UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL AND APPLIED SCIENCES

Electronics and Computer Science

Sharing User Models Between Interactionally-Diverse Adaptive Educational Systems

by

Rikki F. Prince

Thesis for the degree of Doctor of Philosophy

April 2017

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL AND APPLIED SCIENCES
Electronics and Computer Science

Doctor of Philosophy

SHARING USER MODELS BETWEEN INTERACTIONALLY-DIVERSE ADAPTIVE EDUCATIONAL SYSTEMS

by Rikki F. Prince

Adaptive Educational Systems (AES) will become increasingly important in teaching and learning environments over the next decade, as students demand more personalised experiences. These systems reveal, hide, modify and recommend content that is most appropriate for the current user. To do this they rely on an accurate model of the student, their knowledge, experience and goals. With a growing variety of developers of these systems there are more situations where an experienced student will approach a new adaptive system, and it will not have any user model data with which to adapt; this is known as the cold-start problem.

An answer to this is shared user modelling, where data about the student is communicated between adaptive applications. This task becomes more complicated when the applications measure the user in very different ways and therefore have different models to represent the user.

This thesis proposes the design of an intermediary user model system that uses authored rules to map between the user model attributes used by different applications to measure the user. A prototype implementation of this theoretical framework is presented here, called the Interactionally-Diverse Intermediary User Modelling System, or IDIUMS.

Two evaluations of IDIUMS were performed: a simulation and a user trial. The simulation demonstrated that the rule mapping functions as expected, producing user models that are still representative of the user, in relation to all other user models. The user trial showed that use of IDIUMS did not result in the adaptive applications presenting content at a more appropriate level, as perceived by the user.

In determining why the user trial did not demonstrate appropriate adaptations, a review of evaluation methodologies in the AES community was undertaken. This showed that the method implemented for the user trial was in the second most common category of sources of evaluation data, behind expert-measured evaluations like pre-post test.

Contents

Acknowledgements **xvii**

Declaration of Authorship **xxi**

1 Introduction **1**

1.1 Research Questions . . . 2

1.1.1 RQ1: Is it possible and meaningful to make rules to map attributes of a user model between different types of application? . . . 2

1.1.2 RQ2: Can mapping from an attribute in one application to an attribute in another application create adaptations that are perceived by the user as being at an appropriate level? . . . 2

1.1.3 RQ3: What is the appropriate method for evaluating shared user modelling for adaptive educational system? . . . 3

1.1.3.1 RQ3a: What is the range of evaluation techniques used within the adaptive educational systems community? . . . 3

1.1.3.2 RQ3b: Are particular types of evaluation prone to particular influences or biases? . . . 3

1.2 Approach . . . 3

1.3 Summary of Contributions . . . 3

1.4 Structure of Thesis . . . 4

2 Literature Review **7**

2.1 Educational Theory . . . 7

2.1.1 Level of Content . . . 7

2.1.2 Maintaining an Appropriate Level . . . 8

2.2 Adaptivity . . . 10

2.2.1 Games . . . 10

2.2.2 Hypertext . . . 12

2.2.3 Sequencing . . . 13

2.2.3.1 Contrasting Simple Sequencing and Adaptive Hypermedia 14

2.2.4 Summary . . . 14

2.3 User Modelling . . . 15

2.3.1 User Models in Adaptive System Architectures . . . 15

2.3.2 Types of User Model . . . 17

2.3.3 Sharing User Models . . . 18

2.3.3.1 Sharing at the Concept Level . . . 19

2.3.3.2 Sharing at the Attribute Level . . . 19

2.3.4 Summary . . . 21

2.4 Educational Games Review 21
2.4.1 Game Design 21
2.4.2 Integrating Educational Content into Games 22
2.4.3 Authoring Difficulties 22
2.4.3.1 Non-Electronic Game Design 24
2.4.3.2 Video Game Design 26
2.4.4 Summary 27
2.5 Conclusions 27

3 Methodology 29
3.1 Research Questions 29
3.2 Meta-Review Methodology 29
3.2.1 Literature Collection 30
3.2.2 What does “adaptive” mean? 30
3.2.3 Initial Analysis of Papers 31
3.2.4 Categorisation of Papers 31
3.2.4.1 Student 32
3.2.4.2 Logs 32
3.2.4.3 Expert 32
3.2.4.4 Baseline 32
3.2.4.5 System 33
3.2.4.6 None 33
3.3 Results 33
3.3.1 Sources of Papers 33
3.3.2 Sources of evaluation data 33
3.3.2.1 By publication 34
3.3.3 Multiple Categories 34
3.4 Analysis 34
3.4.1 Multiple Categories 36
3.4.2 Influence and Bias 37
3.5 Proposed Methodology 38
3.6 Conclusions 38

4 The Design of the System: IDIUMS 41
4.1 Initial Prototype 41
4.2 Exploring the Problem 42
4.2.1 Aim 42
4.2.2 Method 42
4.2.3 Initial Prototype of Adaptive Applications 43
4.2.3.1 Educational Content 43
4.2.3.2 Adaptive Tutorial 44
4.2.3.3 Adaptive Game 44
Game Flow 45
Game Interface and Graphics 46
Compilation and Execution 46
Implementation 46
4.2.4 Discussion 47

4.2.5 Analysis of Other Educational Applications 50
4.2.5.1 Hypertext 50
4.2.5.2 Games 51
4.2.5.3 Assessment 51
4.2.5.4 Virtual Worlds 51
4.2.5.5 Recommender Systems 52
4.2.5.6 Forums 52
4.2.5.7 Social Networks 52
4.2.5.8 Virtual Learning Environments 52
4.2.5.9 Institutional Records 53
4.2.5.10 Scrutable User Model Editor 53
4.2.5.11 Summary of Generic Attributes 53
4.2.6 Summary of the Problem 54
4.3 Proposed Solution 54
4.3.1 Architecture 55
4.3.2 Overview of IDIUMS Use 56
4.3.3 Concrete Example Scenario 57
4.3.4 Options for Mapping Attributes 59
4.3.5 Issues and Limitations 59
4.3.5.1 Coverage 59
4.3.5.2 Map on read or write 60
Decision 61
4.3.6 Comparison to GUMF 61
4.4 Design 62
4.4.1 Rules 63
4.4.1.1 Types of Rules 63
4.4.1.2 Thresholding 64
4.4.1.3 Straight line graph 64
4.4.1.4 Mathematical equation 64
4.4.1.5 Rule Chaining 65
4.4.1.6 Pseudocode 65
Definitions 65
Code 65
4.4.1.7 Rule Conflict Resolution 66
4.4.1.8 Example Return for Different Resolutions 66
4.4.2 Data Design 69
4.4.3 HTTP API 69
4.4.3.1 Return Type 69
4.4.3.2 Query String Parameters 69
4.4.3.3 Example Responses 69
4.4.3.4 Updating Values in the User Model 71
4.4.3.5 History 71
4.5 Implementation 71
4.5.1 Rule Creation 72
4.5.2 Rule Evaluation and Reversal 72
4.6 Conclusion 74

5 Simulation **75**
5.1 Introduction . . . 75
5.1.1 Research Question . . . 75
5.1.2 Structure of Chapter . . . 76
5.2 Methodology . . . 76
5.2.1 Determining Whether Mapping Works . . . 76
5.2.1.1 Limitations . . . 79
5.2.2 Generation of Virtual Users . . . 79
5.2.3 Constraints . . . 80
5.2.3.1 Range of values . . . 80
5.2.3.2 Relative constraints . . . 80
5.3 Results . . . 80
5.3.1 Graphs . . . 80
5.4 Analysis . . . 82
5.4.1 Effect of duration attribute . . . 83
5.4.1.1 Discussion . . . 84
5.4.1.2 Overall . . . 86
5.5 Conclusions . . . 86
5.5.1 Future Work . . . 86

6 User Trial **89**
6.1 Research Questions . . . 90
6.2 Experimental Design . . . 91
6.2.1 The Applications . . . 91
6.2.1.1 Java Loops Adaptive Educational Hypertext and Assessment . . . 91
6.2.1.2 Java Loops Adaptive Educational Game . . . 93
6.2.1.3 IDIUMS . . . 93
Limitation . . . 93
6.2.2 Experimental Procedure . . . 94
6.2.3 The Participants . . . 94
6.2.4 Order of Use . . . 95
6.2.5 Time . . . 95
6.2.6 Control . . . 96
6.2.7 Data . . . 96
6.2.8 Expected Outcomes . . . 96
6.3 Analysis . . . 97
6.3.1 Samples . . . 97
6.4 Results . . . 98
6.4.1 The May 2013 Experiment . . . 98
6.4.1.1 Differentiating samples by difficulty of content at start of application . . . 98
Test for normality . . . 99
Test for different populations . . . 99
6.4.1.2 Correlation between user self-rated understanding and IDIUMS value . . . 100
6.4.2 The November 2013 Experiment . . . 101

6.4.2.1 Differentiating samples by difficulty of content at start of application . . . 101
Test for normality . . . 101
Test for different populations . . . 101
6.4.2.2 Correlation between user self-rated understanding and IDIUMS value . . . 102
6.5 Discussion . . . 102
6.6 Conclusions . . . 104

7 Conclusion 105
7.1 Addressing the Research Questions . . . 105
7.1.1 RQ1: Is it possible and meaningful to make rules to map attributes of a user model between different types of application? . . . 105
7.1.2 RQ2: Can mapping from an attribute in one application to an attribute in another application create adaptations that are perceived by the user as being at an appropriate level? . . . 106
7.1.3 RQ3: What is the appropriate method for evaluating shared user modelling for adaptive educational systems? . . . 107
7.1.3.1 RQ3a: What is the range of evaluation techniques used within the adaptive educational systems community? . . . 107
7.1.3.2 RQ3b: Are particular types of evaluation prone to particular influences or biases? . . . 107
7.2 Contributions . . . 108
7.3 Future Work . . . 109
7.3.1 Shared User Modelling . . . 109
Cold-start vs hot-start . . . 109
Run user trial with learning gain . . . 109
7.3.2 Further Reviews of the AES Literature . . . 109
What are the key trends in adaptive systems and user modelling? . . . 109
Taxonomy of all adaptive systems . . . 109
Are adaptive systems a comparable to human tutoring? . . . 110
7.3.3 Development of IDIUMS . . . 110
Resolution of Multiple Values . . . 110
Investigate Expert Rule Creation . . . 110
Investigate Educational Data Mining for Rule Creation . . . 110

A Survey Questions 111

B Domain Model 115

C Domain Model Comparison 117

References 119

List of Figures

2.1 The adaptive architecture presented by Prince and Davis (2008). 12
2.2 The ALIGN architecture by Peirce et al. (2008). 13

3.1 Bar chart of count of combinations of categories 36

4.1 A screenshot of the domain model being created in the AHA! Graph Author tool. 45
4.2 A screenshot of the domain modelling editor for GALE (Dicerto and Mazzetti, 2010) 46
4.3 A screenshot of a stage within the *While loops* learning object by Boyle (2006). 47
4.4 A flow diagram depicting the expected order of events in the game. 48
4.5 A screenshot of the Loop Game displaying an example level. 48
4.6 Architecture diagram of the various components of IDIUMs, and how they interface with external applications. 55
4.7 Demonstration of the flow and conversion of attributes between the LoopGame application and GALE hypertext application. 57
4.8 Quadruples produced by LoopGame as Alice plays it. 57
4.9 This graph plots the relationship between successes/attempts and the knowledge value as defined in Table 4.2. Red dashed line demonstrates interpolation of a value between those in definition. 58
4.10 Quadruples produced by IDIUMS to provide to GALE. 58
4.11 A enhanced entity relationship diagram of the database underlying IDIUMS. 68
4.12 Screenshots of adding rules in the IDIUMS `/edit` interface. 73

5.1 Simulation run on pilot experiment data. 82
5.2 Simulation run on pilot experiment data. 85

6.1 Screenshot of the adaptive game running 93
6.2 Questions asked at the end of each application usage. 97

List of Tables

2.1 A comparison of some well known adaptive hypermedia architectures, and the types of model they explicitly define in their architectures. 16

3.1 Sources of adaptive educational systems papers. 33
3.2 Categories by publication. 34
3.3 Count of how many papers fit into multiple categories. 34
3.4 Table of which papers fit into which categories. 35

4.1 Summary of the measurable attributes in common types of educational applications, and whether they are measured explicitly or only inferred. . 53
4.2 Example of an non-linear relationship between success/attempts and knowledge. 58
4.3 Example of a possible relationship between a score-based user model and a stereotype-based user model. 64
4.4 Definitions of the terms *Variable*, *Binding* and *Rule*. 65
4.5 Example resolved values for various suggested resolution rules. 68
4.6 Mapping between URL and return content-type. 69

5.1 Input and output pairs for simulation. 76
5.2 Attributes in each each application's user model (attributes in parentheses were not used). 77
5.3 The rules configured in this version of IDIUMS. 78
5.4 Pearson's correlation of distance between pairs. 81
5.5 Pearson's correlation of distance from origin. 81
5.6 Pearson's correlation of distance between pairs, without duration attribute. 83
5.7 Pearson's correlation of distance from origin, without duration attribute. . 84

6.1 The potential routes through the user trial a participant might be allocated. 95
6.2 Samples gathered during the user trial. 98
6.3 Shapiro-Wilk normality test results for the samples in May 2013 99
6.4 Wilcoxon rank sum test results for the samples in May 2013 99
6.5 Student's t-test results for the samples in May 2013 99
6.6 Pearson's product-moment correlation results for the samples in May 2013 100
6.7 Shapiro-Wilk normality test results for the samples in November 2013 . . 101
6.8 Student's t-test results for the samples in November 2013 101
6.9 Wilcoxon rank sum test results for the samples in November 2013 102
6.10 Pearson's product-moment correlation results for the samples in November 2013 102

Listings

4.1 XML example of multiple values in IDIUMS response 66
4.2 JSON example of multiple values in IDIUMS response 67
4.3 Example of requesting the success attribute for the loops concept in the Loop Game and receiving an HTML page. 70
4.4 Example of requesting the success attribute for the structure concept in the Loop Assessment and receiving a JSON response. 70
4.5 Example of requesting the success attribute for the loop variable concept in the hypertext and receiving an XML page. 70
4.6 Format of rules in IDIUMS. 72

Acknowledgements

Uncountable thanks to Emma, Meredith, Ray, Karin, Sam, Ross, Pat, Charlie, Al, Briony, Jonny, Tom, Sami, Dayah, Tim, The Princes, The Forsmans, The Lee-Choons and all friends and family that have shown me love and support of all forms throughout the time I have been working on my PhD. I know it cannot have been easy all the time and I am sorry for sometimes not giving you the attention you deserve. Now I have acheived this goal, I look forward to making up for that!

Enormous gratitude to my supervisors Hugh and Dave, my examiners Paul and Les, formative advisors and examiners Gary and other Hugh, my colleagues Yvonne and Mark, the members of the Learning Societies Lab (LSL), and the members of the Web and Internet Sciences (WAIS) research group. I consider you all friends first and foremost, and wholeheartedly appreciate the advice, encouragement, guidance, intellectual stimulation and personal support you have contributed during the course of my studies, both undergraduate and postgraduate. I could not have made it here without you!

To everyone else who has been on some part of this journey with me, thank you :-) I set out with the aim of completing this PhD without compromising on the things I enjoy doing, whether that be teaching and learning new things, participating in sports, playing computer, video and board games, or getting to know interesting and fun people. On the whole I think I have achieved that, so thank you for helping me on that path!

For Mormor, Nan, Jim and Morfar.

Declaration of Authorship

I, ..., declare that the thesis entitled ....

and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

1. this work was done wholly or mainly while in candidature for a research degree at this University;
2. where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. where I have consulted the published work of others, this is always clearly attributed;
4. where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. parts of this work have been published as:
   - Prince, R. and Davis, H. C. (2008). The Scope of Adaptive Digital Games for Education. In *Digital Games and Intelligent Toys Based Education, 2008 Second IEEE International Conference on*, pages 167–169
   - Prince, R. (2009). Sharing User Models with Adaptive Games. In *Adjunct Proceedings of User Modeling, Adaptation, and Personalization 2009 Doctoral Consortium*
   - Prince, R. and Davis, H. C. (2011). IDIUMS: sharing user models through application attributes. In Abel, F., Baldiris, S. M., and Henze, N., editors, *Adjunct Proceedings of the 19th International Conference on User Modeling, Adaptation and Personalization (UMAP)*

Signed:..

Date:..

Chapter 1

Introduction

Over the course of studying a PhD, you get asked with trepidation what the topic of your thesis is on, what felt to me like, an infinite number of occasions. Over time, this "elevator pitch" is refined and repeated until only the simplest kernel remains. It is perhaps worthwhile introducing this thesis with such a description, as it defines the scope of this work.

Start by thinking about a system like the Amazon website. On this site you are constantly presented recommended links to new books, CDs and almost every other thing you can buy, based on what you have purchased before. This is an adaptive system, powered by a recommender system.

These same technologies are used in Technology Enhanced Learning (TEL). Instead of suggesting a new DVD to buy, adaptive educational systems (AES) will guide a student to the next topic, question or exercise that is most appropriate for their current stage of learning. Where an e-commerce adaptive system will make decisions based on your purchase history, an adaptive educational system will use the pages you have read, your scores in quizzes and many other metrics to predict your knowledge of a topic, and therefore present appropriate information.

As yet, there is no one learning provider that is as all encompassing as Amazon; examples of AES are drawn from researchers, teachers and projects from a variety of institutions. The result is that each AES becomes a silo for the learner's data. One AES may have recorded what the learner knows, but any other AES will look upon a new user as a novice. This is known in this research field as the *cold-start problem*. It is an active research area to invent suitable technologies, standards and processes to enable and encourage these applications to share their users' data with each other.

Within this thesis I approach this problem within the context of those AES being significantly different in their functionality and interactivity. New technologies have been adopted when building educational systems, so where we once just had digital versions of books, now there are interactive quizzes with immediate feedback and educational serious games with all the degrees of freedom afforded by a virtual environment. The way an application records a user's interaction with a system is no longer a single metric such as whether or not a page has been read, or whether or not a CD has been bought, but is far richer. It is the exploitation of that richer data that will enhance sharing of user data between adaptive educational systems, and result in a more satisfying learning experience.

1.1 Research Questions

This thesis sets out to answer one main question, which consists of two sub-questions, and in answering that main question produced another. These are described here.

1.1.1 RQ1: Is it possible and meaningful to make rules to map attributes of a user model between different types of application?

The premise underlying this research is that is that the user model that an adaptive application measures about its user can be transformed in a way that makes is compatible with a different adaptive application. The approach considered is one utilising expert-authored rules that define logical or mathematical relationships between the user models. This research question asks whether doing this transformation produces user models that still represent the user.

1.1.2 RQ2: Can mapping from an attribute in one application to an attribute in another application create adaptations that are perceived by the user as being at an appropriate level?

The ultimate aim of this thread of research is to discover a mechanism that will allow a user to start using a new adaptive application, and for it to immediately adapt to their level of ability, based on their prior usage of other adaptive applications. This research question asks whether a mapping approach to transforming user model data has an effect on an adaptive application's ability to immediately adapt content to an appropriate level for a new user.

1.1.3 RQ3: What is the appropriate method for evaluating shared user modelling for adaptive educational system?

In trying to answer RQ1, the evaluation of a user trial culminated in an unexpected result. A new research question arose from this, to determine whether the choice of methodology had an impact on the result.

1.1.3.1 RQ3a: What is the range of evaluation techniques used within the adaptive educational systems community?

The evaluation of RQ1 was one of a number of approaches that could be taken to test the performance of adaptive systems. This research question asks what other approaches could be taken and which are most popular in the community.

1.1.3.2 RQ3b: Are particular types of evaluation prone to particular influences or biases?

Each type of evaluation used in the adaptive educational systems community has different factors that may make it more or less susceptible to influences that diminish the accuracy of its findings. This research questions digs deeper into which evaluation criteria are susceptible and why.

1.2 Approach

The approach taken to answering these research questions is to design and develop a piece of software that will implement a solution to the cold-start problem at an attribute level, and then evaluate its effectiveness. It is important to actually build an exemplar of such a system, as this is a very practical problem, so a pragmatic solution is needed. While this work will only produce a prototype of this software, it will enable the research questions to be answered, and will serve as a basis for further work in this area.

1.3 Summary of Contributions

In addressing the research questions above, a number of contributions have been made by this work:

1. An intermediary user model system has been designed, and a reference implementation built.

2. A simulation framework for testing the functional performance of shared user modelling systems has been implemented. From this we know that IDIUMS operates as expected.

3. A user study was performed to test the impact of IDIUMS on the adaptation of content, which revealed that users did not perceive any difference in the adaptation.

4. A meta-review of recent AES literature highlighted that using the participant as a source of data for evaluations (e.g. through self-rating in surveys) is the second most common approach, following expert-led measurements (such as pre-post testing).

1.4 Structure of Thesis

This thesis describes the design and reference implementation of a shared user modelling system, the testing and evaluation of that system through a simulation and a user trial, and a systematic review of evaluation methodologies used in the research community. Over the course of seven chapters, this thesis addresses the research questions detailed above.

Chapter 2 is a literature review of the state of the art within adaptive systems, with a particular focus on user modelling and sharing. Also, included here is some background on educational theory relevant to understanding this area and on the design of educational games, as a basis for understanding the unique challenges of sharing user model data from games to more traditional Adaptive Educational Systems.

Chapter 3 conducts a meta-review of recent Adaptive Educational Systems literature to discover and critically appraise the common methodologies for evaluating AES, and defines a methodology for this piece of work.

Chapter 4 reports on the creation of an initial prototype of some adaptive applications, to tease out the challenges of sharing user model data between applications of different types. Following this, it proposes the design of a system that will allow this sharing and alleviate the cold-start problem for adaptive applications. Finally, it describes the creation of a reference implementation of this system.

Chapter 5 details a simulation that was implemented to test whether the user model sharing system is performing its mapping accurately. It does this by generating a batch of virtual users (based on data gathered from a pilot test), measuring how close or spread the user models of those virtual users is, performing the mapping using the system and repeating the measure afterwards.

Chapter 6 presents a user trial that was conducted twice, to determine whether use of the user model sharing system resulted in adaptive applications performing better adaptations than if they begun in a cold-start situation.

Chapter 7 concludes this thesis, by summarising the chapters that have preceded it and explaining how various threads of that work have addressed the research questions set out earlier in this chapter.

Chapter 2

Literature Review

2.1 Educational Theory

Adaptive Educational Systems (AES) exist to meet the need of automatic, scalable, personalised learning. Before considering why adaptivity is crucial to delivering this, it is important to understand the educational theory that underlies how these systems work.

2.1.1 Level of Content

Over the course of five years, through a series of informal annual meetings, Bloom and his associates produced a taxonomy of educational objectives (Bloom et al., 1956). The taxonomy is their classification of the types of outcomes an educator may intend to meet through the teaching of a particular course or curriculum.

Bloom's work originally specified three domains: the cognitive, the affective and the psychomotor domain. The group's second book, subtitled "The Affective Domain" (Krathwohl et al., 1964), classifies objectives for learners to understand and analyse their feelings and attitudes towards any particular subject. The psychomotor domain, for which Bloom and his colleagues did not create a taxonomy, relates to teaching how to perform and improve physical actions. The cognitive domain is the sort of learning that is seen in most formal education: knowledge and analytical thought processes.

Within the cognitive domain, Bloom identified six categories of "the student behaviors which represent the intended outcomes of the educational process". These are knowledge, comprehension, application, analysis, synthesis and evaluation.

The knowledge category refers to remembering specific information, how to handle that information and abstractions such as principles and theories. Comprehension is "the lowest level of understanding" and as such is simply the use of a concept in isolation. Comprehension includes extrapolating an identified trend and faithfully translating a communication into another format. To meet an application objective, a student must take an abstraction known to them, such as a generalised method, and use it in a real situation, to perform that intended method.

An analysis of a particular communication recognises the structure and the relationships between the elements, and summarises or explains these as appropriate. It should also identify "unstated assumptions" and "implicit structure" to show full understanding of the original communication.

Synthesis is the process of creating a new piece of work, possibly by combining existing elements in a new way. This may take the form of a unique piece of writing or the creation of a plan. Evaluation is the ability to make judgements about a communication, based on evidence within the communication itself, or against a list of expected criteria drawn from other high quality works.

The importance of planning the learning objectives ahead of creating the educational materials is supported by Gilbert and Gale (2007), for the technology-enhanced learning domain. In their Project Development Framework for "e-learning systems engineering" (ELSYE), Gilbert and Gale recognise "top level objectives" and "enabling objectives" as key elements of the front-end analysis and initial design phases of e-learning systems engineering.

2.1.2 Maintaining an Appropriate Level

A number of theories in educational theory and psychology point to the importance and subtleties of providing learning at a level appropriate to the student.

Constructivist learning theory suggests that knowledge is not some existing entity waiting to be learned, but an artefact *constructed* by each individual learner from their experiences in relation to their existing knowledge and previous experiences. Following the constructivist viewpoint ensures an emphasis on the individual learner and requires that learning experiences be matched to the learner: "different students will approach the construction of knowledge in different ways, and the educational environment must be supportive of these differences" (Ben-Ari, 1998).

Vygotsky (1978) declares that it is not enough to judge a pupil's mental development solely on the results of tests of their ability to independently complete tasks predetermined to be appropriate for a child of that age; instead he suggests that it is important to look at what the pupil is able to do with the assistance of an adult or a more able peer. Vygotsky calls the gap between what a child can do independently and what a child can do when assisted, their *zone of proximal development*. In an example from McCarthy

(1930), it is shown that the tasks that children aged 3-5 can do collaboratively, are ones they can do independently when they are aged 5-7, suggesting that students experiencing good learning when performing tasks in their zone of proximal development.

Another theorem where a careful balance between psychological states results in enhanced performance is Csikszentmihalyi's *flow* 2008. Flow occurs when the challenge of a task is perfectly balanced against their current abilities, and the individual performing the task enters a state of full immersion in the activity. In Chen (2006, 2007), where flow is applied to computer games, this is presented to mean that if the task is too difficult the individual is likely to experience anxiety, and if it is too easy they will become bored, but this is simplified representation of the 8-channel model in Massimini et al. (1987).

The complimentary nature of these two theories was identified by Basawapatna et al. (2013), who suggested that the Zone of Proximal Development sits in the region beyond Flow but before anxiety, as even when in the state of Flow, learners may need additional support from a peer or tutor to progress their development. Basawapatna et al. (2013) created a framework that combines thw two theories, and named it the *Zones of Proximal Flow*. They demonstrated its use for teaching game design during the *Scalable Game Design project*, through a combining project-based learning (PBL) to put students in a state of Flow, and just-in-time teaching (JiTT) to return them to Flow if the challenge of the tasks moves them into their Zone of Proximal Development (and to stop them progressing to their zone of anxiety).

In her 1993 book, Laurillard (1993) describes a teaching and learning process called the conversational framework. Within this framework, a teacher and a student engage in a discussion about the topic that the student wishes to learn, starting with the teacher and student exchanging their understanding of the concept being discussed. From the student's attempt to describe the concept, the teacher can then alter their explanation, in a way that may improve the student's understanding.

Based on this discussion, the teacher can, if necessary, provide the student with a series of problems that encourage the student to apply the concept in an imaginary situation they may understand better than an abstract explanation of the concept. Laurillard describes this as the teacher constructing an environment or world, and the student performing actions within this world. Both student and teacher are then able to reflect on the interaction within this environment, so the student can modify their understanding of the concept and the teacher can again explain the concept in light of the student's correct or incorrect application of it.

The lesson from this literature is that good education involves reacting to what the student does and adjusting the teacher's responses to meet the needs and the level of that particular student. The next section will look at some technologies that enable this to be done by a computer.

2.2 Adaptivity

Laurillard's conversational model points towards an ideal education system where each student is mentored by a tutor, who can tailor their teaching material, style and speed of delivery to suit the student's needs.

There is scope for this form of education to be delivered through existing games. For example, Steinkuehler (2004) demonstrates how an existing community, within the game Lineage, provides ad hoc apprenticeships to new players that wish to join their community. Through a player-to-player conversation, set in the world itself, the experienced player scaffolds the beginner by instructing them, demonstrating, allowing the beginner to attempt the task and then offering feedback on how well the beginner performed.

However, in a society that relies on mass education, one-to-one mentoring is clearly not a viable solution. The advent of technology-enhanced learning creates the opportunity to provide personalised learning experiences to anyone with access to a computer. The adjective used to describe such applications is *adaptive*. These adaptive applications take previously prepared resources, and selectively display or modify them based on a model of the user that the application has built up through the user's interactions with the system. Adaptive technology could act as a substitute for the tutor and allow this one-to-one method of teaching to reach anyone with a computer (Brailsford et al., 2001).

This chapter describes the application of adaptive technologies in digital games and hypertext.

2.2.1 Games

The literature reveals some existing attempts at making some part of educational games adaptive.

In 2004, Conati and Zhao presented their work on a pedagogical agent, which augmented *Prime Climb*, an existing two-player game which aims to help eleven to thirteen year olds learn factorisation. In the game, a player can only move to a number that does not have any common factors with their partner's current number. Conati and Zhao's software agent models the student's knowledge of factorisation using a Dynamic Bayesian network, and then uses the predicted probabilities to select an appropriate hint from a set of eight.

A slightly different approach is to adaptively select the most appropriate game for the learner, from a repository. Carro et al. (2006) created a methodology and model for defining an adaptive educational environment. This system allows the learner's age, as well as language and media preferences, to be represented. Correspondingly, games within the system are allocated similar attributes, and can also have goals, difficulty levels and a description of themselves. Activities are then specified, which indicate the type of game that should be played. Decomposition Rules define a how Activities should be combined to form an Activity Group, which also have a theme, or “Scenery”, associated with them.

Moreno-Ger et al. (2007) demonstrated how an adaptive game could be created as an IMS Learning Design Unit of Learning (IMS-LD UoL), allowing it to be re-used anywhere IMS-LD is supported, such as in a virtual learning environment (VLE). Using a game engine called <*e-Adventure*>, which enables the creation of educational point-and-click adventure games, Moreno-Ger et al. created *Paniel and the chocolate-based sauce adventure*, which had three stages in its storyline. Firstly the learner uses virtual library books to discover recipes. Then the learner has to create some sauces, ready for the final exam. Lastly, the learner takes an in-game exam where they have to match the sauces they have created to the tastes of restaurant customers.

The adaptive element in this game is in the form of skipping stages based on previous experience. The Unit of Learning begins with a questionnaire to determine the learner’s knowledge of the subject matter. Based on the score from this, the learner either starts at the beginning, skips to the second level, or goes straight to the final exam. If the system was part of a full virtual learning environment, with a life-long user model, then much of what is gathered from the questionnaire could be filled in by the existing user model.

In addition, researchers in the video games realm have considered how games could adapt so that they are better suited to the player playing them. Charles et al. (2005) suggest that “a game may be adapted through changes to a player’s character, non-player characters in the game and the game environment or state”. Possible ways of doing this include non-player characters providing clues, and increasing or decreasing the number of bonus items available to collect in the world. They propose an approach to player modelling that uses a set of discrete descriptions of playing styles and abilities, and then assigns a value for each to indicate the extent to which the player fits that style.

Hunicke (2005) describes the process of Dynamic Difficulty Adjustment (DDA) as an economic system, where the game engine modifies the supply of resources relative to the demand to balance the difficulty of the game as perceived by each player. The example studied in the paper is to calculate the probability of a player dying in a particular combat encounter, within a first-person shooter (FPS). The DDA system implemented by Hunicke, named *Hamlet*, can be given different policies for detecting when to intervene and strategies for how to intervene. The strategy used by the experimental system is to add health to the player every 3-4 seconds if the “player’s probability of death is greater than 40%”.

As a result of these early explorations into using adaptivity in games, there have been architectures developed to help developers of new adaptive educational games structure their systems. A poster by Prince and Davis (2008) proposed an architecture (see Figure 2.1) heavily influenced by adaptive hypermedia architectures such as AHAM (De Bra et al., 1999). As such, it contains a reference to a lifelong user model, an adaptation engine fed by a set of adaptation rules and a game described in its component parts (rules and goals, controls and display, and digital content). The content of the described game is adapted by the adaptation engine and fed into the game engine. As it is played, data about the interactions and performance of the player is fed into the lifelong user model.

Peirce et al. (2008) present their ALIGN (Adaptive Learning In Games through Non-invasion) architecture (see Figure 2.2), as implemented in the ELEKTRA project. As well as providing a method for adapting non-invasively (by tracking consistency constraints, so that adaptations make themselves obvious by breaking continuity), ALIGN separates the modelling of the user, filtering of adaptive elements and the adaptation decision making process out from the game engine. This allows any game to interface with ALIGN as long it provides the relevant evidence to ALIGN, and can act on the adaptations that ALIGN recommends.

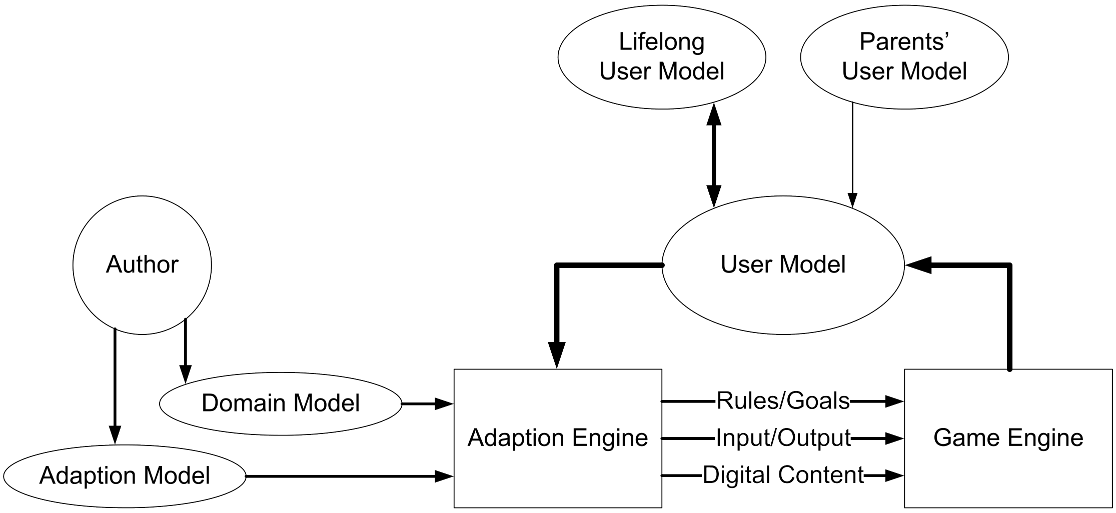


Figure 2.1: The adaptive architecture presented by Prince and Davis (2008).

2.2.2 Hypertext

An area that has benefited from the addition of adaptivity is hypertext, the document linking technology that underpins the World Wide Web, and has been widely used in educational applications (Brusilovsky, 2001; De Bra, 2002).

Adaptive Hypermedia consists of a body of research that has been developing since the early 1990s, as researchers realised that systems that served hypertext content could be

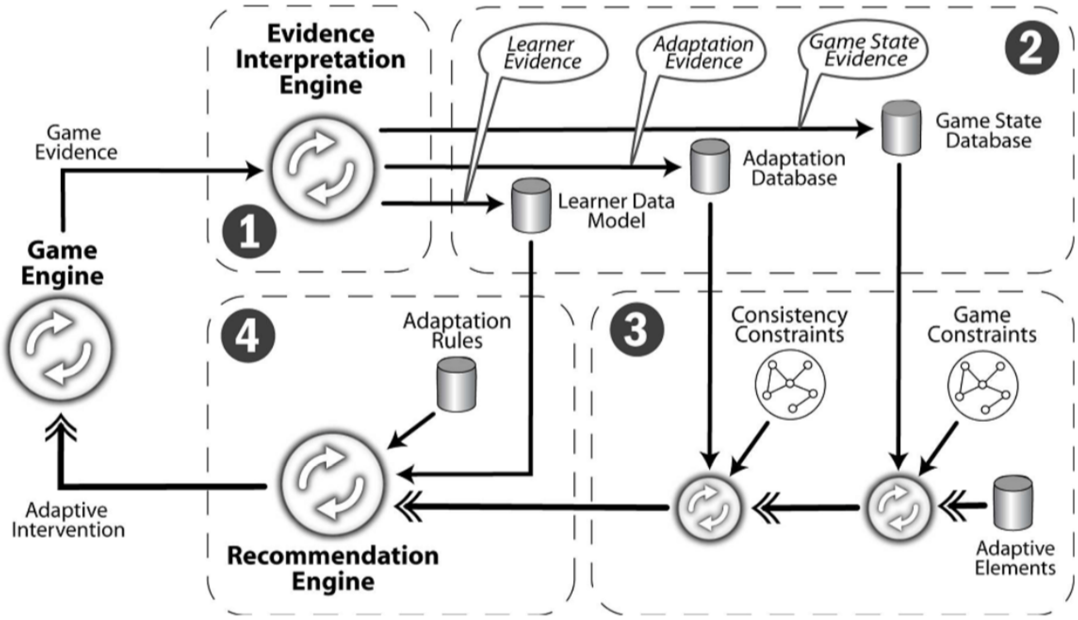


Figure 2.2: The ALIGN architecture by Peirce et al. (2008).

designed to change what text and links were displayed, depending on the user's "goals, preferences and knowledge" (Brusilovsky, 1996).

A number of architectures for constructing Adaptive Hypermedia systems have been proposed, including the Adaptive Hypermedia Architecture Model (AHAM) by De Bra et al. (2003) and the Generic Adaptivity Model (GAM) by de Vrieze et al. (2004). Common to most of these architectures are a domain model, describing the resource being accessed; a user model, which represents the user's "goals, preferences and knowledge"; and an adaptation model, containing rules that determine what material to display or hide based on the state of the user model.

The range of possible adaptations is set out in the taxonomy of adaptive hypermedia technologies described by Brusilovsky (2001). The taxonomy explains that existing fragments of text (represented in the domain model) can be inserted, removed or put into a different order. The links that are already present in the source material can be hidden, sorted or have extra information attached to them. In the extreme cases, the taxonomy even suggests that completely new links could be added or natural language text fragments generated, without the original author of the material ever having to put the information in the system.

2.2.3 Sequencing

IMS Simple Sequencing (IMS Global Learning Consortium, Inc., 2003) is a specification from the IMS Global Learning Consortium, Inc. and is a means of ordering a set of learning activities. IMS have created a custom XML markup that allows the instructor to organise the learning activities they have available to them in a tree structure. The instructor can then create guided routes through the content for students to follow, or allow the student to navigate the material themselves. The instructor can also choose to put conditional rules on the branches of the tree structure. This allows the instructor to limit the student's access to certain material until they have performed a particular learning activity (e.g. read a certain resource) or have achieved a certain level in a learning activity (e.g. 75% in a test).

2.2.3.1 Contrasting Simple Sequencing and Adaptive Hypermedia

Abdullah and Davis (2003) discuss the differences between Simple Sequencing and Adaptive Hypermedia, noting that Simple Sequencing is designed with the instructor in mind, allowing them to order the learning activities in a course, whereas Adaptive Hypermedia is focused on the user, what they already know and what they want to learn.

The two types of system also differ in how they track the user and how they personalise the experience to the user. Simple Sequencing records completion of activities in a Tracking Model, while Adaptive Hypermedia stores attribute-value pairs for concepts in a User Model. Adaptive Hypermedia appears to have a finer grained model of what the user already knows or has experienced. Adaptive Hypermedia also has a more detailed list of techniques for adapting the experience to the user. The taxonomy of adaptive hypermedia technologies (Brusilovsky, 2001) lists numerous ways, such as inserting or removing fragments of text, and sorting and hiding the available hyperlinks.

Prince and Davis (2008) used the analysis of this difference to determine how a game could be made truly adaptive, rather than just sequenced. This involved an architecture similar to Adaptive Hypermedia systems such as AHAM and GAM, and an adaptive game engine which utilised flexible input mechanisms, and procedural generation technologies to produce different gaming experiences for each player.

2.2.4 Summary

Adaptivity is an important feature of educational software in the ongoing pursuit of providing a learning environment that is personalised to each individual learner, so that they construct their understanding of the world with as few barriers as possible. It is a technique used in many areas, and as such, computer science researchers have developed a number of architectures for developing software of this sort. This review focussed on its applicability in hypertexts and computer games.

2.3 User Modelling

A core part of adaptive architectures is the user model, and is the component that is most crucial for sharing as a user moves between adaptive applications. This section of the literature review analyses previous work in the field of user modelling, introduces the challenge of sharing user model data, and the more recent work on interoperability and sharing user model data.

2.3.1 User Models in Adaptive System Architectures

To put the user model in context, it is worth considering where it is used in a system. Adaptive applications are pieces of software that can change the experience they deliver to the user based on the inputs they receive. Many adaptive systems choose to personalise the content they display to the user based on what they determine will be most appropriate to the current user. To be able to make this inference, the system therefore needs to have some data about the user.

Two prevalent areas of research into adaptive applications are Intelligent Tutoring Systems (ITS) and Adaptive Hypermedia (AH). In fact, adaptive hypermedia stemmed from research into Intelligent Tutoring Systems (Sleeman and Brown, 1982), when researchers began to apply the techniques to hypertext systems such as Microcosm (Fountain et al., 1990) and the World Wide Web (Berners-Lee, 1992) in the early 1990s. This relationship with tutoring systems may also explain the large number of adaptive hypermedia systems which are built for education (Brusilovsky, 2001; De Bra, 2002).

By studying the architectures for these adaptive systems, such as the Adaptive Hypermedia Architecture Model (AHAM) by De Bra et al. (2003), the game-based ITS from Mills and Dalgarno (2007) and the Generic Adaptivity Model by de Vrieze et al. (2004), it is possible to list the common elements. These are the domain model, the user model and the adaptation model.

The domain model defines the information that the system can provide. For example, AHA! (De Bra et al., 2003) represents the domain as a set of concepts, which are typically related to a particular page within the hypertext.

The user model is used to represent the "goals, preferences and knowledge" (Brusilovsky, 1996) of the person using the system. As the key to this review, this is discussed in further detail below.

The adaptation model describes to the adaptive engine how to react to the various interactions the user may have with the system, by linking elements of the domain model to states the user model can be in. A simple example is for an adaptive hypermedia

Architecture	Domain Model	User Model	Adaptation Model	Instructional or Teaching Model	Goal and Constraints Model	Interface Model
InterBook[a]	Inferred	Overlay				
AHAM[b]	Yes	Overlay		Yes		
WHURLE[c]	Inferred	Overlay and Stereotype				
ELM-ART[d]	Inferred	Overlay and Episodic				
LAOS[e]	Yes	Yes	Yes		Yes	Yes
GAM[f]	Yes	Overlay	Yes			Yes
GBL-ITS[g]	Yes	Constraint-based		Yes		

[a]Brusilovsky et al. (1997)
[b]De Bra et al. (1999)
[c]Zakaria and Brailsford (2002)
[d]Weber and Specht (1997)
[e]Cristea and de Mooij (2003)
[f]de Vrieze et al. (2004)
[g]Mills and Dalgarno (2007)

Table 2.1: A comparison of some well known adaptive hypermedia architectures, and the types of model they explicitly define in their architectures.

system to hide (not display) a link to a page if the user model believes that the user does not have the prerequisite knowledge for that page.

Table 2.1 summarises the types of model that are explicitly defined as part of seven prevalent adaptive hypermedia architectures. In some cases, the models can be inferred; for example, the existence of an overlay *user model (UM)* suggests that there must be a *domain model (DM)* to overlay, even if it is not explicitly mentioned.

It may also appear odd that any adaptive application may exist without an *adaptation model (AM)*, but the distinction lies in whether the *adaptive engine (AE)* represents the way adaptations occur separate to the algorithm that processes adaptations. By separating these, it allows adaptations to be manipulated externally and authored, and enables different content to have different adaptations associated with them. Adaptive systems without an adaptive model will typically have a fixed algorithm that applies the same rule(s) to each adaptation, using the domain and user models as its inputs.

Instructional models or *teaching models* are a form of specialised adaptation model that are designed around pedagogical rules that define how to adapt at each stage. The *Goals and Constraints* heading in Table 2.1 refers to the fact that some architectures extend their user model with a model of the goals of the user or any restrictions appropriate to be used in the adaptation. A number of generic adaptive architectures also include a separate model for the way the user can interact with the content, labelled here as an *interface model*, to allow adaptive system administrators and authors to have more control over the user interface of their adaptive application. Architectures without an interface model will typically have any functionality that adapts the user interface built into the adaptive engine.

2.3.2 Types of User Model

Researchers have found many ways of representing the user within their adaptive applications. The variety is due to the domains for the applications, the backgrounds of the researchers and iteration on previous ideas.

The overlay model is a common user model implementation (Zakaria and Brailsford, 2002) and as identified in the survey of adaptive architectures recorded in Table 2.1. It is tightly linked to the domain model of the system, as it represents the user's knowledge as a subset of all the possible concepts within the entire domain (Kay, 2000; Nguyen and Do, 2008). Each user has a value against each concept in the domain, which indicates how well they understand the concept. Nguyen and Do (2008) talks about a vector model, which is a simplified version of the overlay model, simply storing a vector of values.

Related to the overlay model is the differential model. Where the overlay model represents what the user knows within the whole domain, the differential model only stores the concepts that the system intends to teach to the user Kay (2000). The distinction is subtle and which of the two is used could be determined by how the system is built. If the domain model is provided by another source, contains every concept relevant to the domain and the system only intends to teach part of it, then the model will be overlay. If the domain model is designed for the system and contains only the concepts being taught, then the model is differential.

Designed almost intentionally as a contrast to the overlay model the buggy (Kay, 2000), fault (Nguyen and Do, 2008) or constraint-based model (Mitrovic et al., 2001) records errors in the user's knowledge.

The stereotype model (Kay, 2000) takes a very simplistic approach by allocating the user to one of a few finite profiles, such as "beginner", "intermediate" and "expert". From the user's interaction with the system they are allocated to one of the stereotypes, which determines what the system presents to the user.

Kavcic (2000) identifies that there are other, more mathematical ways to model the user, including Bayesian, machine learning and logic-based methods.

One of the challenges for systems which model the user is how to initialise the model when the user first uses it. Sosnovsky et al. (2007) identifies three methods for initialising a new user's model and also comments that none of these are a perfect solution. The simplest approach is to leave the user model empty and assume the user knows nothing about the domain, which will result in suboptimal adaptations if the user does have some knowledge of the domain; this is known as the *cold-start problem*. Another solution is to expose the user model and its raw values to the user so they can manually edit them. While this is certainly scrutable Kyriacou and Davis (2008); Kay (2000) and flexible, it also expects the user to be able to interpret the values and understand how the system will use them. Sosnovsky uses ELM-ART as an example of the third type of initialisation: using a detailed questionnaire or quiz to determine the user's knowledge and feed this information into the user model.

Finally, an alternative to having to explicitly initialise a new user model is to retrieve relevant information about the user from other shared, interoperable user models (Carmagnola, 2008).

2.3.3 Sharing User Models

Investigating systems, methods and architectures for sharing data between user models is a particularly active research area. The benefits to sharing information about the user between systems include the ability for systems to efficiently initialise a new user model or to even handle a user jumping straight into the middle of an adaptive system, such as being linked from a search engine to a page in the middle of an adaptive hypermedia system (which might normally expect the user to work in some vague sequence starting from the homepage).

Sharing user model data has the potential to enrich and improve adaptations in the applications sharing the data, as they can adapt based on information they would not normally be able to generate (Carmagnola, 2008). Shared user models may also reduce the chances of data being out of date or incorrect, as long as there is a mechanism for resolving differences between two user models.

There are two distinct architectures for sharing user models between applications. One approach is to have a single user model server which is accessible through a simple interface. The other is to have each application store a segment of the user model and to make the applications interoperable so they can exchange data (Carmagnola, 2008).

Some examples of the single, shared user model are the User Model Web Service (UMoWS) that uses SOAP over HTTP to communicate with adaptive applications (Bieliková and Kuruc, 2005); Personis provides an API using XML-RPC (Remote Procedure Calls) (Kay et al., 2002); and u2m.org which hosts the UserModelService which is accessed in a REST-like fashion (Chepegin et al., 2004; Heckmann et al., 2005).

Carmagnola discusses a number issues related to the interoperability of user models. One is the provenance of a value in a user model. Without any idea of how a value was generated or collected, an application cannot make an informed decision as to whether to trust the value. The solution proposed by Carmagnola and Cena (2006) is a method for interchanging rules (represented in "a semantic web rule language"), which describe the reasoning strategies used by the source application.

Another issue is that the different applications which have user models may use different languages and terminology to describe the concepts and attributes within their domains. Using RDFS as an intermediate language for sharing user model data, Carmagnola (2008) defines algorithms for calculating the relevance value of another user model's data in relation to the current user model query. This value is a combination of object similarity, calculated from the Dice coefficient of the micro-contexts (parent, children and sibling concepts in the domain ontology), and the property similarity, calculated using the Levenshtein distance between the property used in the shared user model and the property requested in the query.

An overlay user model contains two dimensions on which it represents the user: concepts and attributes. Concepts describe the various pieces of potential knowledge in the domain model. Attributes describe the metrics by which the adaptive application measures the user. To successfully share between adaptive applications, either both have to be using the same concepts and attributes, or there needs to be a way of converting between the concepts and attributes of each.

2.3.3.1 Sharing at the Concept Level

When sharing user models between applications, there is also the issue of each application having a different domain model, or at least different names for the concepts within the domain. Sosnovsky et al. (2007) presents a system for translating between overlay user models using an extension of the GLUE system (Doan et al., 2002), which uses machine learning techniques to generate mappings between ontologies. In fact, as the domain model is a hierarchy of concepts, it can be expressed as an ontology and one of the many ontology mapping techniques (Kalfoglou and Schorlemmer, 2003) can be used to allow both applications to understand each other's domain model.

2.3.3.2 Sharing at the Attribute Level

As well as different adaptive applications naming the concepts in their domain model differently, they may also name their attributes differently, or even measure and adapt on completely different properties.

One of the earliest attempts to resolve this was GUMO (Heckmann et al., 2005), which proposed the use of a common ontology for user modelling attributes allowing applications that use this ontology to easily share. GUMO uses the web ontology lanuage OWL

(McGuinness and van Harmelen, 2004) as its representation lanugage for user model attributes and how they relate to each other.

The next source of investigation into this area occured as part of the GRAPPLE EU project. This project refers to the common adaptive system architecture, AHAM (Kravcik et al., 2008; De Bra et al., 1999) (Adaptive Hypermedia Application Model) in a service-oriented manner, producing services for adaptation, user modelling, authoring and more, connected together by an event bus (*Grapple Event Bus*, *GEB*). A key component of the GRAPPLE achitecture is the Grapple User Modeling Framework (GUMF).

GUMF genericises all sources of data from adaptive applications as what it calls "dataspaces". Administrators of client applications (adaptive applications that are using GUMF as a user model data source) can use a web-based interface to subscribe to whichever registered *dataspaces* their instance of the application requires, as well as activate plugins and rules that modify how the *dataspaces* behave (Abel et al., 2009a; Leonardi et al., 2010). Data is modelled as Grapple statements, which take the subject-predicate-object form common in RDF. Querying of data through GUMF is provided by RESTful, SOAP and RSS interfaces (Abel et al., 2009a) though the papers and EU Framework deliverables (Abel et al., 2010a,b) that provide technical details of GUMF only demonstrate SOAP.

By interfacing the sources of primary data as generic *dataspaces*, it allows GUMF to also represent secondary data, such as external data sources, the results of queries, or the result of applying conversion rules (Leonardi et al., 2010), to also be interacted with as *dataspaces*, so client applications do not have to have custom code to interpret these.

Abel et al. (2009a) identifies that this approach "involves *conversion* between the different systems' user models", rather than using a shared language, allowing this shared user modelling approach to be used amongst any adaptive systems, rather than by a predefined set of systems.

Abel and colleagues demonstrated the application of using this by sharing social media profiles between different social adaptive websites. They used GUMF's ability to output user model data as "user pipes" in RSS, to mashup several sources of user model data using the tool Yahoo! Pipes (Abel et al., 2009a,b). They also evaluated the benefits of a similar service, named *Mypes*, which combines profile information from a variety of social networking services. They demonstrated that aggregation of social profile data can improve the range of facets available within a profile (14 of 17 attributes filled with useful data versus "7.6 for Facebook, 8.2 for LinkedIn and 3.3 for Flickr"), and produce a more complete profile than any individual social media service (Twitter profiles doubled in completeness from near 50% after using Mypes, and even the on average more than 80% complete LinkedIn profiles were improved by 7%).

The sharing of user model data by mapping at the attribute level has also been briefly demonstrated in educational systems. Oneto et al. (2009) discusses the benefits of connecting learning management systems (LMSs) to the Grapple Event Bus, and reports that GRAPPLE has integration with 5 existing LMSs. They explain that this allows LMSs to add adaptation by linking to resources in GALE (Grapple Adaptive Learning Environment), and that GALE can use data from the LMSs if they send them to GUMF via the GEB. Leonardi et al. (2010) describes extending GUMF with the Grapple Derivation Rule language (GDR), and then works through an example where an AHA!-based *Urban Geography* adaptive system recommends external links based on data queried from a Moodle-based *Geography* course.

2.3.4 Summary

In this review of the field of user modelling and their place within adaptive systems, the key elements of an adaptive system were identified as the domain model, the user model and an engine to provide adaptations from the normal material based on the state of the user model. The common types of user model were discussed and the overlay model seen as the most popular. The sharing of user models between adaptive systems was seen as a technique for increasing the consistency of data across applications and for bootstrapping an empty user model on a user's first use of an application. The two approaches to shared user modelling are to either have a single user model system that all applications connect to and store data in, or for each application to store their user model, but to expose it through a standard interface so that other applications can request data from it. A number of techniques for mapping between domain models were discussed.

2.4 Educational Games Review

To support the work carried out to investigate sharing data with digital games (Section 4.2) this review discusses games for education and adaptive games.

2.4.1 Game Design

To produce a good quality educational video game it is essential that game design theories are considered. As well as thousands of years of traditional games design, which has provided the world with entertaining and challenging activities such as chess, football, jigsaw puzzles and *Monopoly*, there has been a concerted effort, in recent years, to formalise the techniques used by the masters of modern video game design.

2.4.2 Integrating Educational Content into Games

The video game is a new medium, and as such, researchers are only just beginning to discover how to exploit it. Fisch (2005) believes inspiration can be taken from the television industry, which has had much longer to develop successful techniques for teaching through the medium of television.

Fisch makes a number of interesting observations regarding what is important in an educational video game. Before any game is made, it is important to make sure that a video game is the most appropriate medium for delivering the educational objective. The example given is that teaching the concept of balance is more appropriately done with real blocks than virtual ones, as in the real world there are forms of tactile feedback that are impossible to simulate in software.

Fisch also observes that the purpose of an educational video game is to teach the player something new, so it is likely that the learner will get something wrong on their first attempt. For this reason, it is important to provide appropriate hints and feedback to guide the learner toward the right answer in a way that will help them learn why that is the answer and how to remember it.

A very clear message from the article by Fisch is that it is important to integrate the educational material with the part of the game the learner actually plays. Attempts at making educational games in the past have often resulted in assessment by multiple choice quizzes or fill-the-blanks tests, with the player rewarded by entertaining animations. Fisch warns that although this form of educational game may motivate the player to repeatedly practice what they are learning, educational game designers must be wary of using these "seductive details" as not only does research show that learners remember the entertaining parts instead of the educational material (Fisch, 2005; Garner et al., 1992) but it may even cause the learner to resent the educational material for restricting their access to the "fun stuff" (Fisch, 2005).

To avoid these "seductive details", Fisch makes reference to an approach used in the educational television industry known as "content on the plotline". Essentially, this means integrating the educational content as part of the storyline, in a television programme, or as part of the way the learner plays, in a video game. Some types of video game contain a narrative element, in which case the educational content could be inserted into the game's storyline. However, if the storyline is not integrated with the way the game is played, then maybe an educational video would be the most appropriate medium.

2.4.3 Authoring Difficulties

When considering the practicalities of creating games as educational resources, it soon becomes clear that experienced and capable educators not necessarily be good game designers as well. They may have the potential to create good video games, but modern video games generally require large teams of designers, programmers and artists.

Although acceptable games can be made with small teams of one or two people, would they be up to the demanding standards of the students of the "PlayStation generation"? Add to this the fact that, like most professions, it takes some time to gain the experience to be good at designing and creating games, and it begins to seem unrealistic to expect educators to make good games.

The Federation of American Scientists funded the production of a game called *Immune Attack*, the creation process of which is described by Kelly et al. (2007). *Immune Attack* is a game set inside the human body, which teaches college and high school students about the components of the human immune system, and how they interact to protect the body.

The article by Kelly et al. draws attention to the scale of such a project and the number of different team members required to provide all the skills needed to make an educational game. Kelly et al. describes the team members as skilled in "biology, immunology, pedagogy, game design, and learning science". Not only is this quite an array of abilities to bring together in the first place, but managing people from such varied backgrounds is also a daunting prospect.

An example of how a team of this size and variety can cause conflicts is recounted by Kelly et al. (2007): "Members of the game design team were immediately intrigued by the fact that the immune system is based on rules that lend themselves [...] to battlefield metaphors. [...] However, the immunologists in the group vetoed the idea of using anything resembling a first-person-shooter game [...] which is simply not how the immune system works."

In contrast to the high budget project reported by Kelly et al., Dickey (2006) describes the challenges of building a 3D learning environment with limited resources. Some of the issues she raises include the cost of 3D modelling, the aggravation of converting between proprietary data formats and the inability of many educators to write the complex programming code necessary to make a game work.

The size of projects such as *Immune Attack*, and the difficulty of producing educational games with small teams such as Dickey's, raises the question of how best to proceed if creating an educational game. Either funding has to be sourced to sustain a large development team, the process for creating games has to be simplified or the scope of such games has to be reduced.

One way educational and training games are being funded is by partnerships between research groups and entertainment video games development companies. For example, UK developers, Blitz Game Studios Ltd. have a division called TruSim, which has the remit of creating 'serious' games for training and visualisation. However, this approach relies on the concept being of commercial value so the professional developers can recoup their cost through sales of the game.

For it to be reasonable for a course leader to prepare an educational game, alongside the lecture slides, handouts and additional resources, they must be easier and quicker to make. Systems such as Missionmaker (Buckingham and Burn, 2007) and Alice (Kelleher and Pausch, 2007) demonstrate the potential of simple authoring tools, while Frazer et al. (2007a,b) draw attention to the usefulness of educational minigames, which due to their reduced scope, would be quicker to create.

Another issue facing educators trying to integrate video games into their syllabuses is the amount of course time they could consume. Gee (2004) argues that video games are well suited to education because they make the student "master something that is long and challenging" (Gee, 2003) and promote deep learning (Biggs, 2003) of the topic at hand. A course leader may then struggle to allocate time for their students to play an educational game. Minigames may provide a solution to this, though as this conflicts with Gee's notion that games must be long to make the most of the learning affordances, it suggests there is a fine balance to be struck between the depth of an educational game and the amount of course time a student is allocated to playing it.

2.4.3.1 Non-Electronic Game Design

Games have existed long before computers, and so there are many forms of games that exist solely in a physical form, using custom playing equipment, cards or even the human participants on their own. Ellington, Addinall and Percival discuss their categorisation of game types and formats, a 4-phase process for designing new games and examples of how to apply this process to a variety of game types, in *A Handbook of Game Design* (Ellington et al., 1982).

Ellington et al. claim that games can be pure games, simulations or case studies, or any combination those three descriptions. A case study refers to the situation where the player is provided with a set of evidence and given the opportunity to play the role of someone investigating that evidence, in a similar way to how lawyers and doctors are trained in their respective disciplines. Pure games can involve physical skill, intellectual skill, chance, or a mix of all three.

Although noting that some games can be classified as electronic games, Ellington et al. state that "self-contained electronic games and TV games are best left to the experts". They do however offer advice on how to design a "computer" game. Non-electronic games are sub-categorised by Ellington et al. into psychomotor skill games and manual games.

Psychomotor games are further split into field games, involving any game where the player is physically in the area where the game is played, such as football, and table games, where the action takes place on a surface upon which the player manipulates some items, for example snooker and pool, as well as simulated field games like air hockey.

Similarly, manual games can be further grouped as simple manual games, card games, board games and device-based games. As might be expected, card games are those played with a standard or custom deck of cards, and board games which take place on a playing surface designed for that particular game. Device-based games have a key physical component custom designed for their purposes, such as a Rubik's Cube or roulette. Simple manual games usually do not require any equipment and rely only on the abilities of the players and the instructions, a good example of which is charades.

The 4-phase design process of Ellington et al. involves determining the need for such a game, developing the idea into a suitable concept, creating the materials required for playing the game and, optionally, publishing the game for others to use.

The first phase begins by asking whether the game is for fun, to make money or for educational/training purposes. Either way, the target population and design objectives must be decided, as they are used in the next phase. The process then checks whether a game is the best way of meeting the objectives and whether a full solution, or one that could be altered to suit the objectives, already exists.

In phase two, the design objectives and target population determine the content of the game. Then an appropriate format and structure are chosen, based on the design objectives. The format refers to one of the categories described earlier, and each format typically affords different structures. For example, field games have a number of structures including territorial possession games such as football, and games where opponents take turns to score points against each other such as cricket. Choosing the structure essentially means picking the rules and gameplay mechanisms that will be employed.

Creating, testing and improving the game is the third stage. The designer has to select what the packaged game will contain, which at the least will include the instructions and any special materials required to play. Then the designer can create a prototype of their design and test it within a sample of the target population. These steps can be repeated a number of times until the prototype suitably meets the original design objectives. At this stage, the fourth phase, the game can be published, either by the designer, if it is simple enough to reproduce, or by an external publishing house, if they can forecast making enough profit from sales of the game.

2.4.3.2 Video Game Design

Rollings and Morris (2003) describe their vision for the video game development lifecycle, based on their experiences working in the industry. In their book, they discuss the process of developing an idea into a possible game and set out the documents the game designer must produce to explain their idea to others.

According to Rollings and Morris, an idea for a game is a combination of sources of inspiration that, when synthesised into one whole idea, resonate to be better than just the combination of all the individual ideas. In more concrete terms, an idea should have a style, a plot, a character, a setting and a theme. The plot may be set in action by some form of introduction to the backstory of the characters involved, but to make an interesting game, it has to be the player that acts out the plot, otherwise it's just a fixed story, with a game interrupting the flow. Once the game designer has thought through these elements of the game idea, they are written down in the game treatment document, which can be used to pitch the idea to potential publishers and is used to examine whether the game is feasible commercially and technologically.

Rollings and Morris confess that they think it is more important to make a fun piece of interactive entertainment, than it is for this to necessarily be a game, though do not suggest any direct way of determining how fun something is. However, they do advise that if it is to be a game, it is likely to require the player meet a number of goals which could include gathering tokens or seizing territory, winning a race of some sort, overcoming obstacles, exploring and solving problems, or eradicating opponents.

By this stage, Rollings and Morris expect the game specification document to be produced. It is composed of the game features, descriptions of the gameplay and the interface, a list of the rules and hints on level design. The features, gameplay and rules create a unique harmony. The rules define what is and is not possible in the proposed game world. From these rules, it is hoped that the features of the game will emerge, although these are often subject to fine tuning. The features are supposed to be the differentiating factor from other games and are high level descriptions of what the player can do in the game. Which features are used and the way they are used by the player to attempt to achieve their goals within the game, are what create the gameplay.

The book goes into detail about what gameplay is and how Rollings and Morris think it can be achieved. Drawing on economic game theory, gameplay is described as "the strategies required to reach specific end points". As a way to keep the games entertaining, they discuss gameplay as "a series of interesting choices", based on a quote from Sid Meier (Rollings and Morris, 2003, p61). The best choices are either ones that "should sometimes be taken, [...] depending on other factors" or "whose timing is critical and depends on the context", and as such, should have both positive or negative repercussions, depending on which choice is made.

It is important to avoid dominant or dominated strategies: in circumstances where it is always or never beneficial to select a particular option, then the player is really left with no choice at all. Using intransitive relationships, such as in the game *Rock-Paper-Scissors* is a simple way to avoid dominance, though in some senses is only the beginning of making interesting gameplay. Balancing a game can involve the use of economic game theory by creating a payoff matrix to compare the costs and benefits of each choice in comparison to each choice by the opponent.

Other factors that can make choices more interesting include the requirement to make a choice now that will open up more choices at some point in the future, and the impermanence of some choices, due to the fact that the result of the choice may be destroyed, stolen or may only last for a fixed amount of time. Having a versatile option provides the player with the chance to protect themselves in an unpredictable environment.

Rollings and Morris also discuss the importance of the look and feel of the game for making the game immersive. Ambience should sell and support the backstory, through the sounds and visuals used, as well as the way the player's character interacts with the environment. It is essential that the way the player interfaces with the game is as intuitive as possible, so that the user does not even notice they are using it. They also advocate the use of storytelling mechanisms such as foreshadowing (providing the player of an indication of what might happen later), personalisation, regular plot points, suspense and resolution to enhance the player's experience in the game.

2.4.4 Summary

Games are rich and complex systems that require lots of different types of interaction from their players. The formalisation of how to design games is ongoing, and the emphasis of literature about game design is on lifecycles that improve consistency of the design process, but the structure of what to do in each step of those lifecycles is actively being discussed and encoded.

It is a significant open challenge to find appropriate ways to integrate cognitive learning into games. Whilst many argue that playing games is inherently a learning experience, it is not trivial to align the learning of a game's mechanics and higher level strategies, with the knowledge, understanding and analysis that embodies higher level learning outcomes.

2.5 Conclusions

Much of contemporary learning theory, including *constructivism* and *zone of proximal development*, suggests that new knowledge and understanding is constructed in the context of what the learner already knows and their level of development and ability. If the next task is pitched just above their current level, they will be more likely to be engaged with the task and will learn in an optimal manner. If it is too far above their current level, they will become anxious and frustrated; if it is below, or even at, their current level they will soon get bored.

When the computer science technique *adaptivity* is applied in educational technology, it is usually to provide content and activities at the appropriate level for the student currently using the system. A core component of adaptive architectures is the *user model*, which stores data about the interactions the user has had with the adaptive system and what the adaptive system infers about the user from those interactions; typically this is knowledge, interest or some sort of performance score, but it could be any other metric the application is able to measure through the user's input.

If a learner moves between adaptive applications, typically their user model is lost, and at the start of each move the adaptive application will have no information about the learner. This *cold-start problem* results in the adaptive application targeting learning at the wrong level for the current learner, at least until it has enough interactions to determine the correct level. It also results in attempting to construct new learning based on a very narrow view of what the learner already knows, which risks not only repeating existing learning but also asking the student to learn something they do not have the prerequisite knowledge for. The literature shows that it is an active research problem to enable sharing user models between adaptive applications.

Whilst work on adaptive systems is common in more traditional educational technology, such as hypertexts and computer-aided assessments, the idea has only recently been applied to educational computer games. The user model data generated by computer games, and how to share that with other adaptive applications, has yet to be fully researched by this community.

This literature review has demonstrated that there is scope to research the sharing of user models with a variety of different applications and that there are challenges caused by newer categories of adaptive educational applications. It is this realisation that motivates the research questions RQ1 and RQ2. The next chapter will survey the literature to determine a suitable methodology for conducting research to answer these questions.

Chapter 3

Methodology

A methodology was devised in line with those observed in the user modelling, adaptation and personalisation research community, to answer RQ1 and RQ2 in an appropriate manner, To ensure a suitable methodology was chosen, a meta-review was conducted to investigate the range of evaluations used for adaptive educational systems, the results of which are presented in this chapter.

3.1 Research Questions

The research questions, as defined in Chapter 1, that are guiding this chapter are as follows:

RQ3a: What is the range of evaluation techniques used within the adaptive educational systems community?

RQ3b: Are particular types of evaluation prone to particular influences or biases?

3.2 Meta-Review Methodology

To perform the meta-review, a corpus of academic literature about Adaptive Educational Systems (AES) was collected and the evaluations reported were then categorised to produce the data set with which to work. This data was then used to produce summary statistics and analysed in order to answer the research questions.

3.2.1 Literature Collection

The literature used for the review was collected systematically. This meta-review targets peer-reviewed academic literature, so the articles were sourced from a journal, conference and workshop that represent the research area studied in this thesis.

It was decided to bound the parameters of the review to make the process repeatable and make the time required to complete it predictable. This is in contrast to approaches that begin with a search term and analyse all results and sometimes even recursively search the references in each found result. The parameters used to bound the search were source and time period. This review collected papers during the 5 year period 2011-2015 from the UMAP conference, PALE workshop and AIED journal. There were 331 papers in these publications during this time period.

Clearly not every paper from these sources is about adaptive educational systems, so a filter is applied when reading the abstracts to select those papers which describe the required type of systems. Three questions were asked of every paper:

1. *"Is it a system?"* - theoretical papers and literature reviews were excluded;
2. *"Is it educational?"* - many systems are described in these sources, and this review is only interested in those that attempt to aid learning;
3. *"Is it adaptive?"* - the system described should automatically adjust its behaviour based on a model of the user.

3.2.2 What does "adaptive" mean?

It is perhaps worth refreshing what is considered adaptive, given it is important for selecting the correct papers for this meta-review. Adaptive (Educational) Systems has a close relationship and some overlap with research areas such as Intelligent Tutoring Systems (ITS), recommender systems and Educational Data Mining (EDM), but it is distinct from these.

Adaptive systems tend to use all the information it has about the user (ResearchGate) to provide the user with a presentation of the content or navigation options that are custom to them. This contrasts with ITS which often just react to the last response from the user, to provide appropriate feedback or problem solving support, and continue in a predefined (although possibly branching) sequence. Adaptive systems do not necessarily use any artificial intelligence (AI) techniques in providing adaptations, whereas ITS, EDM and, to an extent, Recommender Systems, are grounded in AI.

There are some indicators to help decide if a system is adaptive, for the purposes of this meta-review. If the paper reports that the system adapts, there is a good chance it is adaptive. A key additional clue is that it has the architecture of an adaptive system: a user model, crucially, a domain/content model and an adaptation engine. Finally, what is the system is producing as output? If the adaptations are changing the presentation and navigation of the content, based on the whole user model, it is likely adaptive. If it is responding with feedback to steps of a problem, it probably is not adaptive.

3.2.3 Initial Analysis of Papers

The filtering resulted in a set of 55 papers that were identified as adaptive educational systems. Each of these resulting papers was read in detail and data gathered on various relevant facets of each paper. The following questions are answered for each paper, recorded in tabular form for further analysis:

1. *"What is the full reference for this paper?"*
2. *"What is the name of the system?"*
3. *"What sort of user model does the system use for adaptation?"*
4. *"How is the system evaluated?"*
5. *"What were the results of the evaluation?"*
6. *"What are they testing?"*

3.2.4 Categorisation of Papers

Following the high-level analysis of the papers, the results were used to categorise papers and perform summary statistics on those categories.

An initial subset of papers were first grouped by similar evaluation methodologies, based on the answer to "How is the system evaluated?". However, because each evaluation methodology is subtly different, a more general way of defining each methodology was required.

Through analysis of the methodologies discovered, it was determined that the source of data gathered and analysed in these evaluations is varied: the data can come directly from the participant, from an expert independent of the participant, or from some computation either in the system or outside of the system.

As more papers were gathered, they were provisionally added to a group based on the data source or sources used in their evaluation. This refined the definition of the group, until the categories described below were finalised.

3.2.4.1 Student

The data used to evaluate the performance of the educational system comes from the user (student) themselves. This is typically in the form of a survey filled in by the student at some point in the experiment.

Rosa et al. (2015) and Tegos et al. (2014) both use questionnaires based on a variation of the Technology Acceptance Model (TAM) to measure user friendliness and perceived utility, while Shi et al. (2013) utilised the SUS usability questionnaire.

3.2.4.2 Logs

Some researchers perform statistical analysis on the system usage logs, that record the user's behaviour, to answer certain questions about the performance of the system.

For example, Muir and Conati (2012) and Conati et al. (2013) used logging of eye-tracking during experiment, so measurements like fixation time could be compared to other measures, such as which type of hint they were looking at.

3.2.4.3 Expert

This category covers any experiment where a human expert has had input into judging the performance of the user. A large number of these are experiments where the user (student) takes a test of some form (often before and after use of the system); as the test and the marking scheme have been written by a subject matter expert, performance is deemed to be judged by the expert. In some cases the system's output for certain users might be reviewed by an expert for effectiveness.

Evaluations with data sourced from an expert are generally tests, like Arroyo et al.'s (2011) mock standardized test administered before and after their intervention. Alternatively, they may be based on testing that is already taking place, such as Greer et al.'s (2015) comparison of end-of-year grades in the year their system was used, against the two previous years' grades.

3.2.4.4 Baseline

A number of systems in this review judge the performance of the algorithm underlying the system against baseline results. The baseline can either come from previous evaluations in the literature, or may be a comparison against chance (i.e. if the system randomly selected an available option).

Source/Year	2011	2012	2013	2014	2015	Totals
AIED	1 / 13	4 / 9	2 / 7	10 / 15	6 / 19	23 / 63
PALE	5 / 12	2 / 9	0 / 4	4 / 8	3 / 6	14 / 39
UMAP	3 / 51	6 / 43	3 / 51	2 / 47	4 / 37	18 / 229
Totals	9 / 76	12 / 61	5 / 62	16 / 70	13 / 62	55 / 331

Table 3.1: Sources of adaptive educational systems papers.

Baselines can either come from other literature, as in Gong et al. (2012) where results are compared to previous results by the same author, or from a selected simple case such as Kardan's (2012) classifier algorithm which was evaluated against a baseline of always selecting the most likely class.

3.2.4.5 System

Some experiments will focus on evaluating which of a number of options are best for their particular algorithm. Often this is where parameters are learned or tuned for systems that base their adaptations on machine learning or artificial intelligence algorithms. These evaluations are asking which set of parameters produce best performance, rather than how well the system performs its intended purpose. For example, Baker and Clarke-Midura's (2013) evaluation reports on the model coefficients trained on real student data and the Leave One Out Cross-Validation correlation, which they report is comparable in performance to Bayesian Knowledge Tree models (baseline).

3.2.4.6 None

If no evaluation at all is presented in the paper, this is also recorded.

3.3 Results

3.3.1 Sources of Papers

Table 3.1 records the ratio of how many papers met the criteria of being an adaptive educational system in each of the selected publications, over the period 2011-2015.

3.3.2 Sources of evaluation data

The primary quantitative output of this review is a measure of how many of the reviewed papers fit into the categories defined above. In Table 3.2

Category	Student	Logs	Expert	Baseline	System	None
AIED	12	9	13	1	3	1
PALE	2	2	2	2	1	6
UMAP	4	1	7	6	5	3
Total	18	12	22	9	9	10

Table 3.2: Categories by publication.

Number of Categories	0	1	2	3	4	5
Count	10	22	18	5	0	0

Table 3.3: Count of how many papers fit into multiple categories.

3.3.2.1 By publication

Number of papers that fit each category, n=55 (note some papers fall into multiple categories)

3.3.3 Multiple Categories

Some papers use multiple data sources for their evaluations, Table 3.3 summarises the count of papers that use multiple categories. The count of how many papers fit into each group of categories is presented in Figure 3.1.

3.4 Analysis

Table 3.2 summarises the number of systems that derive their evaluation data from each category described in the methodology. This demonstrates that the most common form of evaluation (22) involves gathering data from some expert-judged source, such as a pre- and post-test. This is the case overall, and in the last 5 years of UMAP proceedings and AIED journals. It is promising that so many evaluations use objective measures of performance, either of the students using the system or of the system itself.

The second most common source of evaluation data is from students (18), either in the form of subjective surveys or self-assessment of performance. In one sense, this is encouraging in terms of the methodology chosen for the user trial described in Chapter 6. However, given the concerns raised in the discussion in the Categorise section about participant-derived performance evaluations, perhaps this approach should not be as widespread as it is.

Category	References
Student	(Redondo-Hernández and Pérez-Marín, 2011), (Forbes-Riley and Litman, 2012), (Nussbaumer et al., 2012), (Shi et al., 2013), (Arroyo et al., 2014), (Bixler et al., 2014), (Khachatryan et al., 2014), (Pareto, 2014), (Tegos et al., 2014), (Walker et al., 2014), (Alvarez et al., 2015), (Brusilovsky et al., 2015), (Gilbert et al., 2015), (Mohammed and Mohan, 2015), (Rosa et al., 2015), (Vail et al., 2015), (Westerfield et al., 2015)
Logs	(Käser et al., 2012), (Muir and Conati, 2012), (Stamper et al., 2012), (Conati et al., 2013), (Käser et al., 2013), (Shi et al., 2013), (Adamson et al., 2014), (Heffernan and Lindquist Heffernan, 2014), (Pareto, 2014), (Alvarez et al., 2015), (Brusilovsky et al., 2015), (Chaouachi et al., 2015), (Gilbert et al., 2015), (Nava Tintarev, 2015)
Expert	(Arroyo et al., 2011), (Cocea and Magoulas, 2011), (Green et al., 2011), (Dodigovic, 2012), (Forbes-Riley and Litman, 2012), (Stamper et al., 2012), (Baker and Clarke-Midura, 2013), (Conati et al., 2013), (Käser et al., 2013), (Adamson et al., 2014), (Arroyo et al., 2014), (Shareghi Najar et al., 2014), (Pareto, 2014), (Walker et al., 2014), (Brusilovsky et al., 2015), (Chaouachi et al., 2015), (Gilbert et al., 2015), (Greer et al., 2015), (Silva et al., 2015), (Vail et al., 2015), (Westerfield et al., 2015)
Baseline	(Thai-Nghe et al., 2011), (Gong et al., 2012), (Kardan, 2012), (Baker and Clarke-Midura, 2013), (Sabourin et al., 2013), (Baker et al., 2014), (Bixler et al., 2014), (Nye et al., 2014), (Corrigan et al., 2015)
System	(Champaign et al., 2011), (Cocea and Magoulas, 2011), (Green et al., 2011), (Thai-Nghe et al., 2011), (Forbes-Riley and Litman, 2012), (Käser et al., 2012), (Baker and Clarke-Midura, 2013), (Baker et al., 2014), (Corrigan et al., 2015), (Pelánek and Jarušek, 2015)
None	(Frias-Martinez and Virseda, 2011), (Roldán et al., 2011), (Leite et al., 2011), (Berger, 2012), (Kardan, 2012), (Zapata-Rivera, 2012), (Arevalillo-Herráez et al., 2014), (Henning et al., 2014), (Lenat and Durlach, 2014)

Table 3.4: Table of which papers fit into which categories.

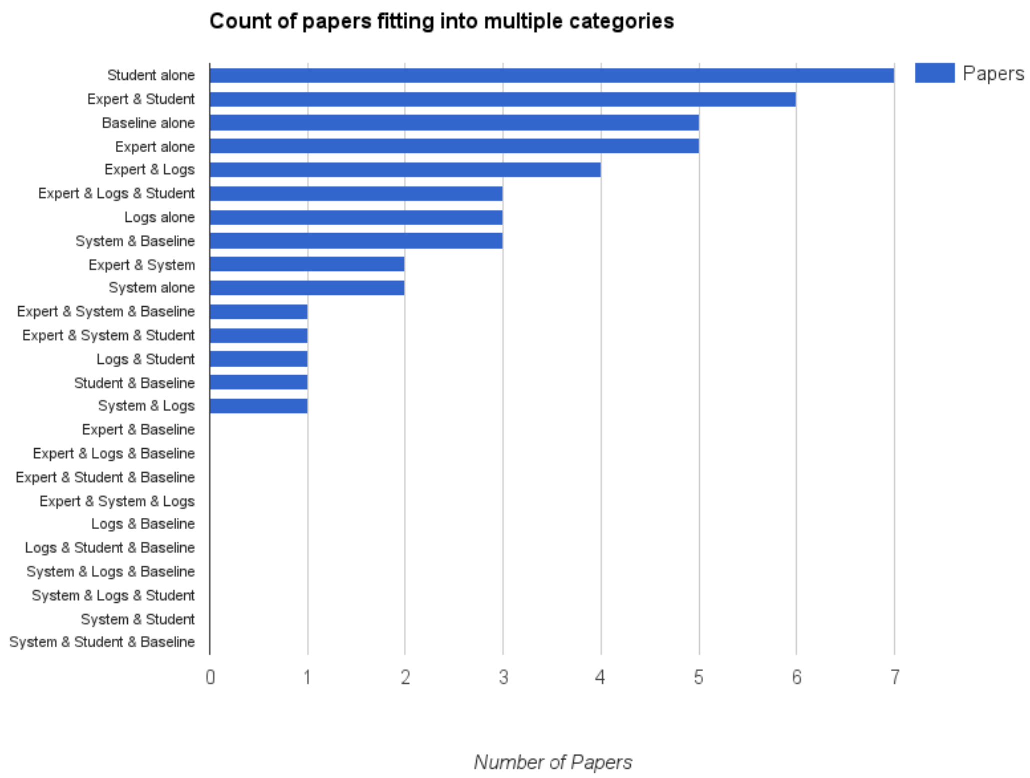


Figure 3.1: Bar chart of count of combinations of categories

Logs (12) from system usage were the third most common source of data, which again supports the use of log data in the experiment in Chapter 6.

The final two categories, Baseline (9) and System (9) had an equal occurrence in the literature and cover a similar mode of evaluation, where the performance of the system or some sub-part of it is evaluated computationally and the result is either compared to existing state of the art baselines or the best of many potential parameters is reported (although there was only an overlap of 2 systems that used both).

On a par with Baseline and System (at least in the scale of this sample) is the number of papers that have no evaluation at all. Given that most of these are from the workshop, it matches with the expected level of the various venues.

3.4.1 Multiple Categories

Of the papers that performed an evaluation, 40% used a single category of data source to evaluate, another 33% used two, and 9% used three. No evaluations used data from

4 or 5 categories in their reported evaluation.

It is interesting to note that the most common pairing of evaluation categories is Expert & Student (13%), though this is still exceeded by the number of evaluations in the review that only use Student data sources (16%).

Although performing evaluations with a pair of data sources is fairly common, there are some pairings that are absent: Expert & Baseline, Student & System and Logs & Baseline. There are very few publications that use 3 sources as part of their evaluation, but those that do always include Expert.

3.4.2 Influence and Bias

One thing to consider about these categories of evaluation is the sources of influence and bias in measurement of performance of the system. Evaluations that are reporting the optimal parameters for their algorithm, or those that are comparing performance to some baseline are relatively free of human influence, except perhaps errors in the code or statistical analysis.

Usage logs could be expected to be independent of bias, as it is representative of what happened in the system, rather than a proxy or for it. However, there are circumstances where a logged piece of information is used as an indicator for some other measure of performance. For example, Gilbert et al. (2015) uses the log of amount of time an author took to construct problems in their system as a proxy for how easy it is, despite there being a number of factors that might affect this.

Evaluations that fall into the Student category are most susceptible to issues with accuracy. Where data is collected from the participant themselves, there is the risk that the participant does not understand the question being asked or the scale on which they are answering or measuring, or may even fall foul of the Hawthorne effect (where participants improve their behaviour when they are aware they are being observed). There is also the potential for inconsistency, between participants, in interpreting these. Where an identical question is asked of all participants (such as in a usability survey) it could be argued that these problems are eased by the statistical analysis of the results over a large enough sample; it is the average response value that is generally reported, after all. However, in situations where the participant is asked to judge their own ability or performance during the experiment, there is scope for inconsistencies in accuracy to arise.

The Expert category tends to involve human evaluation of performance, that is independent of the participant. There is still potential for influence, as there are humans involved in the process, but typically there will be increased consistency between judgements as a small set of experts is used or a criteria for judgement is generated by an expert to be applied by others (for example, a marking scheme for a pre-post test).

3.5 Proposed Methodology

Having explored the breadth of possible evaluation strategies employed by adaptive educational systems researchers, it is possible to devise an appropriate methodology to explore research questions RQ1 and RQ2.

Firstly, a system will be designed and built to enable the mapping and conversion of user model data, between adaptive applications, at the attribute level to better enable the exploration of the best way of doing this. The justification for building a system, rather than just designing a procedure or framework for transferring data between applications, is that the cold-start problem affects existing adaptive applications and functioning impementations can directly benefit the community and provide a better experience for the user and administrators of such applications. The design and implementation for this system is detailed in Chapter 4.

Secondly, the functionality of the system will be evaluated using a simulation. This simulation will test the system using a large number of programmatically generated virtual users. Whilst the expected results from the mapping provided by the system could be proven mathematically, the benefits of simulating against a working version are to provide additional testing against bugs and to gain some understanding of the performance of the system when in use. The simulation and its results are described in Chapter 5.

Thirdly, a user trial will be performed to determine whether the use of the system proposed and built in the first stage has a significant impact on the experience of learners, when it is used to connect together the user models of several adaptive educational systems. This will ask users to report on their experience of using the system; whilst this is a proxy for learning, it is a good mechanism for determining whether the system is effective enough for users to notice and is a popular method of evaluation in the research community, as seen in this chapter. The complete methodology and the results of the user trial can be seen in Chapter 6.

3.6 Conclusions

This chapter sought to develop a methodology to answer the research questions set out in Chapter 1, by considering the common types of evaluations performed in the research area of Adaptive Educational Systems. This was done by performing a meta-review of the literature within three key publications over a 5 year period, and developing a categorisation system for the evaluations present within that literature.

The review discovered that while there is a more prevalent category of evaluations within the community (those that use an Expert to judge performance), evaluations that use the Student's judgement (either in surveys or self-assessment) are the second most common, which supports its use in the proposed methodology.

As a result of this review, a methodology was defined for the rest of this thesis, summarised as the creation of a system to perform mapping between attributes of different adaptive applications, the simulation of this system to ensure it performs as expected and a user trial to determine whether it has a significant impact on the experience of the learners using it.

Chapter 4

The Design of the System: IDIUMS

IDIUMS, the Interactionally-Diverse Intermediary User Modelling System, is the core technical contribution of this thesis. In this chapter, the problem that drives the need for a system like IDIUMS is identified and discussed, the design for IDIUMS' solution to this problem is documented and details of the reference implementation recorded.

4.1 Initial Prototype

As discussed in Chapter 2, adaptive systems are prone to the *cold-start problem*, and to address this there are many proposals to share user model data between adaptive systems. However, with the increasing variety of application types forming these adaptive systems, there is a mismatch in the attributes used to measure the user, particularly in systems using an overlay user model.

It is important to be clear about what data is shared between adaptive applications. In the overlay user model, for every concept in the domain model a number of attributes can be defined. It is to these attributes that values are assigned.

Researchers have already demonstrated methods of translating or mapping between synonymous concepts within domain models of similar applications. For example, if the domain is music, when defining genres of music one domain model main refer to concepts such as "rock and roll", "independent" and "classical", while the other may name them "rock", "indie" and "classic". There are already ways of creating or automatically generating mappings between domains such as these.

The hypothesis here is that even if the domain model concepts are the same in two adaptive applications, or they are subtly different but being mapped between, there are other challenges of sharing user model data between applications. The following section explores the challenge provided by applications which use very different metrics to measure the user.

4.2 Exploring the Problem

In order to explore the challenges involved in sharing user models between different types of adaptive applications, an initial prototype was implemented.

4.2.1 Aim

The aim of building this initial prototype of two applications, between which a user model could be shared, was to discover what challenges are involved. It was intended that this would expose the technical issues involved in interfacing between two adaptive applications, as well as the possible problems related to translating the data generated by each application.

Although some researchers have studied methods of sharing user models between a number of adaptive hypermedia systems (Kuruc, 2005; Bieliková and Kuruc, 2005), less effort has been focused on how to share user models between different types of application, particularly in the educational domain, and what issues arise from attempting to do so. With this in mind an investigation was conducted to build two types of adaptive educational application (an adaptive hypertext and an adaptive game) and investigate how to share the data produced by both applications.

4.2.2 Method

There were five key stages to constructing this initial prototype: designing the domain model, creating the adaptive hypertext, creating the adaptive educational game, designing the user model data interchange and creating the user model sharing interface.

As both applications are for educational purposes, with adaptations working to personalise the educational experience for the user, to share user model data requires the same (or at least a very similar) domain. Although it could be argued that user data could be shared between different domains (for example, knowledge of neuroscience may help the understanding of neural networks in artificial intelligence), it is a hard problem with sparse examples of such overlaps. Methods of mapping between similar domains have already been studied. Instead, it was decided to design one domain model that could be fixed for both applications, so the other interactions involved in sharing a user model could be observed. This is described in Section 4.2.3.1.

The domain model were then imported into the adaptive hypertext engine and pre-requisite rules defined. In addition, the textual content explaining and teaching the concepts were written.

The game was designed as a way to enable a learner to experiment with Java code in a visual, interactive and challenging environment, with a focus on using loops. Level scenarios were set up to challenge the player's knowledge of various concepts in the domain model. The user model records the player's performance in attempting the levels and the game adapts by presenting different levels depending on the user model.

To share user model data, the data that each application produces must be designed, by analysing the capabilities of each application and the interactions available to the user.

Finally, to actually share the data between the applications, a software interface is required to transfer the values between the adaptive hypermedia engine and the game engine.

4.2.3 Initial Prototype of Adaptive Applications

To explore how an adaptive game and adaptive hypermedia can benefit from sharing a user model requires examples of these applications to be built. The game design, adaptive hypermedia tutorial and the common educational content used in this initial prototype are described below.

4.2.3.1 Educational Content

To effectively test a shared user model, the two educational applications need to be designed to teach similar concepts, otherwise there will be little data to share.

With this in mind a common educational objective was devised so both the game and the adaptive hypermedia application could be written to teach it.

Based on teaching experience on undergraduate programming courses and the potential availability of suitable test subjects, an important but difficult to master programming concept was selected: the loop.

To establish the important concepts required for teaching looping, two Java programming books were compared to determine the set of possible concepts. Overlapping concepts were collated into one common concept. These books were Java Gently (Bishop, 2000) and The Object of Java (BlueJ Edition) (Riley, 2002). The result of this comparison can be seen in Appendix C.

The overall concept of a programming loop was broken down into concepts that are either key components of, or contribute to the understanding of the main idea. These, in turn, were broken down into sub-concepts where appropriate. The breakdown and hierarchy of the concepts can be seen in Appendix B.

These concepts were then mapped to pages of text to be presented in a tutorial, as well as scenarios in the game.

The domain model was generated from two Java books, the result of which can be seen in Appendix B. This domain model was then put into an appropriate format for each adaptive application to interpret.

4.2.3.2 Adaptive Tutorial

The adaptive tutorial has been created in GALE De Bra et al. (2003) using the domain model described in the previous section. GALE was selected as it is a mature, stable adaptive hypertext engine, having evolved from four versions of AHA before it. It was also in active development, as part of the EU-funded project "GRAPPLE", during the course of this research.

The AHA Graph Author tool was used to create the domain concepts within AHA! (see Figure 4.1) and to define the relationships between them, then it was ported into GALE [1]. For each concept in the domain model, a page of tutorial XHTML text was written to explain it, so the adaptive hypertext is functioning.

4.2.3.3 Adaptive Game

The inspiration for this game comes from a learning object presented by (Boyle, 2006). As part of a learning object about programming *while loops* in Java, the learner is presented with some Java code which purports to control a submarine, and an image of the submarine floating on the surface of the water. Having read the code, the learner can then press a button to run the code and see how it moves the submarine in an animation. A screenshot of part of the learning object can be seen in Figure 4.3.

The adaptive game presented here is a bespoke game, inspired by Boyle's learning object, which improves upon the learning exercise by allowing the learner to write the code themselves and see the resultant animation. The challenge of the game comes from a variety of different levels which have to be traversed by the player's avatar, each of which are designed to encourage the understanding and use of a particular looping concept.

By breaking the game down into levels which represent certain concepts, the adaptive engine within the game can query the user model to decide which level is most appropriate to present next.

[1] This was due to this work coinciding the timing of the releases of AHA!4 and GALE, not due to any procedural advantage.

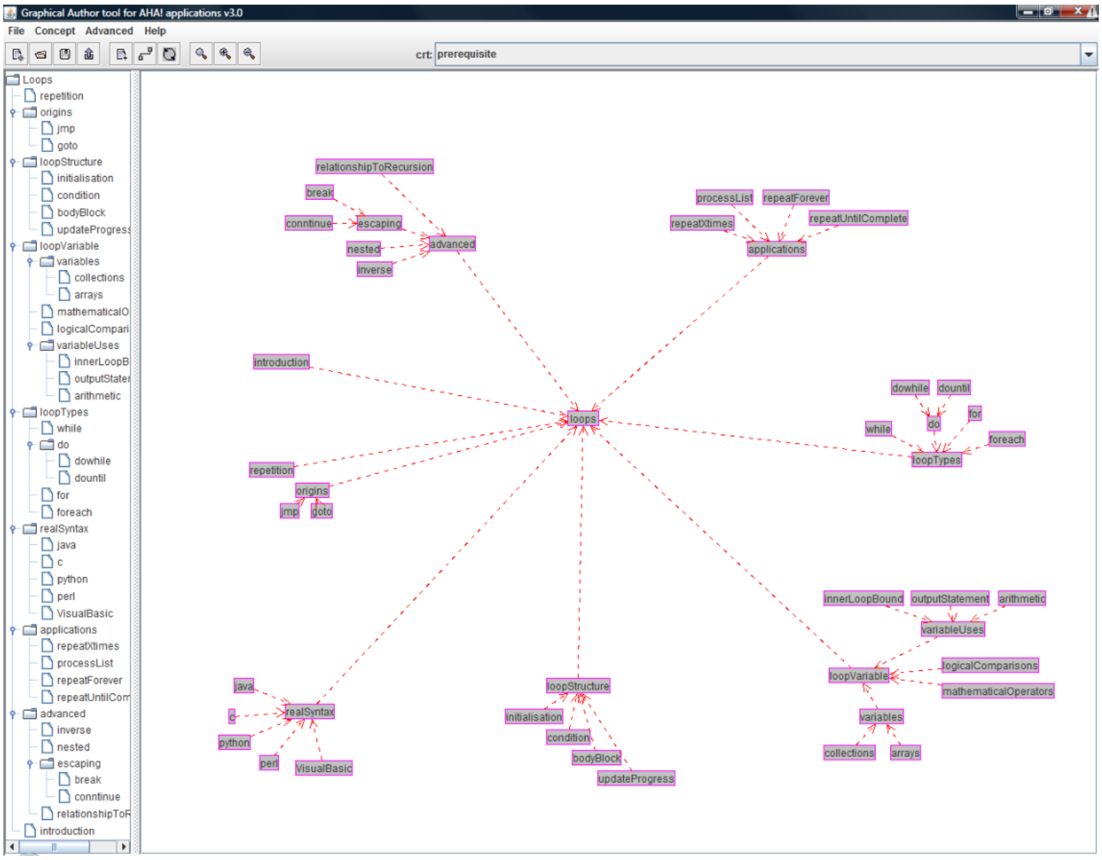


Figure 4.1: A screenshot of the domain model being created in the AHA! Graph Author tool.

Game Flow The expected order of events in the game are described below and are depicted in Figure 4.4.

At the start of the game, the player logs in to identify themselves and prepare the user model. The player is then introduced to the game, which sets the theme and explains the rules and objectives. Once an initial level is selected (either the lowest in difficulty, or a more appropriate one based on the player's user model), it is displayed to the player through the game interface (described below).

Once at the game interface, the player can type in imperative Java code (as what they write will be injected into a pre-defined method and class). The player can then click a button marked 'Execute' to interpret the code and execute it in the animation window.

There are two outcomes that can occur from running the code: either the player's avatar (a submarine, by default) successfully navigates the level and exits through the portal; or the avatar collides with an obstacle (such as a mine). Success takes the player to another level, whereas a collision resets the current level and allows the player to modify their code to try again. In both cases, the user model is updated to reflect the player's understanding of the programming loop concept underlying the current level.

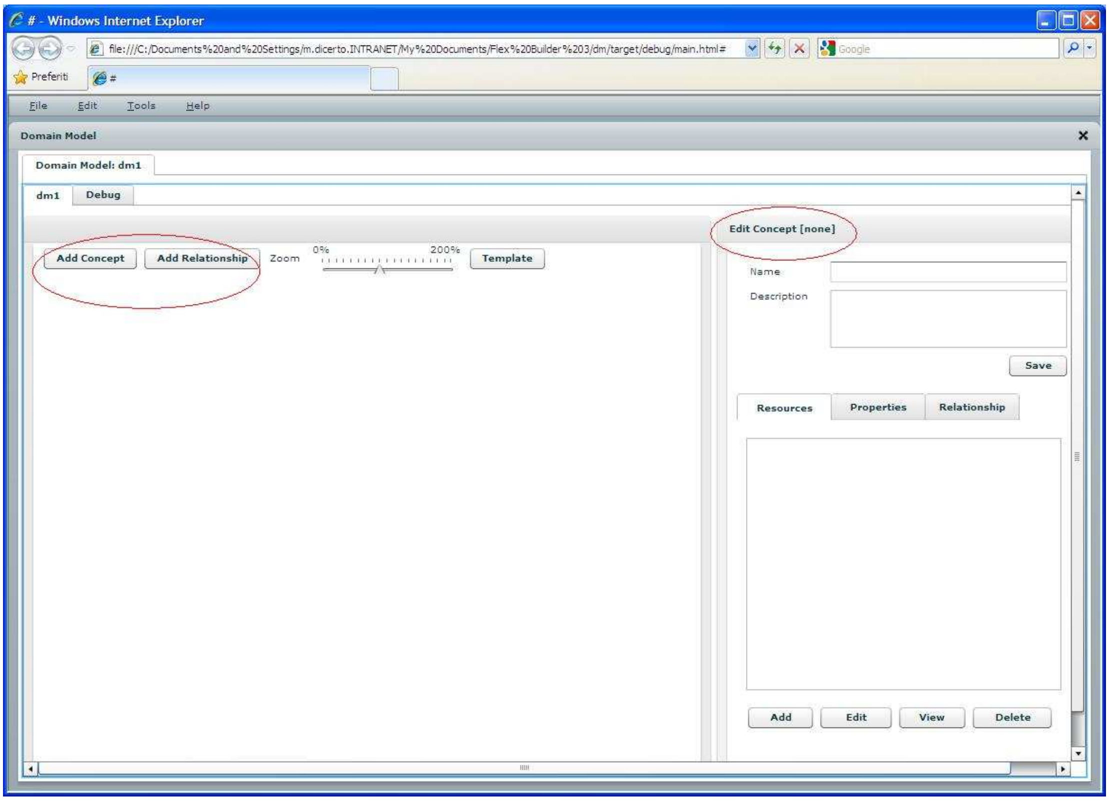


Figure 4.2: A screenshot of the domain modelling editor for GALE (Dicerto and Mazzetti, 2010)

Game Interface and Graphics The user interface for this game is split into two parts. One part contains a text box to enter and edit the Java code. The second part is a two-dimensional (2D) representation of the game world and the current level. In this representation are a number of obstacles, a start and end point and the player's avatar. The interface contains a button which triggers compilation of the code. An example of the layout of the interface can be seen in Figure 4.5.

Compilation and Execution Interpreting the player's Java code on the fly makes use of the `JavaCompiler` interface, available since Java 6. To avoid having to read and write to intermediate files, the `JavaFileManager` and `JavaStandardFile` classes were extended to support files that exist solely in memory (not on disk).

The player's code should contain calls to methods such as `.moveDown(5)` or `.moveRight(x)` on the avatar object, which add those actions to a queue. A timer in the game then animates these actions in sequence, detecting any collisions that might occur.

Implementation The adaptive game was implemented as a Java applet and animates an avatar based on the code written by the user. A number of levels have been created

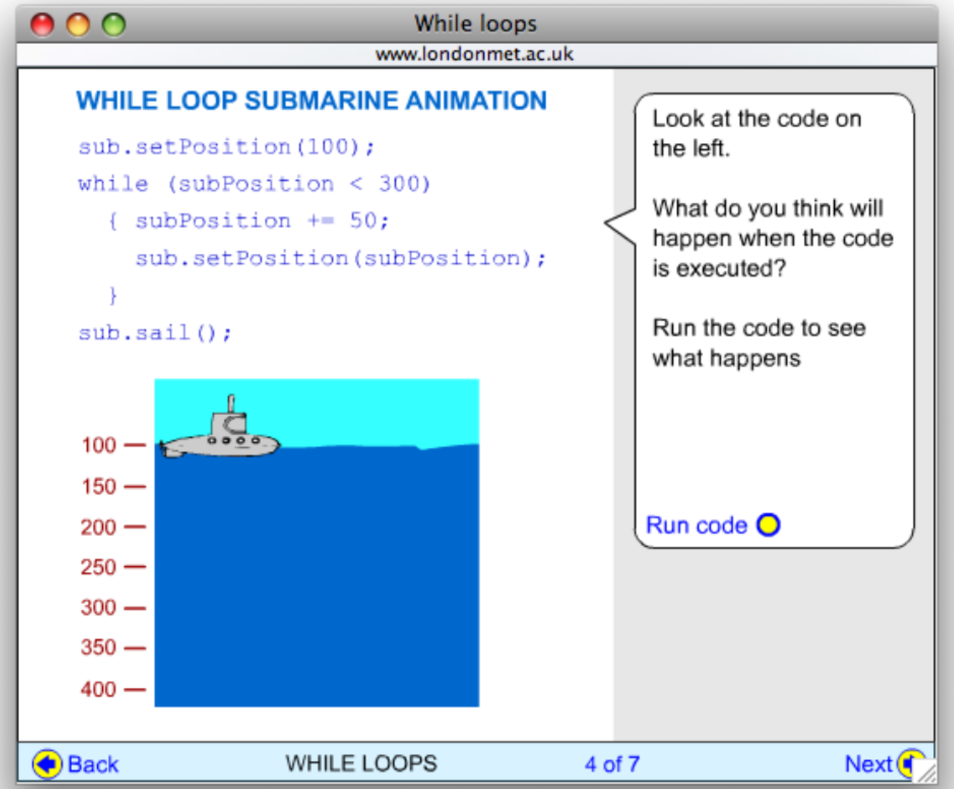


Figure 4.3: A screenshot of a stage within the *While loops* learning object by Boyle (2006).

to test the concepts in the domain model. A screenshot of the game interface, displaying an example level, can be seen in Figure 4.5.

4.2.4 Discussion

The purpose of this initial prototype was to investigate the interchange of data between an adaptive game and an adaptive hypertext. It was therefore necessary to consider what data is available from each application and what approaches could be taken for exchanging the data.

Unfortunately, the documentation for GALE's event bus was insufficient to enable the adaptive game to be programmed to communicate with GALE's user model, so it was not possible to make functional the direct software link between the adaptive hypertext and the game. However, the process of analysing the data produced by each application and designing the interchange procedure was extremely important in exposing the issues in this field, and is discussed below.

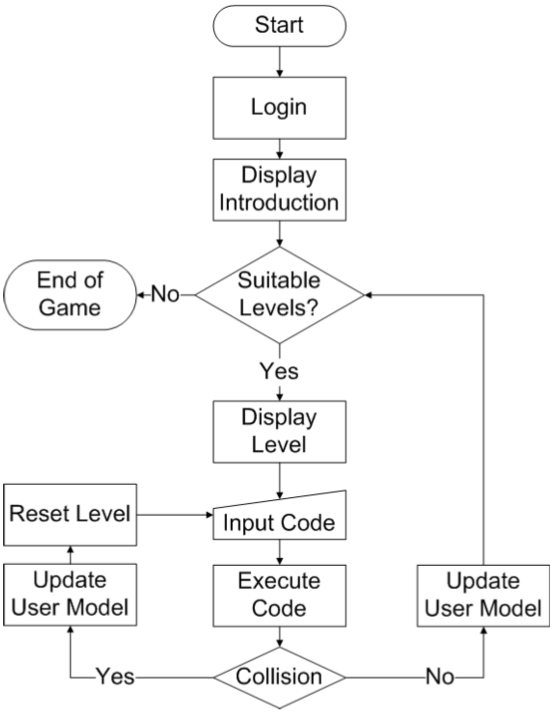


Figure 4.4: A flow diagram depicting the expected order of events in the game.

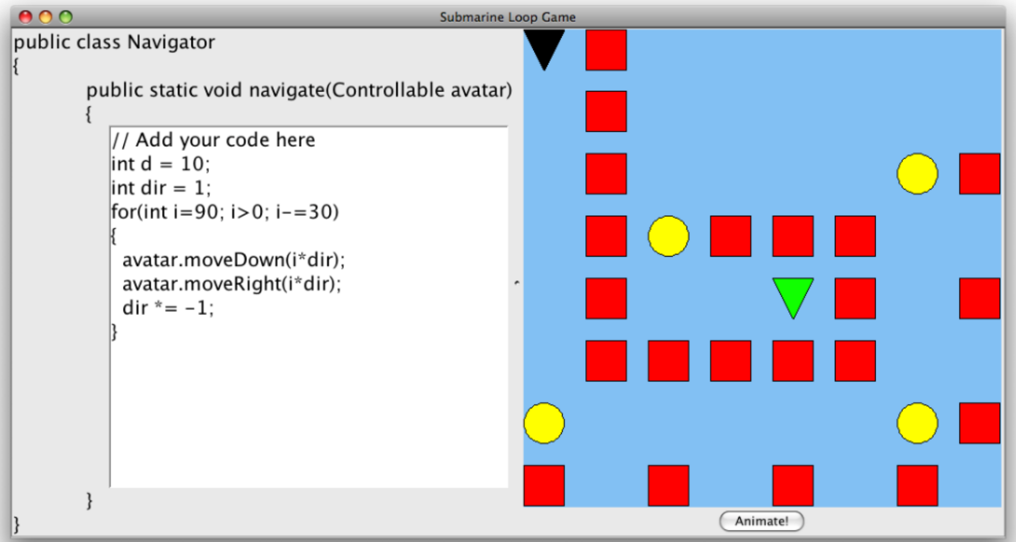


Figure 4.5: A screenshot of the Loop Game displaying an example level.

For each concept in the domain model in GALE, the overlay user model can store a set of attributes of type boolean, integer or string. Typical examples used in the AHA! (a precursor to GALE) tutorial (De Bra et al., 2004) are knowledge, suitability and visited. The interesting thing to note is that the primary interaction between the user and a hypertext system, such as GALE, is clicking on links. Only when a link is clicked can any inference about the user be made. All that is known is that the user has visited the page, possibly a number of times. The examples in the AHA! tutorial update the knowledge attribute when a page is visited, implying that visiting a concept's page is analogous to an increase in knowledge of that concept.

Each level in the adaptive game is designed to teach and assess a particular set of concepts in the domain model. Statistics about the user's attempts at the level make up the attributes in the user model. These values include the number of attempts at completing the level, the amount of time spent doing so, the number of successes (and by inference, number of failures) and a profile of the code (number of lines, statements and loops used).

It is evident that the adaptive hypermedia produces a smaller amount of data about the user than the game, as it can track fewer interactions and in less detail. The solution of requiring the game to compress the rich data available into a single value is undesirable as it puts a load on every new adaptive application to support reducing its user model values into those attributes that other applications currently use. It also limits the ability of any future applications to use the richer attributes; it is the dumbing down of user model data.

One approach to make the GALE hypertext benefit from the extra data that the game provides is to rewrite the GALE hypertext's adaptation rules making use of the new attributes in the user model. The other approach is to rewrite the game adaptation engine, so that rather than putting values for time taken, number of failed attempts and number of loops used into the user model, the game applies an equation to these values to calculate a knowledge attribute, which GALE is able to understand.

However, updating each existing adaptive application to make use of new user model attributes provided by new applications is an unreasonable expectation to make of the developers of the existing applications, if they are even still in active development.

Currently, the game and the hypertext are able to read and write whichever values they choose to the user model. Therefore, given knowledge of the attribute values provided by each application, the adaptation engine and the adaptation rules could be re-written to accommodate them. While this is plausible with two applications sharing user model data, it does not scale particularly well. A better solution would be for each application to write the values it generates to a shared user model, to read the values it wishes to adapt on from that user model and for the user model system to interpret and map the values from each application, so the others benefit from the data, even if they do not read the values the other applications provide. It is already possible to map between related domains, so the next step is to map between the attributes used by diverse applications.

4.2.5 Analysis of Other Educational Applications

After identifying that there is a difference between the types of attributes used in an adaptive hypertext and an adaptive game, it is appropriate to briefly discuss the attributes that occur in other applications typically used in education, particularly those that have the potential to be made adaptive.

Although this work is considering how to merge different attributes from a variety of adaptive applications, a number of applications have been analysed to determine which attributes they could contribute to a shared user model, to see if there are any attributes that can make up a common base language.

The attributes available in these various forms of application are summarised in Table 4.1.

Some adaptive systems described in the literature will contain multiple of these categories; for example, a number of adaptive hypermedia systems will have hypertexts, assessments and scrutable user models all built into one application. In this section, these types of application are treated individually, to assess what they uniquely provide. These hybrid applications demonstrate how they make use of the user model values from the different sub-components of the application, but have the benefit of being developed simultaneously to take advantage of the attributes in each sub-component; the challenge in this thesis is to find a way to replicate this but between applications that are developed independently and without the guarantee that any of the other adaptive applications will be used.

4.2.5.1 Hypertext

Adaptive hypermedia has a rather simple interface and as such provides little information about the user in its basic form. A plain hypertext engine only knows what link the user has clicked on, which page they have gone to and what time they clicked the link.

Some additional information could be inferred from these values, but they require some assumptions to be made and so are not necessarily accurate. When a user clicks on a link, it suggests that the user has some interest in the information at the end of that link. Visiting a page means the user may have increased their knowledge of the concept(s) on that page, assuming they have read and understood it fully. The time between clicks provides some information about how long the user spent looking at a page, but this is dependent on the user clicking away from the page as soon as they have finished with it.

4.2.5.2 Games

Games are not as easy to analyse as a hypertext, as a game can take on many forms, whereas a hypermedia always contains links and document content. At the very least a game (or a level or sub-game) should contain rules and a goal, so a user model can track success at completing this goal. Related to this would be to count the number of attempts at completing the goal. A game can measure the time taken to complete a task, as it is able to (and required to) detect the start and goal completion events. That said, it is still possible that the user spent time away from the computer while the game was still running, so this time may not accurately map to how long the user actually spent solving the problem.

4.2.5.3 Assessment

In an educational context, an adaptive assessment engine provides some of the best user data. Some adaptive hypertext engines even include quiz or questionnaire modules as a way of augmenting their user model data. At a basic level, an assessment quiz engine knows whether the user answered correctly or incorrectly to a question. If a question requires the user to select 3 items out of 5, the assessment engine would also know if the user selects 2 correctly and 1 incorrectly, which might be interpreted as a 2/3 understanding of the concept. Unfortunately, free text answers are very difficult to turn into user model data, without very sophisticated natural language processing.

4.2.5.4 Virtual Worlds

Virtual worlds fall into a space somewhere between games and hypertexts. They share the visual characteristics and collaborative multi-user aspects of some games, but often lack the goals and rules that typify a game. As an environment to display content, though in a visual (usually three-dimensional) way, and with the ability to teleport between locations in the virtual world, they somewhat mirror the textual content and hyperlinking functionality of a hypertext.

Because of these characteristics, it is possible for a virtual world to ascertain that a user has visited a particular area and how long they were there. The virtual world can also detect that the user has communicated with other users, possibly indicating that the concept being learned has been discussed.

The virtual world could also report on the user's interactions with various objects in the world, but there are many permutations of each interaction with each object and their meaning is specific to the area that has been built in the virtual world. That said, a 'look at' interaction is similar to a visit to a hypertext page, as long as the object being looked at has an associated concept. To 'pick up' an object may infer the user's interest in that object.

4.2.5.5 Recommender Systems

Recommender systems are common on online stores such as Amazon.co.uk, where they provide suggestions of new products based on the previous ones bought. To do this, they store a value to indicate interest in a product. They may also track which suggested products were turned down, which essentially results in a negative interest value.

4.2.5.6 Forums

Discussions forums do not tend to adapt to the user, other than due to a few preferences as to how the forum may be displayed. However, when used in an learning context, forum usage may indicate some information about the user. For example, reading particular threads associated with a known concept will affect the user's knowledge of that concept, while starting a thread with questions about a concept indicates an interest to know more about that concept.

4.2.5.7 Social Networks

The rapid increase in social network usage over the past half-decade has resulted in the creation of user models for the users of these websites. However, most of the information stored is for the purpose of encouraging social interaction between the users. Therefore, most of the data is either personal details or the interests of the user.

4.2.5.8 Virtual Learning Environments

Many educational institutions are using software systems to help their learners manage their courses, often known as Virtual Learning Environments (VLEs) or Learning Management Systems (LMS). These systems contain many of the features of the applications already discussed here. They usually contain course materials, hyperlinked together, self-assessment quizzes, records of previous assessment (exams and coursework) and discussion forums. Therefore, VLEs could contribute much about a user's accessing of material, previous results from assessing knowledge of various concepts and interest, derived from access or discussion.

4.2.5.9 Institutional Records

The records held by an institution contain much useful data about a learner. The results of the various forms of assessment, such as coursework and exams, required by the institution measure knowledge and understanding of the concepts assessed, and the learner's choice of modules indicate an interest in the concepts taught on those modules.

4.2.5.10 Scrutable User Model Editor

Scrutability describes the ability of a user to freely view and edit their user model. This allows the user to correct values in the user model and even add new ones. However, it is worth noting that unless the user has an understanding of what the values mean in the context of the applications that use them, any edits by the user may be inaccurate.

4.2.5.11 Summary of Generic Attributes

From the analysis above, Table 4.1 has been constructed. It records which measurements of the user can be made by an application, and whether those measures are explicit from the user's actions, or if they are implicit because they are calculated from some other explicit measurement and some assumption of the relationship. For example, hypertexts can explicitly measure a user visiting the page and what time they visited. Sometimes hypertext applications will infer how long the user spent on a page using the difference from the subsequent page visit (*Time Spent: Implicit*); often hypertexts will assume the user has some level of knowledge because of their visit to a page (*Knowledge: Implicit*).

Application	Interest	Knowledge	Discussion	Time Spent	Visited	Start Time	Success	Attempts
Hypermedia	Implicit	Implicit		Implicit	Explicit	Explicit		
Games	Implicit	Implicit		Explicit		Explicit	Explicit	Explicit
Assessment		Explicit		Explicit			Explicit	Explicit
Virtual Worlds	Implicit		Explicit					
Forums			Explicit					
Rec. Sys.	Explicit							
Social Nets.	Explicit		Explicit					
VLEs	Explicit	Explicit	Explicit		Explicit			
Institutional Records	Explicit	Explicit						

Table 4.1: Summary of the measurable attributes in common types of educational applications, and whether they are measured explicitly or only inferred.

The results in Table 4.1 demonstrate that there is no one attribute that is measured by all commonly used educational applications, making it difficult to suggest one value that all applications could map onto for the purposes of sharing. It also highlights the complexity that would be involved in defining an ontology of attributes to share between applications and their relationships.

4.2.6 Summary of the Problem

This chapter has described an initial prototype created to discover the issues surrounding sharing user model data between different types of application. To do this, an adaptive hypermedia tutorial and an adaptive educational game were created using a common domain model.

As a result of creating these applications and analysing the user model data they produce, a disparity was discovered. The adaptive game could provide richer, more detailed data about the user than the adaptive hypertext, because it allowed and tracked more user interactions with the system.

Neither changing existing applications to cope with new user model data, nor reducing all user model data to a single, catch-all value, are acceptable as scalable, long-term solutions. Instead, a user model which can accept the usual read and write requests of adaptive applications, yet also map values internally so all applications can benefit from the data received from new applications is proposed as a solution.

4.3 Proposed Solution

In order to address the issue of mismatched attributes, the author proposes the IDIUMS framework and reference implementation. IDIUMS is a centralised user modelling system that provides mapping between attribute values in different adaptive systems, based on expert-created rules. To an adaptive application it simply requests a value (for a certain user, concept, attribute) from the IDIUMS user model, and will receive some value back even if is has never recorded any information for this user, concept, attribute.

IDIUMS is designed to provide a conversion between the different attributes that might be provided by various applications. The reason behind this is that it tends to be the attributes that differ between applications, as each application represents the user and their interactions in a different way.

The key principle behind this approach is that applications provide rules that map their attributes onto the attributes of existing applications. When a value is requested from IDIUMS one of three things may happen: (1) a value is already available for that request, so it is returned; (2) there is a rule directly to an application with a value, so is used to convert and return a value, or; (3) rules are chained (to exhaustion or a pre-defined depth) until an application with a value is found, so all the rules are applied one after another to calculate and return a value.

4.3.1 Architecture

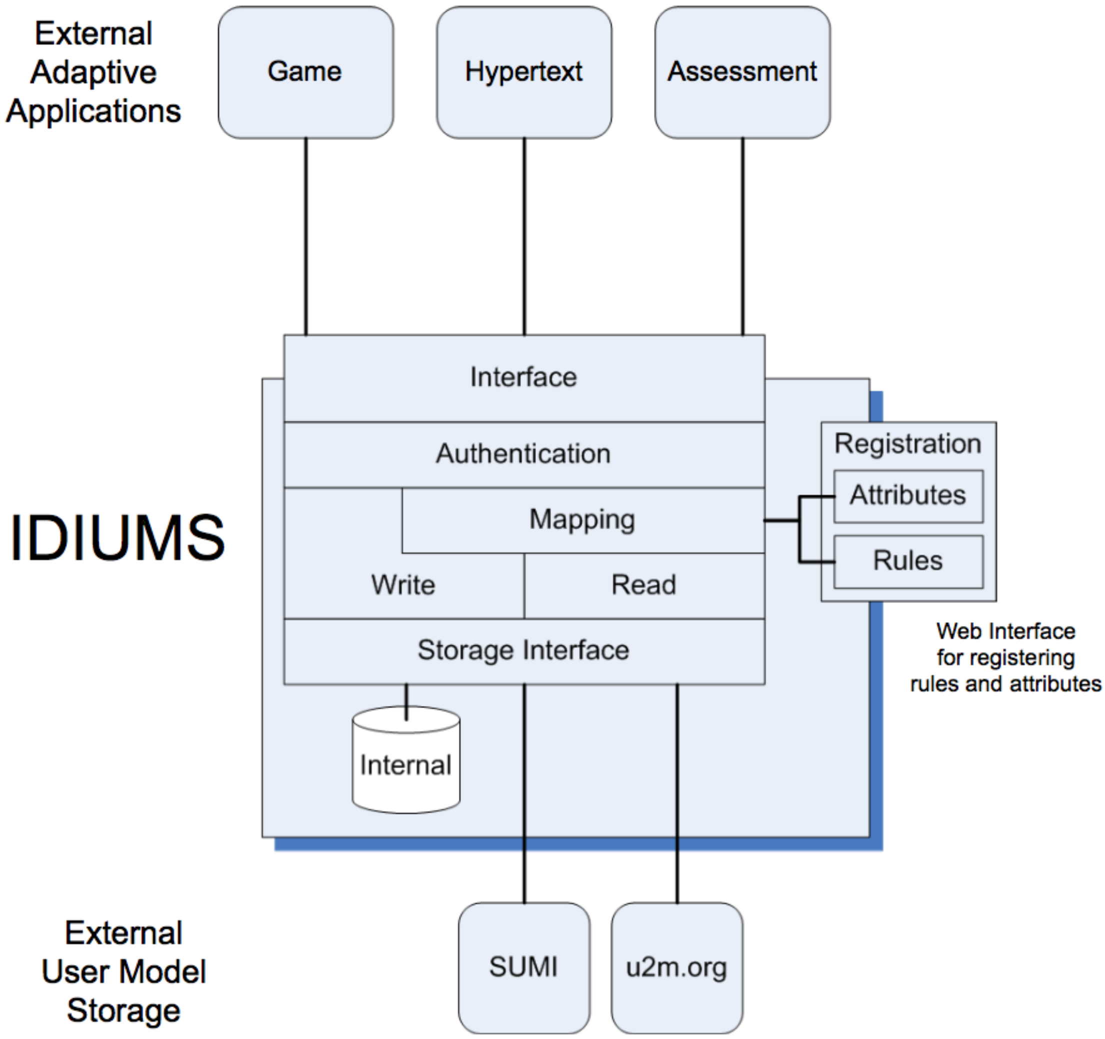


Figure 4.6: Architecture diagram of the various components of IDIUMs, and how they interface with external applications.

IDIUMS has been designed as a centralised user modelling system to limit the scope of this research to just the issue of mapping between attributes. It is feasible to imagine how IDIUMS could work in a form of peer-to-peer network, so each application stores a local user model and contacts other peer user models should it not have a value, though there would likely still need to some central repository to track the network addresses of other applications and possibly the shared rules for mapping between applications. However, it is only worth considering the complex issue of how to make IDIUMS function as peer-to-peer after evaluating that the concept works in the first place, and it is simpler to do this with all the user model data and rules in the same, centralised location.

(As an aside, it is worth pointing out that in practice IDIUMS need not be a centralised bucket of all users' data, as currently there is no explicit interaction between the data of different users. This means applications could be built to take the network address of the current user's IDIUMS installation as a setting, allowing the user to host their own user model data.)

The IDIUMS architecture, as depicted in Figure 4.6, includes a layer which maps between domain models, which could be implemented using one of the existing methods referenced in Section 2.3.3.1. However, for this thesis this layer will be implemented statically; instead of having the additional variable of dynamically mapping between domain models, a common domain model will be used.

4.3.2 Overview of IDIUMS Use

To explain the utility of IDIUMS and give the reader some understanding of how the system will work, an example scenario for use will be stepped through.

1. Adaptive application A is built, and uses a library to integrate IDIUMS as its user model.
2. Developer of A registers A with the IDIUMS system, detailing which attributes it will read and write (data type, range of expected values).
3. A is used by some users, and their data is stored in IDIUMS.
4. New adaptive application B is built, and uses a library to integrate IDIUMS as its user model.
5. Developer of B registers B with the IDIUMS system, detailing which attributes it will read and write, and providing some rules that map from B's attributes to A's attributes.[2]
6. B is used by some users.
   (a) If they have previously used A, IDIUMS uses the values in A combined with the mapping rules, to calculate an appropriate starting value for B.
   (b) If they have never used A or B, IDIUMS will raise an error and the application can use its default 'zero' value.
7. While the user is interacting with B, new values may be written back to IDIUMS.
8. New adaptive application C follows the same process as B did, but may map rules to attributes in A or in B.

[2]Once B has joined the IDIUMS ecosystem, it would also be valid for the author of A to add some rules that map between attributes of A and B, though this cannot be guaranteed as happening.

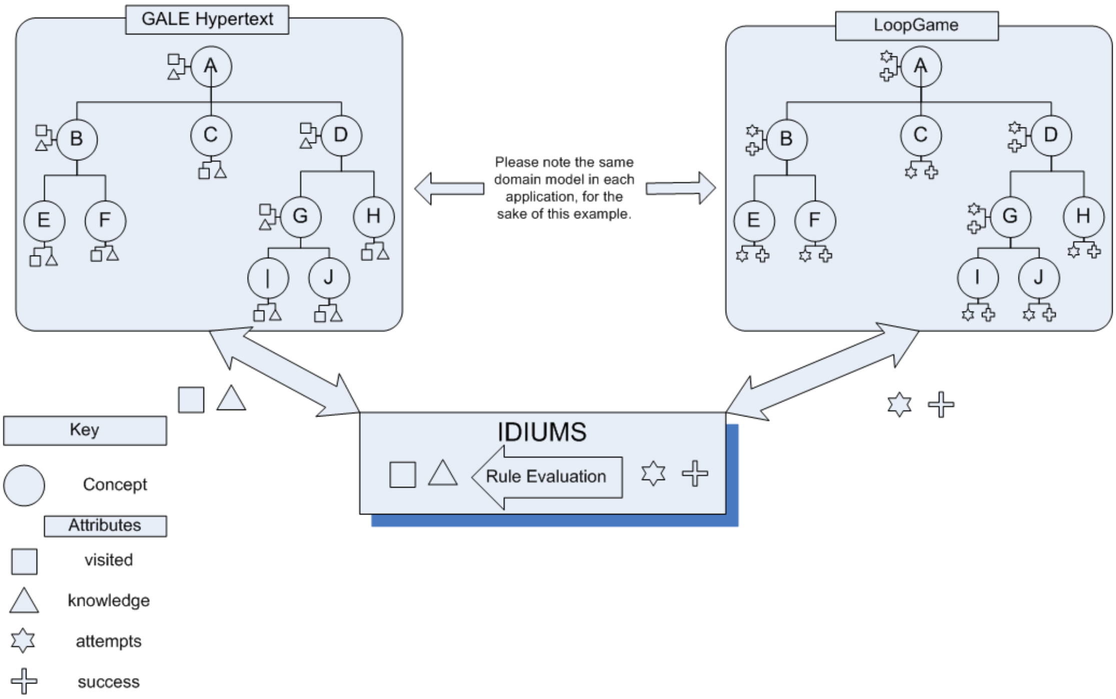


Figure 4.7: Demonstration of the flow and conversion of attributes between the LoopGame application and GALE hypertext application.

(“alice”, “NestedLoop”, “attempts”, 2)
(“alice”, “NestedLoop”, “success”, 1)

Figure 4.8: Quadruples produced by LoopGame as Alice plays it.

4.3.3 Concrete Example Scenario

To illustrate the difference in attributes between applications, and how translating values between the attributes in different applications can help reduce the “cold-start” problem associated with new adaptive systems, the following example has been devised:

If Alice plays LoopGame and for the NestedLoop concept has attempted the level twice and once succeeded at it, this will be recorded in an IDIUMS quadruple (Figure 4.8).

When Alice goes to the hypertext to read about loops, GALE will initially not know anything about her understanding of the concepts in its domain model. GALE could communicate with LoopGame directly, but they speak a different language. GALE adapts on visited (flag of page visit) and knowledge (usually inferred from a visit), whereas LoopGame adapts on attempts (at levels) and success.

IDIUMS will use a rule set to express relationships between attributes provided by different applications.

In this example, IDIUMS might map attempts directly onto visited (playing a level is like visiting a page) while calculating that 1 success in every 2 attempts (i.e. 0.5 success)

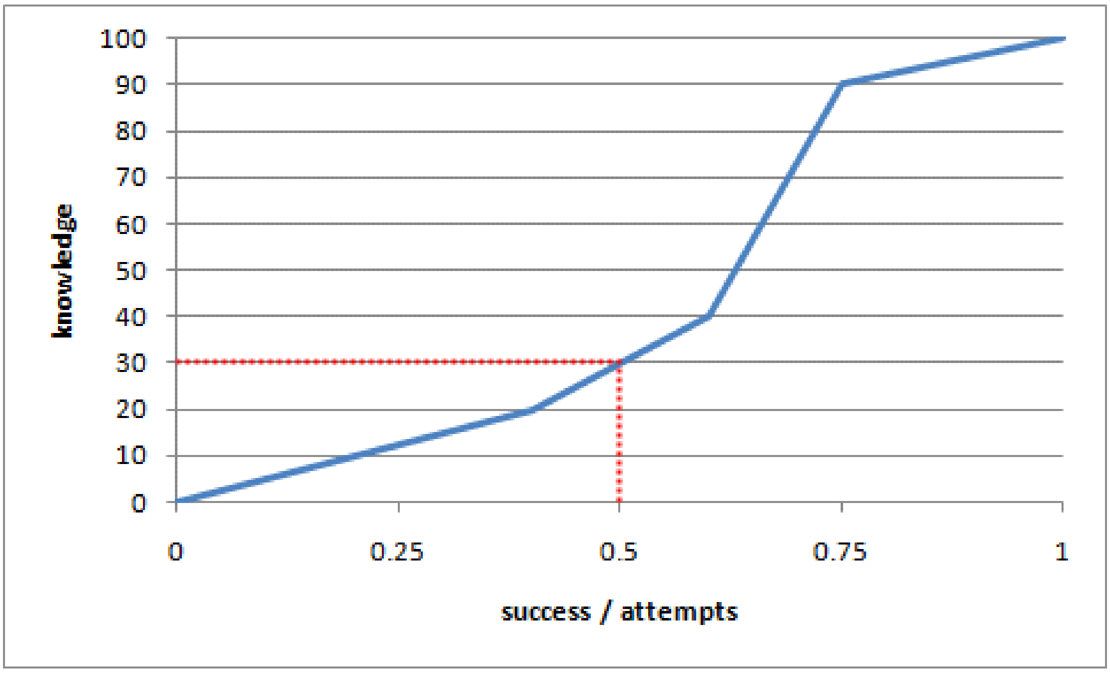


Figure 4.9: This graph plots the relationship between successes/attempts and the knowledge value as defined in Table 4.2. Red dashed line demonstrates interpolation of a value between those in definition.

("alice", "NestedLoop", "visited", 2)
("alice", "NestedLoop", "knowledge", 30)

Figure 4.10: Quadruples produced by IDIUMS to provide to GALE.

is equivalent to 30% knowledge of the concept in question (as shown in. This could be represented by the following rules:

$$visited = attempts \tag{4.1}$$

success/attempts	knowledge
0.00	0
0.40	20
0.60	40
0.75	90
1.00	100

Table 4.2: Example of an non-linear relationship between success/attempts and knowledge.

By applying these two rules, IDIUMS can generate the quadruples in Figure 4.10. GALE is then able to adapt on the attributes it knows about, visited and knowledge.

4.3.4 Options for Mapping Attributes

A core element of IDIUMS is the way it maps between the attributes of one application and those of another. Three methods of doing this are proposed here.

One approach to mapping the attributes is to have a pre-defined ontology which describes all possible attributes and how they map to each other. The advantage of this is that application developers are aware of the attributes available and how they are mapped. However, it would also require regular updating as new applications are added: a requirement IDIUMS is designed to avoid! A variant of this approach is to define an ontology of just those attributes which are deemed core to all known applications and only guarantee mapping onto these attributes.

Another way of eliciting mappings between attributes is to require developers to register the attributes their app will use with IDIUMS before being allowed to access it. In this registration process, developers would have to define which attributes they would contribute to the user model and provide rules to indicate how they map to attributes in the core ontology mentioned above or even onto attributes from other applications.

Finally, mapping between attributes can be derived using a collaborative filter, between users. At the user level, the filter could determine when multiple users have similar attributes for a particular application and infer a value one user does not have for an attribute, based on the value the other user has. For example, say both Anne and Betty completed the nested-loop level of the Loop Game in 5 *attempts*. If Anne has a value for nested-loop *knowledge* of 80, but Betty has no recorded *knowledge* attribute, it could be reasonable to believe that Betty's *knowledge* value will be close to Anne's as they completed the game level in the same number of *attempts*.

Of the three methods, it is the developer-registered rules that has been selected for this first iteration of IDIUMS design. However, the other two approaches should not be dismissed. By providing an interface to the available attributes and rules, IDIUMS in effect enables the ontology approach as developers can choose to mimic existing attributes to make for simpler mapping upon integration with IDIUMS. The collaborative filter is an approach to automatically generating rules, which is desirable but ended up outside the scope of this work; it would however make an excellent piece of future work.

4.3.5 Issues and Limitations

4.3.5.1 Coverage

Consideration is not given to the minimum required coverage of rules. In fact, it is expected that a further research question would be to discover how many rules out of each application are required to ensure the system is stable. For the sake of experimentation, as many rules as possible will be added to the system. However, as the aim is merely to be better than a cold-starting adaptive system, as long as the hypothesis holds, a system which maps any data between two systems should be better than no data sharing at all, as long as that mapping is not intentionally detrimental.

4.3.5.2 Map on read or write

A design decision that must be considered in such an intemediary user modelling system is whether to perform the mapping at the point that a value is requested from the intermediary system, or to convert values as they are written into the system. The primary difference between these two is the common space-time tradeoff.

In terms of storage cost, mapping at the point a client requests a value has zero additional storage cost, only needing to store the values explicitly written to IDIUMS. If mapping is performed when a client writes a value to IDIUMS, then additional storage is required to cache the results of all calculations of rules, and rule chains, related to the attribute being written.

Ideally, IDIUMS needs a minimum of one rule per attribute to be useful, but it is expected that there will often be more than this. Even at 1 rule per attribute, that only relates to one attribute in another application, the chaining algorithm currently caps at a depth of 10 rule links, so this cache will cause an extra 10 values to be stored for every 1 written to IDIUMS. When considering that there may be multiple rules mapping each attribute, and each rule might reason on multiple attributes, caching all possible results of rules becomes relatively expensive.

The benefit of storing these calculated values for each value written to IDIUMS by a client is that it allows the time-cost of those calculations to be shifted. If the mapping is performed when a value is requested by a client, then the execution of those rules, and if necessary all the chained rules, will cause some delay in providing a response to the client, which in turn might make the end use wait and lessen their user experience.

However, if mapping calculations are performed when a value is written to IDIUMS, then an immediate response can be given to the client, and the queue of mappings can be calculated by a pool of worker threads, or a separate process. The effect of this is that dealing with a client request for a value is a trivial operation as IDIUMS has already calculated all possible permutations of rule chains and can instantly respond, and reduce the delay to the end user.

An alternative optimisation, could be to convert all incoming values into a common intermediate set of attributes, based on a common ontology. This would reduce the amount of additional data stored in the cache, and would require a single calculation on both read and write. However, this is expressly against the aim of IDIUMS to avoid having to be updated with new ontologies as new, unpredictable applications introduce new attribute types. The goal is to put the onus of mapping between new attributes and existing ones on the developer that introduces the new attribute, making the ecosystem self-sustaining.

Decision The additional storage required to create a cache of this sort is not likely to be a significant barrier, due to the rapidly decreasing costs of spinning-disk storage. If IDIUMS were to be deployed in a situation with thousands of users and particularly where there are many applications in the ecosystem, this should be reconsidered, because the greater the number of applications increases the likelihood that attributes will have rules related to them, which increases the branching factor of the algorithm. The cache could be adapted to store fewer values by restricting the depth of rule chains followed.

Therefore, the decision had to be made regarding whether the time spent calculating the results of the mapping rules should be performed when values are written to IDIUMS or when they are requested by the client. It was determined that the caching mechanism, and how it interacted with the response generation, would require significant development time. With only a small number of applications, few rules, and a moderate number of users, it was decided that the response time to users would be acceptable, and as such implementing the cache of rule mapping results upon write was outside the scope of this reference implementation and the subsequent experiments. However, future administrators of highly-used IDIUMS-based environments should consider developing a caching feature.

4.3.6 Comparison to GUMF

The principle related work in this area is the Grapple User Modelling Framework (GUMF) described in Section 2.3.3.2. Presented here is an analysis of what is similar and different between IDIUMS and GUMF.

The underlying difference between GUMF and IDIUMS is how they frame the solution, which has repercussions for the design of the architecture. GUMF presents itself as an "intelligent storage and reasoning engine" (Leonardi et al., 2010) and provides this in a very generalised way. This is reinforced by the *Dataspace* aspect of the architecture, as it allows data from any source (whether it be other user models or other data sources, such as DBpedia in the Leonardi et al. (2010) geography example) to be made available to the running instance of the adaptive application. IDIUMS seeks to be a straightforward replacement or addition to an overlay user model, in both existing and new adaptive educational systems.

This decision is reflected in the way data is stored, communicated and reasoned upon in each of the systems. In GUMF, data exists in GRAPPLE statements, a triple-based format that "are basically reified RDF statements". This allows it to make use of other, external data sources that are represented in RDF, to combine using rules to generate new dataspaces. This contrasts with IDIUMS's approach of representing overlay user model data as (user, application, concept, attribute, value) tuples, inspired by earlier adaptive hypermedia systems.

There are also differences in terms of using these two systems as part of an adaptive application. GUMF is part of the larger GRAPPLE framework, and requires the Grapple Event Bus (GEB) to enable the adaptive application to communicate with it. Each of these are complex applications that require compilation and setup. One of the goals of IDIUMS is to be a self-contained application, distributed as an executable JAR file, which can be run with a single CLI command (or double clicking a script) and immediately provides a web (HTTP) interface on the requested port.

After launching the user modelling framework, there is the barrier of integrating it into an existing or new codebase. Both GUMF and IDIUMS provide libraries to assist with this. The use of RDF Grapple statements in GUMF requires client applications to be able to parse, reason with or convert RDF statements and URIs, into the format the application uses to represent user data. Whilst new applications being built can benefit from the encouragement to use such a flexible and standardised data interchange format, it may represent a barrier to existing adaptive applications integrating with GUMF. IDIUMS's use of basic computer data types such as integer, floating point decimals and character strings should reduce the development burden on clients applications when attempting to integrate.

Notably, the example in Leonardi et al. (2010) seems to suggest that the receiving application should be developed to query the new dataset provided by the GDR and GUMF. IDIUMS differs by focusing on mapping onto the attributes an adaptive application already reasons in, with the aim of reducing the amount of work a developer has to do to integrate IDIUMS. While GDR and GUMF could be utilised to map onto existing attributes, applications would have to decode from GUMF's subject-predicate-object Grapple statement format, and the emphasis on dataspaces suggests developers should see GUMF as an extra data source, rather than a utility to feed their existing user model.

4.4 Design

The structure of IDIUMS can be described as a series of layers (as seen in Figure 4.6). Adaptive applications communicate with the interface, via HTTP. In the design, there is a layer of authentication to ensure only user for the appropriate user is shared. Writing data to IDIUMS passes straight through to the storage, but reading from IDIUMS involves a pass through the rule mapping module. Both reading and writing work through the storage interface, which usually attaches to an internal database, but could equally connect to an external user modelling service such as u2m.org or SUMI. The mapping module stores details of the known applications, their attributes and all rules that apply to known attributes; all of this information must be registered with IDIUMS, by application developers.

The rules module and the HTTP interface are described in further detail below, along with details of the data design for the storage and the software design of the entire system.

4.4.1 Rules

One of the underlying tenets of this proposed system is that the developers of adaptive educational systems (AES) that introduce their application into the IDIUMS ecosystem must register their application, the attributes it will read and write, and define some rules that map between values for these new attributes onto the attributes in existing applications. This will allow new applications to use the user's data that has been contributed from other applications.

To better explain how IDIUMS will convert between the attributes from different applications, an explanation of the rules which define the conversion follows.

It is worth noting that rules may be based on mathematical combinations of multiple attributes. An example of this can be seen in Table 4.2 where the knowledge value is not calculated from the successes value alone: it is calculated from the result of successes divided by attempts.

4.4.1.1 Types of Rules

To convert between the different types of data in various adaptive applications, a flexible rule system is required to account for the ways in which a conversion may need to be described. Therefore, three types of rules have been considered here.

- Thresholding - eg. above 20 $\rightarrow$ 1, above 40 $\rightarrow$ 2, above 70 $\rightarrow$ 3, etc.
- Straight line graph (like above but with continuous values between).
- Mathematical equation (continuous conversion over all values).

assessment score	stereotype
$0 <= x < 40$	beginner
$40 <= x < 75$	intermediate
$75 <= x < 100$	expert

Table 4.3: Example of a possible relationship between a score-based user model and a stereotype-based user model.

4.4.1.2 Thresholding

This type of rule would be key in converting between values that are continuous over a particular range (for example, average assessment score) and values that are discrete (such as a game level, a number of attempts or a stereotype).

4.4.1.3 Straight line graph

The straight line graph is defined in a similar way to thresholding, but rather than outputting a single value as long as the input value is above a certain threshold, it interpolates a value. An example of this sort of relationship can be seen in Table 4.2 and Figure 4.9.

4.4.1.4 Mathematical equation

The intention is that any mathematical equation can be used to describe the conversion between the input value and the output. For example, a direct relationship between attributes would be described by Equation 4.2, a conversion which quickly increases but then tapers off (diminishing returns) could be defined by a logarithmic equation (such as in Equation 4.3 or a conversion based on the normal distribution (Equation 4.4.

$$y = x \tag{4.2}$$

$$y = ln(x) + 4 \tag{4.3}$$

$$y = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4.4}$$

Term	Description
Variable	a model class for an attribute attached to a certain application. Also contains information stored in the database about this attribute, such as the application it is associated to, its type, and upper and lower bounds (if it is numerical).
Binding	a representation of a variable within a rule, linking it to its symbol in the rule (e.g. an equation like = 7x + 5, x would be the symbol).
Rule	a representation of the mapping from one or more attributes to another. The supported possible rules are described earlier in the Section 4.4.1 of this chapter.

Table 4.4: Definitions of the terms *Variable*, *Binding* and *Rule*.

4.4.1.5 Rule Chaining

Another important feature of the design of IDIUMS is that where an immediate mapping cannot be made, for example because the application the rule refers to has no data about the user either, the rules from that application to others will be followed, in the search of any data about the user. This chaining will continue until user data is found or a pre-defined depth is reached (10 rules in the prototype).

4.4.1.6 Pseudocode

The core of IDIUMS is its rule traversing algorithm that calculates new values for a user accessing a certain application based on the values entered by other applications. While not necessarily novel itself (it is, after all, a bounded depth first search at its core), it is important to document the way it works in case future researchers wish to reimplement it.

Definitions Definitions of the terms *Variable*, *Binding* and *Rule* can be found in Table 4.4

Code Below is a pseudocode version of the getValue() method in IDIUMSResponse.java (starts line 118).

```
function getValue(user, application, concept, attribute[, datetime])
        Attempt to retrieve the value from the database
        if the value exists
                return the value

        Add unknown variable to queue
```

```
        Repeat while unknowns remain, and while depth < 10
                For each unknown variable, v
                        Retrieve rules that output variable, v
                For each rule, r
                        Retrieve variables that rule, r, relies on
                For each rule variable, rv
                        Attempt to retrieve the value from the database
                        If the value exists
                                Store it a map
                        If the value does not exist
                                Add variable, rv, to the unknowns queue
                For each original unknown
                        If value is in the map
                                Retrieve value from the map
                        If not in map
                                Evaluate variable (this is another method)
                        Add resultant value to list VALUES
        return VALUES

function eval(values, user, concept, seenAlready)
        For each binding
                Evaluate binding (returns list of values)
        Calculate permutations of binding values
        For each permutation
                Evaluate this rule using those permutations
                Add result to list of ALL_RESULTS
        return ALL_RESULTS
```

4.4.1.7 Rule Conflict Resolution

The upshot of chaining rules is the likelihood of exponentially increasing the number of resultant values, and with it the potential of those values not agreeing. A number of methods of resolving between these conflicting values are possible:

4.4.1.8 Example Return for Different Resolutions


```
<xml>
  <value user="thesis" application="Loop Game" concept="variables" attribute="
    successes" datetime="2016-01-16 00:29:57" ruledepth="1">5</value>
  <value user="thesis" application="Loop Game" concept="variables" attribute="
    successes" datetime="2016-01-16 00:29:57" ruledepth="2">8</value>
  <value user="thesis" application="Loop Game" concept="variables" attribute="
    successes" datetime="2016-01-16 00:29:57" ruledepth="1">1</value>
```

```
  <value user="thesis" application="Loop Game" concept="variables" attribute="
    successes" datetime="2016-01-16 00:29:57" ruledepth="4">5</value>
</xml>
```


Listing 4.1: XML example of multiple values in IDIUMS response


```
[
  {
    "user": "thesis",
    "application": "Loop Game",
    "concept": "variables",
    "attribute": "successes",
    "datetime": "2016-01-16 00:29:57",
    "ruledepth": "1",
    "value": "5"
  },
  {
    "user": "thesis",
    "application": "Loop Game",
    "concept": "variables",
    "attribute": "successes",
    "datetime": "2016-01-16 00:29:57",
    "ruledepth": "2",
    "value": "8"
  },
  {
    "user": "thesis",
    "application": "Loop Game",
    "concept": "variables",
    "attribute": "successes",
    "datetime": "2016-01-16 00:29:57",
    "ruledepth": "1",
    "value": "1"
  },
  {
    "user": "thesis",
    "application": "Loop Game",
    "concept": "variables",
    "attribute": "successes",
    "datetime": "2016-01-16 00:29:57",
    "ruledepth": "4",
    "value": "5"
  }
]
```


Listing 4.2: JSON example of multiple values in IDIUMS response

This example of a selection of potential resolution rules demonstrates that even some may need a secondary resolution should they also result in multiple values (such as (b) in Table 4.5).

At present, IDIUMS simply returns all of the matched or calculated values to the client and leaves the job of resolving to the client. However, future designs will incorporate the ability to select a resolution preference in the administration interface.

Resolution	Resolved Value
(a) Accept the first value returned.	5
(b) Accept the value with least rules chained.	5 or 1
(c) Calculate the mean of all the values returned.	4.75
(d) Calculate the mode of all the values returned.	4
(e) Calculate the mean of the distinct values returned.	4.67
(f) Accept the best value (optimistic).	8
(g) Accept the worst value (pessimistic).	1
(h) Accept the value derived from the most recent real data.	
(i) Accept values derived from applications we trust.	
(j) Accept values within some range or tolerance.	
(k) Select a value at random.	N/A

Table 4.5: Example resolved values for various suggested resolution rules.

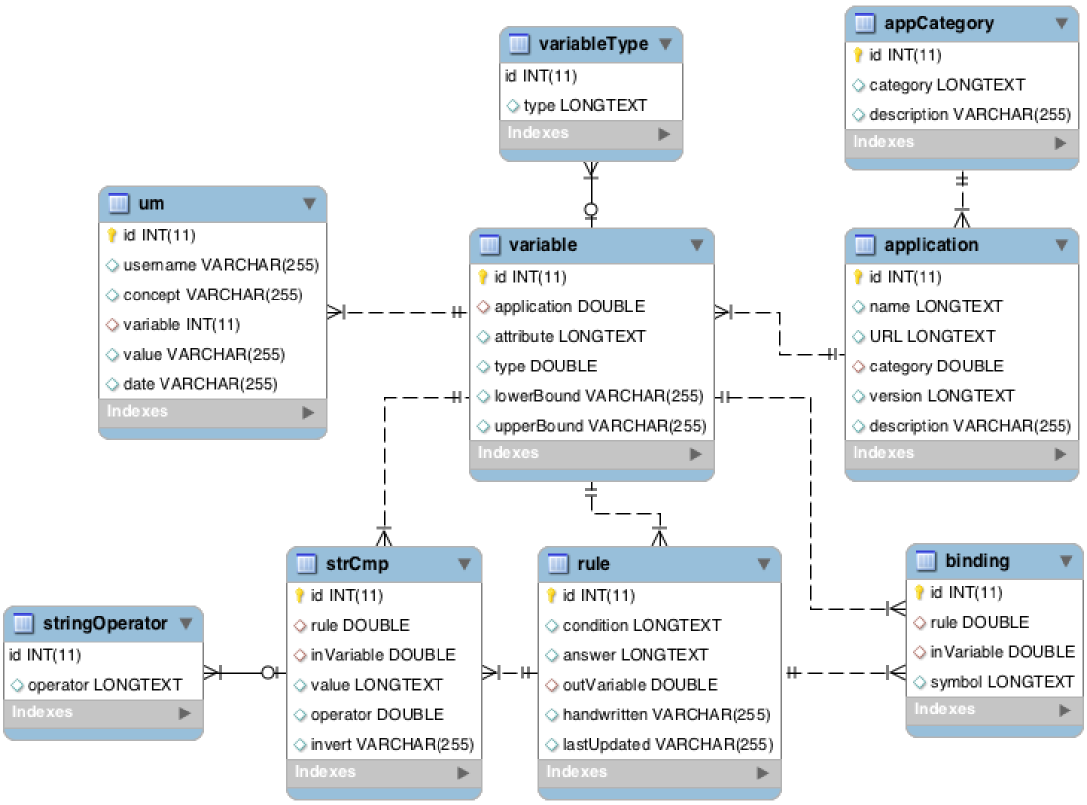


Figure 4.11: A enhanced entity relationship diagram of the database underlying IDIUMS.

URI	Returns
http://phd.idiu.ms:8000/um	HTML
http://phd.idiu.ms:8000/um.xml	XML
http://phd.idiu.ms:8000/um.json	JSON

Table 4.6: Mapping between URL and return content-type.

4.4.2 Data Design

The enhanced entity relationship diagram in Figure 4.11 describes the structure of the database that underlies IDIUMS by storing the registered applications and rules, and the user model data itself.

4.4.3 HTTP API

The method of communicating with IDIUMS is by HTTP requests to an endpoint that consumes `application/x-www-form-urlencoded` and outputs json and xml. It is not strictly a REST architecture (Fielding, 2000) as it does not implement HATEOAS and does not make full use of the HTTP verbs to perform CRUD (create, read, update, delete) operations. However, it does take inspiration from the REST architecture by referencing each ¡user, concept, application, attribute¿ as a unique URI (using parameters rather than clean URIs) and communicates over HTTP.

The endpoint for IDIUMS is `http://phd.idiu.ms:8000/um`. The requested data format is specified by a filename extension, and the resource is identified by query string parameters.

4.4.3.1 Return Type

The client interacting with the IDIUMS server can select which format is returned using file endings on the /um URI.

4.4.3.2 Query String Parameters

The client identifies which part of the user model it is trying to select using query string parameters. Below are the

4.4.3.3 Example Responses

Parameter	Description	Expected Value	Required?
user	The username identifying the user to whom the data relates.	String, e.g.	Yes
concept	What educational topic is IDIUMS recording data for, in this instance?	String, e.g.	Yes?
application	Name of the application.	String	Optional?
attribute	Name of the required attribute.	String	Optional?
variable	An ID to represent the tuple of application and attribute, as each has to be registered with the system.	Integer ID of <application, attribute>	Optional?

```
URI: http://phd.idiu.ms:8000/um?user=thesis\&concept=loops\&application=Loop+Game
    \&attribute=successes

<p>You have requested the value of thesis's variables.successes</p>
<p>getValue() produced:</p>
<ul>
  <li>15 - calculated @ 2016-01-20 21:45:54 - [Loop Game].successes(type:2) = s
    using {Binding[s]=15==thesis's variables.score in [Loop Assessment] @
    2016-01-15 14:20:16} - Rule Depth: 1</li>
</ul>
```

Listing 4.3: Example of requesting the success attribute for the loops concept in the Loop Game and receiving an HTML page.

```
URI: http://phd.idiu.ms:8000/um.json?user=thesis\&concept=loops.structure.
    condition\&application=Loop+Assessment\&attribute=correct

Response:
[
  {
    "user": "thesis",
    "application": "Loop Assessment",
    "concept": "loops.structure.condition",
    "attribute": "correct",
    "datetime": "2016-01-16 00:29:57",
    "ruledepth": "2",
    "value": "4"
  },
]
```

Listing 4.4: Example of requesting the success attribute for the structure concept in the Loop Assessment and receiving a JSON response.

```
URI: http://phd.idiu.ms:8000/um.xml?user=thesis\&concept=loops.loop$_variable.
    variables.arrays\&application=Loop+Hypertext

Response:
```

```
<xml>
  <value user="thesis" application="Loop Game" concept="loops.loop$_variable.
    variables.arrays" attribute="knowledge" datetime="2016-01-16 00:29:57"
    ruledepth="0">60</value>
  <value user="thesis" application="Loop Game" concept="loops.loop$_variable.
    variables.arrays" attribute="visited" datetime="2016-01-16 00:29:57"
    ruledepth="0">6</value>
</xml>
```


Listing 4.5: Example of requesting the success attribute for the loop variable concept in the hypertext and receiving an XML page.

4.4.3.4 Updating Values in the User Model

Changing of user model values is done by performing a POST to the URI of the value that needs replacing. IDIUMS accepts an XML or JSON body representing the value being set.

4.4.3.5 History

When a value is set, it is recorded as a new, timestamped entry in the user model database. The side-effect of this is that all historical values for a particular user, concept, application, attribute are kept, so it is possible to request user model values at a certain point in time. Rule chaining also works based on the user model value at that point in time, so it is possible to determine what IDIUMS would have responded in the past.

This historical storage of values enables simulations using data collected in the wild, allows client applications to adapt based on chronological features of the user's model and potentially allows future extension of IDIUMS to support rules based on historical values or the rate of change of those values.

4.5 Implementation

For the purposes of this thesis and the experiments conducted for it, a reference implementation of IDIUMS was constructed. This was built in Java 6, using a micro-webserver framework so that the whole IDIUMS implementation could be wrapped up in a JAR (Java ARchive) and executed cross-platform. While it is not expected that IDIUMS in this form will be used in any production systems, this ease of setup is important for the repeatability and extensibility of this research.

This initial reference implementation of IDIUMS does not support authentication. Future development of IDIUMS would include the addition of OAuth2.

Libraries for connecting to the HTTP-based API have been created for PHP, Java and R, and the simplicity of the API would make it easy to implement such a library in any other language with HTTP support.

4.5.1 Rule Creation

In IDIUMS, rules are added by administrators or developers of the client applications through the rule editing interface, located at the URI `/edit`.

At this interface, the administrator defines rules of the format depicted in Listing 4.6.

```
if(condition) then
   attribute1 = equation
Where:
   symbol represents attribute2
```

Listing 4.6: Format of rules in IDIUMS.
(Underlining indicates elements that can be modified.)

At its most basic, it provides a rule, in the form of the *equation* (an example is show in Figure 4.12(b)), that defines how to calculate a value for *attribute1* (which is demonstrated in Figure 4.12(c). The equation is expressed as infix notation, using mathematical operators, numbers and symbols that represent other attributes. The *Where* clause defines which attributes (typically in other applications) each symbol refers to, as seen in Figure 4.12(d). Finally, the entire rule can be restricted to only apply in certain circumstances, by setting the *condition*.

4.5.2 Rule Evaluation and Reversal

An open source Java symbolic calculator, named Jasymca[3], was modified into a library to perform evaluation of rules. A side effect of using a symbolic calculator for this is that in certain cases equations can be rearranged for a different variable, which means that rule chaining could work in the opposite direction to that specified by the author, whenever feasible (i.e. when there is only one unknown).

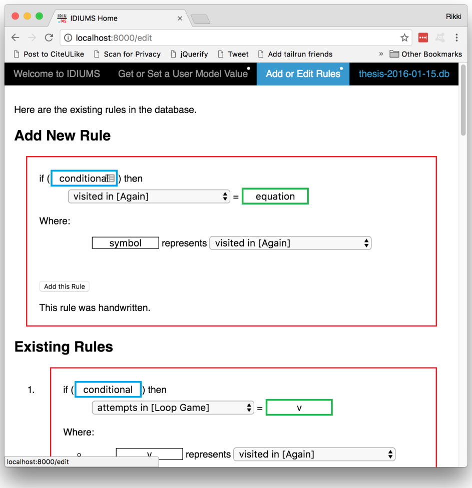
For example, given a rule such as:

$$s = v * k/100$$

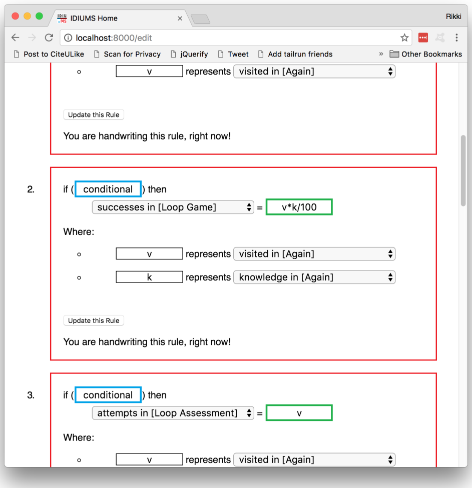
Where:

s is the successes attribute in the game application

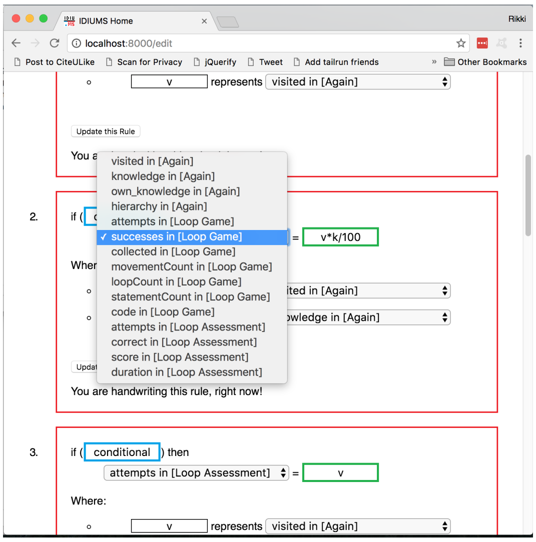
[3] `http://webuser.hs-furtwangen.de/~dersch/jasymca2/indexEN.html`



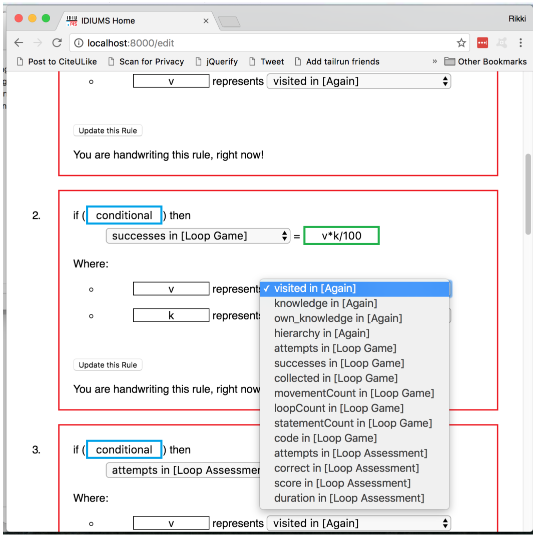
(a) Add a rule



(b) Example rule, $s = v * k/100$



(c) Setting output as *successes* attribute in `Loop Game`



(d) Setting v symbol as *visited* attribute in `Again` (the GALE hypertext)

Figure 4.12: Screenshots of adding rules in the IDIUMS `/edit` interface.

v is the visited attribute in the hypertext application

k is the knowledge attribute in the hypertext application

If s and v are known (the user model has values for these attributes for the current user), and a value is required for k, the symbolic calculator is able to rearrange the equation, without additional input from the rule author, to create a new rule:

$$k = s * 100/v$$

4.6 Conclusion

As a wider range of education applications have adaptive aspects added to them there will be an increasing number of attributes stored in user models that represent how those applications have measured the user. To make use this data in full adds to the complexity of sharing user model data between applications because, without modification, each application only understands its own attributes.

This chapter has demonstrated this problem through the exploration of dissimilar adaptive applications, and then proposed the design of a software solution to this problem. The details of a reference implementation of this design are is also presented, in preparation for its use in later chapters of this thesis.

Chapter 5

Simulation

5.1 Introduction

IDIUMS, as a newly designed and implemented piece of software for mapping user model attribute data between applications, is unproven in performing its primary function. As part of the testing of the prototype implementation of IDIUMS, a simulation was designed and implemented to demonstrate that the relative difference between user models, before and after mapping using IDIUMS, is preserved.

The challenge motivating the creation of IDIUMS is that each adaptive application being considered here (an adaptive game, and a hypertext & assessment) models its user using a different set of attributes, primarily based on what is feasible for that application to measure about the user or the types of interaction the user has with the system. It solves this by performing a mapping of the user's model in one application, into an appropriate model, that still represents the user's ability, in the next application. It is therefore important to test that the user models produced by IDIUMS are still representative of the user.

5.1.1 Research Question

The research question, as defined in Chapter 1, motivating the experiment in this chapter is:

RQ1: Is it possible and meaningful to make rules to map attributes of a user model between different types of application?

Does the act of applying the mapping rules in IDIUMS produce user models that represent the student to the same extent they did when input into IDIUMS? Measuring how well the new user model still represents the user is done by measuring how close

Submit input UM data for	Request output UM data for
hypertext	game, assessment
game	hypertext, assessment
assessment	hypertext, game

Table 5.1: Input and output pairs for simulation.

to other user models it is before and after; the distance between models should remain proportionally the same after mapping. If it does not we assume the mapping has failed.

5.1.2 Structure of Chapter

This chapter explains the methodology used to simulate the use of IDIUMS, including how the virtual users were generated, the analysis performed to determine whether IDIUMS produces the expected output and the results of the simulation.

5.2 Methodology

The simulation randomly generates user models for a large number of virtual users of one of the three systems discussed in this thesis (adaptive hypertext, adaptive game and adaptive assessment), and uses the IDIUMS HTTP API to insert them into a version of IDIUMS that is setup with the same rules as in the experiment (see Chapter 6).

Then the simulation requests each virtual user's model in the context of the other two applications, and compares the resultant output user models to the input user model. This comparison is done by checking the relative position of each user model in multidimensional space; two user models that were close to each (similar) before would be expected to still be similar (close) after IDIUMS has mapped them onto different attributes.

This simulation was built in R, to easily enable statistical analysis of the results and generation of charts. To support this, a library of functions were written in R to encapsulate the HTTP communications with IDIUMS. For the purpose of reproducibility of the simulation, the random seed used to generate the results below has been set to 1.

5.2.1 Determining Whether Mapping Works

The goal of IDIUMS is to translate the attribute values within an adaptive application's user model to suitable values in another adaptive application. This should result in each adaptive application treating the user similarly, within the parameters of that

	Datatype	Minimum	Maximum	Constraints
hypertext				
visited	Integer	0		
knowledge	Double	0.0	100.0	
own_knowledge	Double	0.0	100.0	
assessment				
attempts	Integer	0		>= *correct*
correct	Integer	0		<= *attempts*
score	Integer	0		
duration	Double	0.0		
game				
attempts	Integer	0		>= *successes*
successes	Integer	0		<= *attempts*
collected	Integer	0		
movementCount	Integer	0		
loopCount	Integer	0		
statementCount	Integer	0		
(code)	String			

Table 5.2: Attributes in each each application's user model (attributes in parentheses were not used).

application. To enable the receiving application to treat the user in an appropriate way, it would be expected that the distribution of users, relative to each other, will remain proportionally similar to those user models in the originating application.

To demonstrate this within a simulation, the virtual user models before entering IDIUMS were plotted in multidimensional space (where each discrete or continuous numerical attribute), and the distance between each calculated and stored. This mapping and calculation was repeated for the user models that resulted from requesting the values for other applications from IDIUMS.

Two measurements were made to compare the before-IDIUMS state of the virtual user models to the after-IDIUMS state. Firstly, for the N virtual user models generated (N=1000 reported below), the Euclidean distance between each pair of virtual user models was calculated, and compared to the Euclidean distance between the same pair after the IDIUMS mapping. Secondly, each virtual user model had its Euclidean distance from the origin calculated and compared to its Euclidean distance from the origin after IDIUMS mapping.

The reason for avoiding direct comparison of the values in the user model before and after IDIUMS is that they are by definition incomparable: the assessment has 4 attributes per concept compared to the game's 6 and the hypertext's 3. Each attribute is

Rule	Variable Bindings		
	symbol	attribute	application
1. a = v	a v	attempts visited	game hypertext
2. a = k / 10	a k	attempts knowledge	game hypertext
3. s = v * k / 100	s v k	successes visited knowledge	game hypertext hypertext
4. a = v	a v	attempts visited	assessment hypertext
5. c = k / 10	c k	correct knowledge	assessment hypertext
6. s = k / 10	s k	score knowledge	assessment hypertext
7. A.a = G.a	A.a G.a	attempts attempts	assessment game
8. c = s	c s	correct successes	assessment game
9. A.s = G.s	A.s G.s	score successes	assessment game
10. v = a	v a	visited attempts	hypertext game
11. k = a * 10	k a	knowledge attempts	hypertext game
12. G.a = A.a	G.a A.a	attempts attempts	game assessment
13. s = c	s c	successes correct	ga assessment
14. G.s = A.s	G.s A.s	successes score	ga assessment
15. v = a	v a	visited attempts	hy assessment
16. k = c * 10	k c	knowledge correct	hy assessment
17. k = s * 10	k s	knowledge score	hypertext assessment

Table 5.3: The rules configured in this version of IDIUMS.

measured on quite different scales: hypertext.knowledge is a value from 0-100, whereas assessment.attempts is a positive integer that cannot be larger than assessment.correct, and within a user trial may not even make it into double figures.

To determine whether IDIUMS is performing its task as intended, we would expect the distances (between the pairs and from the origin) before passing through IDIUMS to closely correlate with the distances after IDIUMS has performed its mapping. The intention of this is to ensure that users are similarly clustered before and after using IDIUMS, with the corollary that the adaptive systems will treat similar users similarly after their user model has been through IDIUMS; for example, if they were treated as a novice in the initial application, they would be treated as a novice in the subsequent application.

5.2.1.1 Limitations

This approach to determining whether accurate user models are produced is a relativistic one: it measures IDIUMS' ability to produce a suitable user model by checking the user model relative to other user models before and after transformation. The reason this is a limitation is it does not account for the scale of values, and whether the new values make sense for the next application.

The impact of this is that if a rule in IDIUMS is badly constructed, it might produce models that consistently represent the users as less or more able than they actually are. This might result in the adaptive systems producing content that is too difficult or too easy for the user.

This is a matter of tuning the rules entered into an instance of IDIUMS, so is somewhat outside the scope of this particular simulation. This simulation still functions to test the underlying principle of IDIUMS. It is also expected that in practice (rather than simulation) the adaptive applications would update its model of the user through interactions with them.

5.2.2 Generation of Virtual Users

The user models for the virtual users were created by randomly generating values for the various attributes and concepts available for the application being simulated. To make these values realistic, the mean and standard deviation were calculated for all the concept-attribute pairs from the data in the pilot run of the experiment (Chapter 6). These statistics were then used to generate values from a normal distribution.

5.2.3 Constraints

The act of developing the simulation drew attention to some constraints inherent in the way values in the various applications' user models are generated; these constraints should be duplicated in the simulation.

5.2.3.1 Range of values

Certain attributes have boundaries. For example, in an assessment, the number of attempts cannot be negative, so it has a lower bound of zero; however, there is no realistic upper bound, except perhaps any maximum integer representation issues inherent in the assessment software. Another example is that the hypertext system used, by default, represents the user's knowledge of a certain concept as a value between 0 and 100. The simulation generates values for these attributes within the allowable ranges.

5.2.3.2 Relative constraints

There are some attributes that are tied to others. For instance, the number of questions correctly answered in an assessment, must be equal to or smaller than the number of questions asked. The virtual user model generation in this simulation implements these constraint, which are listed in Table 5.2.

5.3 Results

The result of using IDIUMS to transform user models was evaluated by correlating measurements about the user model before and after passing through IDIUMS. These measurements were (a) the Euclidean distances between each user model and every other user model, and (b) the Euclidean distance of each user model from the origin. Therefore the Pearson's product-moment correlation, its p-value and a scatterplot with a regression line, are presented here for both measurements (a) and (b) for each combination of input and output application.

5.3.1 Graphs

The results diagrams (Figure 5.1) show two kinds of plot. On one, the Euclidean distance between each pair of user models is plotted before-mapping against after-mapping. The second chart plots each input user model's Euclidean distance from the origin against the user model after mapping. Both types of chart include a regression line to demonstrate the correlation between the before and after user models.

		output		
		game	hypertext	assessment
input	assessment	$r_{ag_p} = -0.03$ $p = 0.03$	$r_{ah_p} = -0.03$ $p = 0.07$	-
input	hypertext	$r_{hg_p} = 0.95$ $p < 2.22 \times 10^{-16}$	-	$r_{ha_p} = 1.00$ $p < 2.22 \times 10^{-16}$
input	game	-	$r_{gh_p} = 1.00$ $p < 2.22 \times 10^{-16}$	$r_{ga_p} = 1.00$ $p < 2.22 \times 10^{-16}$

Table 5.4: Pearson's correlation of distance between pairs.

		output		
		game	hypertext	assessment
input	assessment	$r_{ag_o} = 0.14$ $p = 0.17$	$r_{ah_o} = 0.14$ $p = 0.16$	-
input	hypertext	$r_{hg_o} = 0.95$ $p < 2.22 \times 10^{-16}$	-	$r_{ha_o} = 1.00$ $p < 2.22 \times 10^{-16}$
input	game	-	$r_{gh_o} = 1.00$ $p < 2.22 \times 10^{-16}$	$r_{ga_o} = 1.00$ $p < 2.22 \times 10^{-16}$

Table 5.5: Pearson's correlation of distance from origin.

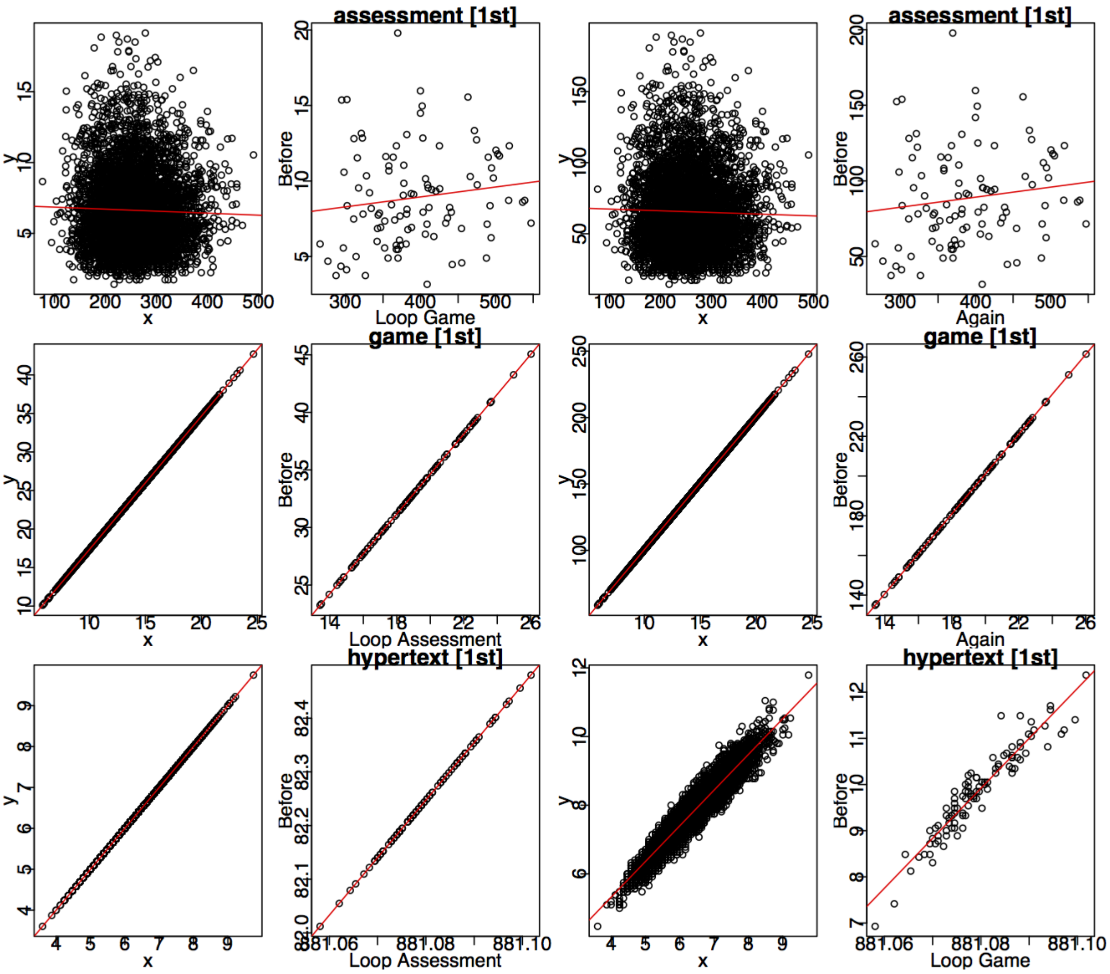


Figure 5.1: Simulation run on pilot experiment data.

5.4 Analysis

The results show very strong positive correlations between the virtual user's model before and after IDIUMS had performed its mappings, except in the case of mapping from the assessment.

The mappings for game-to-assessment, game-to-hypertext and hypertext-to-assessment are near perfect, with every point sitting on the regression line and a statistically significant correlation coefficient of 1 in all cases. This reflects the simplicity of the rules mapping between these applications.

Hypertext-to-game has a little more variation but this appears consistent across the range of user models. The significant, strong, positive correlation of 0.95 suggests that regardless of this variation the mapping is producing user models similar to those inputted.

		output		
		game	hypertext	assessment
input	assessment	$r_{ag'_p} = 0.89$ $p < 2.22 \times 10^{-16}$	$r_{ah'_p} = 0.88$ $p < 2.22 \times 10^{-16}$	-
input	hypertext	$r_{hg'_p} = 0.95$ $p < 2.22 \times 10^{-16}$	-	$r_{ha'_p} = 1.00$ $p < 2.22 \times 10^{-16}$
input	game	-	$r_{gh'_p} = 1.00$ $p < 2.22 \times 10^{-16}$	$r_{ga'_p} = 1.00$ $p < 2.22 \times 10^{-16}$

Table 5.6: Pearson's correlation of distance between pairs, without duration attribute.

5.4.1 Effect of duration attribute

The results above suggest that there is a problem when mapping user model data out of the assessment into the other applications, as there is no correlation between the user model before and after mapping. In the context that the correlations are so significantly positive for the other applications, it is not immediately obvious why the assessment mapping does not work, so further analysis was undertaken.

By systematically removing each attribute from the simulation in turn, it was discovered that the attribute that was having this large effect was duration.

Below are the results of the Pearson's product-moment correlation and the accompanying scatterplots, with the duration attribute removed, followed by a discussion of the cause of this effect and resultant decision regarding the simulation design.

With the duration attribute removed, the assessment-to-game and assessment-to-hypertext now correlate significantly between the user model before and after mapping both when comparing the distance between pairs of user models ($r_{ag'_p} = 0.89, r_{ah'_p} = 0.88$) and distance to the origin ($r_{ag'_o} = 0.87, r_{ah'_o} = 0.87$). The scatterplots (Figure 5.2) indicate there is a more pronounced displacement from the regression line around the smaller distances from the origin.

		output		
		game	hypertext	assessment
input	assessment	$r_{ag'_o} = 0.87$ $p < 2.22 \times 10^{-16}$	$r_{ah'_o} = 0.87$ $p < 2.22 \times 10^{-16}$	-
	hypertext	$r_{hg'_o} = 0.95$ $p < 2.22 \times 10^{-16}$	-	$r_{ha'_o} = 1.00$ $p < 2.22 \times 10^{-16}$
	game	-	$r_{gh'_o} = 1.00$ $p < 2.22 \times 10^{-16}$	$r_{ga'_o} = 1.00$ $p < 2.22 \times 10^{-16}$

Table 5.7: Pearson's correlation of distance from origin, without duration attribute.

5.4.1.1 Discussion

Comparing the correlation coefficient between the results without duration (Tables 5.6 and 5.7) and those with duration (Tables 5.4 and 5.5), demonstrates that including duration in the simulation shows no evidence of a correlation between user models mapped from the assessment to both the game and the hypertext, which contrasts with the significant positive correlation apparent with duration excluded. Considering why this is the case should provide more understanding of the IDIUMS ecosystem.

There are two factors contributing to why duration makes assessment user models not correlate with ones that IDIUMS map into the game or the hypertext: the values duration contains, and the rules set up within IDIUMS.

The duration attribute is a measure of how long a student took to answer a question in the assessment application, in seconds. Across all the different concepts in the domain model of the adaptive applications, the mean average duration varies between 36.5 seconds and 138.0 seconds. The averages for attempts, correct and score (the other attributes in the assessment user model), exist in the range 0.5 to 4.2. The scale of these values are vastly different, and as such if the duration does not correlate with any features the game or hypertext user model, it will have a large effect on the whole user model not correlating.

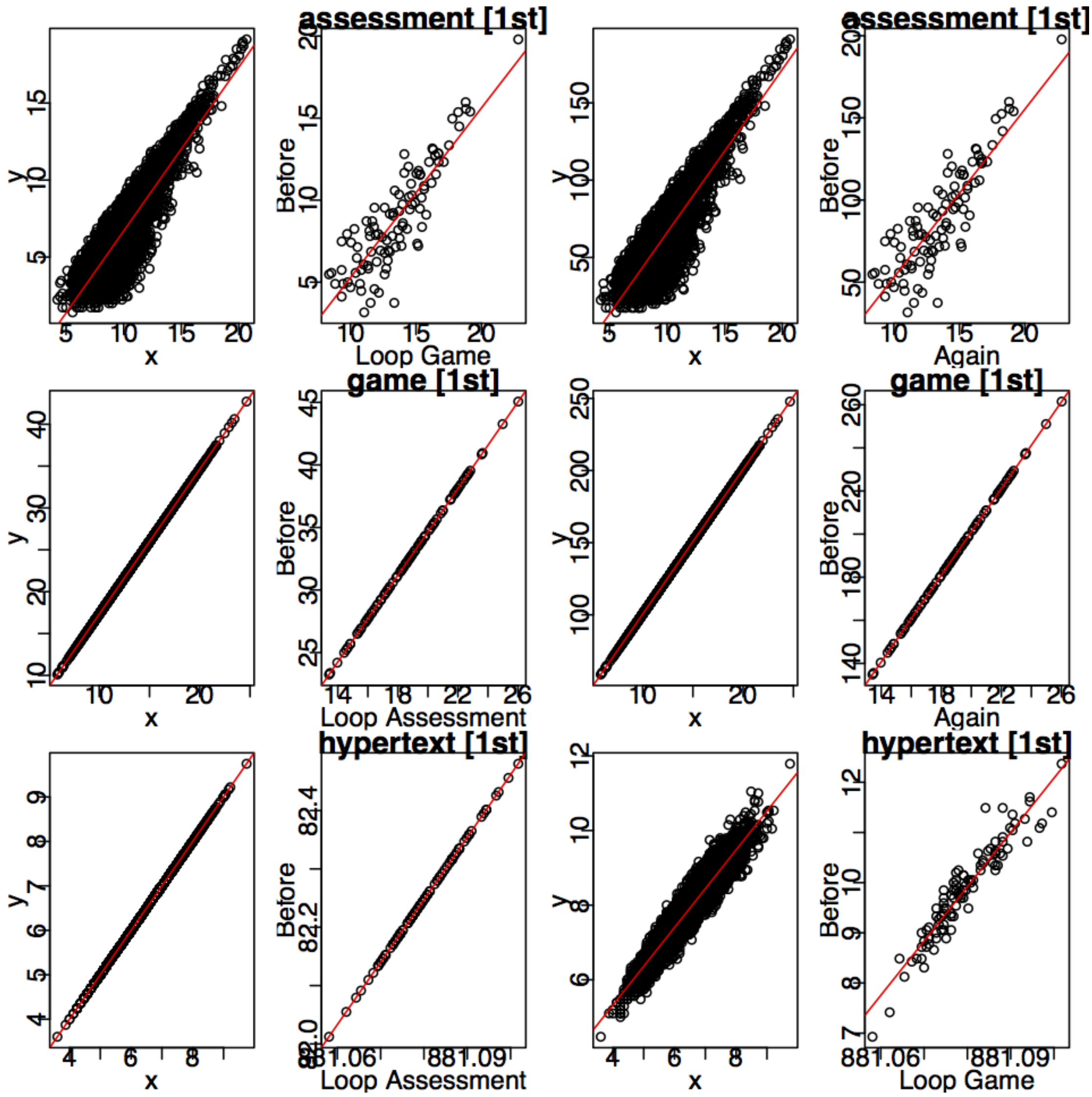


Figure 5.2: Simulation run on pilot experiment data.

The reason that the duration attribute does not seem to correlate with anything in the game or hypertext is that there is no rule mapping onto the duration attribute in the experimental instance of IDIUMS. No rule was created for duration because it is not currently used in the assessment engine's adaption algorithm, and arguably the attribute should not be registered with IDIUMS in the first place. The relationship between the time taken to answer a question and anything else why might know or need to know about a student is a hard one to generalise, and is often a function of the particular problem and the student.

The effect of these two factors is that the duration attribute has a significant effect on the measurements of the assessment user model, but this is not reflected in the user model's output for the game and the hypertext.

5.4.1.2 Overall

The strong positive correlations, all of which were found to be statistically significant, demonstrates that the user models produced by IDIUMS mapping are similar (in relation to other user models in the system) to those input into IDIUMS.

This chapter set out to answer the research question *RQ1: Is it possible and meaningful to make rules to map attributes of a user model between different types of application?* From the results of this simulation we can conclude the rule-based mapping does effectively translate attributes between applications, and that IDIUMS successfully implements this.

5.5 Conclusions

IDIUMS is a new piece of software and has the intention of mapping a student's user model from one application to another. However, before IDIUMS can be used in any experiments to test the effect on a student's adaptive experience, it is important to prove that IDIUMS is performing that mapping reliably and that the new user models are similarly distributed in the new application as they were in the previous application.

This chapter presented an experiment to simulate the real life use of IDIUMS by the generation of one thousand virtual user models, the mapping of these between each application, and the analysis of the resultant output user models.

Strong correlations between the relative positioning of input user models and output user models suggests that the IDIUMS software is performing the job it is supposed to do.

5.5.1 Future Work

The simulation process allowed some observations that might contribute to future work in the area.

In terms of provisioning IDIUMS, it seems some thought needs to be given to the policy for including attributes in the IDIUMS user model: Should an an application register an attribute if it does not actually use it for adaptation? What is the effect of orphaned attributes on the ecosystem? Should such attributes be labelled or highlighted in some way (both to attract mappings and to signal to receiving applications what the intent of this data is).

The realisation that the duration attribute was not being mapped raises the complex question of how timing data relates to other attributes. This is an attribute that is likely to be a differentiator between students, and is relatively easy to measure, but its meaning is strongly context dependent. However, to support this in mapping rules for IDIUMS, there will need to be an investigation into how timing data is used for making educational adaptations, and some design work to add features to IDIUMS' rule engine. To support rules that compare user model values to some aggregation of an attribute across all users (for example, comparing the current user's duration to the average of all durations in the IDIUMS database) would require a new syntax for rule definitions that supports aggregation functions, and implementation within the IDIUMS rule engine for calculating the results of these. Investigation into to how rules of this sort would be used by rule authors would be required, to determine whether aggregation would need to be performed over all users or just the current user, and for a specified concept or all concepts.

Chapter 6

User Trial

This chapter presents the experiment that tests the main research question of this PhD: is it meaningful to share data between adaptive applications which interact with the user in different ways? The experiment is a user trial of the IDIUMS system's ability to share user model data between adaptive applications, with the goal of allowing those applications to provide more appropriate adaptations from the very start of their use.

IDIUMS is a user modelling system designed to act as an intermediary between different types of adaptive applications; its motivation, design and reference implementation is described in detail in Chapter 4. Traditionally, each of these adaptive applications stores its own model of the user, which means when a user starts using a new application, that application knows nothing about the user. This is known as the "bootstrapping" problem. Even if applications have some mechanism for sharing their user model data with other applications, different categories of application (for example hypertexts, assessments, games and recommender systems) have different types of interaction with the user, and as a result they model the user using different attributes. Where they do share information it is often quite basic information such as using common usernames and password to log in to related systems. IDIUMS attempts to provide a solution to this problem by providing a service which enables translations between those attributes, while sharing data between applications.

The experiment presented in this chapter investigates whether the IDIUMS user model system is effective in translating values in one adaptive application to a form that is interpretable in another application. IDIUMS provides a way for (typically educational) adaptive systems with no data about the user's current knowledge of a topic, to infer this from data stored in another adaptive system about that user's knowledge of a topic.

The hypotheses being tested is that it is meaningful to convert between values in multiple different systems, and that the effect of having rules between multiple applications is beneficial in adapting to the user. The proposed experiment will show that sharing data, based on all of the attributes an adaptive system can measure, is better than not sharing any data at all.

This experiment was performed by gathering real user data in two adaptive environments (an adaptive hypertext and assessment, and an adaptive game) and surveying the users about the appropriateness of the adaptations at various stages. The educational content of all three systems will be the same: the programming concept of loops, specifically in the Java programming language.

This chapter will explain the hypotheses being tested by this experiment, describe the experimental design and methodology, and introduce the analyses that will be performed before presenting the results. It will end with a discussion of the meaning of the results, and offer conclusions that can be taken from them.

6.1 Research Questions

The experiment tests two research questions, as defined in Chapter 1.

RQ1: Is it possible and meaningful to make rules to map attributes of a user model between different types of application?

RQ2: Can mapping from an attribute in one application to an attribute in another application create adaptations that are perceived by the user as being at an appropriate level?

A key claim behind this research is that different types of adaptive application have unique ways of modeling their users, because of the unique interaction methods associated with each type of application. Within the popular overlay user model, the data to model the user is stored as values associated with each of the attributes the application can measure. Within a single application, adaption rules may use the value of one attribute to infer the value of another. This research extrapolates that idea to infer the value of an attribute in application A, based on the value of attribute(s) in application B (and C and so on).

The first hypothesis therefore makes two claims. Firstly, that it is possible to define a rule that will translate between attributes in different adaptive applications. Secondly, that the mapping from an attribute in application A to an attribute in application B can be meaningful (in that it creates adaptations that are perceived as useful by the user).

6.2 Experimental Design

An experiment was designed and prepared that would gather data to test the hypothesis. The experiment gathers data about users' interactions with two applications and IDIUMS, their perception of their own knowledge and their perception of whether the applications were adapting appropriately to their ability.

This experiment involves two adaptive applications:

1. An adaptive hypertext, authored in GALE (an adaptive hypertext engine developed by TU/e as part of the GRAPPLE EU project), with an associated adaptive assessment, created using QTIengine to render the questions.
2. An adaptive programming puzzle game, inspired by Boyle's (2006) While loops learning object.

The adaptive assessment and the adaptive game were built from scratch for this experiment, while the adaptive hypertext was authored specifically for this experiment, but delivered through GALE.

The content in each of these adaptive applications was targeted at teaching Java loops to novice programmers at university level.

The meaningfulness of the use of IDIUMS will be measured by controlling for its influence on the participant's experience. Four samples will be gathered, two of which will be from participants using an adaptive application from a cold start, and the other two will be when the participants use an adaptive application following a different adaptive application, with IDIUMS mapping the user model between them. Participants will be pseudorandomly allocated to these conditions.

The data produced will be from the applications' and IDIUMS' own databases, and the results of questionnaires completed by each participant following their use of each adaptive application.

6.2.1 The Applications

6.2.1.1 Java Loops Adaptive Educational Hypertext and Assessment

The first learning environment used in this experiment is an adaptive hypertext and assessment. This is actually made up of two applications, that were presented to the participants in order (hypertext then assessment), partly down to a technical limitation with the hypertext.

It proved technically challenging to extract or inject data from or into the hypertext's user model, despite a number of approaches being considered. This hindered the option to test the hypertext as a separate application interfacing with IDIUMS. Without the hypertext it would leave a situation where the experimental orders are only Game → Assessment and Assessment → Game.

It was deemed unrealistic for a student to immediately perform an assessment on a topic that they are new to, and a far more common scenario would be for a student to use some learning tool in preparation for an assessment. Therefore the decision was made to combine the hypertext and assessment so that the participant performed some learning before being assessed, with their user model from this assessment subsequently submitted to IDIUMS.

The Hypertext part of this environment built in the GALE adaptive hypertext system. It provides textual pages describing each of the concepts in the domain of loop structures in Java, and automatically decides which links are appropriate to highlight, based on the contents of the user model and the relationships between each concept (the domain model).

For every concept in the domain model, a number of attributes are defined by GALE. For a particular user, each of these attributes will store a value (Integer, Boolean or String), based on the participant's use of the system.

In the adaptive hypertext, the user model being constructed is knowledge attained and visits. Visits is an integer count from zero upwards, of the number of times the user has accessed the concept in question. Knowledge is a floating point measure of the user's knowledge of the concept, rated between 0 and 100, calculated based on the number of visits (increasing by 10 each visit).

The Assessment part of this environment is a computer-aided assessment consisting of questions of varying difficulty addressing each of the concepts in the Java loops domain model (as used in the other applications). Questions are rendered by (and responses are processed by) the QTIengine software, via its REST interface.

The adaptive assessment records the number of attempts at a question, the score resulting from those attempts and the duration of time taken to answer each question. The attempts is a positive integer count for the number of times the user has answered a question. The score is an integer measure of whether they answered the question correctly (some questions can be allocated multiple marks). The duration is a floating point measure of the number of seconds taken to answer the question.

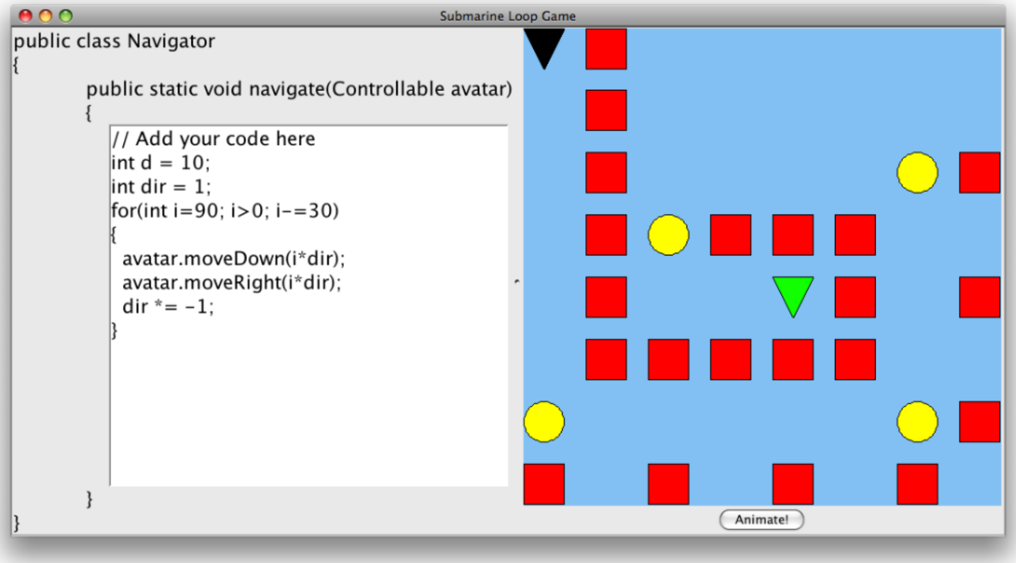


Figure 6.1: Screenshot of the adaptive game running

6.2.1.2 Java Loops Adaptive Educational Game

The Java Loops Adaptive Educational Game is an interactive programming challenge, where the player is tasked with navigating a submarine through an underwater scenario, using Java loops.

The adaptive game stores a value for the number of attempts and successes at game levels related to each concept. It also stores a duration for the amount of time taken to complete these levels. The attempts and successes are positive integer counts for the number of times the player has played and completed a level. The duration is an integer count of the number of milliseconds between the player starting and finishing (either through failure, exiting or winning).

6.2.1.3 IDIUMS

Acting as the user model for both the assessment and the game is the shared user modelling system described in Chapter 4, called IDIUMS. For the purposes of this experiment, IDIUMS was setup with the same rules as described in Table 5.3 in Chapter 5, which are based on the application authors' understanding of how each application works.

Limitation It was not possible to connect GALE into IDIUMS. While the GRAPPLE architecture (Kravcik et al., 2008), of which GALE is a part, has an event bus to allow third party applications to communicate with various parts of the system (including the user model), the protocol for interacting with the GRAPPLE event bus, and the development environment required to do so, was poorly documented. A workaround, which read from and wrote to the GALE MySQL database was developed, but in testing this it was discovered that a live instance of GALE does not appear to access this database when running, presumably because it has an in-memory representation, which meant any changes pushed to the database by IDIUMS were ignored in the actual hypertext application.

6.2.2 Experimental Procedure

The experiment asked users to login and answer a survey rating their self-perceived knowledge of various concepts related to Java loops. Following this they used each adaptive application in turn, in a pseudorandomly chosen order, for 10-15 minutes. Following their use of each application, they answered questions about the level of content at various stages during their usage of that application, and again their self-rated understanding of various concepts related to Java loops.

6.2.3 The Participants

To generate real data from these three adaptive systems, a number of students were asked to use each of the systems. The intention was to construct a realistic learning experience in this domain, using these tools. Therefore, the users will ideally be in a situation where they have not learned the loop control structure in Java yet, or have only recently been introduced to it.

This experiment was run three times, using different cohorts of students from modules taught by the author.

The first time was in October 2012 and was run during the *Ground Controllers* support session of the *COMP1202 Programming 1* module of the Computer Science degree at the University of Southampton. This module teaches the principles of programming through the medium of the programming language Java. The experiment was run in Week 4 of this module, which is when the students are taught the concept of looping. As the students were in an optional support session and were only just learning about loops, they were deemed to have the most scope for improvement, which should make any learning gain more detectable.

The second time was in May 2013 with a cohort of students studying the *COMP1056 Web Design* module of the Information Technology in Organisations degree (ITO) at University of Southampton. This module taught programming using the JavaScript scripting language.

	Application 1	IDIUMS	Application 2
Order 1	Hypertext & Assessment	$\rightarrow$	Game
Order 2	Game	$\rightarrow$	Hypertext & Assessment

Table 6.1: The potential routes through the user trial a participant might be allocated.

The third time was in November 2013. This cohort was similar to the first time (Week 4 of Programming 1 on Computer Science degree), but the following academic year, so had new students.

The participants' anonymity was assured by randomly generating and allocating login details (user name and password) for each participant, so that there is no personally identifiable information in the username or password. As the hypotheses being tested do not make reference to particular characteristics (such as gender or age), these were not recorded, reducing the risk of any participant being identified in relation to their data.

6.2.4 Order of Use

To test the set of all possible orders of use within the constraints of this experiment, as well as assuming each application is used only once, each user was allocated one of the following orders of application use detailed in Table 6.1. Between each use of an application, their user data from the preceding application was translated by IDIUMS, where possible, for use in the following application.

6.2.5 Time

The experiment was conducted in one session, using a web interface to guide the user to each of the applications in turn. The user was expected to use each application for 10-15 minutes, and there was a countdown at the top of the screen to remind them when to move on.

In addition to using the applications, the participants were asked to complete a survey where they rate their own knowledge of the various concepts represented in the applications. This survey was conducted before the experiment started and immediately after each use of an application (four times in total). This data provides a ground truth to compare with the values generated by the applications and IDIUMS. An example of the survey is in Appendix A.

The expected total time commitment from a participant was 1 hour.

6.2.6 Control

To show that IDIUMS allows a new application to adapt better than if it were cold-starting, this experiment requires a control. Whereas IDIUMS provides the currently-used adaptive application with user model values translated from the values already in IDIUMS (from the previously used applications), the control set of participants must be given some values other than those generated by IDIUMS to detect whether using IDIUMS produces the intended effect (better adaptations).

As IDIUMS is attempting to combat the bootstrap problem (where an application assumes the user knows nothing, if they have never used it before), it makes sense to compare it to that situation. Therefore, the participants in the control group will have each application fed with user model values as if they had never used any adaptive application before. For all the attributes used in this experiment this value will be zero, as all the attribute values increase with use.

A experimental group will be asked to use one application, then the other (with IDIUMS translating their user model between applications), and data recorded about their use of the second system. The control group data will be derived from participants use of the first application, at which point the system is definitely in cold-start mode as nobody has used the system yet.

6.2.7 Data

The data generated by the participants' use of the adaptive systems is stored in the databases attached to each application, which runs on a password protected virtual machine. Data was also transferred to the principal investigator's machine for analysis and backup. This computer is also password protected.

The anonymised data was retained and can be found in the accompanying digital storage media, and will provided alongside this document as open data so that future researchers can verify the analysis performed.

6.2.8 Expected Outcomes

The purpose of this experiment was to show that using an external rule-based adaptive engine to convert between the attributes used to store user data in one application and the attributes in another application, enables the application receiving user data to make more appropriate adaptations to the user. Therefore if the hypotheses underlying this work are true, it is expected that the users receiving adaptations based on data generated from the rules in IDIUMS will rate the appropriateness of the materials (assessment questions and game levels) presented to them higher than those in the control group.

- *"At the start of using the Java loops hypertext & assessment how would you rate the difficulty of the questions?"*
- *"At the start of using the Java loops game how would you rate the difficulty of the levels?"*

Figure 6.2: Questions asked at the end of each application usage.

6.3 Analysis

The pilot run of the experiment identified a mistake in the configuration of the rules within IDIUMS, so the only analysis of the data in the pilot was on the initial questionnaire to produce some demographic information about the students taking part. This was corrected in two subsequent runs of the experiment.

Two approaches have been taken to analyse the data generated from this experiment and determine whether the hypotheses hold true. These analyses were repeated for both the second and third run of the experiment.

Firstly, to answer the main hypothesis of this experiment, four samples were produced that include both uses of adaptive applications from cold-start and uses of adaptive applications having had their user model modified by IDIUMS. These samples were then compared to determine whether the use of IDIUMS allowed applications to adapt their content more appropriately, as perceived by the user.

Secondly, for each attribute in each application, the relationship was calculated between the value stored in IDIUMS and the user's self-rated understanding for each concept, with the expectation that there would be a correlation.

6.3.1 Samples

Two analyses have been performed on the data gathered from the user trial. Firstly, the level of adaptation perceived by the user in each application is compared between the sample of those users who were using that application from a cold-start and the users whose user model had been informed by IDIUMS. Table 6.2 puts this in more concrete terms. This takes the participant's answer to the questions in Figure 6.2 and tests the null hypothesis that the two samples come from the same population.

These questions were answered on a scale of 1 (one) to 5 (five):

1. One - Have not heard of this concept.
2. Two - Do not know what this concept means.

	Source of user model		Measuring user perception of level in
Sample 1:	cold-start	→	Hypertext & Assessment
	compared with		
Sample 2:	Game	→	Hypertext & Assessment
Sample 3:	cold-start	→	Game
	compared with		
Sample 4:	Hypertext & Assessment	→	Game

Table 6.2: Samples gathered during the user trial.

3. Three - Understand what it means but cannot use it.

4. Four - Can use it, but may make mistakes.

5. Five - Fully understand this concept and can use it correctly.

If the use of IDIUMS is an improvement over a cold-start, then we would expect to reject the null hypothesis.

The second analysis performed is a calculation of the relationship between the values recorded or calculated by IDIUMS about the user's ability, and the user's own self-rating of their understanding. This is done by testing if there is a correlation between the IDIUMS value and the self-rating, for each concept that exists within the domain model.

6.4 Results

The experiment described above was run three times: October 2012 (with Computer Science first year undergraduates), May 2013 (with ITO first year undergraduates) and November 2013 (with Computer Science first year undergraduates).

The results from October 2012 were discarded due to problems with the configuration of the rules within IDIUMS. All analysis is based on the two subsequent runs of the experiment.

6.4.1 The May 2013 Experiment

6.4.1.1 Differentiating samples by difficulty of content at start of application

After the receiving application in each of the four samples (either the one immediately after cold-start, or the one that has received user model data from IDIUMS), the user

Sample	Statistic, W	p-value
cold-start → Hypertext & Assessment	0.630	0.001
Game → Hypertext & Assessment	Not enough samples	
cold-start → Game	0.961	0.814
Hypertext & Assessment → Game	0.895	0.406

Table 6.3: Shapiro-Wilk normality test results for the samples in May 2013

Samples	Statistic, W	p-value
cold-start → Hypertext & Assessment Game → Hypertext & Assessment	4.5	1

Table 6.4: Wilcoxon rank sum test results for the samples in May 2013

Samples	Statistic, t	p-value
cold-start → Game Hypertext & Assessment → Game	-1.050	0.333

Table 6.5: Student's t-test results for the samples in May 2013

was asked the questions in Figure 6.2. This analysis seeks to determine whether there is a significant difference between Sample 1 & 2, and between Sample 3 & 4.

Test for normality All four samples (see Table 6.2) were tested first for normality, to determine the appropriate statistical method to use for testing whether they were from the same population. The results of the Shapiro-Wilk normality test can be found in Table 6.3. The null hypothesis is that the data are from a normal distribution, so a significant result (p-value is less than or equal to 0.05) will indicate that the data is non-normal.

The samples resulting from use of the Hypertext & Assessment have a p-value less than 0.05, which means the null hypothesis is rejected, asserting that the data are not normally distributed. However, the samples for using the Game have a p-value greater than 0.05, so there is a failure to reject the null hypothesis; as the data is not confirmed as not-normal, a Student's t-test will be performed.

Test for different populations In the Hypertext & Assessment condition, the samples are not normally distributed, so a Wilcoxon rank sum test was performed to calculate the likelihood of the two samples being from the same population. A Student's t-test was performed on the participant data from users of the Game. The results can be seen in Table 6.4 and Table 6.5.

Sample	Attribute	Pearson correlation, r	Sig.	N
cold-start → Hypertext & Assessment	attempts	0.145	0.621	14
	correct	-0.338	0.238	14
	score	-0.232	0.427	14
	duration	0.122	0.818	6
Hypertext & Assessment → Game	attempts	-0.055	0.852	14
	successes	-0.375	0.186	14
cold-start → Game	attempts	-0.457	0.157	11
	successes	-0.330	0.322	11
cold-start → Game	attempts	0.622	0.187	6
	successes	0.680	0.137	6
Game → Hypertext & Assessment	attempts	0.962	0.002	6
	correct	0.962	0.002	6
	score	0.962	0.002	6

Table 6.6: Pearson's product-moment correlation results for the samples in May 2013

In both hypothesis tests, the p-values are larger than the alpha level of 0.05, so the user trial has failed to reject the null hypothesis (that both samples are from the same population), so no conclusion can be drawn about the effect of using IDIUMS had on this population.

6.4.1.2 Correlation between user self-rated understanding and IDIUMS value

To help provide information for better rule design, and to inform the investigation into why there was no difference in the populations, the level of correlation was calculated between the values held by IDIUMS for each concept, and the user's self rating of knowledge, following their use of the receiving system (either the one immediately after cold-start, or the one that has received user model data from IDIUMS).

The only instances where the data in IDIUMS correlates (at the 99% confidence level) with the participant's own self-rating of understanding is in each of the attributes, attempts ($r = 0.962$, $p = 0.002$), correct ($r = 0.962$, $p = 0.002$), and score ($r = 0.962$, $p = 0.002$), of the Hypertext & Assessment after IDIUMS has mapped the user's model from the Game.

All other results were extremely weak correlations, or there were not enough pairs of data to calculate a correlation.

Sample	Statistic, W	p-value
cold-start → Hypertext & Assessment	0.866	0.212
Game → Hypertext & Assessment	0.964	0.637
cold-start → Game	0.859	0.029
Hypertext & Assessment → Game	0.770	0.031

Table 6.7: Shapiro-Wilk normality test results for the samples in November 2013

Samples	Statistic, t	p-value
cold-start → Hypertext & Assessment Game → Hypertext & Assessment	-0.892	0.450

Table 6.8: Student's t-test results for the samples in November 2013

6.4.2 The November 2013 Experiment

6.4.2.1 Differentiating samples by difficulty of content at start of application

This analysis seeks to determine whether there is a significant difference between Sample 1 & 2, and between Sample 3 & 4 (see Table 6.2), which would suggest that the use of IDIUMS has had an impact on the experience of the users.

Test for normality The null hypothesis of the Shapiro-Wilk test is that the data are normally distributed. Therefore the test is seeking to find significant evidence to reject this hypothesis, indicating that the data is not-normally distributed and that a non-parametric test must be used to determine whether the two samples are from the same population. The result of a Shapiro-Wilk normality test can be seen in Table 6.7.

The two samples where the user was surveyed after use of the Game have a p-value less than 0.05, which means the null hypothesis is rejected suggesting that the data are *not* normally distributed. However, both of the samples in which the user was surveyed following the use of the Hypertext & Assessment have a p-value greater than 0.05, so there is a failure to reject the null hypothesis; due to this result, a Students t-test will be used to determine if the samples are from different populations.

Test for different populations In both hypothesis tests, the p-values are larger than the alpha level of 0.05, so the user trial has failed to reject the null hypothesis (both samples are from the same population), so no conclusion can be drawn about the effect of using IDIUMS had on the users' perception of the difficulty.

Samples	Statistic, W	p-value
cold-start → Game Hypertext & Assessment → Game	41.5	1

Table 6.9: Wilcoxon rank sum test results for the samples in November 2013

Sample	Attribute	Pearson correlation, r	Sig.	N
cold-start → Hypertext & Assessment	attempts	-0.100	0.714	16
	correct	0.131	0.627	16
	score	0.155	0.567	16
	duration	0.168	0.533	16
Game → Hypertext & Assessment	attempts	0.316	0.541	6
	correct	0.250	0.633	6
	score	0.086	0.872	6
	duration	-0.138	0.794	6
cold-start → Game	attempts	0.645	0.239	5
	successes	0.645	0.239	5
	collected	0.408	0.495	5
	movementCount	0.577	0.230	6
	loopCount	0.408	0.495	5
	statementCount	0.408	0.495	5
Hypertext & Assessment → Game	attempts	-0.071	0.894	6
	successes	0.120	0.822	6
	collected	NA	NA	3
	movementCount	NA	NA	3
	loopCount	NA	NA	3
	statementCount	NA	NA	3

Table 6.10: Pearson's product-moment correlation results for the samples in November 2013

6.4.2.2 Correlation between user self-rated understanding and IDIUMS value

The expectation of this analysis was that the data in IDIUMS and the user's self-rating of understanding, at the same point in the experiment, would show some correlation, however none of these tests reveal any correlations that are statistically significant.

6.5 Discussion

Analysis of the results in the previous section suggest that the expected outcomes of this experiment did not occur. When learners rate the appropriateness of the adapted content, it makes no difference whether the adaptive system was fed with data from IDIUMS (and therefore derived from another application) or whether they had a cold start in that application.

One potential reason, given a research software project underlies this experiment, is that the IDIUMS software was not working as expected. Any bugs within the system would potentially provide an application with incorrect data about the user, and therefore could skew the experience the user gets. However, as the system was successfully simulated in Chapter 5, there is reasonable confidence that a bug is not the cause of this result.

It is possible that the choice of evaluation criteria were inappropriate, and may never have been able to provide the result expected. This experiment asked the participant to rate their understanding of each concept being taught. However, it is feasible to believe that students in their first year of learning a topic might not be experienced enough to judge their own understanding of those topics. However, the meta-review conducted in Chapter 3 suggests that the methodology used in this thesis is consistent of those used in this research field.

As seen in the literature review, there are several different approaches to evaluating adaptive systems used by the community, but perhaps some of the alternatives would have been a more appropriate evaluation of IDIUMS effects than the one performed above. A review of existing evaluations would need to be performed to find this out.

IDIUMS relies on the authors of new adaptive applications to register rules that map the attributes of existing applications onto those of the new application. This proposed process requires that the author have some understanding of the range of possible values and the behaviour of those attributes during use of the application. From that, they are expected to define an appropriate set of rules to map from one or more application to their own. While in itself it may be contentious to suggest that application authors would know enough about someone else's application to define sensible rules, that is not what was being tested here. It does however raise the question of whether the rules were correct in this instance. The author of the rules was also the creator of both the game and the assessment, as well as the author of the content of the hypertext; if there was ever a scenario where this author-created rule mapping should work, it is this. A suggested way of answering this question would be to use data mining techniques to learn the mapping rules, based on the data collected through a user trial (probably a new trial designed for this purpose, as existing data is somewhat influenced by IDIUMS' existing rules), and comparing them to the author-created rules. Unfortunately, this fell outside the scope of this thesis, but would be an excellent starting point for future work.

Finally there is the adaptive applications themselves. It was assumed that they work and adapt as expected. While built to best principles of adaptive applications, it is feasible that IDIUMS is functioning correctly but the applications are not adapting appropriately to the information they are receiving from IDIUMS.

In future work, this could be mitigated by running a user trial for each application individually, to ensure each is meeting the user's level and how rapidly it is achieving that. Alternatively, existing applications that have already been evaluated and shown to work could be adapted for use in evaluating IDIUMS.

6.6 Conclusions

This chapter has described an experiment designed and performed to gather data to test the hypothesis that authored rules for mapping between the attributes of interactionally-diverse adaptive applications allows those applications to provide more appropriate content for users when starting a new application. This was done by running a user trial which asked participants to use two bespoke adaptive applications, some backed by IDIUMS and some not, and surveyed them as to the appropriateness of the content in each application and a self-rating of their knowledge of each concept in the domain model.

The data gathered from this experiment was analysed in two ways. Firstly, samples of users' measure of content appropriateness were taken for those situations where the application had user model data provided by IDIUMS, and those situations where it had a cold start. These samples were compared to attempt to show that these samples come from significantly different populations. Secondly, the relationship between users' self-rating of their knowledge of each concept and the values within IDIUMS at that point in time was analysed to determine whether there was a correlation between any of the attributes and the users' ratings.

The results of the comparison of samples suggest that they are both from the same population. This implies that there is no effect of using IDIUMS to bootstrap applications, as the difference in content presented is not significant enough for users to notice.

The measurement of relationships between attributes and user ratings demonstrated only one significant correlation: the attributes *attempts*, *correct* and *score* in the adaptive assessment, after the user had used the game and the hypertext & assessment, and that was only apparent in the first full run of the experiment.

Reasons suggested for these negative results include using an inappropriate evaluation criteria, errors in the construction and configuration of IDIUMS (which seems unlikely given the simulation and pilot test) and that the applications are not adapting appropriately to the data received.

Chapter 7

Conclusion

This thesis proposes an approach to solving the cold-start problem faced by adaptive applications (where there is no user model in place when the user first interacts with the system), by introducing an intermediary user modelling service called IDIUMS, that uses authored rules to map values from the user model of one adaptive application, into values in the user model of a different adaptive application.

The work performed here included the design and implementation of an intermediary user modelling system, a simulation and user trial to test the performance of that system, and a systematic review of the literature, in order to address the three research questions laid out in the introduction.

7.1 Addressing the Research Questions

At the start of this thesis, three core research questions were set out, to guide this work. Each of these is addressed below using the evidence gathered from the research work that has been presented in this thesis.

7.1.1 RQ1: Is it possible and meaningful to make rules to map attributes of a user model between different types of application?

The approach of having authored rules for mapping between attributes of user models is a novel one. However, this meant the premise that had to be evaluated as part of this research, and three parts of this thesis contribute to answering the question.

Firstly, an initial prototype determined what attributes are stored by user models in common types of adaptive educational system, and that it would be possible to design rules to map between many of these. This led onto the design for a software system, IDIUMS, that would perform this mapping, and a reference implementation of that system. See Chapter 4 for more information.

Secondly, a simulation of IDIUMS' functionality was performed to ensure that user models output by IDIUMS are still as representative of the user as the ones input, at least in relation to each other (Chapter 5). This simulation demonstrated that largely IDIUMS performs its function very well, resulting in new user models that are similarly distributed. However, it also identified that some attributes can have a significantly negative impact on these output user models, particularly if there are no rules mapping them to other attributes. The simulation showed that if these attributes are removed from the system, then mapping was successful across all cases tested.

The last piece of work that contributes to answering this question, was the user trial, as described in Chapter 6, that measured the user's perception of adaptation, having used IDIUMS to provide their initial user model. However, this trial found that the use of IDIUMS resulted in no significant change in adaptive applications being able to provide content at a level appropriate for the user.

These first two points suggest that it is possible to map attributes between different types of application. It is not clear, however, from the result of the user trial, whether it is meaningful to do this mapping because there was no evident improvement in the level of the content provided by the adaptive applications.

7.1.2 RQ2: Can mapping from an attribute in one application to an attribute in another application create adaptations that are perceived by the user as being at an appropriate level?

The goal of shared user modelling is to allow adaptive applications to instantly adapt their content appropriately for a new user, on the proviso they have used a different application. To find out whether this mapping approach allows this to happen, the result of the adaptations performed by applications must be measured.

This question was addressed by performing the user trial in Chapter 6. This experiment found that the intervention of IDIUMS had no significant impact on participants rating the level of content within an adaptive game and an adaptive hypertext & assessment.

While this might suggest that this sort of mapping is inappropriate for allowing an adaptive application to provide content at an appropriate level and avoid the cold-start problem, it does not conclusively show this. The simulation demonstrates that IDIUMS functions as expected, which means that either the methodology used to run the user trial was not sensitive enough to detect a change, or the adaptive applications did not adapt appropriately to the new user models. This thesis proceeds by questioning whether using a methodology that asks the participant to rate (directly or indirectly) the performance of the system is open to bias or the observer effect. This led to the addition of a third research question.

7.1.3 RQ3: What is the appropriate method for evaluating shared user modelling for adaptive educational systems?

To answer the first two research questions, a methodology needed to be devised. This was approached as two sub-questions.

7.1.3.1 RQ3a: What is the range of evaluation techniques used within the adaptive educational systems community?

The first issue in dissecting whether a different evaluation would have been more appropriate is to survey the full range of evaluations used within the community. This was done by performing a meta-review the 55 of papers about adaptive educational systems that were published in UMAP, PALE and AIED during the time period 2011-2015.

In analysing these papers, a categorisation system was developed to group papers by their evaluation method, and this was refined to being a categorisation of the sources of data used to perform the evaluation.

Further analysis demonstrated that the largest category was that where data was gathered from an Expert source, typically through a pre-post test but occasionally via an expert review. This was closely followed by papers that based their evaluation on data gathered directly from the participant, largely through questionnaires or surveys. Other categories of sources of data included system logs, computation within the system itself, and baselines from the literature or elsewhere.

7.1.3.2 RQ3b: Are particular types of evaluation prone to particular influences or biases?

The analysis of the meta-review in Chapter 3 identified that some of the categories defined may be open to influence or bias.

Of primary concern is the Student category of evaluation data sources, which measure their system's performance by asking the participant to rate it in some way. This approach is susceptible to the Hawthorne effect, if the participant knows they are rating the system they have just used, and has other potential sources of inconsistencies due to different interpretation of the questions or scales by the participants.

The largest category of evaluation data sources, Expert, counters these biases somewhat by having a consistent, independent expert make the judgements on the performance of the system. The other categories (Logs, Baseline, System), which are direct measures of the user actions or system computation are used far less than Expert and Student.

The conclusion from this meta-review, and response to the research question here, is that evaluations of Adaptive Educational Systems should, at the least, be conducted using a method that tests learning directly and independently. Other data sources, in combination, make these evaluations richer, but the basic measure should be one that measures learning.

7.2 Contributions

In answering the research questions, a number of pieces of work have been conducted resulting in a number of contributions to the Adaptive Educational Systems area of research.

1. After identifying that there is an issue with sharing user model data between applications with different interaction mechanisms, a design for a system that would map between the attributes of these applications was specified and a reference implementation built. This reference implementation is available for others in the community to extend, modify or even just use as an overlay user model in their own work.
2. A process for simulating shared user modelling systems to test their functionality was devised and used to show that the system described here works as expected.
3. This thesis contributes a systematic review of the Adaptive Educational Systems research literature to discover what evaluation techniques are most appropriate. This both informs new researchers of the possible avenues they might follow for evaluating their systems, and highlights to existing researchers that some of methods are potentially liable to bias and influence, so encourages them to consider one of the other methods available.
4. A user trial conducted using the IDIUMS system identified that in its current configuration the mapping approach does not have a significant impact on the appropriateness of the level of adapted content, as perceived by the user. Given the simulation demonstrated IDIUMS functions, and the meta-review indicates the evaluation was one of the most commonly used in the community, this results in a new potential direction for researching whether applications need to adapt differently to a hot-start as opposed to a cold-start.

7.3 Future Work

The act of answering the research questions in this thesis has inspired, that can be broadly categorised as work on evaluating rule mapping, other avenues for reviewing the literature and further work on the design and implementation of IDIUMS.

7.3.1 Shared User Modelling

Cold-start vs hot-start The combined results of the user trial, simulation and meta-review leaves the adaptive application as the potential cause of the users not perceiving any difference in adaptation. This raises the question of whether adaptive applications require a different design for handling cold-start and hot-start scenarios, which is an avenue of research as yet unstudied.

Run user trial with learning gain Needless to say, that more work on this thread of research would involve repeating the user trial but with a measure of true learning gain, rather than asking the user to rate the adaptivity or estimate their understanding of the educational concepts.

7.3.2 Further Reviews of the AES Literature

Reviewing the Adaptive Educational Systems literature for the past five years raised a number of questions that could be answered, but were not relevant for the scope of this thesis.

What are the key trends in adaptive systems and user modelling? The last major reviews of the field were Brusilovsky's papers in the early 2000s (Brusilovsky, 2004, 2001), and a lot has happened to in the 15 years since then. As well as summarising the key contributions over the period, it would be useful to see the trends: which research areas have faded, and which have have come in to replace them? There has also been a lot of work in the lifelong user modelling field, which would be valuable to survey.

Taxonomy of all adaptive systems Brusilovsky's original taxonomy, and the update by Knutov et al. are an excellent starting point for anyone beginning in the Adaptive Hypermedia (AH) research space, but as the field has developed, a number of areas of AH have branched off, as well as a number of related fields developing in parallel. Having a full history of the various technologies, what they have developed from and how they are utilised would be hugely valuable for anyone coming from outside the field, or even outside of research, to discover what the state of the art is.

Are adaptive systems a comparable to human tutoring? In 2011, VanLehn published a meta-review in the Intelligent Tutoring Systems (ITS) field, which suggests that ITSes are closer to the level of human tutoring than was originally believed, and it would be interesting to perform a similar review of AES results to see if the same can be said. There will be challenges, as fewer of the AES evaluations perform learning gain measurements, so it is more complicated to compare between systems. However, it will still yield interesting results, and might possibly encourage more researchers in this space to start measuring their learning gain.

7.3.3 Development of IDIUMS

Resolution of Multiple Values In Chapter 4, it is identified that IDIUMS may return multiple values for a particular request. There is scope to research a number of different possible approaches for resolving which of these values to accept and use within an adaptive application, with either a simulation or a user trial to determine which resolutions are most effective.

Investigate Expert Rule Creation For the purposes of the user trial in Chapter 6, the rules for mapping between values in each of the adaptive applications were manually written by the author, based on knowledge of how the applications work. A significant piece of research could be performed to study how expert developers and administrators of adaptive systems create rules for mapping in IDIUMS, and to create guidance as to the most effective approaches.

Investigate Educational Data Mining for Rule Creation As well as studying how humans create rules for IDIUMS, data mining and machine learning techniques could be used to generate suitable mapping rules, after collecting a suitable corpus of data in existing adaptive applications.

Appendix A

Survey Questions

The survey questions as they appeared to participants of the User Trial.

1 Initial self-rating of previous programming experience

This questionnaire begins with a few questions about your previous programming experience.

1. Have you ever used a programming language before today? ◯ Yes ◯ No
2. How many years have you been learning programming?

   Less than 1 year

3. Please tick which programming languages you have programmed with:
   - ☐ BASIC
   - ☐ C
   - ☐ C++
   - ☐ Java
   - ☐ Pascal
   - ☐ Visual Basic
   - ☐ Pascal
   - ☐ PHP
   - ☐ Perl
   - ☐ Cecil
   - ☐ JavaScript
   - ☐ C#
   - ☐ Python
   - ☐ Ruby
   - ☐ Delphi

1 Rating of suitability of previous application

You have just finished using the Java loops [hypertext/game/assessment]. Please answer these questions on the appropriateness of the level of material presented to you. Was it too difficult, too easy or just right?

1 (one) means "too easy", 3 (three) means "just right" and 5 (five) means "too hard".

	1	2	3	4	5
At the start of using the Java loops [hypertext/game /assessment] how would you rate the difficulty of the [text/levels/questions]?	◯	◯	◯	◯	◯
At the end of using the Java loops [hypertext/game /assessment] how would you rate the difficulty of the [text/levels/questions]?	◯	◯	◯	◯	◯
Overall, how would you rate the difficulty of the [text/levels /questions]?	◯	◯	◯	◯	◯

1 Self-rating of concept understanding

Before you use the next adaptive educational systems in this experiment, we would like you to estim
concepts which are related to using loops in Java.

Please rate your understanding of these concepts on a scale of 1 (one) to 5 (five). Choosing the value
programming loops. Choosing the value 5 indicates that you believe you fully understand the concep
when writing programming code.

Below are the meanings of the values 1 to 5:

1. One - Have not heard of this concept.
2. Two - Do not know what this concept means.
3. Three - Understand what it means but cannot use it.
4. Four - Can use it, but may make mistakes.
5. Five - Fully understand this concept and can use it correctly.

Concept	1	2	3	4	5
Initialisation	○	○	○	○	○
Condition	○	○	○	○	○
Loop body block	○	○	○	○	○
Update and progress towards completion	○	○	○	○	○
Collections	○	○	○	○	○
Arrays	○	○	○	○	○
Mathematical operators	○	○	○	○	○
Logical comparisons	○	○	○	○	○
Using the loop variable as part of a second inner loop	○	○	○	○	○
Using the loop variable as part of an output	○	○	○	○	○
Using the loop variable in an arithmetic calculation	○	○	○	○	○
while loop	○	○	○	○	○
do until loop	○	○	○	○	○
for loop	○	○	○	○	○
foreach loop	○	○	○	○	○
Using a loop to repeat an action X times	○	○	○	○	○
Using a loop to process a list of items	○	○	○	○	○
Using a loop to repeat something forever	○	○	○	○	○
Using a loop to repeat until completion	○	○	○	○	○
Creating a loop that counts downwards	○	○	○	○	○
Nesting one loop inside another loop	○	○	○	○	○
breaking out of a loop	○	○	○	○	○
Making a loop continue to the next iteration	○	○	○	○	○
How a loop relates to recursion	○	○	○	○	○

Appendix B

Domain Model

For the purposes of running an experiment using adaptive educational systems, it was decided to fix the domain model of the content to remove that as a variable. This is the hierarchical structure of the domain model created for this experiment, based on researching the key topics presented in published text books. The result of that research can be see in Appendix C.

- repetition
- origins
  - jmp
  - goto
- loop structure
  - intialisation
  - condition
  - body block
  - update progress
- loop variable
  - variables
    - collections
    - arrays
  - mathematical operators
  - logical comparisons
  - uses
    - inner loop bound

        * output statement
        * arithmetic

- loop types
    - while
    - do
        * while
        * until
    - for
    - for each
- real syntax
    - Java
    - C/C++
    - Python
    - Perl
    - Visual Basic
- basic applications
    - repeat actions x times
    - process list of items
    - repeat forever
    - repeat until completion (sentinel)
- advanced topics
    - inverse loops
    - nested loops
    - escaping loops
    - relationship to recursion

Appendix C

Domain Model Comparison

Java Gently	The Object of Java
	While loop has: * loop condition * loop body For loop has: * initialisation * loop condition * progress instruction * statement sequence
Parts of a loop: * check expression * process body * update value	Loops contain: * initialisation code * primary work * progress toward termination
Use loop variable for: * in output statement * in simple arithmetic * as bounds of another loop	
Types of loop: * backwards * empty (skip body) * nested * endless * unfinished (break)	Types of loop: * counting loop * sentinel loop * nested loop
	Cautions: * check variables are in a state to start and end loop * using != has potential to never finish * if condition already false, loop will not start * beware "off-by-1" loops
	Testing coverage: * body is skipped * body executes once * multiple executions
	Versus recursion: * tail recursion can be rolled into loop * if recursion first, hard to make loop

References

Abdullah, N. A. and Davis, H. (2003). Is simple sequencing simple adaptive hypermedia? In *HYPERTEXT '03: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, pages 172–173, New York, NY, USA. ACM.

Abel, F., Heckmann, D., Herder, E., Hidders, A. J. H., Houben, G. J. P. M., Krause, D., Leonardi, E., and van der Sluijs, K. A. M. (2009a). A framework for flexible user profile mashups. In Dattolo, A., editor, *Proceedings of the International Workshop on Adaptation and Personalization for Web 2.0 (APWEB 2.0 2009, Trento, Italy, June 22, 2009; in conjunction with UMAP'09).*

Abel, F., Heckmann, D., Herder, E., Hidders, J., Houben, G.-J., Krause, D., Leonardi, E., and van der Sluijs, K. (2009b). Mashing up user data in the Grapple User Modeling Framework. In Hauger, D., Köck, M., and Nauerz, A., editors, *17th Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS).*

Abel, F., Heckmann, D., Herder, E., Hidders, J., Houben, G.-J., Krause, D., Leonardi, E., and van der Sluijs, K. (2010a). Tool for user event detection. Technical Report 2b.

Abel, F., Herder, E., Hidders, J., Houben, G.-J., Krause, D., Leonardi, E., and van der Sluijs, K. (2010b). Tool for reasoning about user observations. Technical Report 3.

Adamson, D., Dyke, G., Jang, H., and Rosé, C. (2014). Towards an Agile Approach to Adapting Dynamic Collaboration Support to Student Needs. 24(1):92–124.

Alvarez, N., Sanchez-Ruiz, A., Cavazza, M., Shigematsu, M., and Prendinger, H. (2015). Narrative Balance Management in an Intelligent Biosafety Training Application for Improving User Performance. 25(1):35–59.

Arevalillo-Herráez, M., Arnau, D., Marco-Giménez, L., González-Calero, J. A., Moreno-Picot, S., Moreno-Clari, P., Ayesh, A., Santos, O. C., Boticario, J. G., Saneiro, M., Salmeron-Majadas, S., Cabestrero, R., and Quirós, P. (2014). Providing Personalized Guidance in Arithmetic Problem Solving. *CEUR workshop proceedings*, 1181.

Arroyo, I., Royer, J. M., and Woolf, B. P. (2011). Using an Intelligent Tutor and Math Fluency Training to Improve Math Performance. *International Journal of Artificial Intelligence in Education*, 21(2):135–152.

Arroyo, I., Woolf, B., Burelson, W., Muldner, K., Rai, D., and Tai, M. (2014). A Multimedia Adaptive Tutoring System for Mathematics that Addresses Cognition, Metacognition and Affect. 24(4):387–426.

Baker, R. and Clarke-Midura, J. (2013). Predicting Successful Inquiry Learning in a Virtual Performance Assessment for Science. In Carberry, S., Weibelzahl, S., Micarelli, A., and Semeraro, G., editors, *User Modeling, Adaptation, and Personalization*, volume 7899 of *Lecture Notes in Computer Science*, pages 203–214. Springer Berlin Heidelberg.

Baker, R. S., Ocumpaugh, J., Gowda, S. M., Kamarainen, A. M., and Metcalf, S. J. (2014). Extending Log-Based Affect Detection to a Multi-User Virtual Environment for Science. In Dimitrova, V., Kuflik, T., Chin, D., Ricci, F., Dolog, P., and Houben, G.-J., editors, *User Modeling, Adaptation, and Personalization*, volume 8538 of *Lecture Notes in Computer Science*, pages 290–300. Springer International Publishing.

Basawapatna, A. R., Repenning, A., Koh, K. H., and Nickerson, H. (2013). The Zones of Proximal Flow: Guiding Students Through a Space of Computational Thinking Skills and Challenges. In *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, ICER '13, pages 67–74, New York, NY, USA. ACM.

Ben-Ari, M. (1998). Constructivism in Computer Science Education. *SIGCSE Bull.*, 30(1):257–261.

Berger, F. (2012). Evaluating an Implementation of an Adaptive Game-based Learning Architecture. In *Proceedings of the 20th International Conference on User Modeling, Adaptation, and Personalization*, UMAP'12, pages 351–355, Berlin, Heidelberg. Springer-Verlag.

Berners-Lee, T. J. (1992). The world-wide web. *Computer Networks and ISDN Systems*, 25(4-5):454–459.

Bieliková, M. and Kuruc, J. (2005). Sharing User Models for Adaptive Hypermedia Applications. In *ISDA '05: Proceedings of the 5th International Conference on Intelligent Systems Design and Applications*, pages 506–513, Washington, DC, USA. IEEE Computer Society.

Biggs, J. (2003). *Teaching For Quality Learning at University (Society for Research Into Higher Education S.).* Open University Press, second edition.

Bishop, J. (2000). *Java Gently, 3rd Ed.* Addison Wesley, third edition.

Bixler, R., Kopp, K., and D'Mello, S. (2014). Evaluation of a Personalized Method for Proactive Mind Wandering Reduction. In Kravcik, M., Santos, O. C., and Boticario, J. G., editors, *4th International Workshop on Personalization Approaches in Learning Environments (PALE 2014)*, volume 1181.

Bloom, B., Englehart, M. D., Furst, E. J., Hill, W. H., and Krathwohl, D. R. (1956). *Taxonomy of Educational Objectives: The Classification of Educational Goals - Handbook 1: Cognitive Domain.* David McKay Company, Inc., New York.

Boyle, T. (2006). The Design and Development of Second Generation Learning Objects. In Pearson, E. and Bohman, P., editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2006*, pages 2–12, Chesapeake, VA. AACE.

Brailsford, T. J., Moore, A., Stewart, C. D., Zakaria, M. R., Choo, B. S., Davies, P. M. C., and Zakaria, M. R. (2001). Towards a Framework for Effective Web-Based Distributed Learning.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2):87–129.

Brusilovsky, P. (2001). Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2):87–110.

Brusilovsky, P. (2004). Adaptive Educational Hypermedia: From Generation to Generation. In *Proceedings of 4th Hellenic Conference on Information and Communication Technologies in Education, Athens, Greece, September 29 - October 3, 2004*, pages 19–33.

Brusilovsky, P., Ritter, S., and Schwarz, E. (1997). Distributed intelligent tutoring on the Web. In *Proceedings of the 8th World Conference of the AIED Society, Kobe, Japan, 18-22 August 1997.*

Brusilovsky, P., Somyürek, S., Guerra, J., Hosseini, R., and Zadorozhny, V. (2015). The Value of Social: Comparing Open Student Modeling and Open Social Student Modeling. In Ricci, F., Bontcheva, K., Conlan, O., and Lawless, S., editors, *User Modeling, Adaptation and Personalization*, volume 9146 of *Lecture Notes in Computer Science*, pages 44–55. Springer International Publishing.

Buckingham, D. and Burn, A. (2007). Game literacy in theory and practice. *Journal of Educational Multimedia and Hypermedia*, 16(3):323–349.

Carmagnola, F. (2008). The Five Ws in User Model Interoperability. In *Proceedings of the 13th international conference on Intelligent user interfaces (UbiqUM 2008 workshop).*

Carmagnola, F. and Cena, F. (2006). From Interoperable User Models to Interoperable User Modeling. pages 409–413.

Carro, R. M., Breda, A. M., Castillo, G., and Bajuelos, A. L. (2006). A Methodology for Developing Adaptive Educational-Game Environments. In De, Bra, a. u. l., Brusilovsky, P., and Conejo, R., editors, *Adaptive Hypermedia and Adaptive Web-Based Systems: Second International Conference, AH 2002 Malaga, Spain, May 29 - 31, 2002. Proceedings*, volume 2347 of *Lecture Notes in Computer Science*, pages 90–99, Berlin, Heidelberg. Springer Berlin / Heidelberg.

Champaign, J., Zhang, J., and Cohen, R. (2011). Coping with Poor Advice from Peers in Peer-Based Intelligent Tutoring: The Case of Avoiding Bad Annotations of Learning Objects. In Konstan, J., Conejo, R., Marzo, J., and Oliver, N., editors, *User Modeling, Adaption and Personalization*, volume 6787 of *Lecture Notes in Computer Science*, pages 38–49. Springer Berlin Heidelberg.

Chaouachi, M., Jraidi, I., and Frasson, C. (2015). MENTOR: A Physiologically Controlled Tutoring System. In Ricci, F., Bontcheva, K., Conlan, O., and Lawless, S., editors, *User Modeling, Adaptation and Personalization*, volume 9146 of *Lecture Notes in Computer Science*, pages 56–67. Springer International Publishing.

Charles, D., Mcneill, M., Mcalister, M., Black, M., Moore, A., Stringer, K., Kücklich, J., and Kerr, A. (2005). Player-Centred Game Design: Player Modelling and Adaptive Digital Games. In *Digital Games Research Association 2005 Conference: Changing Views - Worlds in Play.*

Chen, J. (2006). Flow in Games. Master's thesis, University of Southern California.

Chen, J. (2007). Flow in Games (and Everything else). *Commun. ACM*, 50(4):31–34.

Chepegin, V., Aroyo, L., De Bra, P., and Heckmann, D. (2004). User Modeling for Modular Adaptive Hypermedia. In *SWEL'04 Workshop at the AH2004 Conference, TU/e Computing Science Report 04-19*, pages 366–371.

Cocea, M. and Magoulas, G. D. (2011). Context-Dependent Feedback Prioritisation in Exploratory Learning Revisited. In Konstan, J. A., Conejo, R., Marzo, J. L., and Oliver, N., editors, *User Modeling, Adaption and Personalization*, volume 6787 of *Lecture Notes in Computer Science*, pages 62–74. Springer Berlin Heidelberg.

Conati, C., Jaques, N., and Muir, M. (2013). Understanding Attention to Adaptive Hints in Educational Games: An Eye-Tracking Study. 23(1-4):136–161.

Conati, C. and Zhao, X. (2004). Building and evaluating an intelligent pedagogical agent to improve the effectiveness of an educational game. In *IUI '04: Proceedings of the 9th international conference on Intelligent user interfaces*, pages 6–13, New York, NY, USA. ACM.

Corrigan, S., Barkley, T., and Pardos, Z. (2015). Dynamic Approaches to Modeling Student Affect and its Changing Role in Learning and Performance. In Ricci, F., Bontcheva, K., Conlan, O., and Lawless, S., editors, *User Modeling, Adaptation and Personalization*, volume 9146 of *Lecture Notes in Computer Science*, pages 92–103. Springer International Publishing.

Cristea, A. I. and de Mooij, A. (2003). LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators. In *WWW'03 (The Twelfth International World Wide Web Conference), Alternate Track on Education.*

Csikszentmihalyi, M. (2008). *Flow: The Psychology of Optimal Experience.* Harper Perennial Modern Classics, 1st edition.

De Bra, P. (2002). Adaptive educational hypermedia on the web. *Commun. ACM*, 45(5):60–61.

De Bra, P., Aerts, A., Berden, B., de Lange, B., Rousseau, B., Santic, T., Smits, D., and Stash, N. (2003). AHA! The Adaptive Hypermedia Architecture. In *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*, HYPERTEXT '03, pages 81–84, New York, NY, USA. ACM.

De Bra, P., Houben, G.-J., and Wu, H. (1999). AHAM: a Dexter-based reference model for adaptive hypermedia. In *HYPERTEXT '99: Proceedings of the tenth ACM Conference on Hypertext and hypermedia : returning to our diverse roots*, pages 147–156, New York, NY, USA. ACM.

De Bra, P., Stash, N., and Smits, D. (2004). *Creating Adaptive Applications with AHA! Tutorial for AHA! version 3.0.*

de Vrieze, P., van Bommel, P., and van der Weide, T. (2004). A Generic Adaptivity Model in Adaptive Hypermedia. pages 344–347.

Dicerto, M. and Mazzetti, A. (2010). Final implementation of the DM definition tool. Technical Report 1c.

Dickey, M. D. (2006). "Ninja Looting" for instructional design: the design challenges of creating a game-based learning environment. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Educators program*, New York, NY, USA. ACM.

Doan, A., Madhavan, J., Domingos, P., and Halevy, A. (2002). Learning to map between ontologies on the semantic web. In *Proceedings of the 11th international conference on World Wide Web*, pages 662–673.

Dodigovic, M. (2012). Intelligent Sentence Writing Tutor: A System Development Cycle. (22).

Ellington, H., Addinall, E., and Percival, F. (1982). *A Handbook of Game Design.* Kogan Page Ltd.

Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures.* PhD thesis.

Fisch, S. M. (2005). Making educational computer games 'educational'. In *Proceedings of the 2005 conference on Interaction design and children*, IDC '05, pages 56–61, New York, NY, USA. ACM.

Forbes-Riley, K. and Litman, D. (2012). When Does Disengagement Correlate with Performance in Spoken Dialog Computer Tutoring? *International Journal of Artificial Intelligence in Education*, 22(2):19–41.

Fountain, A., Hall, W., Heath, I., and Davis, H. (1990). Microcosm: An Open Model for Hypermedia with Dynamic Linking. In *Hypertext: Concepts, Systems and Applications, Proceedings of ECHT'90, Paris, November 1990*, pages 298–311. Cambridge University Press.

Frazer, A., Argles, D., and Wills, G. (2007a). Assessing The Usefulness Of Mini-games As Educational Resources. In *ALT-C 2007: Beyond Control, 4-6 September 2007, Nottingham, UK.*

Frazer, A., Argles, D., and Wills, G. (2007b). Is Less Actually More? The Usefulness Of Educational Mini-games. In *The 7th IEEE International Conference on Advanced Learning Technologies, July 18-20, 2007, Niigata, Japan.*

Frias-Martinez, V. and Virseda, J. (2011). Personalization Of Mobile Learning Tools For Low-Income Populations. *CEUR workshop proceedings*, 732.

Garner, R., Brown, R., Sanders, S., and Menke, D. J. (1992). *"Seductive Details" and Learning from Text*, chapter 10, pages 239–254. Lawrence Erlbaum Associates, Mahwah, NJ.

Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment*, 1(1).

Gee, J. P. (2004). *What Video Games Have to Teach Us About Learning and Literacy.* Palgrave Macmillan.

Gilbert, L. and Gale, V. (2007). *Principles of E-learning Systems Engineering.* Chandos Series for Information Professionals. Chandos Publishing (Oxford) Ltd.

Gilbert, S. B., Blessing, S. B., and Guo, E. (2015). Authoring Effective Embedded Tutors: An Overview of the Extensible Problem Specific Tutor (xPST) System. 25(3):428–454.

Gong, Y., Beck, J., and Ruiz, C. (2012). Modeling Multiple Distributions of Student Performances to Improve Predictive Accuracy. In Masthoff, J., Mobasher, B., Desmarais, M., and Nkambou, R., editors, *User Modeling, Adaptation, and Personalization*, volume 7379 of *Lecture Notes in Computer Science*, pages 102–113. Springer Berlin Heidelberg.

Green, D. T., Walsh, T. J., Cohen, P. R., Beal, C. R., and Chang, Y.-H. (2011). Gender Differences and the Value of Choice in Intelligent Tutoring Systems. In Konstan, J. A., Conejo, R., Marzo, J. L., and Oliver, N., editors, *User Modeling, Adaption and Personalization*, volume 6787 of *Lecture Notes in Computer Science*, pages 341–346. Springer Berlin Heidelberg.

Greer, J., Frost, S., Banow, R., Thompson, C., Kuleza, S., Wilson, K., and Koehn, G. (2015). The Student Advice Recommender Agent: SARA. In Kravcik, M., Santos, O. C., Boticario, J. G., Bielikova, M., and Horvath, T., editors, *23rd International Conference User Modeling, Adaptation, and Personalization UMAP 2015*, pages 16–23.

Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., and von Wilamowitz-Moellendorff, M. (2005). GUMO The General User Model Ontology. In Ardissono, L., Brna, P., and Mitrovic, A., editors, *User Modeling 2005*, volume 3538 of *Lecture Notes in Computer Science*, pages 428–432. Springer, Berlin / Heidelberg.

Heffernan, N. T. and Lindquist Heffernan, C. (2014). The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching. 24(4):470–497.

Henning, P. A., Heberle, F., Streicher, A., Zielinski, A., Swertz, C., Bock, J., Zander, S., and the INTUITEL collaboration (2014). Personalized Web Learning: Merging Open Educational Resources into Adaptive Courses for Higher Education. In Kravcik, M., Santos, O. C., and Boticario, J. G., editors, *4th International Workshop on Personalization Approaches in Learning Environments (PALE 2014)*, volume 1181.

Hunicke, R. (2005). The case for dynamic difficulty adjustment in games. In *ACE '05: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 429–433, New York, NY, USA. ACM.

IMS Global Learning Consortium, Inc. (2003). IMS Simple Sequencing Best Practice and Implementation Guide. Website: http://www.imsglobal.org/simplesequencing/index.html.

Kalfoglou, Y. and Schorlemmer, M. (2003). Ontology Mapping: The State of the Art. *Knowl. Eng. Rev.*, 18(1):1–31.

Kardan, S. (2012). Data Mining for Adding Adaptive Interventions to Exploratory and Open-Ended Environments. In Masthoff, J., Mobasher, B., Desmarais, M., and Nkambou, R., editors, *User Modeling, Adaptation, and Personalization*, volume 7379 of *Lecture Notes in Computer Science*, pages 365–368. Springer Berlin Heidelberg.

Käser, T., Baschera, G.-M., Busetto, A. G., Klingler, S., Solenthaler, B., Buhmann, J. M., and Gross, M. (2012). Towards a Framework for Modelling Engagement Dynamics in Multiple Learning Domains. *International Journal of Artificial Intelligence in Education*, 22(2):42–70.

Käser, T., Busetto, A., Solenthaler, B., Baschera, G.-M., Kohn, J., Kucian, K., von Aster, M., and Gross, M. (2013). Modelling and Optimizing Mathematics Learning in Children. 23(1-4):115–135.

Kavcic, A. (2000). The role of user models in adaptive hypermedia systems. In *Electrotechnical Conference, 2000. MELECON 2000. 10th Mediterranean*, volume 1, pages 119–122 vol.1.

Kay, J. (2000). Stereotypes, Student Models and Scrutability. In *Intelligent Tutoring Systems: Proceedings of 5th International Conference, ITS 2000 Montréal, Canada, June 1923, 2000*, pages 19–30. Springer.

Kay, J., Kummerfeld, B., and Lauder, P. (2002). Personis: A Server for User Models. In De Bra, P., Brusilovsky, P., and Conejo, R., editors, *Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 2347 of *Lecture Notes in Computer Science*, pages 203–212. Springer, Berlin / Heidelberg.

Kelleher, C. and Pausch, R. (2007). Using storytelling to motivate programming. *Commun. ACM*, 50(7):58–64.

Kelly, H., Howell, K., Glinert, E., Holding, L., Swain, C., Burrowbridge, A., and Roper, M. (2007). How to Build Serious Games. *Commun. ACM*, 50(7):44–49.

Khachatryan, G. A., Romashov, A. V., Khachatryan, A. R., Gaudino, S. J., Khachatryan, J. M., Guarian, K. R., and Yufa, N. V. (2014). Reasoning Mind Genie 2: An Intelligent Tutoring System as a Vehicle for International Transfer of Instructional Methods in Mathematics. 24(3):333–382.

Knutov, E., De Bra, P., and Pechenizkiy, M. (2009). AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. *New Review of Hypermedia and Multimedia*, 15(1):5–38.

Krathwohl, D. R., Bloom, B. S., and Masia, B. (1964). *Taxonomy of Educational Objectives: The Classification of Educational Goals - Handbook 2: Affective Domain.* Longman, London, UK, 1 edition.

Kravcik, M., Burgos, D., Hummel, H., Fernandez, D., Oneto, L., and Specht, M. (2008). Integrated model of adaptation on learning with specifications. Technical Report 1.

Kuruc, J. (2005). Sharing User Model between Several Adaptive Hypermedia Applications. In Bieliková, M., editor, *Proceedings of IIT.SRC 2005: Student Research Conference in Informatics and Information Technologies, Bratislava*, pages 249–256. Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava.

Kyriacou, D. and Davis, H. C. (2008). Moving Towards Life-Long User Modeling. In *Advanced Learning Technologies, 2008. ICALT '08. Eighth IEEE International Conference on*, pages 647–648.

Laurillard, D. (1993). *Rethinking University Teaching: A Framework for the Effective Use of Educational Technology.* Routledge.

Leite, I., Pereira, A., Castellano, G., Mascarenhas, S., Martinho, C., and Paiva, A. (2011). Social Robots in Learning Environments: a Case Study of an Empathic Chess Companion.

Lenat, D. B. and Durlach, P. J. (2014). Reinforcing Math Knowledge by Immersing Students in a Simulated Learning-By-Teaching Experience. 24(3):216–250.

Leonardi, E., Abel, F., Heckmann, D., Herder, E., Hidders, J., and Houben, G. J. (2010). A Flexible Rule-based Method for Interlinking, Integrating, and Enriching User Data. In *Proceedings of the 10th International Conference on Web Engineering*, ICWE'10, pages 322–336, Berlin, Heidelberg. Springer-Verlag.

Massimini, F., Csikszentmihalyi, M., and Carli, M. (1987). The monitoring of optimal experience. A tool for psychiatric rehabilitation. *The Journal of nervous and mental disease*, 175(9):545–549.

McCarthy, D. A. (1930). *The Language Development of the Preschool Child.* University of Minnesota, Minneapolis.

McGuinness, D. L. and van Harmelen, F. (2004). OWL Web Ontology Language Overview. Technical report.

Mills, C. and Dalgarno, B. (2007). A conceptual model for game based intelligent tutoring systems. In *ICT: Providing choices for learners and learning Proceedings ascilite Singapore 2007*, pages 692–702.

Mitrovic, A., Mayo, M., Suraweera, P., and Martin, B. (2001). Constraint-Based Tutors: A Success Story. pages 931–940.

Mohammed, P. and Mohan, P. (2015). Dynamic Cultural Contextualisation of Educational Content in Intelligent Learning Environments using ICON. 25(2):249–270.

Moreno-Ger, P., Burgos, D., Sierra, J., and Manjón, B. (2007). A Game-Based Adaptive Unit of Learning with IMS Learning Design and <e-Adventure>. In Duval, E., Klamma, R., and Wolpers, M., editors, *Creating New Learning Experiences on a Global Scale*, volume 4753 of *Lecture Notes in Computer Science*, pages 247–261. Springer, Berlin / Heidelberg.

Muir, M. and Conati, C. (2012). Understanding Student Attention to Adaptive Hints with Eye-Tracking. In Ardissono, L. and Kuflik, T., editors, *Advances in User Modeling*, volume 7138 of *Lecture Notes in Computer Science*, pages 148–160. Springer Berlin Heidelberg.

Nava Tintarev, M. G. (2015). Benefits and risks of emphasis adaptation instudy workflows. In Milos Kravcik, O. C., editor, *23rd International Conference User Modeling, Adaptation, and Personalization UMAP 2015*, pages 8–15.

Nguyen, L. and Do, P. (2008). Learner Model in Adaptive Learning.

Nussbaumer, A., Kravcik, M., and Albert, D. (2012). Supporting Self-Reflection in Personal Learning Environments Through User Feedback. *CEUR workshop proceedings*, 872.

Nye, B. D., Graesser, A. C., and Hu, X. (2014). AutoTutor and Family: A Review of 17 Years of Natural Language Tutoring. 24(4):427–469.

Oneto, L., Abel, F., Herder, E., and Smits, D. (2009). Making today's Learning Management Systems adaptive. In *Proceedings of 1st Workshop on Learning Management Systems meet Adaptive Learning Environments*.

Pareto, L. (2014). A Teachable Agent Game Engaging Primary School Children to Learn Arithmetic Concepts and Reasoning. 24(3):251–283.

Peirce, N., Conlan, O., and Wade, V. (2008). Adaptive Educational Games: Providing Non-invasive Personalised Learning Experiences. In *Digital Games and Intelligent Toys Based Education, 2008 Second IEEE International Conference on*, pages 28–35. IEEE.

Pelánek, R. and Jarušek, P. (2015). Student Modeling Based on Problem Solving Times. 25(4):493–519.

Prince, R. (2009). Sharing User Models with Adaptive Games. In *Adjunct Proceedings of User Modeling, Adaptation, and Personalization 2009 Doctoral Consortium*.

Prince, R. and Davis, H. C. (2008). The Scope of Adaptive Digital Games for Education. In *Digital Games and Intelligent Toys Based Education, 2008 Second IEEE International Conference on*, pages 167–169.

Prince, R. and Davis, H. C. (2011). IDIUMS: sharing user models through application attributes. In Abel, F., Baldiris, S. M., and Henze, N., editors, *Adjunct Proceedings of the 19th International Conference on User Modeling, Adaptation and Personalization (UMAP)*.

Redondo-Hernández, A. and Pérez-Marín, D. (2011). Automatic generation of questions adapted to the personality and learning style of the students. *CEUR workshop proceedings*, 732.

Riley, D. D. (2002). *The Object Of Java: Introduction to Programming Using Software Engineering Principles, BlueJ Edition*. Addison Wesley, bluej ed edition.

Roldán, D., Martín, E., Haya, P. A., and García-Herranz, M. (2011). Adaptive Activities for Inclusive Learning using Multitouch Tabletops: An approach. In *1st International Workshop on Personalization Approaches in Learning Environments (PALE 2011)*, volume 732.

Rollings, A. and Morris, D. (2003). *Game Architecture and Design: A New Edition*. New Riders Games.

Rosa, J. a. H., Barbosa, J. L. V., Kich, M., and Brito, L. (2015). A Multi-Temporal Context-aware System for Competences Management. 25(4):455–492.

Sabourin, J., Mott, B., and Lester, J. (2013). Utilizing Dynamic Bayes Nets to Improve Early Prediction Models of Self-regulated Learning. In Carberry, S., Weibelzahl, S., Micarelli, A., and Semeraro, G., editors, *User Modeling, Adaptation, and Personalization*, volume 7899 of *Lecture Notes in Computer Science*, pages 228–241. Springer Berlin Heidelberg.

Shareghi Najar, A., Mitrovic, A., and McLaren, B. (2014). Adaptive Support versus Alternating Worked Examples and Tutored Problems: Which Leads to Better Learning? In Dimitrova, V., Kuflik, T., Chin, D., Ricci, F., Dolog, P., and Houben, G.-J., editors, *User Modeling, Adaptation, and Personalization*, volume 8538 of *Lecture Notes in Computer Science*, pages 171–182. Springer International Publishing.

Shi, L., Al Qudah, D., Qaffas, A., and Cristea, A. (2013). Topolor: A Social Personalized Adaptive E-Learning System. In Carberry, S., Weibelzahl, S., Micarelli, A., and Semeraro, G., editors, *User Modeling, Adaptation, and Personalization*, volume 7899 of *Lecture Notes in Computer Science*, pages 338–340. Springer Berlin Heidelberg.

Silva, R. C., Direne, A. I., and Marczal, D. (2015). Using Problem Statement Parameters and Ranking Solution Difficulty to Support Personalization. In Kravcik, M., Santos, O. C., Boticario, J. G., Bielikova, M., and Horvath, T., editors, *23rd International Conference User Modeling, Adaptation, and Personalization UMAP 2015*, pages 46–51.

Sleeman, D. H. and Brown, J. S. (1982). Intelligent Tutoring Systems: an overview. In Sleeman, D. H. and Brown, J. S., editors, *Intelligent Tutoring Systems*, pages 1–11. Academic Press.

Sosnovsky, S. A., Dolog, P., Henze, N., Brusilovsky, P., and Nejdl, W. (2007). Translation of Overlay Models of Student Knowledge for Relative Domains Based on Domain Ontology Mapping. In Luckin, R., Koedinger, K. R., and Greer, J. E., editors, *AIED*, volume 158 of *Frontiers in Artificial Intelligence and Applications*, pages 289–296. IOS Press.

Stamper, J., Barnes, T., Croy, M., and Eagle, M. (2012). Experimental Evaluation of Automatic Hint Generation for a Logic Tutor. *International Journal of Artificial Intelligence in Education*, 22(1):3–18.

Steinkuehler, C. A. (2004). Learning in massively multiplayer online games. In *Proceedings of the 6th international conference on Learning sciences*, ICLS '04, pages 521–528. International Society of the Learning Sciences.

Tegos, S., Demetriadis, S., and Tsiatsos, T. (2014). A Configurable Conversational Agent to Trigger Students' Productive Dialogue: A Pilot Study in the CALL Domain. 24(1):62–91.

Thai-Nghe, N., Horvath, T., and Schmidt-Thieme, L. (2011). Context-Aware Factorization for Personalized Student's Task Recommendation. In *1st International Workshop on Personalization Approaches in Learning Environments (PALE 2011)*.

Vail, A. K., Boyer, K. E., Wiebe, E. N., and Lester, J. C. (2015). The Mars and Venus Effect: The Influence of User Gender on the Effectiveness of Adaptive Task Support. In Ricci, F., Bontcheva, K., Conlan, O., and Lawless, S., editors, *User Modeling, Adaptation and Personalization*, volume 9146 of *Lecture Notes in Computer Science*, pages 265–276. Springer International Publishing.

VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*, 46(4):197–221.

Vygotsky, L. (1978). Interaction Between Learning and Development. *Mind and Society*, pages 79–91.

Walker, E., Rummel, N., and Koedinger, K. (2014). Adaptive Intelligent Support to Improve Peer Tutoring in Algebra. 24(1):33–61.

Weber, G. and Specht, M. (1997). User Modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems.

Westerfield, G., Mitrovic, A., and Billinghurst, M. (2015). Intelligent Augmented Reality Training for Motherboard Assembly. 25(1):157–172.

Zakaria, M. R. and Brailsford, T. (2002). Technical Note User modelling, and adaptive hypermedia frameworks for education. *New Review of Hypermedia and Multimedia*, 8(1):83–97.

Zapata-Rivera, D. (2012). Adaptive Score Reports. In Masthoff, J., Mobasher, B., Desmarais, M., and Nkambou, R., editors, *User Modeling, Adaptation, and Personalization*, volume 7379 of *Lecture Notes in Computer Science*, pages 340–345. Springer Berlin Heidelberg.