELOAD] available online at http://www.reload.ac.uk/

[RuleML] Rule Mark-up Language, available online at http://www.ruleml.org/


[SCORM] Sharable Content Object Reference Model available online at http://www.adlnet.org/


[UML] Unified Modelling Language, information available online at http://www.uml.org/


[WebCT] available online at http://www.webct.com/


[Xindice] an XML Database, available online at http://www.xindice.org

[XPath] XML Path Language, available online at http://www.w3.org/TR/xpath

[XSLT] Language for transforming XML, available online at http://www.w3.org/TR/xslt

### GLOSSARY

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstraction</td>
<td>When applied to narratives abstraction is the mechanism that keeps the concepts used in the narrative separate from the identifiers for individual learning resources. This enables narratives to be repurposed across different content bases.</td>
</tr>
<tr>
<td>ACCT</td>
<td>Adaptive Course Construction Toolkit. The first prototype of the Personalised eLearning Development Environment.</td>
</tr>
<tr>
<td>Activity</td>
<td>Activity, in the context of this thesis, describes a series of tasks to be completed, a workflow or process sequence for how those tasks should be addressed and a series of communication tools and services which can support these tasks. The goal of which is to more actively engage the learner.</td>
</tr>
<tr>
<td>Adaptive</td>
<td>Adaptive programs adapt to a particular model or set of models, e.g. the skill level or preferences of the learner.</td>
</tr>
<tr>
<td>Adaptive Course Construction Methodologies</td>
<td>An extension of a Course Construction Methodology which incorporates additional processes of: pedagogical theory and instructional design modelling; personalisation axes modelling; domain and service identification and selection; and user modelling. Abbrev. ACCM.</td>
</tr>
<tr>
<td>Adaptive Educational Hypermedia</td>
<td>Involves the specific application of the architectures, the methods and the techniques of AH to eLearning and e-education. There are subtle differences here in that the end user, the person who should be empowered by such systems, is the learner. The type of adaptivity that is used in such systems should be based on the application of a certain type of strategic knowledge, namely educational strategy or pedagogy. Abbrev. AEH.</td>
</tr>
<tr>
<td>Adaptive Hypermedia</td>
<td>Adaptive Hypermedia systems tailor information to the user and may guide the user in the information space to present the most relevant material. Abbrev. AH, AHS.</td>
</tr>
</tbody>
</table>
| Associationist/Empiricist perspective to understanding learning | As described in [Mayes & de Freitas, 2004]: Learning is viewed as activity, the process of connecting the elementary mental or behavioural units, through sequences of activity. In this approach, knowledge is an organised accumulation of associations and skill-components. The basic principle is that competence in advanced and complex tasks is built step by step from simpler units of knowledge or skill, finally adding coordination to the whole structure. This involves several steps such as:  
1) Analyse the domain into a hierarchy of small units,  
2) Sequence the units so that a combination of units is not taught until its component units are grasped individually,  
3) Design an instructional approach for each unit in the sequence. |
<table>
<thead>
<tr>
<th>Candidate</th>
<th>One of the disparate design elements of personalised eLearning development (Narrative Structure, Selector, Learning Resource, Narrative Attribute, Activity etc.) that fulfils a particular (usually educational or technical) function. Candidates are considered a model and have associated metadata.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate Group</td>
<td>A group of candidates. Groups are formed by grouping candidates that fulfil the same function, but perform it in different ways. Candidate Groups are metadata-only models. Their metadata contains the identifiers of their constituent candidates.</td>
</tr>
<tr>
<td>Candidate Selector</td>
<td>A form of narrative that is capable of reasoning across the metadata of candidate models in a candidate group. This reasoning may utilise any model in the multi-model approach to select the most appropriate candidate(s) for a given function.</td>
</tr>
<tr>
<td>Cognitive perspective to understanding learning</td>
<td>As described in [Mayes &amp; de Freitas, 2004]: Learning is viewed as achieving understanding, the output of an individual’s attention, memory and concept formation processes. This approach provides a basis for analysing concepts and procedures of subject matter curricula in terms of information structures. The design principles include i) ownership of the task, ii) coaching and modelling of thinking skills, iii) scaffolding, iv) guided discovery, v) opportunity for reflection, vi) ill-structured problems.</td>
</tr>
<tr>
<td>Competencies</td>
<td>A structured list of knowledge, skills and attitudes that are possessed or required by a learner. Competencies are used as the foundation to guide needs analyses and evaluations.</td>
</tr>
<tr>
<td>Concept Space</td>
<td>A framework for representing knowledge about a subject area. This means that the conceptual knowledge about subject concepts and their relationships with other concepts in the area can be consolidated in ontological and accessible form.</td>
</tr>
<tr>
<td>Constructive Pedagogies</td>
<td>Pedagogies which focus on constructivist learning. See Associationist/Empiricist for a description of this perspective on learning.</td>
</tr>
<tr>
<td>Course Construction Methodology</td>
<td>A process of creating an educational experience which involves the constructive alignment of learning outcomes with assessment metrics and activity sequences. Abbrev. CCM.</td>
</tr>
<tr>
<td>Curriculum (Constructive) Alignment</td>
<td>Involves the alignment of learning outcomes with assessment metrics and activity sequences.</td>
</tr>
<tr>
<td>Hypermedia</td>
<td>The word hypermedia implies a combination of “Hyperspace” and “Multimedia”. Hyperspace prescribes an existence in more than three dimensions. Using a hyperlinked structure provides the user with the ability to “jump” across links to reach resources at the far reaches of hyperspace. Multimedia refers to applications that combine text, graphics, full-motion video, and sound in an integrated package.</td>
</tr>
<tr>
<td>Learning Management System</td>
<td>A program that manages the administration of training and eLearning. Typically includes functionality for course catalogues, launching courses, registering students, tracking student progress.</td>
</tr>
<tr>
<td><strong>Learning Outcomes</strong></td>
<td>A statement which clearly identifies the skills, knowledge and/or understanding that a learner will be able to demonstrate as a result of successfully completing an identified part of a learning programme in order to gain credit for a unit.</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Metadata</strong></td>
<td>Information that provides macro-level details about an object, such as author, title, subject, date created, etc. Usually used to describe models.</td>
</tr>
<tr>
<td><strong>Model</strong></td>
<td>A representation of an object, process, behaviour or attitude.</td>
</tr>
<tr>
<td><strong>Multi-Model, Metadata-driven approach</strong></td>
<td>An approach for producing adaptive systems. This approach separates elements of adaptivity into discrete models. Utilises metadata to reason about models. Uses a generic engine to execute process models (narratives) and access other model metadata.</td>
</tr>
<tr>
<td><strong>Narrative</strong></td>
<td>A process model that in personalised eLearning systems, such as APeLS, reconciles input models to form an output. May be considered the embodiment of adaptive logic and reasoning.</td>
</tr>
<tr>
<td><strong>Pedagogic Diversity</strong></td>
<td>Typically, pedagogical models address a single educational perspective. However, in reality, learning can rarely be placed into a single category. Therefore pedagogic diversity describes an ability to accommodate the dynamic nature of learning by not limiting pedagogic approaches.</td>
</tr>
<tr>
<td><strong>Pedagogy</strong></td>
<td>Theory of education. The educational strategies, instructional techniques, and teaching approaches that teachers can use to facilitate learning.</td>
</tr>
<tr>
<td><strong>Personalisation</strong></td>
<td>The process of being adaptive towards a user's characteristics and preferences. Personalised eLearning generally refers to personalisation towards a learner's objectives, prior knowledge and learning preferences.</td>
</tr>
<tr>
<td><strong>Personalised eLearning</strong></td>
<td>The application of personalisation to online learning experiences. The goal here is to ensure that the application of personalisation does not undermine the pedagogy of the eLearning.</td>
</tr>
<tr>
<td><strong>Situative</strong></td>
<td>As described in [Mayes &amp; de Freitas, 2004]: Learning is viewed as being influenced by the social and cultural setting in which the learning occurs (socio-psychological and communities of practice). This approach emphasises assessing participation, authenticity of practice, peer assessment.</td>
</tr>
<tr>
<td><strong>Virtual Learning Environment</strong></td>
<td>A virtual learning environment is a set of teaching and learning tools designed to enhance a student's learning experience by including computers and the Internet in the learning process. Abbrev. VLE. See Learning Management System, also.</td>
</tr>
</tbody>
</table>
APPENDICES

APPENDIX I – Psychology and Cognition

Blooms Taxonomy

Blooms' taxonomy of educational objectives, describes six different levels of objectives as a hierarchy, namely, knowledge or recall, comprehension, application, analysis, synthesis and evaluation.

Knowledge, in blooms' taxonomy, is measured as the learner’s ability to recall or state information and forms the lowest level of the hierarchy. An example of this would be the ability to recite a poem or the ability to state a mathematical formula. Comprehension, in bloom's taxonomy, is the ability to give meaning to information. More often this level occurs concurrently with knowledge but in certain cases, knowledge is a prerequisite of comprehension. An example of this would be the ability to explain what a line in a poem means or the ability to paraphrase a given mathematical formula. Application, in Bloom’s taxonomy, is the ability to use knowledge or principles in new or real-life situations, in essence rules using not problem solving. An example of this would be to identify an example of a metaphor in a poem or to recognise an isosceles triangle. Analysis, in bloom’s taxonomy, is the ability to decompose complex information into smaller parts and understand the relationships between the parts. An example of this would be the ability to identify the poetic strategies used in a given poem or the ability to determine the strategies that would be necessary to solve a mathematical word problem. Synthesis, in Blooms’ taxonomy, describes the ability to create something novel by integrating information that has been learned through the lower levels of the taxonomy. An example of this would be to write an essay or poem or integrate several
different strategies to solve a mathematical problem. Evaluation, in Blooms' taxonomy, describes the ability to perform a comparison based on cognitive standards, previously developed through the lower levels of the taxonomy. An example of this would be the ability to analyse a peer's essay based on principles of composition or the ability to determine the degree of efficiency when solving a mathematical problem.
APPENDIX II – Models, Schema and Transforms

Narrative Structures

Model
<concept xmlns:xlink="http://www.w3.org/1999/xlink" id="001" type="holder">
  <name lang="en">Sample Pedagogical Narrative Models</name>
  <description lang="en" />
  <relationships>
    <relationship xlink:href="#010">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#1-N" />
      <relationship_description lang="en" />
    </relationship>
    <relationship xlink:href="#020">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#1-N" />
      <relationship_description lang="en" />
    </relationship>
    <relationship xlink:href="#030">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#1-N" />
      <relationship_description lang="en" />
    </relationship>
  </relationships>
</concept>

<concept id="010" type="model">
  <name lang="en">Case Study</name>
  <description lang="en" />
  <relationships>
    <relationship xlink:href="#0101">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#1-1" />
      <relationship_description lang="en" />
    </relationship>
    <relationship xlink:href="#0102">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#1-1" />
      <relationship_description lang="en" />
    </relationship>
    <relationship xlink:href="#0103">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#1-1" />
      <relationship_description lang="en" />
    </relationship>
    <relationship xlink:href="#0104">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#1-1" />
      <relationship_description lang="en" />
    </relationship>
    <relationship xlink:href="#0105">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#1-1" />
      <relationship_description lang="en" />
    </relationship>
    <relationship xlink:href="#0106">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#1-1" />
      <relationship_description lang="en" />
    </relationship>
    <relationship xlink:href="#0107">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#1-1" />
      <relationship_description lang="en" />
    </relationship>
    <relationship xlink:href="#0108">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#1-1" />
      <relationship_description lang="en" />
    </relationship>
    <relationship xlink:href="#0109">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#1-1" />
      <relationship_description lang="en" />
    </relationship>
  </relationships>
</concept>
</relationship>
<concept id="0101" type="narrative">
  <name lang="en">Introduction</name>
  <description lang="en"/>
  <relationships/>
</concept>
</candidates/>
</concept>
<concept id="0102" type="narrative">
  <name lang="en">Context</name>
  <description lang="en"/>
  <relationships/>
</concept>
</candidates/>
</concept>
<concept id="0103" type="narrative">
  <name lang="en">Problem</name>
  <description lang="en"/>
  <relationships/>
</concept>
</candidates/>
</concept>
<concept id="0104" type="narrative">
  <name lang="en">Resources</name>
  <description lang="en"/>
  <relationships/>
</concept>
</candidates/>
</concept>
<concept id="0105" type="narrative">
  <name lang="en">Activities</name>
  <description lang="en"/>
  <relationships/>
</concept>
</candidates/>
</concept>
<concept id="0106" type="narrative">
  <name lang="en">Case Tools</name>
  <description lang="en"/>
  <relationships/>
</concept>
</candidates/>
</concept>
<concept id="0107" type="narrative">
The WebQuest Pedagogical Model, as described by Bernie Dodge, is illustrated through the following conceptual layout.
This document should be written with the student as the intended audience. Write a short paragraph here to introduce the activity or lesson to the students. If there is a role or scenario involved (e.g., "You are a detective trying to identify the mysterious poet.") then here is where you'll set the stage. If there's no motivational intro like that, use this section to provide a short advance organizer or overview. Remember that the purpose of this section is to both prepare and hook the reader. It is also in this section that you'll communicate the Big Question (Essential Question, Guiding Question) that the whole WebQuest is centered around.

To accomplish the task, what steps should the learners go through? Use the numbered list format in your web editor to automatically number the steps in the procedure. Describing this section well will help other teachers to see how your lesson flows and how they might adapt it for their own use, so the more detail and care you put into this, the better. Remember that this whole document is addressed to the student, however, so describe the steps using the second person. First you'll be assigned to a team of 3 students... Once you've picked a role to play....... and so on. Learners will access the on-line resources that you've identified as they go through the Process. You may have a set of links that everyone looks at as a way of developing background information, or not (if you break learners into groups, embed the links that each group will look at within the description of that stage of the process. (Note, this is a change from the older WebQuest templates which included a separate Resources section. It's now clear that the resources belong in the Process section rather than alone.) In the Process block, you might also provide some guidance on how to organize the information gathered. This advice could suggestions to use flowcharts, summary tables, concept maps, or other organizing structures. The advice could also take the form of a checklist of questions to analyze the information with, or things to notice or think about. If you have identified or prepared guide documents on the Web that cover specific skills needed for this lesson (e.g. how to brainstorm, how to prepare to interview an expert), link them to this section.
<concept id="0304" type="narrative">
  <name lang="en">Resources</name>
  <description lang="en">List, describe and link Web sites here.</description>
  <relationships>
    <relationship xlink:href="#0305">
      <relationship_type xlink:href="#FOLLOWEDBY" />
      <relationship_cardinality xlink:href="#1-1" />
      <relationship_description lang="en" />
    </relationship>
  </relationships>
</concept>

<concept id="0305" type="narrative">
  <name lang="en">Conclusion</name>
  <description lang="en">Put a couple of sentences here that summarize what they will have accomplished or learned by completing this activity or lesson. You might also include some rhetorical questions or additional links to encourage them to extend their thinking into other content beyond this lesson.</description>
  <relationships>
    <relationship xlink:href="#0306">
      <relationship_type xlink:href="#FOLLOWEDBY" />
      <relationship_cardinality xlink:href="#1-1" />
      <relationship_description lang="en" />
    </relationship>
  </relationships>
</concept>

<concept id="0306" type="narrative">
  <name lang="en">Evaluation</name>
  <description lang="en">Describe to the learners how their performance will be evaluated. Specify whether there will be a common grade for group work vs. individual grades.</description>
</concept>

Narrative Attributes

Model

<narrative_attribute xmlns:xlink="http://www.w3.org/1999/xlink" id="willAutomateLater">
  <name lang="en">Narrative Attributes</name>
  <type lang="en" ref="" />
  <description lang="en">Container for all Narrative Attributes</description>
  <parameters />
  <relationships>
    <relationship xlink:href="#001">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#0-N" />
      <relationship_description lang="en" />
    </relationship>
  </relationships>
</narrative_attribute>

<narrative_attribute id="001">
  <name lang="en">Adaptive Axes</name>
  <type lang="en" ref="" type="adaptiveaxes" />
  <description lang="en">Adaptive Mechanisms can be applied to NarrativeConcepts</description>
  <parameters />
  <relationships>
    <relationship xlink:href="#0010">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#0-N" />
      <relationship_description lang="en" />
    </relationship>
  </relationships>
</narrative_attribute>

<narrative_attribute id="0010">
  <name lang="en">Actor</name>
  <type lang="en" ref="" type="adaptiveaxes.actor" />
  <description lang="en">This can be used to request that actor-based adaptation be applied to a NarrativeConcept</description>
  <parameters />
  <relationships>
    <relationship xlink:href="#00102">
      <relationship_type xlink:href="#CONTAINS" />
      <relationship_cardinality xlink:href="#0-N" />
      <relationship_description lang="en" />
    </relationship>
  </relationships>
</narrative_attribute>
Subject Matter Concept Space

Model - Electricity and Electromagnetism

Electricity and Electromagnetism

Magnetism
Activities

Model

<learning_activities xmlns:xlink="http://www.w3.org/1999/xlink">
  <learning_activity>
    <name>Peer Review</name>
    <type>Composite</type>
    <description>The peer review learning activity involves several sub activities, as described below, and requires the inputs from multiple roles.</description>
    <activities>
      <activity>
        <concept id="1.1" type="narrative">
          <name lang="en">Submission</name>
          <description lang="en"/>
        </concept>
        <relationships>
          <relationship xlink:href="#1.1" xlink:label="local">
            <relationship_type xlink:href="#FOLLOWEDBY"/>
            <relationship_cardinality xlink:href="#1-1"/>
            <relationship_description lang="en"/>
          </relationship>
        </relationships>
        <candidates/>
        <communications>
          <ipc xlink:href="communications.xml#1" xlink:label="remote">email</ipc>
          <ipc xlink:href="communications.xml#2" xlink:label="remote">post</ipc>
        </communications>
      </activity>
      <activity>
        <concept id="1.2" type="narrative">
          <name lang="en">Technical Corrections</name>
          <description lang="en"/>
        </concept>
        <relationships>
          <relationship xlink:href="#1.1" xlink:label="local">
            <relationship_type xlink:href="#FOLLOWEDBY"/>
            <relationship_cardinality xlink:href="#1-1"/>
            <relationship_description lang="en"/>
          </relationship>
        </relationships>
        <candidates/>
        <communications/>
      </activity>
      <activity>
        <concept id="1.1" type="narrative">
          <name lang="en">Open Discussion</name>
          <description lang="en"/>
        </concept>
        <relationships>
          <relationship xlink:href="#1.1" xlink:label="local">
            <relationship_type xlink:href="#FOLLOWEDBY"/>
            <relationship_cardinality xlink:href="#1-1"/>
            <relationship_description lang="en"/>
          </relationship>
        </relationships>
        <candidates/>
        <communications>
          <ipc xlink:href="communications.xml#1" xlink:label="remote">chat</ipc>
          <ipc xlink:href="communications.xml#2" xlink:label="remote">forum</ipc>
        </communications>
      </activity>
    </activities>
  </learning_activity>
</learning_activities>
Resource Candidates

Model

<learning_resource>
    <metadata_location>http://naxalite.cs.ted.ie:8080/exist/xml/db/ae/skoool/content/USMLC01.xml</metadata_location>
    <learning_resource_location>http://naxalite.cs.ted.ie:8080/skoool/shockwave/USMLC/USMLC01.swf</learning_resource_location>
    <learning_resource_title>Introduction to Magnets</learning_resource_title>
</learning_resource>

<learning_resource>
    <metadata_location>http://naxalite.cs.ted.ie:8080/exist/xml/db/ae/skoool/content/Magnets7.xml</metadata_location>
    <learning_resource_location>http://naxalite.cs.ted.ie:8080/skoool/shockwave/USMLC/Magnets7.swf</learning_resource_location>
    <learning_resource_title>Understanding Magnetism</learning_resource_title>
</learning_resource>

<learning_resource>
    <metadata_location>http://naxalite.cs.ted.ie:8080/exist/xml/db/ae/skoool/content/USMLC03.xml</metadata_location>
    <learning_resource_location>http://naxalite.cs.ted.ie:8080/skoool/shockwave/USMLC/USMLC03.swf</learning_resource_location>
    <learning_resource_title>Magnetics</learning_resource_title>
</learning_resource>

<selector href="http://134.226.38.81:8080/exist/servlet/db/ae/acct_developer_support/narrative_attributes/selectors/prior_knowledge_1.xml" />

PEDE Narrative

Model

<concept xmlns:xlink="http://www.w3.org/1999/xlink" id="1" type="narrative">
    <description lang="en">Personalised Science Course</description>
    <relationships />
    <adapt />
    <candidates />
    <concept id="030" type="model">
        <name lang="en">Webquest</name>
        <description lang="en" />
        <relationships>
            <relationship xlink:href="#object_id">
                <relationship_type xlink:href="#CONTAINS" />
                <relationship_cardinality xlink:href="#1-N" />
                <relationship_description lang="en" />
            </relationship>
        </relationships>
        <adapt />
        <candidates />
    </concept>
    <concept id="0301" type="narrative">
        <name lang="en">Introduction</name>
        <description lang="en" />
        <relationships />
        <adapt />
        <candidates />
        <concept id="364325608" type="subjectmatter" xlink:href=""/>
        <name lang="en">Electricity and Electromagnetism</name>
        <description lang="en">empty</description>
        <relationships>
            <relationship xlink:href="#364292591">
                <relationship_type xlink:href="#CONTAINS" />
                <relationship_cardinality xlink:href="#1-1" />
                <relationship_description xlink:href="en" />
            </relationship>
            <relationship xlink:href="#364292591">
                <relationship_type xlink:href="#CONTAINS" />
                <relationship_cardinality xlink:href="#1-1" />
                <relationship_description xlink:href="en" />
            </relationship>
            <relationship xlink:href="#364292591">
                <relationship_type xlink:href="#CONTAINS" />
                <relationship_cardinality xlink:href="#1-1" />
                <relationship_description xlink:href="en" />
            </relationship>
        </relationships>
    </concept>
</concept>

253
This concept is Magnetism. This can be used to request that actor-based adaptation be applied to a NarrativeConcept. The type of adaptivity relates to the Theory of Knowledge Spaces.

The learners' activities will be. The task could be a problem or mystery to be solved; position to be formulated and defended; product to be designed; complexity to be analyzed; personal insight to be articulated; summary to be created; persuasive message or journalistic account to be crafted; a creative work, or anything that requires the learners to process and transform the information they've gathered. If the final product involves using some tool (e.g., HyperStudio, the Web, video), mention it here. Don't list the steps that students will go through to get to the end point. That belongs in the Process section.

To accomplish the task, what steps should the learners go through? Use the numbered list format in your web editor to automatically number the steps in the procedure. Describing this section well will help other teachers to see how your lesson flows and how they might adapt it for their own use, so the more detail and care you put into this, the better.
better. Remember that this whole document is addressed to the student, however, so describe the steps using the second person. First you'll be assigned to a team of 3 students... Once you've picked a role to play...... and so on. Learners will access the on-line resources that you've identified as they go through the Process. You may have a set of links that everyone looks at as a way of developing background information, or not. If you break learners into groups, embed the links that each group will look at within the description of that stage of the process. (Note, this is a change from the older WebQuest templates which included a separate Resources section. It's now clear that the resources belong in the Process section rather than alone.) In the Process block, you might also provide some guidance on how to organize the information gathered. This advice could suggestions to use flowcharts, summary tables, concept maps, or other organizing structures. The advice could also take the form of a checklist of questions to analyze the information with, or things to notice or think about. If you have identified or prepared guide documents on the Web that cover specific skills needed for this lesson (e.g. how to brainstorm, how to prepare to interview an expert), link them to this section.
<activities></activities>
</concept><concept id="0304" type="narrative">
  <name lang="en">Resources</name>
  <description lang="en">List, describe and link Web sites here.</description>
  <relationships />
  <adapt />
  <candidates />
</concept><concept id="-180124918" type="subjectmatter" xlink:href=""><name lang="en">Ohm' s Law</name>
  <description lang="en">empty</description>
  <relationships />
  <adapt />
  <candidates />
</concept><concept id="-180035171" type="subjectmatter" xlink:href=""><name lang="en">PS Symbol</name>
  <description lang="en">empty</description>
  <relationships />
  <adapt />
  <candidates />
</concept><concept id="-179700382" type="subjectmatter" xlink:href=""><name lang="en">Transfer of Energy</name>
  <description lang="en">empty</description>
  <relationships />
  <adapt />
  <candidates />
</concept><concept id="0305" type="narrative">
  <name lang="en">Conclusion</name>
  <description lang="en">Put a couple of sentences here that summarize what they will have accomplished or learned by completing this activity or lesson. You might also include some rhetorical questions or additional links to encourage them to extend their thinking into other content beyond this lesson.</description>
  <relationships />
  <adapt />
  <candidates />
</concept><concept id="-364325608" type="subjectmatter" xlink:href=""><name lang="en">Electricity and Electromagnetism</name>
  <description lang="en">empty</description>
  <relationships />
  <adapt />
  <candidates />
</concept><concept id="0306" type="narrative">
  <name lang="en">Evaluation</name>
  <description lang="en">Describe to the learners how their performance will be evaluated. Specify whether there will be a common grade for group work vs. individual grades.</description>
  <relationships />
  <adapt />
  <candidates />
</concept>

**gAPeLS Narrative**

**Model**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<narrative>
  <id>ACCT Narrative</id>
  <type>JSRS</type>
  <call_name />
  <code><![CDATA[
(deffunction add-simple (?sectionName ?conceptId ?conceptDescription
  (update-model ?modelName "concept")
  (update-model ?modelName "name" ?sectionName)
  (update-model ?modelName "id" ?conceptId)
  (update-model ?modelName "description" ?conceptDescription)
)
(deffunction add-prior-knowledge (?sectionName ?conceptId ?conceptDescription ?listCandidates
  ?returnType ?transform ?smcs)
  (call "add-simple" ?sectionName ?conceptId ?conceptDescription)
  (foreach ?var ?listCandidates
    (update-model ?modelName "learning_resource")
    (cd ?modelName "..")
  )
  (return 0)
](deffunction add-prior-knowledge (?sectionName ?conceptId ?conceptDescription ?listCandidates
  ?returnType ?transform ?smcs)
  (printout t "sn" ?sectionName crlf)
  (printout t "cid" ?conceptId crlf)
  (printout t "cdescription" ?conceptDescription crlf)
  (printout t "can" ?listCandidates crlf)
  (printout t "rt" ?returnType crlf)
  (printout t "t" ?transform crlf)
)
]]>
```

258
(bind ?conceptsRequired (get-related-concepts ?smcs ?transform ?conceptId "#PREREQUISITE"))
(printout t "CONCEPTSPACE.conceptsRequired" ?conceptsRequired crlf)

(bind ?conceptsLearned (search-model "learner" "langstring"))
(printout t "LEARNER.conceptsLearned" ?conceptsLearned crlf)

(if (< (length$ (intersection$ ?conceptsLearned ?conceptsRequired ) ) 0 )
then
(printout t "******» "Found a PK Match" crlf)
(bind ?returnedCandidates ?listCandidates)
else
(return 1 )
)

(if (eq ?returnType "model")
then
  (update-model ?modelName "concept")
  (update-model ?modelName "name" ?sectionName)
  (update-model ?modelName "id" ?conceptId)
  (update-model ?modelName "description" ?conceptDescription)
  (foreach ?can ?returnedCandidates
    (update-model ?modelName "learning_resource")
    (update-model ?modelName "href" ?can)
    (cd ?modelName "..")
  )
  (return 0)
else
  (return ?returnedCandidates)
)

(bind ?modelName "acct_course_narrative")
(create-model ?modelName)
(update-model ?modelName "course")
(update-model ?modelName "name" "  Personalised Science Course")

(update-model ?modelName "concept")
(update-model ?modelName "name" "Introduction")
(update-model ?modelName "description" "")


(if (neq ?returnV 1) then (cd ?modelName "..") )


(if (neq ?returnV 1) then (cd ?modelName "..") )


(if (neq ?returnV 1) then (cd ?modelName "..") )

(bind ?returnV (add-prior-knowledge "Mains Electricity" "" "empty" (create$ http://localhost:8080/skoool/shockwave/ECULC/ECULC06.swf ) "model" ".../webapps/acct/transforms/prerequisites_from_smcs.xsl" "http://localhost:8080/exist/servlet/db/ae/acct/acct_smcs/acct_distribution_course_skoool_smcs.xml" ":CONTAINS")

(if (neq ?returnV 1) then (cd ?modelName "..") )

(cd ?modelName "..")
(update-model ?modelName "concept")
(update-model ?modelName "name" "Context")

259
Transforms

PEDE Narrative to gApELS Narrative

```xml
<?xml version='1.0' encoding='utf-8'?><xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:fn="http://www.w3.org/2004/10/xpath-functions" exclude-result-prefixes="fn xlink">
    <xsl:output method="html"/>
    <xsl:param name="smcs"/>
    <xsl:param name="add_simple_selector"/>
    <xsl:param name="model_name"/>
    <xsl:template match="/">
        <narrative xmlns:fn="http://www.w3.org/2004/10/xpath-functions"
            xmlns:xlink="http://www.w3.org/1999/xlink">
            <id>ACCT Narrative</id>
            <type>JESS</type>
            <call_name/>
            <code><![CDATA[
                (bind ?modelName "acct_course_narrative")
                (create-model ?modelName)
                (update-model ?modelName "course")
                (update-model ?modelName "name" "<xsl:value-of select="concept/name"/>")
                <!-- Create Concepts logical structure -->
                <xsl:apply-templates select="concept"/>
                (store-model "coursesConnection" ?modelName "<xsl:value-of select="$model_name"/>") ]]]
            </code></narrative>

    </xsl:template>
    <xsl:template match="concept">
        <xsl:variable name="concept_id" select="@id"/>
        <xsl:variable name="concept_type" select="@type"/>
        <xsl:variable name="concept_name" select="name"/>
        <xsl:variable name="concept_description" select="description"/>
        <xsl:variable name="num_child_concepts" select="count(concept)"/>
        <xsl:if test="$concept_type=' narrative'">
            (update-model ?modelName "concept")
            (update-model ?modelName "name" "<xsl:value-of select="$concept_name"/>")
            (update-model ?modelName "description" "<xsl:value-of select="$concept_description"/>")
        </xsl:if>
        <xsl:if test="$num_child_concepts = 0 and $num_learning_activities = 0">
            (cd ?modelName ".")
        </xsl:if>
        </xsl:if>
        <xsl:if test="$concept_type='subjectmatter'">
            (bind ?returnV (<xsl:call-template name="get_call_name">
                <xsl:with-param name="c_id" select="$concept_id"/>
            <xsl:if test="$num_child_concepts = 0">
                (if (neg ?returnV 1) then (cd ?modelName ".") )
            </xsl:if>
        </xsl:if>
        <xsl:if test="count(child::concept)=0">
            (apply-templates select="concept"/>
        </xsl:if>
        <xsl:if test="count(child::learning_activity)=0">
            (apply-templates select="learning_activity"/>
        </xsl:if>
    </xsl:if>
</xsl:template>
    <xsl:template match="learning_activity">
        (update-model ?modelName "name" "<xsl:value-of select="name"/>")
    </xsl:template>
</xsl:stylesheet>
```

261
### Parameters

<table>
<thead>
<tr>
<th>Parameter Type</th>
<th>Value</th>
<th>Range</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept Name</td>
<td>Yes</td>
<td>No</td>
<td>Required</td>
</tr>
</tbody>
</table>

### Concept Name

- Yes/no selection is required.
- The range of values is from min to max.
- The attributes are name, type, and the restrictions.

**Example:**

- **Concept Name:**
  - **Type:** Name
  - **Value:**
    - **Range:**
  - **Required:** Required

---

**Note:**

- The table entries are based on the XSLT template provided, which includes functionality for rendering parameters and concepts with their respective attributes.
- The template dynamically generates the table entries based on the XML data structure provided by the XSLT transformation.
- The inclusion of the `@type`, `@min`, and `@max` attributes in the table entries provides a clear representation of the parameter types, their expected values, and the required status.

---

**Source:**

The XSLT code snippet is used to generate the HTML table entries, ensuring a consistent and structured presentation of the parameter details.

---

**Further Exploration:**

- Understanding the role of XSLT in data transformation and web content generation.
- Exploring the integration of XSLT with XML data for dynamic table generation.
- Delving into the specifics of how XSLT templates can be customized to suit various data presentation needs.

---

**Conclusion:**

The XSLT template effectively leverages XML data to create a structured and user-friendly table representation, highlighting the importance of XML and XSLT in modern web development and data visualization.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>unit</code></td>
<td></td>
<td>radio</td>
</tr>
<tr>
<td><code>all</code></td>
<td></td>
<td>radio</td>
</tr>
</tbody>
</table>

**Concept**

- **IsRequired**: Required status
- **Natype**: Type of attribute

**HTML Template**

```html
<xsl:template name="processna">
  <xsl:if test="parameters/@set='unit'">
    <xsl:call-template name="unit_q" />
  </xsl:if>
  <xsl:if test="parameters/@set='all'">
    <xsl:call-template name="all_q" />
  </xsl:if>
</xsl:template>

<xsl:template name="concept">
  <xsl:param name="isRequired" />
  <xsl:param name="natype" />
  <tr class="h4" style="background-color:#FFFFEE">
    <td><xsl:value-of select="name" /></td>
    <td align="center">
      <input type="radio" value="yes" name="name" select="$natype" />.
      <xsl:value-of select="@id" />
    </td>
    <td align="center">
      <input type="radio" value="no" name="name" select="$natype" />
      <xsl:value-of select="@id" />
    </td>
    <td align="center">
      <xsl:if test="count(child::concept)!=0">
        <xsl:call-template name="concept">
          <xsl:with-param name="isRequired" select="$isRequired" />
        </xsl:call-template>
      </xsl:if>
    </td>
  </tr>
</xsl:template>
```
This site was entirely produced by the Adaptive Course Construction Toolkit (ACCT). This generic web application can deliver an infinite variety of courses produced by the ACCT. The following functionality is offered through this web interface:

- From the adaptive course that you have composed an auto-generated learner modeling questionnaire is produced. This modelling mechanism allows you to test your adaptive course by creating instances of learner models for the course to reconcile. By reconciling the learner model against the adaptive course model, a personalised learning path is produced for this instance of the learner. This gives you the opportunity to validate the strategic effectiveness of the course while employing the personalising power offered by adaptivity.
- Once you have created a populated instance of a learner model, the adaptive course can be executed. This web application uses an Adaptive Engine (AE3), developed at the Knowledge and Data Engineering Group (KDEG), to execute your adaptive course. The engine is responsible for reconciling the models of your adaptive course to produce a personalised learning path.
- The course developer can easily view the semantics of the course that they have composed. This echo's the adaptive course view provided by the Narrative Builder of the ACCT. It allows you to identify the types of adaptivity that you have applied to a course, the types of concepts you wish to teach and the candidate learning resources that you have associated with the course concepts.

To evaluate your adaptive course, carry out the following steps:

- Build an instance of the learner model using "Build Learner Model" from the application menu on the left.
- Execute the adaptive course on the supplied learner model using "Execute Adaptive Course" from the application menu on the left.
Hello there! This is an adaptive course for Personalised Science Course. To navigate the course, use the menu on the left.

<table>
<thead>
<tr>
<th>Introduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
</tr>
<tr>
<td>Poles/Repulsion</td>
</tr>
<tr>
<td>Poles</td>
</tr>
<tr>
<td>Repulsion</td>
</tr>
<tr>
<td>Process</td>
</tr>
<tr>
<td>Peer Review</td>
</tr>
<tr>
<td>Submission</td>
</tr>
<tr>
<td>Technical Corrections</td>
</tr>
<tr>
<td>Open Discussion</td>
</tr>
<tr>
<td>Final Response</td>
</tr>
<tr>
<td>Submission of Revised</td>
</tr>
<tr>
<td>Work</td>
</tr>
<tr>
<td>Peer Review</td>
</tr>
<tr>
<td>Completion</td>
</tr>
<tr>
<td>Publication and Conclusions</td>
</tr>
<tr>
<td>Resources</td>
</tr>
<tr>
<td>Ohm's Law</td>
</tr>
<tr>
<td>PS Symbol</td>
</tr>
<tr>
<td>Power</td>
</tr>
<tr>
<td>Transfer of Energy</td>
</tr>
<tr>
<td>Conclusion</td>
</tr>
<tr>
<td>Electricity and Electromagnetism</td>
</tr>
<tr>
<td>Evaluation</td>
</tr>
</tbody>
</table>

All Content Supplied by Intel Ireland through SkoolLa Initiative
Concept Description:

![Diagram of a circuit with two poles and a repulsion symbol.](image)

**Introduction**
- Conduction of Electricity
- Circuits

**Task**
- Poles/Repulsion
- Poles
- Repulsion

**Process**
- Peer Review
- Submission
- Technical Corrections
- Open Discussion
- Final Response
- Submission of Revised Work
- Peer Review
- Completion
- Publication and Conclusions

**Resources**
- Ohm's Law
- PS Symbol
- Power
- Transfer of Energy

**Conclusion**
- Electricity and Electromagnetism

**Evaluation**

*All Content Supplied by Intel Ireland through Skoolie Initiative*

Screenshot: gAPeLS Adaptive Course Page (All yes's selected)
APPENDIX IV – Trial Questionnaire

Brief Background

Question 1: How many online eLearning courses have you developed?
0  1  2-5  6 or more

Question 2: How many times have you used a course composition tool?
0  1  2-5  6 or more

Question 3: How would you rate your knowledge of Adaptive Hypermedia?
none  novice  intermediate  expert

Question 4: How would you rate your knowledge of Personalised eLearning?
none  novice  intermediate  expert

Task Oriented Questions

Question 5: Starting the Adaptive Course Construction Toolkit (ACCT) was...
very easy  easy  difficult  very difficult

Question 6: Creating a course package using the ACCT was...
very easy  easy  difficult  very difficult

Question 7: Loading a course package using the ACCT was...
very easy  easy  difficult  very difficult

Question 8: Importing models from an external course package using the ACCT was...
very easy  easy  difficult  very difficult

Question 9: Opening the Subject Matter Concept Space Builder was...
very intuitive  intuitive  Complex  very Complex
Question 10: Editing a subject matter concept space using the ACCT was
very intuitive intuitive Complex very Complex

Question 11: Saving a subject matter concept space using the ACCT was...
very intuitive intuitive complex very complex

Question 12: Opening the Narrative Builder was...
very intuitive intuitive complex very complex

Question 13: Searching for learning resources using the ACCT...
very easy easy difficult very difficult

Question 14: Selecting learning resources using the ACCT...
very easy easy difficult very difficult

Question 15: Designing an adaptive course narrative using the ACCT was...
very intuitive intuitive complex very complex

Question 16: Editing an adaptive course narrative using the ACCT was...
very easy easy difficult very difficult

Question 17: Saving an adaptive course narrative using the ACCT was...
very easy easy difficult very difficult

Question 18: Publishing your adaptive course using the ACCT was...
very easy easy difficult very difficult

Question 19: The role of the Subject Matter Concept Space in the process of generating a personalised course was...
very clear clear confusing very confusing
Question 20: The role of Narrative Attributes in the process of generating a personalised course was...
very clear clear confusing very confusing

Question 21: In your own words explain how the Subject Matter Concept Space is used in the process of generating a personalised course - OPEN

Question 22: In your own words explain how Narrative Attributes are used in the process of generating a personalised course - OPEN

Question 23: The Adaptive Course Construction Methodology is...
very clear clear confusing very confusing

Question 24: The ACCTs implementation of elements of the Adaptive Course Construction Methodology was...
very clear clear confusing very confusing

Question 25: In your own words describe the sufficiency of the ACCTs implementation of elements of the Adaptive Course Construction Methodology - OPEN

Question 26: In your opinion, what was the easiest part of adaptive course composition using the ACCT? - OPEN

Question 27: In your opinion, what was the most difficult part of adaptive course composition using the ACCT? - OPEN

Question 28: In your opinion, what was your favourite aspects of the ACCT interface? - OPEN

Question 29: In your opinion, what was your least favourite aspects of the ACCT interface? - OPEN
Usability Questions

Question 30: The terminology used in the ACCT was...
very consistent consistent inconsistent very inconsistent

Question 30a: The Adaptive Hypermedia terminology used in the ACCT was...
very consistent consistent inconsistent very inconsistent

Question 30b: The Personalised eLearning Terminology used in the ACCT was...
very consistent consistent inconsistent very inconsistent

Question 30c: Messages which appeared on the screen were...
very consistent consistent inconsistent very inconsistent

Question 30d: Position of instructions on the screen was...
very consistent consistent inconsistent very inconsistent

Question 31: The use of shortcuts in the ACCT was...
very sufficient sufficient insufficient very insufficient

Question 32: The informative feedback offered by the ACCT was...
very appropriate appropriate inappropriate very inappropriate

Question 32a: Instructions for correcting errors were...
very clear clear confusing very confusing

Question 32b: System keeps you informed about what it is doing...
always almost always almost never never

Question 32c: Animated cursors keep you informed...
always almost always almost never never
Question 33: The ACCT’s use of system dialogues to bring closure to action sequences was...
very sufficient sufficient insufficient very insufficient

Question 33a: Performing an operation leads to a predictable result...
always almost always almost never never

Question 34: The use of error prevention and error handling was...
very appropriate appropriate inappropriate very inappropriate

Question 34a: Operations were...
very dependable dependable undependable very undependable

Question 34b: System failures occurred...
very seldom seldom frequently very frequently

Question 34c: System warns you about potential problems...
always almost always almost never never

Question 34d: Instructions for correcting errors were...
very clear clear confusing very confusing

Question 35: Recovering from a logical error was...
very easy easy difficult very difficult

Question 35a: Correcting your mistakes was...
very easy easy difficult very difficult

Question 35b: Correcting typos was...
very simple simple complex very complex
Question 35c: Undo/Redo actions was...
very easy  easy  difficult  very difficult

Question 36: Within the adaptive course composition process using the ACCT made you feel...
very empowered  empowered  disenfranchised  very disenfranchised

Question 36a: Unexpected system actions occurred...
always  almost always  almost never  never

Question 36b: Ability to perform desired task...
always  almost always  almost never  never

Question 36c: Data entry sequences were...
very efficient  efficient  tedious  very tedious

Questions about the Workshop

Question 37: Your overall opinion of the workshop was...
very productive  productive  unproductive  very unproductive

Question 38: The presentations made at the beginning of the workshop were...
very helpful  helpful  unhelpful  very unhelpful

Question 39: The jargon buster presentation was...
very helpful  helpful  unhelpful  very unhelpful

Question 40: The tasks of the workshop were...
very concise  concise  unclear  very unclear

Question 41: The time allotted for each task of the workshop was...
very sufficient  sufficient  insufficient  very insufficient
APPENDIX V – Questionnaire Results

Question 1: How many online eLearning courses have you developed?

- 0: 4
- 1: 0
- 2-5: 0
- 6 or More: 0

- 0: 2
- 1: 1
- 2-5: 1
- 6 or More: 0
Question 2: How many times have you used a course composition tool?

- None: 4
- Novice: 0
- Intermediate: 0
- Expert: 0

Question 3: How would you rate your knowledge of Adaptive Hypermedia?

- None: 3
- Novice: 1
- Intermediate: 0
- Expert: 0

Question 4: How would you rate your knowledge of Personalised eLearning?
Question 5: Starting the Adaptive Course Construction Toolkit (ACCT) was...

Question 6: Creating a course package using the ACCT was...
Question 7: Loading a course package using the ACCT was...

Question 8: Importing models from an external course package using the ACCT was...
Question 9: Opening the Subject Matter Concept Space Builder was...

Question 10: Editing a subject matter concept space using the ACCT was...
Question 11: Saving a subject matter concept space using the ACCT was...

Question 12: Opening the Narrative Builder was...
Question 13: Searching for learning resources using the ACCT...

Question 14: Selecting learning resources using the ACCT...
Question 15: Designing an adaptive course narrative using the ACCT was...

Question 16: Editing an adaptive course narrative using the ACCT was...
Question 17: Saving an adaptive course narrative using the ACCT was...

Question 18: Publishing your adaptive course using the ACCT was...
Question 19: The role of the Subject Matter Concept Space in the process of generating a personalised course was...
Question 20: The role of Narrative Attributes in the process of generating a personalised course was...

Question 21: In your own words explain how the Subject Matter Concept Space is used in the process of generating a personalised course - OPEN

Question 22: In your own words explain how Narrative Attributes are used in the process of generating a personalised course - OPEN

Question 23: The Adaptive Course Construction Methodology is...
Question 24: The ACCTs implementation of elements of the Adaptive Course Construction Methodology was...

Question 25: In your own words describe the sufficiency of the ACCTs implementation of elements of the Adaptive Course Construction Methodology - OPEN

Question 26: In your opinion, what was the easiest part of adaptive course composition using the ACCT? - OPEN

Question 27: In your opinion, what was the most difficult part of adaptive course composition using the ACCT? - OPEN

Question 28: In your opinion, what was your favourite aspects of the ACCT interface? - OPEN

Question 29: In your opinion, what was your least favourite aspects of the ACCT interface? - OPEN
Question 30: The terminology used in the ACCT was... 

Question 30a: The Adaptive Hypermedia terminology used in the ACCT was...
Question 30b: The Personalised eLearning Terminology used in the ACCT was...

Question 30c: Messages which appeared on the screen were...
Question 30d: Position of instructions on the screen was...

Question 31: The use of shortcuts in the ACCT was...
Question 32: The informative feedback offered by the ACCT was...

Question 32a: Instructions for correcting errors were...
Question 32b: System keeps you informed about what it is doing...

Question 32c: Animated cursors keep you informed...
Question 33: The ACCT's use of system dialogues to bring closure to action sequences was...

Question 33a: Performing an operation leads to a predictable result...
Question 34: The use of error prevention and error handling was...

Question 34a: Operations were...
Question 34b: System failures occurred...

Question 34c: System warns you about potential problems...
Question 34d: Instructions for correcting errors were...

Question 35: Recovering from a logical error was...
Question 35a: Correcting your mistakes was...

Question 35b: Correcting typos was...
Question 35c: Undo/Redo actions was...  

Question 36: Within the adaptive course composition process using the ACCT made you feel...
Question 36a: Unexpected system actions occurred...

Question 36b: Ability to perform desired task...
Question 36c: Data entry sequences were...

Question 37: Your overall opinion of the workshop was...
Question 38: The presentations made at the beginning of the workshop were...

Question 39: The jargon buster presentation was...
Question 40: The tasks of the workshop were...

Question 41: The time allotted for each task of the workshop was...

Question 42: Comments on the workshop – OPEN
APPENDIX VI - Key Design Elements of Personalised eLearning

Information models have been the intelligent driving force behind business systems, computer and software engineering systems and educational systems for centuries. The idea is to capture and represent the knowledge of a system in an effort to increase its reusability and durability. Systems built on an information-model paradigm are inherently more flexible and usable. Personalised eLearning is a relatively new field emerging from the world of Intelligent Tutoring Systems (ITS) and Adaptive Hypermedia Systems (AHS). The goal of personalised eLearning is to support e-learning content, activities and collaboration, adapted to the specific needs and influenced by specific preferences of the learner and built on sound pedagogic strategies [Dagger et al., 2005]. In the area of personalised eLearning, systems like APeLS [Conlan et al., 2002a] have pioneered architectures and methodologies which are built upon multi information models. These systems can then reconcile and reason about the semantics of and the knowledge contained within these models. This approach feeds on knowledge and is completely flexible and extensible; in essence anything that can be modelled can be reconciled and reasoned about.

When this information model paradigm is applied to personalised education, there are core intelligence sources that must be represented, i.e. the disparate design elements of personalised eLearning. This intelligence relates to pedagogy, activity, subject matter, personalisation, learning resources, services, learners, educators and a whole plethora of other information. Note however that a key conscious influential factor involving all information models is granularity. Granularity relates to the conceptual “size” used to aggregate the building blocks of the information model. In a pedagogical model, granularity pertains to the aggregation of the educational concepts and activities in the model. In an activity model, granularity pertains to how fine or coarse the composite and/or atomic activities in the model are. In a subject matter area model, granularity pertains to the coarseness of concepts and relationships used to represent the modelled elements of that subject matter domain. Granularity of personalisation models indicate the scoping of the parameters that the personalisation axes can understand.

The learner and teacher are core information sources needed by the personalised eLearning services, however the personalised eLearning development environment does not need to either explicitly model nor have access to specific instances of learner or teacher information. What the personalised eLearning development environment does is specify the types of information that is needed and the sources which can provide that information.

For example, in an eLearning experience developed to be personalised based on a learner’s prior knowledge of a subject matter area, the personalised eLearning design indicates that “sections” of the eLearning experience are to be personalised towards the individual’s prior knowledge of the subject matter in that “section”. It also provides an identifier for the model type, i.e. learner model, which can provide the required information. This mechanism can also be used to provide a base schema for the learner model that is needed by this eLearning design. The base schema can be used to “auto-generate” a learner modelling mechanism which the personalised eLearning services can then use. This is detailed in section 5.3.8.

The following sections provide detailed descriptions of the core disparate design elements of personalised eLearning development. Following these descriptions, an insight into the design and modelling process for representing the disparate elements is provided.
Pedagogies (Narrative Structures)

Recent trends in "personalised eLearning" indicate a focus on adaptively retrieving rich multimedia content and describing it as a base for educational engagement [Brusilovsky et al., 2002] [De Bra et al., 2003]. The lack of pedagogy and activity in the learning process inevitably leads to weak educational offerings. One of the key factors influencing this trend is the lack of pedagogical support for course developers. By making pedagogical strategies more accessible we can facilitate and promote wider use of pedagogy in personalised eLearning environments, support the course development process with pedagogically rich development guidelines and promote the sharing of pedagogy within the educational community. Recent progress in the instructional design community has lead to the creation of a design principles database “as an infrastructure for designers to publish, connect, discuss and review design ideas” [Design Principles for Educational Software]. These information sources provide invaluable descriptive and usage guidelines for instructional design which could be used in the support framework for personalised eLearning development environments.

A pedagogic strategy usually consists of an arranged sequence of conceptual tasks and activities that need to be performed. This workflow is usually accompanied with best practice principals and use case guidelines to illustrate the maximum potential benefit offered by the strategy. Through a pedagogical modelling mechanism we can create an accessible and flexible instance of the pedagogy. The model contains descriptive and usage information for each of the high level concepts/activities of the pedagogical strategy and suggests a possible sequencing of these abstract elements. Typically, pedagogical strategies can be represented as a series of high-level descriptive concepts representing learning activities to be undertaken.

The main benefit in facilitating and supporting the reuse of educational strategies is the ability to share successful and proven pedagogical approaches to personalised eLearning with the educational community. Educators can take and modify these strategies as desired, but have the benefit of a proven basis upon which those changes can be made. In this way, institutions across different countries can benefit from each others “knowledge” pertaining to teaching experiences. They may share both strategy and content, or strategy alone. Through facilitating this reuse, pedagogical expertise may be shared in the same way that content currently is shared. These mechanisms allow the strategy to be applied across educational, institutional, geographical and cultural boundaries.

There are a number of key potential facilitators to the reuse of educational strategies. The primary of these is the use of standards to represent the model in order to increase accessibility. Standards impact the reuse of educational strategies in two ways. If the strategies are written in a standards compliant format they may be utilised by personalised eLearning development environments that support the “importing” of that particular standards specification. The second impact that standards have on the reuse of educational strategies is their description towards discovery. Strategies may be described in a similar manner or style as existing Learning Objects (LO) and Learning Resources (LR). The associated metadata describes the use and requirements of the LO but is separated from the actual content objects. If appropriate standards are used the potential discoverability and, hence, reusability of a strategy is greatly increased.

Another key facilitator in the reuse of educational strategies is the utilization of a commonly understood vocabulary. This may take the form of either a shared vocabulary or of mappings
between separate vocabularies. There exists similar work in the area of semantic interoperability within the semantic web community [O'Sullivan & Lewis, 2003]. The vocabulary is used to describe pedagogical concepts, activities, collaboration, services and personalisation in the educational strategy. A common understanding of these provides rich semantic information and can help to facilitate the reuse of the strategy across different personalised eLearning development environments and different personalised eLearning services.

Pedagogical strategies are fundamental to the success of future eLearning applications, in both personalised and non-personalised settings. A key enabler to promoting a pedagogically aware eLearning development community is the ability to model pedagogical strategies, making the modelled strategies accessible and promoting their reuse on a global scale. This ability to actively practice pedagogic course development should facilitate the production of more effective, active and engaging eLearning experiences.

Activities
Evolving online education requires the changing and modification of the building blocks of instructional design [Reigeluth, 1999]. In order for the learner to acquire higher order cognitive skills (analysis, synthesis and evaluation) based on Bloom’s taxonomy, the need for instructional design which facilitates, promotes and supports activity based learning must be realized. Through online learning and eLearning we can provide a more active learning experience, promote active learner involvement and encourage self motivation.

Learning activities typically consist of some form of task(s), associated tools which could be used to perform the task(s), a description of roles within the task and appropriate learning content. Learning activities require some intuitive sequencing of operations. The sequencing describes the workflow between sub-activities within the learning activity. These elements and attributes of learning activities can be modelled to produce an accessible and reusable form of the knowledge incorporated in that activity. The workflow information can provide the base knowledge for sequence recommendations within the learning activity model. This information, used in a personalised eLearning development environment, can support a course developer in producing activity-oriented personalised eLearning. The model should contain a description of the Learning Activity, the type of the Learning Activity (atomic or composite), the types of outcomes it can provide, the types of communications tools available, the different roles in the activity and best practice guidelines on how to use the activity. This flexible modelling approach increases potential for reusability, accessibility and interoperability of Learning Activities.

Subject Areas
One of the key sources of information in any educational experience is knowledge about the subject area. This information contains conceptual and relational descriptions about the subject matter. It provides scope to the personalised eLearning experience, it provides knowledge to the personalisation axes and it supports the course developer when designing personalised eLearning offerings. By modelling this information, the knowledge pertaining to the semantics of the subject matter area can be reused, repurposed and shared amongst the educational community. A standards based representation, in Web Ontology Language [OWL] for example, promotes the accessibility of the underlying information model.
A key factor in the successful representation of subject area is the use of commonly understood vocabularies and semantics to describe the domain. There are certain standards which actively promote the use of common vocabularies in describing a subject area. For example, the Library of Congress Subject Headings (LCSH) provides an up to date and comprehensive list of subject headings which could provide a solid and distinct taxonomy for describing subject area information.

**Personalisation Axes (Narrative Attributes)**

In face-to-face learning scenarios such as the classroom or the corporate training room, the tutor or facilitator can interact with the students and perceptually gage the competency levels and the learning motivations based on those interactions and "adjust" their teaching appropriately. With standard eLearning environments these teacher dynamics are not possible since a "one size fits all" mentality is applied. However a "one size fits one" approach is more realistic. In reality different learners bring, for example, different learning outcomes, different prior knowledge and past experiences and different cognitive preferences to the learning experience. This can only be accommodated for by supporting personalisation of the learning experience towards the individual. As previously described in section 2.3, some of the different types of personalisation axes which exist are prior knowledge, learning outcomes, learner preferences, learning history, cultural background, communication styles and learning styles.

These personalisation axes can then be modelled in order to represent the inherent knowledge in a form that is reusable and accessible to; course developers using personalised eLearning development environments; and learners using personalised eLearning services. Personalised eLearning development environments make use of modelled personalisation axes while supporting the course developer in composing personalised eLearning experiences. This illustrates that personalisation axes can be successfully modelled and reused [Dagger et al. 2005b].

**Learning Resources**

Vast quantities of learning materials currently exist in both open and closed corpus settings. Institutions such as universities and schools, for example, house massive quantities of closed corpus learning resources. Open Corpus Content, which is free for use by any educational institution or system, is available in public digital repositories such as Connexions and EducaNext, commercial digital repositories such as ContentDM and Lydia, and via the World Wide Web. Countries such as the UK, Ireland, USA, Canada and Australia are increasingly investing in national digital repositories to encourage the re-use of digital content. Examples of such initiatives include ARIADNE Foundation, Education Network Australia Online, EduSource, and BELLE, Multimedia Educational Resources for Learning and Online Teaching and Gateway to Educational Materials and the National Institute of Multimedia Education (JAPAN). The metadata, were present, used to describe these learning resources can provide for example technical, educational, copyright, lifecycle, relation and classification information about the resource. However the various standards, vocabularies and taxonomies which exist and which are used to describe learning resources can sometimes restrict their potential for reuse. The use of standards such as SCORM, IEEE LOM, IMS LRM and Dublin Core
Dublin Core are widespread. Although these standards share some similarities they are quiet different. Mapping between the different syntax and schema can produce a common canonical form but this is time consuming. However, a much more crucial issue than syntax drift exists. Semantic drift is very difficult to remedy. The use of diverse vocabularies and taxonomies to describe the content is a global trend. These can differ from school to school, university to university, company to company, and so on. Therefore a key issue to address with regard to learning resource discovery and reuse is not only syntactic interoperability but semantic interoperability.

In situations where metadata does not exist, classification services can auto-generate it. Annotea [Annotea], developed by the W3C project, uses an RDF based annotation schema to describe annotations as metadata. Metasaur [Metasaur], developed by the University of Sydney, supports the creation of metadata for learning objects, using a standard vocabulary ontology that is generated automatically. This ontology can then have terms added to improve accuracy. This ensures consistency in the terms used to tag content. Semtag [Dill et al., 03] is a system that has been developed using Seeker, which is a platform for large-scale text analytics. Semtag was created by the IBM Almaden Research Centre. The system crawls through web pages and performs an automated semantic tagging of each page using the TAP ontology. Semtag uses a Taxonomy Based Disambiguation (TBD) algorithm to ensure the correct classification of content in its tagging. Thorough and consistent markup of data not only ensures the correct interpretation and use of content, it also provides a first step towards and entirely machine-comprehendible “semantic web”.

The information stored in a learning resource’s metadata can be used by personalised eLearning development environments to identify similarities between it and the information stored in the personalised eLearning designs. For example, if the metadata for a piece of content describes it as being:

a) pedagogically neutral
b) a certain technology type such as flash or html
c) a certain modality type such as visual or aural
d) not copyright protected

then it may be offered as appropriate candidate content for a personalised course using a Webquest pedagogy, designed for diverse VARK [VARK] learners in a university curriculum. So the metadata of a learning resource can support the process of selecting appropriate content. The inverse of this is also true, meaning that a personalised eLearning design provides the blue prints for the content types which it requires to fulfil its purpose. In this way, it can provide the semantic information required in order to source appropriate learning content.

Another key issue here is the granularity of the learning resource. As mentioned earlier with regard to granularity of pedagogical, activity, subject area and personalisation information, granularity of learning resources is a difficult problem to address and is outside the scope of this thesis, however, there is notable research on learning resource granularity which paves the way for affective solutions [Verbert & Duval, 2004] [Duval et al. 2003] [Duval, 2004].

Learners (LIP)

Constructivism involves the learner becoming active and interactive within their own learning experiences to develop their own understanding of the knowledge domain [Jonassen (1999)]. An active learning process of engaging prior knowledge and competencies with current learning
promotes the active development of new knowledge through cognitive processing. This theory describes situations where the learner is in control of the pace, mode, frequency and so on of their learning. Control is passed to the learner by using the information about the learner to orchestrate the educational experience. The learner information can relate to prior knowledge and competencies, prior misconceptions, learning history, current learning objectives, preferred learning style, cultural background, and so on. Capturing this vast amount of information can be a daunting task and typically a personalisation system only needs a subset of this information to achieve its strategic goals. However, specifying that subset of required information is a difficult task.

One method of specifying the minimum required subset of information uses the personalised eLearning design. A personalised eLearning design provides a rich source of information regarding the types of personalisation required. Through analysing the applied personalisation in a personalised eLearning design, the base set of learner requirements can be gathered. For example, in a personalised eLearning design teaching electromagnetism through a Webquest pedagogical theory might apply personalisation based on prior knowledge to the “Introduction” section. Now let’s assume that this section contains three concepts, “conduction of electricity”, “magnetism” and “circuits”. This information now describes a scenario where the engaging learner requires knowledge pertaining to the “prerequisite” requirements of these concepts, based on their specification in the subject area description, in order for them to be appropriate. Inferring across this scenario, we can produce requirements for a subset of learner information based on the “concepts” that are being personalised in the personalised eLearning design. A mechanism for storing this learner information is ePortfolios [ePortfolio].

Through learner empowerment [Bajraktarevic et al. (2003)] the reach and effectiveness of adaptive personalised eLearning can be extended [Conlan et al. (2004)].

**Personalised eLearning Designs**

When the course developer uses the PEDE to build an educational experience, the embodiment of that experience is a personalised eLearning design. Based on this, the personalised eLearning design contains information about the course developers chosen pedagogy, activity types, subject area, personalisation axes, selected candidate learning resources and any other information sources used. It is therefore a collection of the various disparate design elements of personalised eLearning, a PEDE Narrative.
APPENDIX VII – Phase 1 Trial and Evaluation

- Overall, did you feel that you had enough time to create an adaptive course?
  - Yes
  - No
  Comments:

- Overall, were you satisfied with the support information (online-help, messages, documentation) when completing the tasks?
  - Yes
  - No
  Comments:

- Did you feel that the trial and evaluation process was adequate?
  - Yes
  - No
  Comments:

- Did you feel that the ACCT empowered and supported you during the creation of your adaptive course?
  - Empowered
  - Disenfranchised
  Comments:

- Was it easy to use the ACCT?
  - Easy
  - Difficult
  Comments:

- Could you effectively create your adaptive course using the ACCT?
  - Yes
  - No
  Comments:

- Could you efficiently (time, complexity, reusability) create your adaptive course using the ACCT?
  - Yes
  - No
  Comments:

- Did you feel comfortable using the ACCT?
  - Comfortable
  - Uncomfortable
  Comments:
• Was it easy to learn to use the ACCT?
  - Easy to use
  - Difficult to use
  *Comments:*

• If you made a mistake using the ACCT, was it quick and easy to recover from it?
  - Quick recovery
  - Impossible to recover
  *Comments:*

• Was the information (such as online help, on-screen messages, and other documentation) provided by the ACCT clear?
  - Very clear
  - Very unclear
  *Comments:*

• Did the ACCT have all the functionality and capability you expected it to have?
  - Yes
  - No
  *Comments:*

• Was the ACCT interface consistent through the course development process?
  - Consistent
  - Inconsistent
  *Comments:*

• Do you think that the ACCT interface is appropriate for adaptive course construction?
  - Appropriate
  - Inappropriate
  *Comments:*

• Do you understand what a subject matter concept space (SMCS) is?
  - Yes
  - No
  *Comments:*

• Do you understand what the role of subject matter concept space (SMCS) is?
  - Yes
  - No
  *Comments:*

• Did you understand the logical representation of the SMCS presented by the ACCT?
  - Yes
  - No
  *Comments:
• Did you find it easy and intuitive to add “concepts” to the SMCS?
  o Easy and intuitive to add Concepts
  o Difficult and unintuitive to add Concepts
  Comments:

• Did you find it easy and intuitive to add relationships to the concepts of the SMCS?
  o Easy and intuitive to add Relationships
  o Difficult and unintuitive to add Relationships
  Comments:

• Did you feel empowered by the ability to create and customise your own relationship definitions?
  o I felt empowered
  o I felt disenfranchised
  Comments:

• Do you understand what a Narrative Model is?
  o Yes
  o No
  Comments:

• Do you understand what the role of the Narrative Model, in the context of adaptive course creation, is?
  o Yes
  o No
  Comments:

• Did the visual representation of the Narrative make it easier to understand the logic and purpose of the Narrative within the scope of an adaptive course?
  o Yes
  o No
  Comments:

• Do you understand the purpose of pedagogy in the course development process?
  o Yes
  o No
  Comments:

• Did you like the functionality of the “palette” of tools offered by the Narrative Builder?
  o Yes
  o No
  Comments:

• Did you like the drag and drop interface of the custom Narrative Builder?
  o Yes
  o No
Comments:

- Did you feel empowered by the ability to create your own customised Narrative?
  - I felt empowered
  - I felt disenfranchised
  Comments:

- Did you feel that the support offered during the creation of the custom Narrative was helpful?
  - Very helpful
  - Very unhelpful
  Comments:

- Did you find it easy to search for learning resources?
  - Very easy
  - Very difficult
  Comments:

- Do you think that the searching mechanism is powerful enough?
  - Yes
  - No
  Comments:

- Would you like to see extra search functionality?
  - Extra functionality
  - No more functionality
  Comments:

- Do you know what adaptivity is?
  - Yes
  - No
  Comments:

- Do you understand how adaptivity is represented in the ACCT (i.e. Narrative Attributes)?
  - Fully understand
  - Do not understand
  Comments:

- Do you understand the role of adaptivity in the Narrative Model?
  - Fully understand
  - Do not understand
  Comments:

- Do you know what pedagogy is?
  - Yes
  - No
Comments:

- Have you ever used a pedagogical strategy before (during the creation of a course)?
  - Yes
  - No
  Comments:

- The ACCT provides modelled versions of pedagogy. Do you feel that the pedagogical representations (in model form) strongly provide a support framework for building a course based on the chosen pedagogy(s)?
  - Provides a strong pedagogical framework
  - Provides no pedagogical framework
  Comments:

- Do you feel that a course designed using the ACCT will be more pedagogically effective than a course created without using the ACCT?
  - Pedagogically effective
  - Pedagogically ineffective
  Comments:

- Do you feel that having the ability to test and verify your adaptive course in real-time will improve the course consistency and effectiveness?
  - Yes
  - No
  Comments:

- Do you feel empowered by the ability to test and verify your adaptive course?
  - Yes
  - No
  Comments: