



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

Copyright statement

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

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

Liability statement

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

Access Agreement

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

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

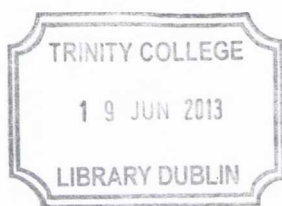
Domain-Aware User Model Interoperability

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

Edmond Walsh

Knowledge and Data Engineering Group,
School of Computer Science and Statistics,
Trinity College Dublin,
Ireland

2013



Thesis 10099

Declaration

This thesis has not been submitted as an exercise for a degree at this or any other University. It is entirely the candidate's own work. The candidate agrees that the Library may lend or copy the thesis upon request. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Summary

Exchanging user information between multiple sources can potentially bring many benefits that enhance the user experience in software applications. Richer and more dynamic user models can be constructed allowing more tailoring of content and services to the specific needs of each individual user. However, providing effective interoperability is a complex challenge as there can be significant heterogeneity between user models. As user model interoperability provides more expansive and detailed user models, it results in user models that contain information from multiple overlapping domains. Current generic approaches and tools offer insufficient support for the complex domain-specific nature of many user model interoperability tasks.

This thesis presents the design, implementation and evaluation of the FUSE domain-aware approach to user model interoperability. This novel approach differs from existing generic approaches by incorporating domain knowledge in processes and tools to support complex user model interoperability tasks in multiple overlapping domains. The FUSE Mapping Tool supports mapping between heterogeneous user model representations and the FUSE Translation Service supports user model translation between multiple systems. Both of these tools provide domain knowledge through two domain-aware mechanisms: a canonical user model and user model mapping types. These mechanisms provide knowledge of multiple domains which can support the complex tasks involved in the mapping and translation of user models.

Acknowledgements

The author would like to acknowledge the invaluable support and guidance of his Supervisor, Prof. Vincent Wade, in undertaking this work. The author wishes to thank his Assistant Supervisor, Dr. Alexander O'Connor, for his help and advice. Finally, the author would like to express special thanks to Anne-Marie for her continued encouragement and support.

This work was funded by the Irish Research Council for Science, Engineering and Technology, and by Science Foundation Ireland.

Edmond Walsh

University of Dublin, Trinity College

2013

Contents

Summary	v
Acknowledgements	vii
List of Figures	xiii
List of Tables	xv
1 Introduction	1
1.1 Motivation	1
1.2 Research Question	3
1.2.1 Objectives	4
1.3 Contribution	4
1.4 Methodology	6
1.5 Thesis Overview	7
2 User Model Interoperability	11
2.1 Introduction	11
2.2 Interoperability Definitions and Methods	12
2.2.1 Mapping Tools	14
2.3 User Modelling	19
2.3.1 Key Issues for User Model Interoperability	22
2.4 User Model Interoperability Approaches	26
2.4.1 Personis	26
2.4.2 CUMULATE	27
2.4.3 GUMF	27
2.4.4 UUCM	28
2.4.5 GUMO/UserML	28
2.4.6 GUC	29
2.4.7 Dolog and Schafer	29
2.4.8 Carmagnola and Dimitrova	29
2.4.9 Cena	30
2.4.10 Berkovsky	31
2.4.11 Comparison of Surveyed Approaches	31
2.5 Evaluation Metrics	42
2.6 Summary	45
3 FUSE: A Domain-Aware Approach to User Model Interoperability	47
3.1 Introduction	47

3.2	Methodology	48
3.3	Selection of Key Issues, Methods and Metrics	48
3.4	Domain-Aware User Model Interoperability	53
3.4.1	Application in Education	56
3.5	FUSE Domain-Aware Approach	58
3.5.1	Canonical User Model	58
3.5.2	User Model Mapping Types	65
3.6	FUSE Processes	71
3.6.1	User Model Mapping	73
3.6.2	User Model Translation	74
3.7	Summary	75
4	Development and Evaluation of the FUSE Mapping Tool (Cycle 1)	77
4.1	Introduction	77
4.2	Development of the FUSE Mapping Tool	78
4.2.1	Methodology	78
4.2.2	Architecture	81
4.2.3	Implementation	83
4.3	Preliminary Analysis of Usability in User Model Mapping	93
4.3.1	Objectives and Methodology	93
4.3.2	Semi-Structured Interviews	96
4.4	Discussion	98
4.5	Summary	100
5	Development and Evaluation of the FUSE Mapping Tool (Cycle 2)	101
5.1	Introduction	101
5.2	Development of the FUSE Mapping Tool	102
5.2.1	Selected Modifications	102
5.3	Mapping Tool Walkthrough	107
5.3.1	Basic User Model Mapping Example	108
5.3.2	More Complex Mapping Examples	113
5.4	Analysis of Accuracy and Duration in User Model Mapping	116
5.4.1	Objectives and Methodology	116
5.4.2	Mapping Accuracy	121
5.4.3	Mapping Duration	123
5.5	Analysis of Usability in User Model Mapping	124
5.5.1	Objectives and Methodology	124
5.5.2	Prior Experience Questionnaire	125
5.5.3	SUS Questionnaire	127
5.5.4	Task Questionnaire	128
5.5.5	Comparison Questionnaire	130
5.6	Discussion	131
5.7	Summary	135
6	Development and Evaluation of the FUSE Translation Service	137
6.1	Introduction	137
6.2	Development of the FUSE Translation Service	138
6.2.1	Architecture	138
6.2.2	Implementation	140

6.3	Translation Service Walkthrough	143
6.3.1	User Model Translation Example	143
6.4	Analysis of Performance in User Model Translation	146
6.4.1	Objectives and Methodology	146
6.4.2	User Model Schemas	148
6.4.3	User Model Instances	150
6.5	Discussion	152
6.6	Summary	155
7	Evaluation of the FUSE Domain-Aware Approach	157
7.1	Introduction	157
7.2	Comparison with Surveyed Approaches	158
7.3	Discussion	165
7.4	Summary	166
8	Conclusions	167
8.1	Research Question and Objectives	167
8.2	Contribution	171
8.3	Future Work	174
8.4	Final Remarks	176
	Bibliography	177
	Glossary	187
A	User Model Examples	189
B	Mapping Examples	199
C	Evaluation Results	205

List of Figures

2.1	Stylus Studio	15
2.2	Altova Mapforce	16
2.3	COMA++	17
2.4	CogZ	18
3.1	Example of education-based user model with overlapping domains	54
3.2	Direct and canonical model mapping approaches	59
3.3	Top-level IMS LIP elements	63
3.4	Canonical user model elements	64
3.5	Generic mapping types	66
3.6	User model mapping process	72
3.7	User model translation process	72
4.1	Mapping Tool development methodology	79
4.2	FUSE architecture	82
4.3	Iteration 1 of the Mapping Tool	85
4.4	Iteration 2 of the Mapping Tool	86
4.5	Iteration 3 of the Mapping Tool	87
4.6	Mapping example in XML representation	90
4.7	Mapping example in XQuery representation	92
5.1	Version 2 of the Mapping Tool	107
5.2	Mapping example 1	108
5.3	Navigation of user model 1 to “firstname” element	109
5.4	Navigation of canonical user model to “firstname” element	109
5.5	Selection of “equivalent first name” user model mapping type	110
5.6	Selection of user model 1 “firstname” element	110
5.7	Steps 1-5 repeated to select “firstname” element from user model 4	111
5.8	Selection of testing area to validate created mappings	111
5.9	Selection of source and target for mapping testing	112
5.10	Result of mapping testing	112
5.11	Mapping example 2	113
5.12	Mapping example 2 in the Mapping Tool	114
5.13	Mapping example 3	114
5.14	Mapping example 3 in the Mapping Tool	115
5.15	Mapping example 4	115
5.16	Mapping example 4 in the Mapping Tool	116
5.17	Average task accuracy results	122
5.18	Average task duration results	123

5.19	Average participant prior knowledge results	126
5.20	SUS results	128
5.21	Average task-based user satisfaction results	129
6.1	FUSE architecture	139
6.2	APeLS-based SQL web course	142
6.3	Test Service for the Translation Service	143
6.4	Translation example 1	144
6.5	User model translation steps in the Translation Service	145
6.6	Mapping complexity and quantity	148
6.7	User model quantity	149
6.8	User model size	150
6.9	User model instance quantity 1	151
6.10	User model instance quantity 2	151
A.1	Example of user model 1	189
A.2	Example of user model 2	191
A.3	Example of user model 3	194
A.4	Example of user model 4	194
A.5	Example of canonical user model	195
B.1	Example of XML mappings	199
B.2	Example of source to canonical XQuery mappings	201
B.3	Example of canonical to target XQuery mappings	202

List of Tables

2.1	Comparison of surveyed approaches	32
3.1	User model interoperability key issues	49
3.2	Preliminary and selected systems	57
3.3	Examples of user model mapping types	68
4.1	Trial user models	93
4.2	Semi-structured interview results	97
5.1	Selected modifications in the Mapping Tool	103
5.2	Experiment user models	118
5.3	Gold standard mapping steps in the Mapping Tool and Altova Mapforce	120
5.4	Prior experience questionnaire	125
5.5	SUS questionnaire	127
5.6	Task questionnaire	128
7.1	Comparison of FUSE with surveyed approaches	163
C.1	Semi-structured interview	205
C.2	Mapping accuracy and duration for task 1	206
C.3	Mapping accuracy and duration for task 2	207
C.4	Mapping accuracy and duration for task 3	208
C.5	Mapping accuracy and duration for task 4	209
C.6	Mapping accuracy and duration for tasks 1-4	210
C.7	Paired t-tests for accuracy and duration	211
C.8	Prior experience questionnaire	212
C.9	Participant responses to prior experience questionnaire	213
C.10	SUS questionnaire	214
C.11	Participant responses to SUS questionnaire	215
C.12	Task questionnaire	216
C.13	Participant responses to questions 1-6 of task questionnaire	217
C.14	Paired t-tests for questions 1-6 of task questionnaire	218
C.15	Participant responses to question 7 of task questionnaire	219
C.16	Participant responses to question 8 of task questionnaire	220
C.17	Participant responses to question 9 of task questionnaire	221
C.18	Participant responses to question 10 of task questionnaire	222
C.19	Comparison questionnaire	223
C.20	Participant responses to question 1 of comparison questionnaire	223
C.21	Participant responses to question 2 of comparison questionnaire	224
C.22	Number/complexity of mappings	225

C.23 Number of exchanging user models 225

C.24 Size of user models 226

C.25 Percentage of concurrent user model requests processed 226

Chapter 1

Introduction

1.1 Motivation

A vast amount of information about users is now being accumulated within organisations and on the open web [Carmagnola et al., 2011]. Within organisations such as those in corporate, academic, medical or governmental areas, extensive user information is a central component of many systems. For example, in recent surveys 93% of responding companies use administration systems to manage human resources [Martin, 2012] and 79% have in place or plan to use a learning management system to manage employee educational profiles [IMC (UK) Learning Ltd, 2010]. Similarly, on the open web, users provide and manage their own information through the many services they use. For example, social networking websites such as Facebook and LinkedIn allow users to maintain extensive personal and professional profiles. These social network profiles are increasingly becoming the focus for identity and personalisation on the web, allowing other websites to personalise content for users based on their profile.

In many cases, these various online systems and services are gathering related information about the same user; however, there are many organisational and technological boundaries that inhibit the sharing and reuse of this information [Carmagnola et al., 2011]. Exchanging user information between multiple sources can potentially bring many benefits to enhance the user experience in software applications [Aroyo et al., 2006]. Richer and more dynamic models of the user can be constructed and made available to each system, thereby allowing improved user-centric functionality while at the same time reducing time-consuming and repetitive user interactions [Berkovsky et al., 2007]. Potentially, one of the main benefits of user model interoperability is in

the area of personalisation. In this case, richer user models allow for more tailoring of content and services to the specific needs of each individual user [Berkovsky et al., 2007].

Resolving the heterogeneity of the user models is the key requirement to provide effective sharing of user information. The heterogeneity of the various systems and their representations of user information can be significant. The diverse nature of the web naturally results in user data that is designed by people with different perspectives, for different purposes and can differ greatly in structure, syntax and semantics. There are also many other challenges related to the provision of user model interoperability such as privacy and security in the exchange of user data. This research is fundamentally focused on the core technical, syntactic and semantic challenges in overcoming user model heterogeneity and, as a result, related challenges such as privacy and security are considered outside the scope of this thesis.

There has been significant effort made in the development of standardised user models, for example IMS LIP in education, but heterogeneity remains a major challenge [Van Der Sluijs and Houben, 2006]. Mediation approaches have often been adopted when attempting to overcome user model heterogeneity [Carmagnola et al., 2011]. This can consist of a manual mapping process where a mapping administrator identifies equivalent attributes between data models. This is often accomplished using custom-made translation scripts or one of the many generic mapping tools available, such as Altova Mapforce [Altova, 2012]. The manual process can also be combined with an automatic matching process in which algorithms are used to automatically find potential matches, typically assigning them a confidence score [Shvaiko and Euzenat, 2012]. Automatic matching has been a focus for significant research, however, at present the complexity of many existing data interoperability tasks means that matching cannot complete the entire mediation process automatically [Falconer and Storey, 2007]. As a result, manual mapping provided by an administrator continues to be a significant requirement for user model interoperability.

As user model interoperability provides more expansive and detailed user models, it also results in user models that contain information from multiple overlapping domains. In this research, a domain is defined as “a group of related terms that represent a particular field of knowledge”. For example, a user model can contain general user information from the user domain (e.g. identification, interests, preferences). This can overlap with user information from other domains such as education (e.g. learning preferences, assessments) or employment (e.g. performance, responsibilities). Subsequently, these domains can also overlap with related user information from particular subject domains (e.g. competencies in mathematics, assessments in computer literacy).

This overlap of domains within user models increases the complexity of the manual mapping task: requiring mapping administrators that possess increasingly specialised domain knowledge. Current generic mapping approaches and tools offer insufficient support for the domain-specific nature of many user model interoperability tasks.

The primary motivation of this research is to investigate the impact of a domain-aware approach on key aspects of mapping and translation for user model interoperability. A domain-aware approach can be defined as:

The incorporation of exchangeable knowledge from one or more domains into generic tools and processes to support complex interoperability tasks.

Domain knowledge can be applied to support complex user model interoperability tasks in particular application areas, for example, education. However, the domain knowledge is also exchangeable and extendable to support interoperability tasks in other application areas.

This research develops a domain-aware approach to user model interoperability called FUSE (Federated User Exchange). The approach uses knowledge from multiple domains to support both manual mapping and automatic translation for user model interoperability. The FUSE approach consists of two different domain-aware mechanisms; (i) a canonical user model and (ii) user model mapping types, which tailor the process to specific domains. All mappings are created to the canonical user model which is a consistent shared user model representation. The user model mapping types are mapping components specifically created and used for mapping between user models. The combination of the domain-aware approach in both the mapping and translation processes is novel and potentially improves key aspects of the provision of user model interoperability.

1.2 Research Question

How does the domain-aware approach developed in FUSE affect the accuracy, duration and usability of user model mapping, the performance of user model translation, and the overall provision of user model interoperability?

1.2.1 Objectives

The following three objectives were derived to address the research question:

1. Conduct a review of the state of the art in user model interoperability to determine key issues and methods.
2. Design and develop the FUSE domain-aware approach to user model interoperability and implement it in the form of mapping and translation tools.
3. Evaluate the tools to determine the effect on accuracy, duration and usability of user model mapping, the performance of user model translation, and the overall provision of user model interoperability.

Education was chosen as the test application area to develop and evaluate the FUSE domain-aware approach. This area was selected as there is a rich collection of existing educational services which contain representative user models across multiple domains.

1.3 Contribution

The main contribution of this thesis is that **for user model interoperability tasks in the application area of education, the FUSE domain-aware approach and tools demonstrate improved accuracy and usability for user model mapping in multiple overlapping domains while also maintaining mapping duration times.** This contribution of the research was supported by the development of the Mapping Tool for administrators to manually resolve heterogeneity between user models. This tool contrasts with the state of the art in graphical mapping tools by incorporating two domain-aware mechanisms; a canonical user model and user model mapping types, in a web-based tool. These mechanisms provide knowledge of multiple overlapping domains and allow administrators to create increasingly specialised user model mappings to a consistent and detailed canonical user model. Experiment results indicate that the accuracy and usability of the Mapping Tool was superior when compared with a mapping tool using currently established approaches. The duration of the mapping tasks was similar in both tools despite the additional requirements of the domain-aware approach. These indicative evaluations were conducted with multiple small user populations, in the application area of education, and with selected user models, domains, and mapping tasks.

The first minor contribution of this thesis is that **the FUSE domain-aware approach and tools provide efficient performance across multiple key aspects of user model translation**. This contribution of the research was supported by the development of the Translation Service which provides automatic exchange of user models between multiple heterogeneous web systems. The two domain-aware mechanisms; the canonical user model and user model mapping types, are also applied in this tool to support the execution of specialised user model mappings to a consistent, shared canonical user model. Experiment results indicate efficient translation performance across multiple key aspects relating to user models and mappings at the schema and instance levels.

The second minor contribution of this thesis is that **the overall FUSE domain-aware approach and tools provide a comprehensive form of user model interoperability when compared to existing state of the art approaches**. This contribution of the research was supported by a systematic functional analysis of the overall FUSE approach, consisting of the Mapping Tool and the Translation Service. Results indicate that the FUSE approach provides a comprehensive form of user model interoperability when compared with a selection of existing state of the art approaches.

Evidence of the contribution to the state of the art comes in the form of the following publications:

1. **Walsh E.**, O'Connor A., and Wade V., "Evaluation of a Domain-Aware Approach to User Model Interoperability", 23rd ACM Conference on Hypertext and Social Media, June 2012, Milwaukee, Wisconsin, USA. [Walsh et al., 2012]
2. **Walsh E.**, O'Connor A., and Wade V., "Supporting Learner Model Exchange In Educational Web Systems" 7th International Conference on Web Information Systems and Technologies, May 2011, Noordwijkerhout, The Netherlands. [Walsh et al., 2011a]
3. **Walsh E.**, and Wade V., "Lowering the Barriers to User Model Interoperability" Ubiquitous User Modeling Workshop at the Conference on User Modeling, Adaptation, Personalization, June 2009, Trento, Italy. [Walsh and Wade, 2009]
4. **Walsh E.**, Dagger D., and Wade V., "Supporting "Personalisation for All" through Federated User Model Exchange Services (FUMES)" Towards User Modelling and Adaptive Systems for All Workshop at User Modelling 07, June 2007, Corfu, Greece. [Walsh et al., 2007]

1.4 Methodology

The goal of this research was to identify how the domain-aware approach applied in FUSE affects the accuracy, duration and usability of user model mapping, the performance of user model translation, and the overall provision of user model interoperability. To achieve this, many different but related stages were required such as the development and evaluation of graphical mapping tools, translation tools and multiple user models and mapping types. As a result, a combination of different research methods were needed to achieve all the research objectives [Cohen et al., 2007].

In the initial stages of the research exploratory methods were applied [Babbie, 2010]. The field of interoperability was reviewed, including current mapping techniques and tools. The field of user modelling was reviewed and a list of key issues for user model interoperability was derived and used as criteria to compare a collection of state of the art approaches. Existing evaluation methods were also reviewed to support the subsequent evaluation of this research.

The exploratory methods led to the formulation of a new approach to user model interoperability. Firstly, appropriate key issues and methods from the state of the art review were selected. These choices were made based on the findings of the analysis of the surveyed approaches and based on the overall objectives of this research. One of the key findings of the state of art review was the requirement for support in the complex tasks inherent in user model interoperability by providing domain knowledge in the processes and tools. To address this, a novel domain-aware approach to user model interoperability was formulated and designed.

A top-down design approach was used where the main concepts and key aspects of the approach were defined and then refined in a representative application area. The selection of an application area allowed in-depth analysis to be conducted. Education was selected because of the availability of systems, user models, and the use of multiple domains which are also relevant to other application areas. A review of selected educational systems was conducted, ultimately leading to a collection of user models, domains and interoperability tasks that could be used during the design, implementation and evaluation of a domain-aware approach to user model interoperability. Following this analysis, the results were used to design the FUSE domain-aware approach. This consisted of two domain-aware mechanisms: the canonical user model and the user model mapping types. Each of these mechanisms was designed to provide knowledge of multiple overlapping domains to support complex interoperability tasks. Finally, two FUSE processes were

defined; a manual user model mapping process and an automatic user model translation process. These processes specify in detail how the FUSE approach is applied to provide domain-aware user model interoperability between existing systems.

For the user model mapping process, the FUSE Mapping Tool was developed using an iterative design approach based on the selected systems and user models. Once an initial prototype was designed and implemented, user feedback was collected using a number of different methods. Firstly, users were asked to complete a number of tasks using the tool. Semi-structured interview sessions were then conducted and the collected data was used to enhance the Mapping Tool as part of the iterative development process. The final experiments consisted of tasks and questionnaires completed by users to evaluate the accuracy, duration and usability of the Mapping Tool. The users also completed the same tasks and questionnaires in a widely used, commercial mapping tool allowing for a direct comparison between tools. To determine the mapping accuracy, the user mappings were compared to a set of “gold standard” benchmark mappings which were developed for both tools. The gold standard mappings were defined and categorised using a practical test-based approach. To analyse the significance of the evaluation results, commonly used statistical methods were employed.

For the user model translation process, the FUSE Translation Service was developed in stages using the selected systems and user models. A prototype tool was developed to translate user information between the heterogeneous user models. The evaluation of the Translation Service consisted of a performance analysis in which the impact of user models and mappings at both the schema and instance levels was assessed. The results were analysed in relation to recommended performance guidelines for web-based systems.

A functional comparison of the overall FUSE domain-aware approach and the state of the art approaches was also conducted to examine the tools in the context of the field of user model interoperability. Overall, the applied methods allowed the research question and objectives to be addressed by providing representative and existing interoperability challenges throughout the design, implementation and evaluation of this research.

1.5 Thesis Overview

This section describes the layout of this thesis and the topics covered in each of the chapters.

- **Chapter 2** describes the field of user model interoperability. It presents definitions and key methods for interoperability including mapping techniques and some existing commercial and research-based mapping tools. It details general user modelling principles and the key issues for user model interoperability. The identified key issues are then used as a basis to analyse and compare a number of key approaches in the area and draw conclusions about the current state of the art in user model interoperability. Finally, some evaluation methods from the related research are described to support the evaluation of this research.
- **Chapter 3** describes the FUSE approach to user model interoperability. To define the approach, a number of appropriate key issues, methods and metrics were selected from the state of the art review. The concept of domain-aware user model interoperability is presented and refined in the application area of education. A selection of education-based user models and domains were identified to support in-depth development and evaluation. The FUSE domain-aware approach is presented, including its two domain-aware mechanisms; a canonical user model and user model mapping types. Finally, the FUSE approach is applied in two detailed processes; a user model mapping process and a user model translation process. The combination of the domain-aware approach in both user model mapping and translation processes is novel and differentiates this work from the related state of the art approaches.
- **Chapter 4** describes the design, implementation and evaluation of the first cycle of the FUSE Mapping Tool. The Mapping Tool applies the FUSE user model mapping process. The design methodology and architecture of the tool are presented followed by its implementation which includes the development of various user models and mapping types. The preliminary analysis of the usability of the tool is then described using the results of semi-structured interviews performed with early users of the prototype Mapping Tool.
- **Chapter 5** describes the design, implementation and evaluation of the second cycle of the FUSE Mapping Tool. This includes the use of the preliminary evaluation results to make enhancements in the development of the second version of the Mapping Tool. A walkthrough of the Mapping Tool functionality is presented using the selected user models to demonstrate user model mapping examples. Finally, two experiments are detailed where the domain-aware Mapping Tool is compared to a generic approach applied using a commercial mapping tool. The first experiment compares the accuracy and duration in both tools using tasks completed by users. The second experiment compares the usability in both tools using a selection of questionnaires. The results of both of these experiments demonstrate improvement in key

aspects of the user model mapping process.

- **Chapter 6** describes the design, implementation and evaluation of the FUSE Translation Service. This service applies the FUSE user model translation process. The architecture and implementation of the tool is described in detail. A walkthrough of the tool functionality is presented using the selected user models to demonstrate user model translation examples. An experiment is presented which analyses many different aspects of the performance of the Translation Service and compares them with recommended performance guidelines. The results of this experiment demonstrate efficient performance across multiple key aspects of the user model translation process.
- **Chapter 7** describes the functional comparison of the FUSE domain-aware approach, consisting of both the Translation Service and the Mapping Tool, with the state of the art approaches to user model interoperability. The state of the art approaches were comparatively assessed in Chapter 2 and the same categorisation of key issues is used to assess the FUSE approach. The results of this comparison demonstrate that FUSE provides a comprehensive approach to user model interoperability across a selection of key issues.
- **Chapter 8** describes the main conclusions drawn from this research. The research is assessed in terms of the original research question and objectives. The key contributions are described in the context of the field of user model interoperability. The contributions are focused on the FUSE domain-aware approach and tools and the benefits in key aspects of both the user model mapping and translation processes when compared against state of the art approaches and tools. Finally, some potential areas for future work are presented followed by some final remarks about the research.

Chapter 2

User Model Interoperability

2.1 Introduction

Providing interoperability between heterogeneous data models is a complex topic which encompasses elements from a number of related fields. It consists of many general interoperability issues that have been extensively examined from the perspective of relational databases [Sheth and Larson, 1990] and, more recently, the field of ontology engineering [Shvaiko and Euzenat, 2012]. User model interoperability is a subset of these generic approaches and attempts to reconcile the issues at a finer-grained level based around the user.

This chapter details the current state of the art in user model interoperability. Firstly, it describes some of the main aspects, methods and tools required for interoperability. It also details approaches to user modelling and the key issues in the provision of user model interoperability. A selection of current state of the art approaches to user model interoperability are then described. Using the identified key issues, the state of the art approaches are reviewed and compared to identify the main strengths and weaknesses in the current research into user model interoperability. Finally, a number of evaluation metrics are presented that can be used in the evaluation of user model interoperability approaches.

2.2 Interoperability Definitions and Methods

The provision of interoperability is a broad and complex field that has produced various terminologies and methods. Wegner [Wegner, 1996] describes interoperability as:

“The ability of two or more software components to cooperate despite differences in language, interface, and execution platform.”

Sheth [Sheth, 1999] describes four different levels of interoperability:

1. **System interoperability** refers to the ability to exchange data across different software and hardware boundaries.
2. **Syntactic interoperability** refers to the ability to exchange data across differences in the machine-readable aspects of data representation formats.
3. **Structural interoperability** refers to the ability to exchange data across data modelling constructs of different arrangement and granularity.
4. **Semantic interoperability** refers to the ability to exchange and interpret across differences in the meaning of information.

These interoperability levels have been the focus of significant research in different fields. In particular, relational databases were the source of much of the early research into data interoperability. The emergence of the web and web-centric standards such as XML have greatly pushed the boundaries of interoperability at all levels by adding syntax and structure. More recently, the move towards the semantic web and the emergence of new data standards with a focus on semantics such as RDF [World Wide Web Consortium (W3C), 2012b] and OWL [World Wide Web Consortium (W3C), 2012a] has seen significant focus on semantic interoperability.

Additional higher levels of interoperability have also been defined, for example, pragmatic interoperability [Asuncion and Van Sinderen, 2010]. This states that the resolution of system, syntactic, structural and semantic heterogeneity is not sufficient to achieve complete interoperability. Understanding how the data is used in another system and the intended versus actual effect of data exchange is also necessary. However, there is still significant debate about the concept of pragmatic interoperability and, as a result, it is considered outside the scope of this research.

Attempts to provide syntactic, structural and semantic interoperability have generally supported two high-level methods. The first is the use of shared common representation of information to which all interoperable systems adhere. For widespread use, this can often take the form of a specification or standard defined by a body such as the ISO [International Organization for Standardization (ISO), 2012]. An example in the area of user information is the IMS Learner Information Package (IMS LIP) [IMS Global Learning Consortium, 2012]. This specification supports the definition of some common user information constructs in the area of education, for example, learning competencies and preferences. However, heterogeneity at the syntactic, structural and semantic levels will still occur as not every system will fully adhere to the standard. At the semantic level, the development of ontologies is an attempt to provide shared explicit representations of domains for interoperability. However, as a single overall ontology is infeasible, multiple ontologies for the same domain concepts can also exist requiring interoperability across heterogeneous ontologies [O'Sullivan et al., 2007].

The lack of development and adoption of standards and shared representations in many areas has led to other methods being developed to provide interoperability. The second primary technique that is widely used is mediation or mapping between different data models. This method is used to provide interoperability in various fields such as database and ontology engineering and, as a result, varying terminology has emerged. Kalfoglou and Schorlemmer define many of the terms used in the context of ontology mapping [Kalfoglou and Schorlemmer, 2003]. They include:

1. **Matching** is the process of identifying matches between individual data model elements and often involves assigning a confidence score for each match.
2. **Mapping** is the process of creating the actual equivalence between the data model elements using a mapping representation format.
3. **Alignment** is the process of specifying all of the appropriate mappings between two data models.

Matching is typically an automatic process and many algorithms and tools have been developed to support it such as the Alignment API [David et al., 2011] and S-Match [Giunchiglia et al., 2004]. These tools take different data models and derive a list of matches including a confidence score. These matches can then be used to create mappings in an executable format such as XQuery [World Wide Web Consortium (W3C), 2012e] or XSLT [World Wide Web Consortium (W3C), 2012f]

in the case of XML-based data models. For RDF and OWL-based models, languages such as SWRL [World Wide Web Consortium (W3C), 2012c] and EDOAL [David et al., 2011] have been used.

However, due to the heterogeneity of real-world data models, the automatic matching process can rarely specify a full alignment [Falconer and Storey, 2007]. As a result, manual intervention is usually required in the mapping process. This often takes the form of an expert administrator manually examining the data models for matches and creating mappings. This can be done at a low-level using a translation language such as XSLT, but for large numbers of mappings this can become an extremely complex and intensive task. As a result, many tools have been developed to support users in specifying these mappings [Kalfoglou and Schorlemmer, 2003] [Haslhofer, 2008]. Manual mapping tools use graphical methods to make it easier to analyse data models and create mappings.

Different tools have been developed for different purposes such as database, XML and ontology mapping. In the following section, four mapping tools that can support XML-based data models are described, particularly from the perspective of their visualisation techniques.

2.2.1 Mapping Tools

To examine mapping visualisations, four mapping tools were selected which represent both commercial and research approaches. The selected commercial tools were Altova Mapforce [Altova, 2012] and Stylus Studio [Progress Software, 2012]. The selected research mapping tools were COMA++ [Aumueller et al., 2005] and PROMPT/CogZ [Noy and Musen, 2003] [Falconer, 2009]. The commercial tools were selected because they are consistently cited within related research as being among the most common and comprehensive commercial mapping tools available [Haslhofer, 2008] [Bernstein et al., 2006]. The research tools were selected as they are two of the most highly cited research approaches available.

1. **Stylus Studio** is a commercial development environment for working with XML. One of Stylus Studio's main capabilities is a mapping interface as shown in Figure 2.1. Source and target XML data models can be added to start the mapping task. The visualisation used to represent the source and target data models is the common hierarchical tree-based view. In this interface, the source model is displayed on the left side and the target model is displayed on the right side of the interface.

In the central area, the mappings are created graphically by dragging line connectors between equivalent elements of the data models. Mapping type boxes which represent XQuery functions can be selected and dropped in the central area. These mapping type boxes can then be connected to represent the creation of more complex mappings. There are a wide range of different generic mapping types available in Stylus Studio to perform various manipulations of data.

The graphically created mappings can then be validated using testing. This is performed by executing the mappings to perform a translation between sample instances of the data models. Sample instances can be provided for each source and target and when the mappings are executed the resulting translation of data is displayed in an updated instance of the target data model.

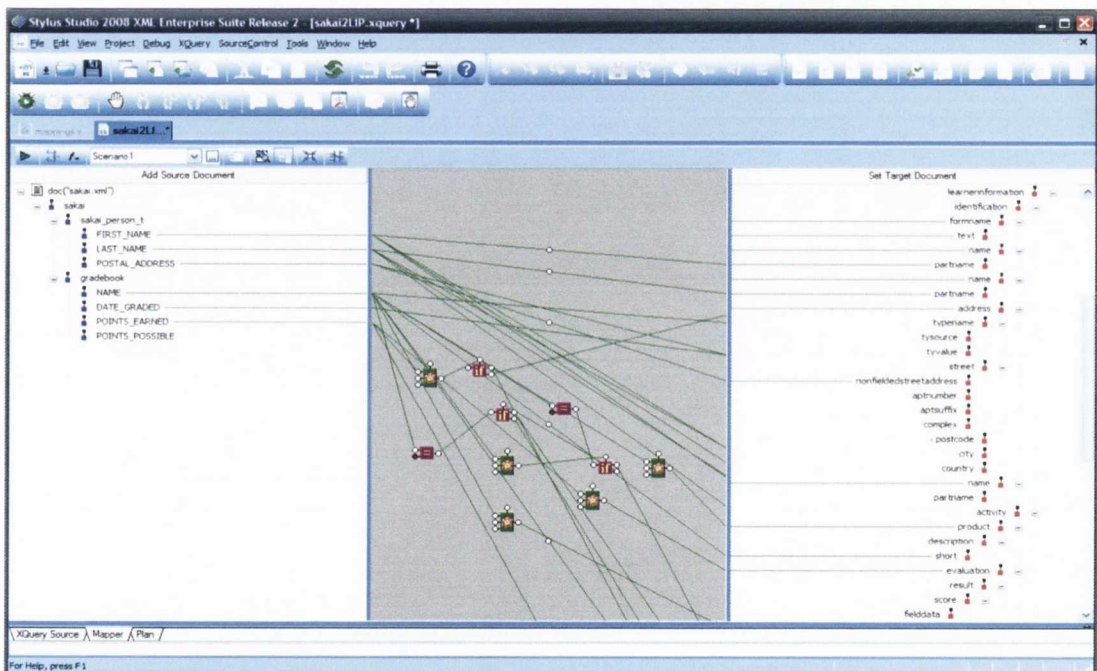


Figure 2.1: Stylus Studio

2. **Altova Mapforce** is another example of a commercial system and has similar functionality to Stylus Studio. Again, it uses tree-based views for visualisation of the XML source and target data models as shown in Figure 2.2. However, the tree visualisations are placed on a larger main canvas area which allows them to be moved and positioned to the user's preference. The main canvas area also allows multiple source and target data models to be visualised and mapped concurrently.

Mappings are created by connecting lines between matching elements of the tree-based source and target data models. More complex mappings can be created by selecting required mapping types from a list and dragging and dropping boxes which represent the mapping types onto the main canvas. Multiple mapping types can be added to the canvas and lines can be connected between the various inputs and outputs to the source and target data models. Mapforce contains a wide range of different generic mapping types to perform various manipulations of data.

The graphical mappings can then be validated using testing. Similarly to Stylus Studio, the mappings are executed to perform a translation between sample instances of the source and target data models. When the mappings are executed the resulting translation of data is displayed in an updated instance of the target data model.

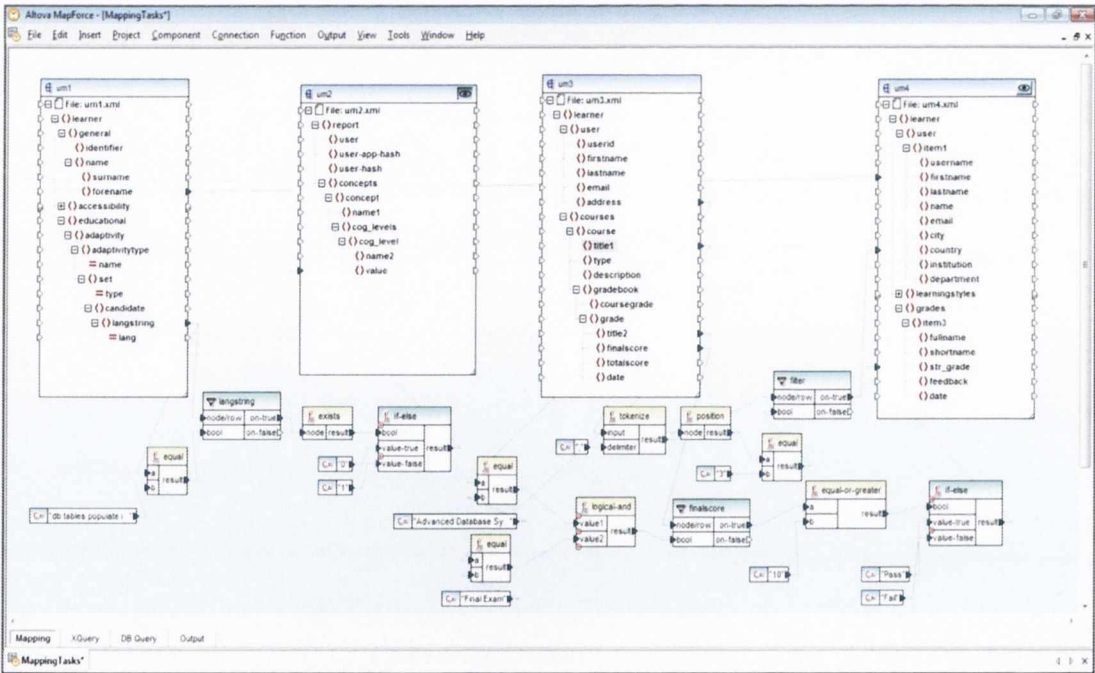


Figure 2.2: Altova Mapforce

3. **COMA++** is one of the most highly cited tools that has emerged from mapping-based interoperability research [Aumuellner et al., 2005]. It can support both schema and ontology models and utilises tree-based visualisations as shown in Figure 2.3. The interface layout is similar to Stylus Studio and Mapforce, consisting of two hierarchical tree visualisations with lines connecting equivalent elements.

The tool is semi-automated and generates candidate matches using selected matching algorithms. As it is a matching-based tool, it does not have a central area to graphically create mappings. The user can review and edit the resulting matches using text-based inputs in the user interface.

COMA++ is a matching-based tool for schema and ontologies and does not provide a testing area for instance data translation. It offers a number of functions to manipulate the mappings such as determining differences and commonalities between mappings, merging mappings, transitively composing matches and comparing mappings. The mappings are maintained in a repository and can be imported and exported for use in various interoperability tasks.

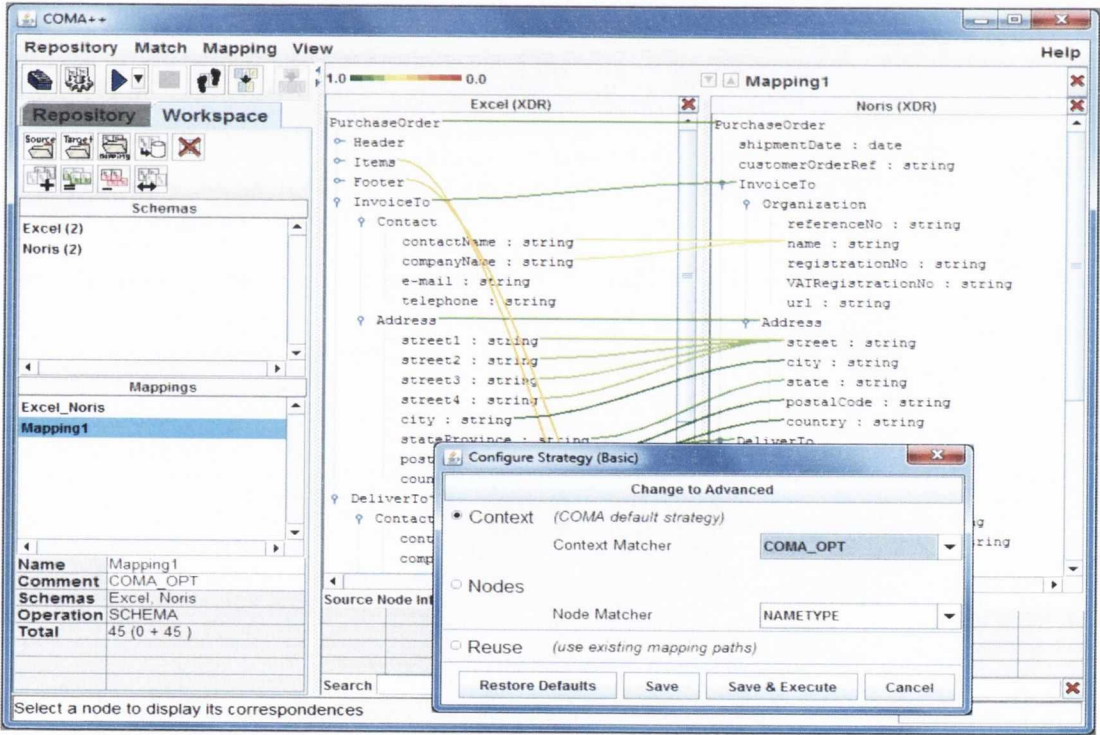


Figure 2.3: COMA++

4. **PROMPT/CogZ** is another highly cited research tool. PROMPT is a plug-in for the Protege ontology editor which allows the management of multiple ontologies, including the creation of matches and mappings [Noy and Musen, 2003]. PROMPT uses a simple text-based visualisation and allows the user to specify two ontologies to match. It then presents a number of candidate matches in a list view that the administrator can review and add to as needed. The tool supports equivalence-based matching and mapping of ontology concepts.

Extending this work, Falconer [Falconer, 2009] has developed CogZ which adds enhanced visualisations to PROMPT as shown in Figure 2.4. It uses a combination of tree-based and graph-based views to display the two input ontologies. The graph-based display allows the ontologies to be viewed as a series of boxes connected with lines that can be expanded to navigate through the ontologies and find matches. The identified matches and the candidate matches suggested by PROMPT are combined visually using the common tree-based view.

PROMPT/CogZ is an ontology matching and mapping tool and does not provide a testing area in the same way as Stylus Studio or Altova Mapforce. The main purpose of the tool is not to provide instance data translation but to align ontologies for potential use in a number of different processes such as ontology merging or transformation.

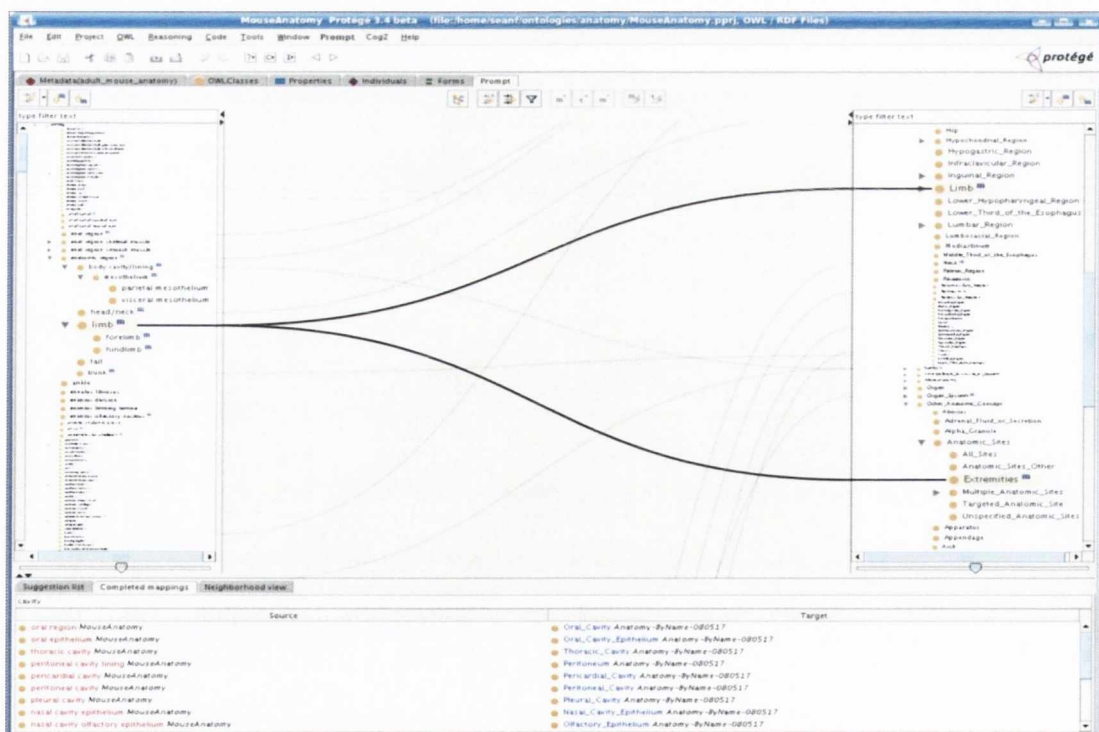


Figure 2.4: CogZ

In summary, it is usually not possible to provide the interoperability of heterogeneous data models fully automatically. As a result, mapping tools are often a key component in the successful delivery of any form of data interoperability. The tools reviewed in this section represent some of the most popular commercial and research approaches currently being used.

In terms of the overall visualisation of the source and target models, tree and graph-based approaches are the most common in both research and commercial tools. Some of the tools also provide concurrent visualisation of various source and target data models which supports multi-system interoperability.

In the creation of mappings, the commercial mapping tools are more focused towards data model instance translation. The research mapping tools have a greater focus on semi-automatic matching and mapping in schemas and ontologies. As a result, the commercial tools allow more complex mapping to be defined that can translate heterogeneous data model representations. They have a central area between the source and target models where mapping types are graphically defined and configured. The research tools provide equivalence-based mappings semi-automatically using matching algorithms. They are presented graphically or in a text-based form and can be edited by the user to create valid mappings.

The testing of the mappings is mainly conducted in the commercial mapping tools for the purpose of instance translation. Sample source and target data model instances are used to test the mappings and confirm that they perform the required translation. The research tools are mainly focused on aligning the schemas and ontologies for various purposes. However, providing some form of validation for the defined mappings would be a useful addition to these tools.

Overall, the intended purpose of interoperability tasks has a significant impact on the features and functionality provided in different mapping tools. The commercial and research mapping tools have different objectives and show differences in key aspects relating to complex mapping creation and testing. However, there are also many similarities in the various tools, in particular the visualisation approaches for source and target data models and the resulting mappings.

2.3 User Modelling

User modelling is the attempt to create structured representations of people within systems to allow the provision of better, more personalised user experiences. User models can take many

different forms depending on their purpose. This section outlines some of the current approaches to user modelling and how user models have evolved over time.

User Models

The **feature-based user model** approach attempts to model characteristics about individual users such as their knowledge, interests, goals, background and individual traits [Brusilovsky and Millán, 2007]. A feature-based user model can contain both static and dynamic user information, for example, identification information does not change frequently while in contrast, interests, goals and knowledge can be continuously changing as the user progresses over time.

In adaptive educational systems, user knowledge is particularly important as it forms the foundation for personalisation based on the user's understanding of concepts. The simplest way of representing the user's knowledge in a feature-based learner model is the **scalar model** approach [Brusilovsky and Millán, 2007]. This presents the user's knowledge as a single value on a scale. The scalar model is very basic as it represents the user's understanding as a single average score across an entire subject domain.

This has led to the development of more complex structural models that allow the representation of user's knowledge of multiple concepts in a subject domain. The structural user model is often formed using an **overlay model** approach [VanLehn, 1988]. This is where the concepts from the subject domain are overlaid with each user's knowledge of those concepts. This produces a user model that consists of a collection of attribute/value pairs where the attribute is the subject domain concept and the value is the user's knowledge of that concept. The user's knowledge can be represented as a qualitative value, for example good/average/poor, or as a quantitative value, for example 0-100.

Users can also have misconceptions in their knowledge which can be represented in feature-based user models. For example, the **constraint-based user model** approach represents knowledge as a set of constraints [Ohlsson, 1994]. Constraint violations on the part of the user indicate incorrect or incomplete knowledge. User models have also been developed to deal with uncertain or imprecise information. The **uncertainty-based user model** approach often uses probabilistic methods based on Bayesian Networks [Horvitz et al., 1998] or Fuzzy Logic [Kavcic, 2004] to represent user knowledge.

The **stereotype user model** approach attempts to assemble users into predefined groups and

provide personalisation based on those classifications [Brusilovsky and Millán, 2007]. This was a common approach in early adaptive systems as it allowed personalisation based on simple groupings such as the user's role. The stereotype approach is still used, often as a means to reduce the impact of the "cold start" issue. This occurs when there is insufficient user information present initially in the user models of new users thereby making it difficult to provide personalisation [Van Der Sluijs and Houben, 2006].

User models can also be represented as **system user profiles**. An example in education would be the learning management system (LMS). LMS are widely used systems for the centralised management of learning activities within educational institutions and as a result collect rich information about each user such as identification, preferences and assessments. LMS gather user information across a number of different tools and components, for example, assessment information in a gradebook tool. As a result, this user information is not always structured in a single profile representation and can be distributed throughout the LMS database. If this information is extracted it can be used by other educational systems and personalisation services. Similar examples of user profiles exist in administration and management systems in various other application areas.

To facilitate interoperability between systems, **standardised user profiles** have also been developed. An example in education is the IMS Learner Information Package (IMS LIP) [IMS Global Learning Consortium, 2012]. This specification is used to create a common representation of a user's learning history and preferences so that they can be more easily exchanged between different educational systems.

User models are also increasingly being created in the form of **social-based user profiles** on social networking websites such as Facebook and LinkedIn. These profiles are created and maintained by users for the purposes of sharing and connecting with other users. Recently, Facebook and others have begun to offer new forms of functionality such as personalisation based on the contents of the user profile. This functionality is made available to other websites and is an attempt to make social networks such as Facebook a centralised global source of user information.

Event-Based User Data

In addition to user models, many systems also gather less structured **event-based user data**. An example of this is the tracking of usage within a system by logging user actions. This data

can offer valuable information about a user's behaviour in a system but it must be analysed and potentially extracted into a more structured user model form. LMS are one example of this, as users interact with a series of different tools for different purposes while generating usage data within the system. Another example of this approach is in recommender systems where preference-based user models are defined based on the analysis of user actions [Berkovsky et al., 2008]. In this case, recommendations can then be provided based on the similarity between different user models.

Large amounts of user event data can be generated over time. Extracting and filtering this user data is becoming a more important challenge as the number of systems and components users access continues to increase. This has led to approaches that shift the focus from user models as single entities to **user model processes** that act across large collections of user data [McCalla et al., 2000]. It is suggested that traditional user modelling will become impractical over time due to the rapid growth of user data and defining and exchanging the processes for user modelling is a more manageable method in the long-term.

Overall, there are many different types of user models that are created for various purposes such as personalisation, standardisation and user scrutability. Exchanging user models between the various heterogeneous representations is important to enhance knowledge of the user and personalisation across systems in increasingly distributed environments. In this research, the focus is primarily on the structural, syntactic and semantic challenges in exchanging user models. The extraction and analysis of event-based user data is outside the scope of this research but it is a topic of interest for future work. The following section describes the key issues for the provision of interoperability between different types of user models.

2.3.1 Key Issues for User Model Interoperability

User model interoperability has been described as “the process of exchanging distributed user data across applications” [Carmagnola et al., 2011]. However, attempting to reconcile between the various heterogeneous types of user data is a significant challenge. A number of key issues for the successful provision of user model interoperability have been defined by Carmagnola [Carmagnola et al., 2011]. These key issues have been generated from a review of state of the art approaches and encompass a wide range of areas that affect the provision of user model interoperability. In this section, eight identified key issues are described and explored in detail and, in later sections,

will be used to comparatively analyse a selection of state of the art approaches to user model interoperability. The eight key issues are:

1. **Interoperability task** describes the intended purpose of the interoperability system and the main function that will be performed. This is one of the most fundamental aspects of any interoperability approach as the main function and purpose of the system dictates the relevant designs and methods that are used to provide interoperability. Generally, approaches have either facilitated the exchange of user models or provided centralised user modelling/adaptation services that store and maintain user information for individual systems.
2. **Interoperability architecture** specifies the overall architectural design that is used to exchange the user models. This key issue is closely related to the interoperability task. The main approaches are the centralised, decentralised and mixed architectures.

In the centralised approach, a single central system manages the user modelling functionality for multiple distributed systems. The benefit of this approach is that information from multiple systems is easily shared and analysed because it is stored in a shared location using the same user model representation. However, each system is typically developed to use this central repository and not retain user models locally. The most common form of the centralised approach is the user model server [Carmagnola et al., 2011].

In the decentralised approach, the distributed systems maintain their own individual user models and attempt to exchange information directly with each other. This can lead to much greater heterogeneity between user information that must be rectified in some way such as the use of a standardised user model or a translation process. The decentralised approach is potentially more flexible than the centralised approach, allowing systems to maintain their own user models while avoiding the need for the architecture to use a central repository. The disadvantage to this approach is that it is significantly more complex, placing more of a burden on each system to manage the exchange of user information directly.

There are also approaches that use aspects of both the decentralised and centralised methods to create mixed approaches. The mixed approach allows external systems to maintain their own heterogeneous user models but exchange using a centralised system often using a shared user model. The benefit of this approach is that it provides the flexibility of the decentralised approach but reduces the complexity as the exchange is performed via a single location.

3. **Protocols and languages for communication** describes the methods used to exchange the

user models between systems. As more web-based systems that employ user models are being developed, HTTP and web service protocols such as SOAP and REST are becoming more widely used as methods for the exchange of user models.

This key issue also describes the languages for the exchange of user model data. Data interchange formats such as XML are common in existing systems and standards [IMS Global Learning Consortium, 2012]. In general, JSON is also becoming a popular alternative to XML due to its ease of use [Nurseitov et al., 2009]. The development of semantic web technologies has led to more recent research approaches using RDF and OWL to represent user models. In some cases, user model specific languages such as UserML [Heckmann and Krueger, 2003] have also been developed.

4. **Representation of data exchange** describes how the user models are represented during the exchange process from a syntactic and semantic perspective. The issues of syntactic and semantic heterogeneity are well-known from related research into interoperability in relational databases, XML schemas and, more recently, ontologies. As a specific subset of these areas, user model interoperability is also affected by the same issues. The intended purpose of the user model can vary depending on the type of system that incorporates it. This can be a fundamental difference greatly influencing the information, the way it is represented and the potential to translate that information. User models are also designed by different people who use different terminology to describe the information. This results in user models of varying complexity and expressiveness, which use inconsistent vocabularies and have different structures to the user data. Attempts to overcome these issues have commonly involved either the adoption of a common user model representation such as IMS LIP [IMS Global Learning Consortium, 2012] or GUMO [Heckmann et al., 2005] or the use of a translation approach which uses mediation components to resolve heterogeneities and translate user information between heterogeneous systems. The combination of common models and translation is also another possible approach [Berkovsky et al., 2009].
5. **Exchanged data type** specifies the types of data that can be exchanged by the interoperability system. The type of system ultimately determines the representation and content of the user model and, as a result, the user data can be varied. In many cases, user models consist of feature-based user information in areas such as identification, knowledge, skills, interests and preferences [Brusilovsky and Millán, 2007]. However, user data can also extend into other areas, for example, usage data which is the actions of individual users and how they use various

tools, environment user data which is related to a user's context such as location and device, domain data which is the data related to the domains of the system, inferred data which is new data that has been generated by the system for a particular function such as personalisation, reasoning data which is data that describes where the inferred information was generated, for example the reasoner used, and social data which is related to a user's social connections to other people and groups.

6. **Integration of exchanged data** describes the approach of incorporating the user data into the systems. In general there are two high level approaches, either the exchanged user model data is not merged into a system's existing user data but is just retrieved as needed or the exchanged user model data is merged into existing user data.

The first approach of not merging requires an on-demand exchange of user data. This method can be achieved using entire user models or individual fragments of user model data. Exchanging the entire user model is a simpler approach where an application can request and receive an entirely updated user model at any time. This approach is much more performance intensive if there are a large number of user models being exchanged at the same time. The exchange of user model fragments occurs when small sections of user models, which have been updated by an end-user, are exchanged with other systems' user models. This approach is more complex but also more flexible and less performance intensive as updates can be provided to the user model on a continuous basis.

The second approach of merging requires the resolution of conflicts with existing data. Providing interoperability of multiple user models creates issues of overlapping and incomplete user data. When user data from several systems is combined and shared with other systems it can overlap in places. This overlap can result in repeated or conflicting user data which can cause errors in the interoperable systems. In contrast, when user data from multiple systems is combined and shared there is also a potential deficiency of user data. This occurs when there is not enough data about a particular user to fully populate the instance of a system's user model which can result in errors in the interoperable systems.

7. **Privacy** describes the protection of a user's privacy when their information is being exchanged between multiple systems. Many methods have been developed to provide privacy in user model interoperability. Some of the methods include using different access rights for different types of users, encryption of user data, perturbation where some changes are made to the values

in the exchanged user model, scrutable user models, third-party guarantors, and pseudonymous personalisation where the user data is stored using non personally identifiable keys.

8. **Pre and post interoperability task** encompasses some of the other key issues that exist outside of the main exchange process of user model interoperability. They are tasks performed before and after the exchange process such as the discovery of new systems for interoperation, the initial matching of user identifiers across systems, and the evaluation of any improvements in systems that retrieve and apply updated user models.

This research is fundamentally focused on the core technical, syntactic and semantic challenges in overcoming user model heterogeneity. As a result, the final two key issues, privacy and pre and post interoperability task, are considered outside the scope of this thesis. However, they are included in the review of state of art approaches in the next section in order to gain a broader and more comprehensive understanding of these approaches and their methods. Overall, to provide advanced forms of user model interoperability many of the outlined key issues must be addressed. In the following sections, the key issues are used as the basis to describe and compare the main state of the art approaches.

2.4 User Model Interoperability Approaches

Ten approaches have been selected for review from the related state of the art research. They represent the most influential current approaches and demonstrate the diverse nature of the research in this field. In the following sections, a summary of each approach is presented and a comparison is performed using the key issues for user model interoperability.

2.4.1 Personis

Kay et al. [Kay et al., 2002] developed a centralised user model server called Personis to provide user modelling for adaptive systems. Personis consists of a database that applications can send and receive user information using a custom API. The user information is submitted in the form of a list of evidence for individual components. User information can then be retrieved using views which allows a specified collection of user information to be gathered and returned to the requesting application. One of the main goals of this system is to provide a greater focus on scrutability and security. As a result, users can view and edit their user information using user interfaces. The user

information also includes additional information to control the access of applications and users. Communication between the client applications and the server is performed using the XML-RPC protocol over SSL. More recently, Assad et al. have extended this system to provide PersonisAD, a framework for building context-aware, ubiquitous applications [Assad et al., 2007]. This approach allows scrutable modelling of people, sensors, devices and places and supports distributed models with active elements which can trigger when relevant events occur.

2.4.2 CUMULATE

Brusilovsky et al. [Brusilovsky et al., 2005] have created an approach based on a number of distributed, reusable learning activities that use a centralised student modelling server called CUMULATE as the common repository for all user information. CUMULATE collects learning events from each activity and stores them in a single event repository. Events can consist of any component-specific information such as the user's progress in a specific step within a learning activity. These events can then be processed to generate inferenced user models which can represent information such as the user's interests and knowledge in name-value pairs. These newly created user models are accessible by external applications. The event reporting and exchange of data is performed using a HTTP-based protocol combined with XML representations of the user models. Yudelson et al. have extended this approach to support push-based propagation of user information to improve performance with larger user models [Yudelson et al., 2007].

2.4.3 GUMF

Abel et al. [Abel et al., 2009] [GRAPPLE, 2012] have developed a user modelling framework for distributed applications called GUMF, the GRAPPLE User Modelling Framework. GUMF uses a centralised RDF-based repository to collect and store user information from external applications. These applications can also maintain their own user models and only share selected user information. It contains a web interface to register and configure new applications for sharing. Registered applications can submit user information using RDF-based GRAPPLE Statements. The user can then configure GRAPPLE Derivation Rules (GDR) which can be applied to perform a number of tasks. User profile reasoning can be conducted where a description of a user can be deduced from the statements. Conflict resolution can be achieved where GDR rules can specify conditions for the selection of statements. GDR-based profile mapping can also be performed for

schema conversion by an administrator using a web interface. Queries of the user model repository can be executed using SPARQL and the user information in the repository can be viewed using a web interface.

2.4.4 UUCM

Mehta et al. [Mehta et al., 2005] proposes a common format to represent user information for cross-system personalisation called the UUCM, Unified User Context Model. The UUCM is an ontology-based user context model that applications can use to represent and share their user information between systems. To support user information in multiple systems, the model is designed to be flexible and extensible and can be structured to deal with a variety of personalisation approaches and various aspects relevant for capturing user characteristics and context. The UUCM encompasses an extensible set of facets which represent various user information. Both the names and values of user facets can refer to vocabularies or ontologies. This allows the use of shared vocabularies between different interpretations of the UUCM and, as a result, potentially supports the interpretation of heterogeneous user information across multiple systems.

2.4.5 GUMO/UserML

Heckmann et al. [Heckmann et al., 2005] [Heckmann and Krueger, 2003] have developed an approach to exchange user models between different systems which is based on the combination of GUMO and UserML. UserML is an RDF-based user model exchange language and was developed to allow systems to communicate user model information in a syntactically uniform way. GUMO is an attempt to define an ontology which represents a collection of the main user dimensions. Both GUMO and UserML are based on the concept of the division of user model dimensions into three parts, auxiliary, predicate and range. For example, the user's interest in football could be expressed as; auxiliary=hasInterest, predicate=football, range=low-medium-high. GUMO defines a number of user dimensions using this approach. Both GUMO and UserML are applied in a user model service which allows distributed applications to exchange user information. The concepts of multiple heterogeneous applications are mapped to the GUMO ontology to provide a unified understanding of the user information. The applications can then retrieve or add information to the server using UserML-based HTTP requests.

2.4.6 GUC

Van der Sluijs and Houben [Van Der Sluijs and Houben, 2006] have proposed a system called GUC, the Generic User model Component. This approach utilises semantic web technologies to support the exchange of user model data between applications. GUC provides a centralised user model server and applications can subscribe to request and upload user data. GUC supports user models represented in OWL and to support interoperability between heterogeneous representations it allows the configuration of mappings between user models. For the most common user model information, mappings are created to a shared user model such as GUMO and additional mappings can then be created directly between different systems' user models. The mappings are represented using SWRL and stored in a mapping repository where they can be made available to the translation service component. The mappings can be generated using tools such as COMA++. Scrutability is also provided through a user model editor which allows users to access and edit their information in the central user model server.

2.4.7 Dolog and Schafer

Dolog and Schafer [Dolog and Schafer, 2005] have developed a framework which provides a common representation to exchange user models between multiple sources. The learner profile exchange model is based around a shared learner ontology which is derived from a number of standards such as IMS LIP and is represented in RDF. To participate in an exchange of learner profiles, external applications with different internal data models have to provide mappings between their internal data model and the exchange model. An API is designed and implemented to create, export and manipulate learner profiles. The API is implemented in two ways, as a Java API that can provide access between distributed applications and as a web service which acts as a user model server. Another approach which is proposed is the exchange of user information through a peer-to-peer environment consisting of a query-based infrastructure for RDF-based user models.

2.4.8 Carmagnola and Dimitrova

Carmagnola and Dimitrova [Carmagnola and Dimitrova, 2008] have proposed an evidence-based approach to support semantic interoperability and provide automated user model exchange

across applications. This approach uses semantic web technologies and evidential reasoning techniques to map heterogeneous user data between applications. An initial SeRQL [Broekstra and Kampman, 2003] query is performed by a receiver system across multiple provider systems to retrieve user models for a particular user. The provider systems submit their user models to the receiver system in the form of an RDFS user model and domain ontology. The retriever system then uses evidential reasoning algorithms to determine the relevancy and credibility of the various responses when compared against local ontologies or WordNet [Fellbaum, 1998]. To measure the relevancy of the user model statements, the algorithms calculate semantic similarity of both the object and property of the statements. To measure the credibility of the user model statements the algorithms use descriptions of both the provider and statement to determine values of trust in the each statement. Both the relevancy and credibility measures are then used to select the best response to use in the retriever system's user model.

2.4.9 Cena

Cena [Cena, 2011] has proposed an approach to user model interoperability that utilises web services and semantic technologies to provide negotiated communication between systems. Two systems can implement a dialogue model to clarify the meaning of requested concepts. The main components in the dialogue model are communicative acts, dialogue rules, and dialogue games. The communicative acts represent the requestor system, the responder system, the move (the intention of the communication), and the statement (concepts in the user and domain models). Dialogue rules define the allowed moves in communicative acts and how to sequence them. Dialogue games are templates defining the sequence of communicative acts to be exchanged for reaching a particular goal. This approach has been developed in a semantic conversation framework for user model interoperability on the web. The interoperable systems using the framework should express their user models and domain models in RDFS and should refer to external ontologies such as GUMO for other user and domain models. Partial overlap of user information is possible and ambiguities in concept meaning can be addressed by using the dialogue model.

2.4.10 Berkovsky

Berkovsky [Berkovsky et al., 2008] proposes a general framework and specific methodologies for enhancing the accuracy of user modelling in recommender systems by importing and integrating data collected by other recommender systems. This approach is defined as user model mediation and is applied in the form of a user model mediation framework. It provides a generic user data representation model for recommendation systems based on three dimensions: user, item and context. It also describes four major types of mediation: cross-user, cross-item, cross-context, and cross-representation. The framework is applied in the form of a mediator component which performs the exchange of user data between the requesting target system and the source systems. The mediator consists of two principal components. The integration mechanism is a component responsible for resolving conflicts and heterogeneities in the obtained user modelling data using various reasoning and inference mechanisms. The knowledge base is a component used by the integration mechanism. It contains semantically enhanced inter-domain and intra-domain knowledge bases representing dependencies and relationships between various user, item and context features. The data stored in the knowledge base facilitates resolving the heterogeneities in the obtained user modelling data.

2.4.11 Comparison of Surveyed Approaches

A variety of approaches to user model interoperability have been attempted. These approaches have had different objectives and as a result they have examined the key issues for user model interoperability in various levels of detail. In this section, the state of the art approaches are analysed and compared using the user model interoperability key issues described in Section 2.3.1. The comparison of the state of the art approaches is summarised in Table 2.1.

Table 2.1: Comparison of surveyed approaches

Approach	Interoperability task	Interoperability architecture	Protocols and languages for communication	Representation of data exchange	Exchanged data type	Integration of exchanged data	Privacy	Pre and post interoperability task
Personis [Kay et al., 2002] [Assad et al., 2007]	User modelling service	Centralised	HTTP JSON	Common user model	User, environment, inferred	No merge of data	Different access rights, scrutable user model	Service discovery
CUMULATE [Brusilovsky et al., 2005] [Yudelsohn et al., 2007]	User modelling service	Centralised	HTTP XML	Common user model and translation	User, usage, domain, inferred	No merge of data	n/a	n/a
GUMF [Abel et al., 2009]	User modelling service	Centralised	REST/SOAP RDF	Common user model and translation	User, usage, environment, domain, inferred	No merge of data	Different access rights	Service discovery
UUCM [Mehta et al., 2005]	Exchange user models	Decentralised	HTTP CSCP	Common user model	User, usage, environment, social	No merge of data	Encryption [Mehta, 2007]	User identification [Mehta, 2008]
GUMO/UserML [Heckmann and Krueger, 2003] [Heckmann et al., 2005]	Exchange user models	Decentralised	HTTP UserML, OWL	Common user model	User, usage, environment, inferred	No merge of data	Scrutable user model [Heckmann, 2005]	n/a
Carmagnola and Dimitrova [Carmagnola and Dimitrova, 2008]	Exchange user models	Decentralised	HTTP RDF, RDFS	Translation	User, usage, domain, inferred, reasoning	Merging of data	n/a	User identification [Carmagnola and Cena, 2009], user model evaluation

Approach	Interoperability task	Interoperability architecture	Protocols and languages for communication	Representation of data exchange	Exchanged data type	Integration of exchanged data	Privacy	Pre and post interoperability task
Cena [Cena, 2011]	Exchange user models	Decentralised	SOAP RDF, RDFS	Common user model and translation	User, domain	No merge of data	n/a	Service discovery, user identification [Carmagnola and Cena, 2009]
Dolog and Schafer [Dolog and Schafer, 2005]	Exchange user models	Centralised or Mixed	SOAP RDF, RDFS	Common user model	User, usage, environment, domain, inferred	Merging of data	n/a	Service discovery
GUC [Van Der Sluijs and Houben, 2006]	Exchange user models	Mixed	HTTP OWL	Common user model and translation	User, usage, environment, inferred	No merge of data	Scrutable user model	n/a
Berkovsky [Berkovsky et al., 2008]	Exchange user models	Mixed	n/a	Common user model and translation	User, usage, environment, inferred, social	No merge of data	Perturbation [Berkovsky et al., 2007]	User model evaluation

In the following sections, the analysis of the surveyed approaches is described in more detail for each of the eight key issues:

1. **Interoperability task** specifies the intended purpose of the interoperability system. In this review, the selected systems are categorised as either providing user modelling or adaptation services across multiple systems or exchanging user models between multiple systems.

Personis, CUMULATE and GUMF have a goal of providing a user modelling service across multiple systems. Personis is more focused on providing user modelling services to individual components and does not provide a single combined user model representation or methods to merge external system's user models. CUMULATE and GUMF provide user model services for other components and also allow merging of other user models into a single, centralised user model representation.

The remaining approaches have a goal of providing exchange of user models between multiple systems. UUCM and GUMO provide common representations that allow systems to exchange user models. Carmagnola and Dimitrova's approach, Cena's approach, GUC and Berkovsky's approach all attempt to provide different forms of translation between user model representations.

2. **Interoperability architecture** specifies the overall architectural design. There are three architecture types commonly applied in the development of user model interoperability approaches; centralised, decentralised and mixed.

In early systems, the centralised user model server was the most common architecture type. Centralised user model servers such as Personis and CUMULATE have advanced this approach and provide some of the most fully developed implementations currently available. Other notable approaches which are similar to those surveyed include MUMS [Brooks et al., 2004] and UMS [Kobsa and Fink, 2006]. GUMF is a more recent approach and provides a greater focus on semantic interoperability through the use of RDF. Some approaches, such as CUMULATE and GUMF, have attempted to expand the approach by allowing systems that maintain their own user models to interoperate with the centralised server.

A decentralised architecture has also been applied in a number of approaches. UUCM and GUMO are common user model representations that allow participating systems to have the same understanding of the user information and, as a result, exchange it directly. Other approaches such as those by Carmagnola and Dimitrova and Cena apply mediation techniques

to allow systems to derive a common understanding of user information so that it can be translated directly between them.

Finally, the mixed architecture uses aspects of both the centralised and decentralised architectures, allowing systems to maintain their own individual user models but share information by referring to a common user model representation. Mixed approaches include GUC which uses mappings between heterogeneous user models to translate user information using a central component and Berkovsky's approach which uses a central mediator component.

3. **Protocols and languages for communication** are common techniques and technologies that allow systems to interact and exchange user model information. Some older approaches used remote procedure calls to access user models in different systems. Other methods have also been attempted, such as the use of LDAP as a basis for communication with a user model server [Kobsa and Fink, 2006]. However, the evolution of web-based technologies has made it significantly easier to share user models using a common communication protocol. The most basic form is the use of HTTP to request and send user models between systems. Personis is based on simple HTTP-based requests using JSON as the data representation. CUMULATE uses simple HTTP GET/POST requests combined with XML responses. More complex forms of web-based communication are also employed through the use of web service technologies such as WSDL/SOAP. In Dolog and Schafer's approach, the system can be set up as a user model server and accessed using web service protocols. Cena's approach requires WSDL/SOAP-based communication and also includes a UDDI registry to aid discovery of new services. The GUMO/UserML approach allows applications to communicate with HTTP or a UserML-based web service. UUCM uses the Cross System Communication Protocol (CSCP) which employs XML and web service functionality to exchange user models [Mehta, 2008].

To represent the user model data in the exchange process, a number of different formats are employed. CUMULATE and Personis use XML and JSON respectively. Semantic web languages such as RDF are also widely used in the various research approaches. These allow for more explicit semantics to be defined and provide potential benefits such as reasoning about the user. GUMF is largely based on user data in RDF, as is Carmagnola and Dimitrova's approach and the closely related approach by Cena which both use RDF and RDFS to represent user data. GUMO and GUC provide user models developed in the ontology description language OWL.

4. **Representation of data exchange** describes how the user models are represented during the exchange process. Two different approaches are generally used to represent the exchange of user models: a common user model or a translation approach. The common user model is a shared representation of the user model that is applied across the interoperable systems. The translation approach allows systems to maintain their own user models and translate between the different data representations.

Approaches such as UUCM and GUMO represent the user model in the form of an ontology that systems can use to share a common representation of user data. Another common user model representation based on existing user model standards such as IMS LIP was developed in Dolog and Schafer's approach. In this approach, tools that maintain their own user models are responsible for the provision of mappings to the common user model.

Some approaches have combined aspects of both the common user model and the translation approach. In CUMULATE, there is a common domain model ontology which is the basis for the user model. This approach has been developed further to support a translation approach to take user information from other systems. GUMF deploys a centralised user model server approach where there is a common representation of the user information. However, GUMF also provides a translation approach through a mapping tool to translate from other user model representations into its centralised repository. Cena's approach also uses a combination of the common user model and the translation approach. Two heterogeneous systems share some common understanding of the user models using external knowledge models such as GUMO and other domain-specific models. However, there can still be ambiguity in the exchange of user data that the systems can attempt to resolve by communicating directly and negotiating the meaning of terms. GUC adopts a mapping-based approach but uses a common mediator component where mappings can be created to a common user model and which manages the exchange of user data. More complex mappings can also be created directly between systems. Berkovsky's approach uses a mediator component to exchange user data. It relies on a knowledge base to manage the exchange between heterogeneous user models. The knowledge base represents dependencies and relationships between various user, item and context features and allows conversion of terms to a common representation.

Carmagnola and Dimitrova's approach does not rely on a common user model representation and uses mapping algorithms that attempt to automatically identify and map equivalent user data directly between systems. This is achieved by exchanging the user and domain models as

RDFS ontologies and matching to the local ontologies in the receiver system.

5. **Exchanged data type** specifies the types of data that can be exchanged by the interoperability system. Carmagnola [Carmagnola et al., 2011] has defined a number of data types that include user, usage, environment, domain, inferred, reasoning and social data.

Personis supports information about users. PersonisAD adds environment data from sensors, devices and places. Inferred data is also supported through rules for the analysis of evidence. CUMULATE receives user data and usage actions that each interactive component provides. Domain data is a core feature of the system as the user knowledge is represented using an overlay model approach of the subject domains. Inferred data is also generated as the event storage in CUMULATE is open to a variety of inference agents that process this data in different ways and convert it into an inferred user model. GUMF, UUCM and GUMO represent user data such as knowledge and interests, usage data such as user actions and tasks and environment data such as the user's location and device. UUCM can also include relationships to provide the exchange of social data and, as an OWL ontology, GUMO also supports the exchange of inferred user data. GUMF can also infer new user data and exchange data about the subject domains.

Carmagnola and Dimitrova's approach supports the exchange of user data such as interests and knowledge about particular subject domains. Usage and inferred data are also supported. Cena's approach uses common models such as GUMO and domain-specific ontologies and, as a result, can exchange user information such as interests and knowledge and domain data. Dolog and Schafer's approach uses a common user model derived from the combination of aspects of various standards such as IMS LIP. This allows the approach to support the exchange of user data such as competencies, usage activity data, preferences relating to the user's environment, domain data and inferred data. GUC employs a mapping-based approach with a centralised user model server to provide the exchange of any type of user, usage, environment and inferred data. Berkovsky's approach supports the exchange of user and usage data related to items, the context and evaluation of that use in the form of environment and inferred data and social data in the form of user connections.

6. **Integration of exchanged data** specifies challenges related to the exchange of user data between multiple systems. Generally, systems can either retrieve user data as it is needed or they can merge the exchanged user data they have received into an existing user model

representation. Merging the exchanged user data requires methods to resolve conflicts. The integration of exchanged data is an important aspect for comprehensive user model interoperability. However, it is often seen as external to the central process of user model exchange and the responsibility of the receiving systems to manage the exchanged data [Carmagnola et al., 2011].

Most of the approaches do not provide methods to merge the exchanged user data. In these systems, the user data is either stored in a central repository or it remains with the provider systems. The exchange then happens on-demand, initialised by a request from the receiver system. It is suggested that GUC could support the removal of repeated and conflicting user data through intelligent mappings, for example mappings that use dates or precedence to select the most up-to-date data.

Some approaches do support the merging of the exchanged user data. Dolog and Schafer's approach proposes the exchange of evidence of how the user model was derived to allow other systems to determine how user model fragments contribute to an overall merged user model. Carmagnola and Dimitrova's approach uses algorithms to calculate a credibility for each user model statement. The statement with the highest credibility is selected in the event of conflicts in the merged data.

7. **Privacy** is a key issue when attempting to provide user model interoperability between multiple systems. As described in Section 2.3.1, several methods to provide user privacy have been defined: different access rights, pseudonymous personalisation, encryption, perturbation techniques, scrutable user models, and third-party guarantors [Carmagnola et al., 2011].

Personis is an approach which places significant focus on privacy. It gathers access control information which establishes access rights for applications and users. It also allows for scrutable components that enable the user to view and control their own information. GUMF provides different access rights to user information through a request and approval process between the various dataspace administrators. GUC also allows for scrutability via a user model editor which gives the user access to view and modify all their data within the system.

For some of the approaches, the authors have conducted relevant related research in the area of privacy. Mehta has examined encryption techniques [Mehta, 2007], Heckmann has proposed tools for user model scrutability [Heckmann, 2005] and Berkovsky et al. have proposed an approach based on user data perturbation [Berkovsky et al., 2007]. Other notable work in the

area of privacy is provided by Kobsa [Kobsa, 2007].

8. **Pre and post interoperability task** relates to a number of tasks that can be completed before and after an exchange of user data has been performed such as service discovery, user identification and evaluation of updated user models. Often these types of tasks are considered outside the core user model exchange process and, as a result, not many solutions have been presented.

Service discovery for user model interoperability allows systems to identify other relevant systems that can be used to exchange user data. Personis provides service discovery through the DNS-based Bonjour service discovery system. GUMF provides a central registry of all interoperable services to facilitate the discovery of appropriate systems to exchange user data. Cena's approach uses UDDI which registers services and allows them to be discovered by other relevant services to exchange user models. An approach based on the use of peer-to-peer technologies to identify user models in different locations is proposed in Dolog and Schafer's approach.

User identification is a key challenge that is a prerequisite to any exchange of user data. The same user must be identified in two heterogeneous system representations before their user models can be exchanged. In work related to the UUCM, Mehta defines a unified profile "Context Passport" that users bring with them across different systems. This unified profile allows the user to be identified in heterogeneous system representations [Mehta, 2008]. In related work by both Carmagnola and Cena, a set of identification properties are defined including an algorithm that uses the various properties to identify users across different systems. This approach is implemented in a centralised identification registry to which systems can subscribe to manage user identification [Carmagnola and Cena, 2009].

Another important aspect is the evaluation of user models which have been enhanced with user data from external sources to determine if they have been improved. Evaluations have been performed on both Carmagnola and Dimitrova's approach and Berkovsky's approach and both have demonstrated improvements in the quality of the user model after the exchange of user data has been performed.

Key Strengths in the Surveyed Approaches

There are overall key strengths that can be identified from the survey approaches. They include:

- **Diversity of interoperability methods** - The surveyed approaches have demonstrated a wide variety of methods and techniques for user model interoperability. This diversity is one of the main overall strengths in the surveyed approaches and provides many potential methods that can be selected based on individual interoperability objectives. As can be seen in Table 2.1, the interoperability task, the interoperability architecture and the representation of the data exchange are the key aspects that fundamentally define the approaches to user model interoperability. Approaches that provide a user modelling service across multiple sources generally use a centralised architecture. These approaches also typically maintain a common user model representation for all interoperable systems. Approaches that provide exchange of user models use either a decentralised or mixed architecture. These approaches use a common model or translation to exchange user models. While the surveyed approaches are diverse in their methods, many have been applied in the application area of education to provide a consistent structure for in-depth development and evaluation.
- **Emergence of common communication and representation methods** - While diversity in the methods provides a rich collection of different approaches, it is important that common communication and representation methods are applied to allow more widespread provision of user model interoperability. For the communication protocols, Table 2.1 shows that a small collection of standardised languages and protocols are now being used. Web-based approaches such as HTTP and web service technologies such as SOAP are the main methods to exchange user information between different systems. To represent user information in systems, web-based standards such as XML are common. Semantic web technologies based on XML are also widely used, in particular RDF. Some other approaches such as UserML and GUMO have also been specifically designed for the representation and exchange of user models and these types of approaches should be developed further to support richer forms of user model interoperability.

Key Weaknesses in the Surveyed Approaches

There are also overall key weaknesses that can be identified from the survey approaches. They include:

- **Lack of support for existing systems and user models** - Many of the surveyed approaches have been developed to perform user model interoperability between specially developed custom systems. These approaches are not always tested with or applicable to existing systems that manage user data or information. However, there are many existing systems that could provide and receive significant benefits from exchanging their user information. For example in education, learning management systems and student administration systems can be used to manage the learner data of large amounts of students within an institution. There are significant benefits in providing user model interoperability approaches that are designed to support these existing systems rather than only supporting specially designed custom systems.
- **Lack of complex exchange between user models** - The surveyed approaches that provide translation often allow only basic relationships to be defined between user models, for example, specifying that two concepts are equivalent. However, higher levels of complexity and heterogeneity are often encountered when providing translation of user model instances between existing systems. One example of a more complex translation would be the tokenisation of an instance value such as the user's address and the extraction of a particular segment such as the country. Another example would be the translation between ranges of text-based and numeric instance values such as those used in different educational grading scales. To support these types of complex translations between heterogeneous user models requires the use of more advanced mediation components. Also, methods to validate these translations through testing are important to deliver practical approaches. Testing functionality is currently provided in commercial mapping tools such as Altova Mapforce but is not a main aspect of the surveyed approaches.
- **Lack of custom domain-specific processes and tools** - Many of the surveyed approaches that are based on translation do not provide any custom tools to support the process. Instead they rely on existing commercial or research tools, for example Altova Mapforce [Altova, 2012] or COMA++ [Aumueller et al., 2005]. While these tools can be used to provide user model interoperability, the overall approach is restricted by the capabilities and outputs of the chosen tool. For example, the tool may not support the required mapping types, translation testing or

executable mapping representations. Providing translation between multiple user models can also become a very complex task that overlaps many different but related domains of user data. Administrators are typically required to manually perform some aspects of mediation such as mapping tasks. The existing mapping tools are generic and do not support the type of complex multi-domain tasks which are required in user model interoperability [Falconer et al., 2007].

Overall, the review of the state of the art approaches has demonstrated the wide variety of methods used to provide user model interoperability and has also revealed a number of key aspects about each of these approaches. The main overall strengths and weaknesses in the surveyed approaches have also been described. In the next section, some of the relevant evaluation techniques that have been encountered during the state of art review are presented.

2.5 Evaluation Metrics

Evaluation in user model interoperability is a complex issue due to the variety of possible approaches and the different aspects of those approaches that can be evaluated. Many of the surveyed approaches have presented systems or frameworks for user model interoperability but do not describe any extensive evaluations that were conducted. In this section, the evaluation metrics from the surveyed approaches are presented along with some other potential metrics derived from the key issues in user model interoperability.

Evaluation Metrics from the State of the Art Approaches

The surveyed approaches have conducted evaluations that mainly address key issues 1 and 2: the overall process and architecture of the approach. The evaluated metrics in these approaches include:

- **Overall improvement in interoperable user models** from the perspective of the end-user is the most common evaluation metric among the surveyed approaches. In theory, user model interoperability should improve the user models because there is more data to inform the user and to allow the system to make more accurate decisions. This type of evaluation was performed within the approaches of Berkovsky, Carmagnola and Dimitrova, and Cena to determine if recommendations have been improved by user model interoperability. The evaluations were conducted using user trials of the systems. Results for the pre and post

exchange of user data were gathered and used to calculate a Mean Absolute Error (MAE). Each of these approaches has shown an improvement in MAE and therefore determined that user model interoperability does improve user models.

- **Overall interoperability performance** has been evaluated in some of the state of the art approaches, in particular the centralised approaches: Personis and CUMULATE. In Personis, the performance was assessed for the number of transactions per second that could be processed. CUMULATE was assessed for the response time of its read and write actions. Other aspects that could be measured in relation to overall performance are the size and quantity of the interoperable user models and the number of concurrent requests for user model updates that can be processed. The evaluation of response times can include comparisons between systems or against generally accepted performance guidelines.
- **Overall provision of user model interoperability** can be qualitatively evaluated using direct comparisons of functionality between existing user model interoperability approaches. The approaches can be varied and focus on different aspects of the interoperability process. As a result, a set of key issues, such as those defined by Carmagnola [Carmagnola et al., 2011] and explored in Section 2.3.1, must be used to perform a systematic comparison of functionality to evaluate an approach relative to the current state of the art approaches.

Additional Evaluation Metrics

There are many other specific aspects of user model interoperability approaches that can be assessed. However, they are highly dependent on which key issues the approach is focusing upon and the methods used to address those issues. Some metrics that can be identified from the key issues for user model interoperability include:

- **Performance of individual components** is related to key issue 3: the protocols and languages for communication. The chosen methods and technologies have an individual impact on the performance of the interoperability process and measuring this can identify specific areas of the approach that could perform more efficiently.
- **Accuracy of exchange method** is related to key issue 4: the representation of data exchange. If a mediation-based translation approach is taken, measuring the accuracy of the defined mediation is an important metric to determine the quality of the exchange between user models.

- **Duration of exchange method** is related to key issue 4: the representation of data exchange. The duration is the time taken to define the mediation between heterogeneous user models.
- **Usability of interoperability tools** is related to key issue 4: the representation of data exchange. Providing tools to support the manual creation of mediation is important. Measuring and improving the usability of these tools can have a significant impact on the quality of the exchange between user models.
- **Coverage of new user models** is related to key issue 5: the exchanged data types. Measuring the ability of an approach to support new user models which contain a variety of different data types would provide useful information for the long-term development and evolution of the approach.
- **Overlap in exchanged user models** is related to key issue 6: the integration of exchanged data. If a merging approach is taken to the exchange of user models, various inconsistencies can develop. An aspect that can be measured relates to the amount of overlap that occurs between merged user models.
- **Incompleteness in exchanged user models** is related to key issue 6: the integration of exchanged data. The amount of required user information that is not provided during the exchange of user models can also be measured.
- **Access to user models** is a metric related to key issue 7: privacy in user model interoperability. User model privacy can be measured in many different ways and depends upon the selected methods in each approach. For example, the access rights of different types of user could be assessed.
- **Accuracy of user identification** is a metric related to key issue 8: pre and post interoperability task. There are many potential evaluation metrics in this key issue that depend on the selected methods in each approach. For example, the accuracy of the automatic matching of user identifiers could be measured.

To assess the significance of the various quantitative evaluation results, statistical analysis methods are important. Not many of the state of the art approaches have described the use of statistical analysis techniques in their evaluations except in the case of the MAE calculations. One statistical method that is commonly used to evaluate the significance of sample differences is the paired-t test. This has been used successfully in many related areas such as ontology mapping [Fu et al.,

2011].

Overall, detailed evaluations are lacking from many state of the art approaches. There are many different metrics that can be assessed in user model interoperability approaches. However, they are highly dependent on the selected key issues and methods in each approach. In this section, a selection of these metrics have been identified and described based on the key issues for user model interoperability. These metrics are subsequently used in the selection of appropriate evaluation methods for this research.

2.6 Summary

This chapter presented a detailed review of the field of user model interoperability. User model interoperability is a complex topic that encompasses aspects of both the broad field of interoperability and the specialised area of user modelling. Firstly, interoperability was defined and the main methods were detailed. One of these methods is the use of mappings to provide executable connections between equivalent data models. Mapping is a complex task that currently cannot be fully automated. As a result, some current mapping tools were described that are typically used to support a mapping administrator. Following this, the area of user modelling was defined including typical techniques used in the development of user models. Next, a number of key issues for user model interoperability were presented and explored. These key issues reflected the main challenges in providing processes for the exchange of user models.

Following the description of the user model interoperability key issues, a number of state of the art approaches were selected and reviewed. The review consisted of a summary of the approaches and the main methods used. Finally, a comparison of the selected approaches was completed using the user model interoperability key issues to provide a common basis for analysis. The findings of this analysis were discussed to aid the identification of key aspects to focus on in the design of a new approach to user model interoperability. Finally, some evaluation techniques that were encountered during the review process were presented to assist the subsequent evaluation methods of this research.

Chapter 3

FUSE: A Domain-Aware Approach to User Model Interoperability

3.1 Introduction

Providing user model interoperability between existing heterogeneous systems can require complex tasks across multiple domains. This chapter presents the design of a new approach to user model interoperability, collectively called FUSE (Federated User Exchange). The FUSE approach is novel because it uses domain knowledge within an advanced combination of new processes and tools to support complex interoperability tasks. The FUSE approach is developed to advance many aspects of user model interoperability such as the accuracy, efficiency and usability of interoperability tools while leveraging the best practices in the state of the art.

In this chapter, appropriate key issues, methods and evaluation metrics are selected to provide the basis of the new domain-aware approach to user model interoperability. The concept of domain-aware user model interoperability is described and refined in the application area of education. The FUSE approach is presented including descriptions of two domain-aware mechanisms: a canonical user model and user model mapping types. Finally, this approach is applied in two detailed interoperability processes which deliver the FUSE domain-aware approach for user model mapping and translation.

3.2 Methodology

A combination of different research methods were used to design the domain-aware approach to user model interoperability. Firstly, appropriate key issues and methods from the state of the art review were selected. These choices were made based on the findings of the analysis of the surveyed approaches and based on the overall objectives of this research. This led to the basis of a new approach to user model interoperability which leverages the best practices in the state of the art. Following the selection of the key issues and methods, the appropriate evaluation metrics for this approach were also selected from those defined in the state of the art review. These evaluation metrics form the basis for the evaluation of this new approach.

A novel domain-aware approach to user model interoperability was formulated and designed. A top-down design approach was used where the main concepts and key aspects of the approach were defined and then refined in a representative application area. The selection of an application area allowed in-depth analysis to be conducted. Education was selected for a number of reasons. It contains many available systems which are rich sources of user models and user information. These user models represent multiple domains and provide a high level of relevancy to other application areas. Education is also the most cited application area for user model interoperability research and one of significant value to application developments. A review was conducted, ultimately leading to a collection of education-based systems, user models and domains that could be used during the design, implementation and evaluation of the domain-aware interoperability processes and tools. Following this, the findings were used to design the detailed mechanisms and processes of the FUSE domain-aware approach to user model interoperability.

3.3 Selection of Key Issues, Methods and Metrics

The list of key issues presented in Chapter 2 describes the aspects that must be addressed in an effective approach for user model interoperability. This is an extensive list that encompasses many research fields that influence user model interoperability. Not all of the identified key issues are within the scope of this research and in this section the most relevant key issues are selected to address the research objectives. The selected key issues are summarised in Table 3.1. The first six key issues are all highly relevant to this research and its stated objectives. Key issue 7 which is related to privacy and key issue 8 which is related to pre and post interoperability task are

considered outside the scope of the research in this thesis.

Table 3.1: User model interoperability key issues

Key Issue	Methods from the State of the Art
1. Interoperability task	Exchange of user models, User modelling/adaptation service
2. Interoperability architecture	Centralised, Decentralised, Mixed
3. Protocols and languages for communication	HTTP, SOAP, LDAP, XML, JSON, UserML
4. Representation of data exchange	Common user model representation, Translation approach
5. Exchanged data type	User data, Usage data, Environment data, Domain data, Inferred data, Reasoning data, Social data
6. Integration of exchanged data	Not-merging the collected data, Merging the collected data
7. Privacy	Different access rights, Encryption, Perturbation, Scrutable user model, Third-party guarantors and standards, Pseudonymous personalisation
8. Pre and post interoperability task	Service discovery, User identification, Evaluation of interoperable user models

To design a user model interoperability approach which supports the selected key issues, a number of appropriate methods have been chosen from those identified in the state of the art review. The various methods from the surveyed approaches are also summarised in Table 3.1. For each key issue, appropriate methods were selected using the findings from the state of the art analysis. The selected methods for each key issue are:

1. **Interoperability task** describes the intended purpose of the interoperability system and the main function that will be performed. **The purpose of the domain-aware approach presented in this thesis was to facilitate the exchange of heterogeneous user models between existing systems.**
2. **Interoperability architecture** specifies the overall architectural design that is used to exchange the user models. **A mixed architecture was selected because it provides beneficial aspects of both the centralised and decentralised architectures.** A shared centralised system is used to manage and control complex interoperability between multiple systems. However, each existing system can still maintain ownership of the individual user models.
3. **Protocols and languages for communication** describes the methods used to exchange the user models between systems. **A REST web service was chosen as the method for**

communication. REST was selected for its relative ease-of-use and applicability to user model interoperability in existing web-based systems. **XML was chosen as the language for the exchange of user model data.** This is one of the most common exchange formats on the web and, along with RDF, is commonly used to represent user models. XML was chosen over RDF for its wider applicability to other types of existing systems and user information.

4. **Representation of data exchange** describes how the user models are represented during the exchange process from a syntactic and semantic perspective. **To address this key issue, a mapping-based translation method was selected in this approach.** This was chosen because mappings are highly flexible, reusable and can support the type of complex interoperability that is required between existing systems. In this approach, complex mappings are created manually by an expert administrator and executed in an automatic translation process.

To support the mapping-based translation, a canonical model approach was selected to provide a consistent, shared user model representation for the mapping of each external system. It also provides benefits in the addition of new interoperable user models and the management of mappings in multi-system interoperability. The canonical model approach is described in more detail in Section 3.5.1.

A domain-aware approach was developed to support the type of complex tasks from multiple overlapping domains involved in user model interoperability. A domain-aware approach can be defined as the incorporation of exchangeable knowledge from one or more domains into generic tools and processes to support complex interoperability tasks. In this case, the domain-aware approach is the application of knowledge of multiple user-related domains to support the mapping and translation of user models. The domain-aware approach is described in more detail in Section 3.4.

5. **Exchanged data type** specifies the types of data that can be exchanged by the interoperability system. As described in the previous key issue, the method for the representation of data exchange is a mapping-based translation approach. As a result, this method provides high levels of flexibility for the exchanged data type as new mappings can be created to support the various data types. **Different forms of user and domain data were selected as the main focus of exchange in this thesis, however, the mapping-based translation approach provides support for the majority of data types.**
6. **Integration of exchanged data** describes the approach of incorporating user data into the

systems. **An on-demand translation method was selected in this approach which allows interoperable systems to request user model updates as needed. The selection of the mapping-based translation approach also allows the merging of user information directly into existing user models provided by the interoperable systems.** Potential inconsistencies in the user model translation can be addressed by the administrator during the creation and testing of the mappings.

7. **Privacy** describes the protection of a user's privacy when their information is being exchanged between multiple systems. Privacy protection is not within the scope of the research in this thesis. When exchanging the personal data of people it is important that the privacy and security of the data is protected. However, the main objectives of this research are fundamentally focused on the technical, syntactic and semantic challenges in overcoming heterogeneity in user models. In the future, it is envisaged that methods developed from the large amount of related research in this area can be adopted to protect user data as it is being exchanged.
8. **Pre and post interoperability task** encompasses some other aspects of user model interoperability that exist outside of the main exchange process. These aspects are not within the scope of the research in this thesis. This key issue is focused on tasks that are completed before and after the exchange of user models such as the discovery of new systems to interoperate with and the automatic matching of individual user identifiers. These are relevant tasks and would be important in any large scale deployment of user model interoperability. However, this research is focused on the challenges in the core interoperability process and, as a result, approaches to these tasks are not considered. Again, it is anticipated that methods developed in related research could be employed to support these tasks.

The selected key issues and methods allow the development of a novel approach to user model interoperability that addresses the stated research objectives. The evaluation of the approach is also a key objective of this research and in the following section appropriate evaluation metrics are selected.

Selection of Evaluation Metrics

A review of evaluation metrics from the state of the art approaches along with other potential metrics identified from the key issues was presented in Section 2.5. In this section, five of those metrics are selected for use in the evaluation of this approach. The selected metrics are:

- **Accuracy of exchange method** refers to the precision of the interoperation between user models provided by, in this case, a mapping-based translation approach. This metric is often evaluated in related fields such as ontology and database mapping. In these cases, it is typically measured by comparing manually or semi-automatically created mapping results to a set of benchmark or “gold-standard” results [Noy and Musen, 2002]. This process will be used to evaluate user model mapping accuracy in this approach.
- **Duration of exchange method** refers to the time taken to create the interoperation between user models. In relation to the mapping-based approach, it is the time taken by an administrator to manually define the required mappings.
- **Usability of interoperability tools** refers to the usability of tools used by administrators of the interoperability process. As this approach will require manual mapping, the usability of the mapping tool will be assessed. Usability is defined in the international standard ISO 9241-11 [International Organization for Standardization (ISO), 1998] as:

“The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.”

In the context of this approach, the effectiveness and efficiency of the tool can be determined by the previously described accuracy and duration metrics. The user satisfaction is then typically determined by using techniques such as user interviews and questionnaires. In particular, the System Usability Scale (SUS) questionnaire is a widely used method of determining a single numeric metric for usability [Brooke, 1996]. A combination of these techniques will be used to assess usability.

- **Overall interoperability performance** refers to the overall time taken by the interoperability system to exchange user models between multiple systems. This metric can be measured in a number of different ways to assess different variables in user model interoperability. Performance of the translation within the approach will be assessed at both the schema and instance levels. At the schema level, the impact of the number and complexity of mappings between user models, the number of user models and the number of elements within the user models will be measured. At the instance level, the impact of the quantity of concurrent requests for user model instances will be measured.
- **Overall provision of user model interoperability** refers to the evaluation of functionality in the user model interoperability approach. This can be achieved using a common set of key

issues to systematically compare between existing approaches. The key issues presented in Chapter 2 will be used to evaluate the approach against the existing state of the art approaches.

These metrics will be used to evaluate the domain-aware approach for user model interoperability. In combination, they provide comprehensive evidence about the ability of the approach to provide both manual user model mapping and automatic user model translation. To assess the significance of the various evaluation results, statistical analysis methods such as the paired-t test will be used. This has been employed successfully in other mapping-based interoperability approaches [Fu et al., 2011].

Overall, the selected key issues, methods and metrics will be used to develop and evaluate a novel domain-aware approach to user model interoperability. The approach will consist of new processes and tools that allow an administrator to manually construct complex mappings between heterogeneous user models and then apply them to automatically translate user information between existing systems. The approach will also include domain-aware mechanisms which provide knowledge of multiple user-related domains to support the mapping and translation of user models. In the following sections, the domain-aware approach is presented in detail starting with the main concepts and key aspects. Following this, the approach is refined using an in-depth analysis in the application area of education. Finally, the resulting mechanisms and processes, collectively called FUSE, are presented to provide the detailed functional aspects of the domain-aware approach to user model interoperability.

3.4 Domain-Aware User Model Interoperability

In this section, the concept of domain-aware user model interoperability is described in detail, including the fundamental aspects of a domain-aware approach and its application in the area of education.

Overlapping Domains in User Models

A user model is typically designed and constructed for use in systems in a particular application area, for example education. However, a user model can contain information relating to multiple different overlapping domains within an application area. In this research, a “domain” is described as:

A group of related terms that represent a particular field of knowledge.

An example of a user model in the application area of education which encompasses three overlapping domains is shown in Figure 3.1.

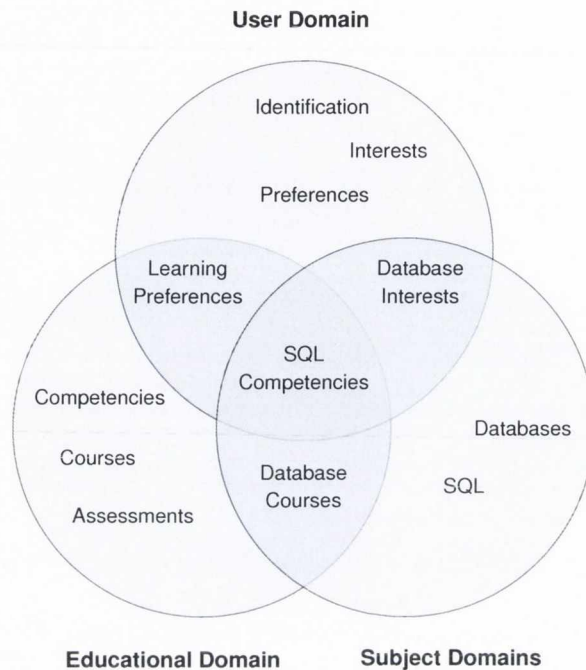


Figure 3.1: Example of education-based user model with overlapping domains

In this example, the user model contains overlapping information from user, educational and subject domains. In the user domain, information about the user such as identification, interests and preferences are represented. In the educational domain, information such as competencies, assessments and courses are represented. The subject domains represent information such as particular topics in databases or SQL. The terms and information in the user model can overlap across each of these domains. For example, the user can have preferences that are specifically learning-based. They can also specify that they have interests in databases. All three domains can also overlap, for example in SQL competencies which represent an assessment of a user's knowledge of a subject.

There are many other domains that can appear in a user model, for example, device, context and tool domains. A second example of a user model with overlapping domains would be in the application area of corporate sales. In this case, the overlapping domains within the user model could potentially include employee, customer, product and company domains.

Requirement for a Domain-Aware Approach

These types of rich, multi-faceted user models bring added complexity for user model interoperability. In particular, higher levels of expertise are required from any administrator of the interoperability process. When a mapping approach is used, expertise from a number of converging areas is required to map accurately. This expertise can be generally separated into two main groups: technical knowledge and domain knowledge. Technical knowledge includes knowledge of the interoperable systems' underlying technologies and data models, knowledge of mapping techniques and mapping tools and knowledge of user modelling techniques. Domain knowledge includes knowledge of any domains used in the interoperable user models.

Large scale mapping of the highest quality would realistically require multiple collaborators with different skill sets and domains of expertise. However, this is not always possible and often one person with some of the required expertise will perform the mapping tasks with potentially additional input from domain experts. As a result, providing support to people who perform mapping tasks is crucial in improving user model interoperability. One of the key methods to support the user is through providing them with additional knowledge of the various user model domains.

Definition of a Domain-Aware Approach

The primary motivation of this research is to investigate the impact of a domain-aware approach for user model interoperability. A domain-aware approach can be defined as:

The incorporation of exchangeable knowledge from one or more domains into generic tools and processes to support complex interoperability tasks.

The domain knowledge can be applied in many areas of the interoperability process, however, it is highly dependent on the selected interoperability methods. In this case, the domain-aware approach provides knowledge of multiple overlapping domains to support manual mapping and automatic translation for user model interoperability.

While the domain-aware approach incorporates knowledge from multiple domains into the interoperability tools, it does not restrict the tools to only those domains. The systems, user models and domain knowledge can be extended or changed to support required interoperability tasks in different application areas. The main challenge in providing this is the development of new domain

knowledge for the interoperability tools. In some cases, this can be a complex and time consuming task if the domain knowledge is newly designed. However, the domain knowledge applied in the tools can often be based on existing knowledge structures from other sources. The domains can also apply across more than one application area. For example, the domain of databases may be used in the user model of a learner but it may also apply in the user model of a professional engineer. The applied domain knowledge is also highly reusable and can be shared within the interoperability tools amongst groups of different administrators.

3.4.1 Application in Education

To support the design, development and evaluation of a domain-aware approach to user model interoperability it was necessary to provide an in-depth analysis of multiple systems, user models and domains in an appropriate application area. Education was selected as the application area in this approach for a number of reasons:

- There are many available and accessible education-based systems.
- There are rich collections of user models and other forms of user information.
- There are multiple potential domains within those user models.
- Many of the domains also represent other application areas.
- The user models are consistent with and representative of other application areas.
- There are many potential interoperability tasks between the user models.
- There is significant related research allowing comparisons of approaches and tools.

A preliminary review of a variety of existing education-based systems and formats was conducted to identify what types of user information were available and how they were structured. As shown in Table 3.2, these systems included adaptive and user modelling systems that maintain clearly defined and structured user models. They also included learning management and administration systems that maintain rich user profile information stored in databases. Standardised representation formats for learner information such as IMS LIP and IEEE PAPI were also analysed.

Table 3.2: Preliminary and selected systems

Name	Description	
APeLS	Adaptive educational system	[Conlan and Wade, 2004]
AHA!	Adaptive educational system	[De Bra et al., 2003]
CUMULATE	User modelling service	[Brusilovsky et al., 2005]
Sakai	Learning management system	[Sakai, 2012]
Moodle	Learning management system	[Moodle, 2012]
OpenSIS	Student administration system	[Open Solutions for Education, 2012]
IMS LIP	Learner information format	[IMS Global Learning Consortium, 2012]
IEEE PAPI	Learner information format	[IEEE, 2012]
LDAP Schema	Directory information format	[Internet Engineering Task Force, 2012]

Following the preliminary review, a selection of these systems (highlighted in Table 3.2) were chosen to provide feature-based user model examples and practical user model interoperability tasks in education. Four systems were selected that offered the most potential for user model interoperability. The multiple overlapping domains of user, education, databases and SQL were also chosen to represent the type of complex interoperability tasks that are commonly required in user models. To exchange competencies in SQL, APeLS and CUMULATE were selected. APeLS is the basis for an adaptive SQL web course used to teach students in Trinity College Dublin. CUMULATE is a web-based user modelling system that is the basis for many educational components used in the University of Pittsburgh including some in the area of SQL. Both of these systems are established educational tools that use highly structured overlay user models to provide personalisation functionality.

To exchange identification and database course assessment information, the learning management systems Sakai and Moodle were selected. These systems were chosen as they are widely used and are often a key source of course and learner information in educational institutions. To generate sample courses in Sakai and Moodle, database course structures and terminology were retrieved from online sources such as MIT OpenCourseWare [MIT, 2012]. The four selected systems were used to provide user models and practical interoperability tasks that are applied throughout the design, implementation and evaluation of this research. Examples of each of the four user models can be seen in Appendix A. In the next section, the specific domain-aware approach to user model interoperability applied in this research, collectively called FUSE, is presented in detail.

3.5 FUSE Domain-Aware Approach

In this research, a domain-aware approach to user model interoperability was developed called FUSE (Federated User Exchange). The FUSE approach uses knowledge from multiple overlapping domains to support manual mapping and automatic translation for user model interoperability. The domain-aware approach is applied in user model mapping and user model translation in the form of two domain-aware mechanisms. These mechanisms, a canonical user model and user model mapping types, tailor the process to the relevant domains. The canonical user model is a shared user model representation to which all mappings are created and the user model mapping types are mapping components that are specifically designed for mapping user information and facilitating reuse.

Both of these mechanisms allow the addition of a domain-aware approach in key areas of the interoperability process but do not restrict the tool to a single application area or set of domains. The two mechanisms allow interchangeability to reuse the tools with other domains or in other application areas. Some examples of other potential domain-aware mechanisms include a mapping visualisation designed specifically for user models in a particular application area or a specialised matching algorithm that derives suggestions for user model mappings in particular domains. However, in contrast with the two chosen mechanisms, these mechanisms would restrict the applicability of the tool in other domains and application areas. In the following sections, both of the chosen domain-aware mechanisms in the FUSE approach are described in more detail.

3.5.1 Canonical User Model

As described in Section 2.3.1, to exchange user models successfully an approach to overcome heterogeneity must be developed. In education, there has been significant effort made in the development of a standardised learner model such as IMS LIP but heterogeneity is still a major challenge [Van Der Sluijs and Houben, 2006]. This is largely due to the diversity of educational systems and their resulting learner representations. The monolithic nature of standards can result in inflexible and excessively complicated implementations. Conversely, it is also difficult for a standard to encapsulate all potential aspects of a learner. Specialist systems can often require domain-specific learner aspects that make the use of a standard unsuitable. While some standards, such as IMS LIP, support the concept of extensions, this leads to further heterogeneity where the individually developed extensions are inconsistent across different learner models. Potentially, this

can lead to situations where systems are using the same standardised user models but they are still largely incompatible for exchange purposes. In the past, the existence of other competing learner standards, such as IEEE PAPI [IEEE, 2012], has hindered the adoption of a single prevailing standard.

The lack of widespread adoption of a learner model standard has led to significant research into other methods to overcome heterogeneity. One approach that is common in other areas such as the federation of relational database systems is the provision of mappings. Mapping consists of creating direct associations between equivalent elements of individual data models. The process typically requires an administrator to create the mappings manually using a mapping tool. The mappings are then applied to perform an interoperability task such as automatic translation between the different data models. In terms of user models, Figure 3.2 shows the direct mapping approach using four user models as an example. In this case, mappings are created from user model 1 to user model 2, 3 and 4 directly. Similarly, the remaining user models are all mapped to each other in this approach. In the user model translation process, these mappings can be executed to exchange the relevant data directly between different user model representations.

A variation of this approach takes aspects of both standardisation and mapping to provide canonical model mapping. In this form, mappings are created from each user model solely to a canonical model, which provides a common user model representation. This approach is also shown in Figure 3.2. In the user model translation process, the mappings can be executed to exchange data into a canonical model representation before exchanging that data into the chosen target user model representation.

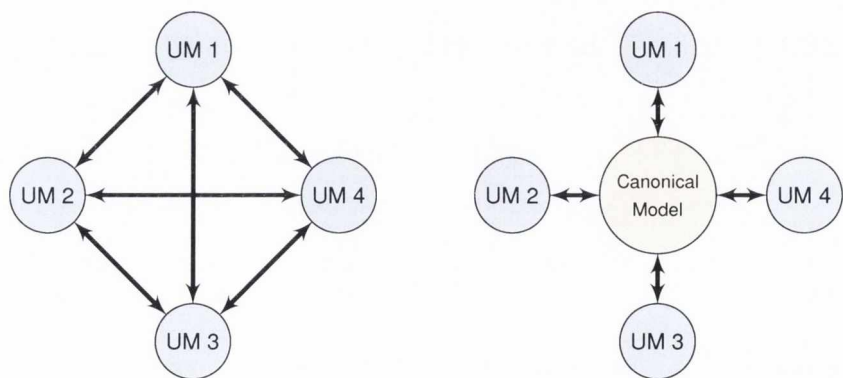


Figure 3.2: Direct and canonical model mapping approaches

Advantages and Disadvantages of Canonical User Model

There are advantages and disadvantages to the canonical model approach when compared with the direct mapping approach. They include:

- **Mapping accuracy and duration** - The canonical model approach provides a consistent, shared user model which represents the domains across the user model mapping tasks. Administrators can gain knowledge and familiarity in the domain structures and terms used in the canonical user model. The canonical user model also provides a consistent target model which can result in more similarity in the mappings required between user models. As a result, the use of the canonical model can potentially assist administrators in performing user model mapping tasks more quickly and accurately. This contrasts with the direct mapping approach where there is no domain knowledge or consistent user model representations provided. In this case, the administrator must contend with analysing the user model structures and identifying new mappings each time.
- **Mapping tool usability** - The consistency of the domains in the canonical model also allows for richer visualisation approaches to be adopted in mapping tools. As the canonical model is the focus for the mapping each time, it can be represented in different ways that improve the overall usability of the mapping tool. In the direct approach, the mapping tools must be generic to apply mappings between various types of source and target user models. This results in mapping tool visualisations which are typically tree-based such as those reviewed in Section 2.2.1.
- **User model translation performance** - The canonical model approach affects the quantity of mappings required. To perform a one-to-one exchange from a source to a target user model requires two mapping sets in the canonical approach but only one in the direct approach. However, this changes as the number of interoperable systems increases over time to many-to-many exchange scenarios. In the direct mapping approach, each user model is directly mapped to every other requiring $N(N-1)/2$ bi-directional mapping sets for N user models. Each time a new system is added to an interoperability scenario, the number of bi-directional mapping sets required is N ; the number of previously mapped user models. Thus, the total number of mapping sets required will increase every time a new system is added. The canonical model mapping approach results in the number of mapping sets being controlled. Each time a new application is added it only requires one set of bi-directional mappings to the canonical

model. It is then, as a result of previous mappings to the canonical model, mapped to every other application automatically.

The quantity of mappings in each approach would suggest that over time, the direct mapping approach is potentially slower than the canonical model approach to perform automatic translation in multi-system user model interoperability. However, the canonical model also requires a two-step translation approach consisting of source to canonical and canonical to target. The performance of the approaches is also highly dependent on the relative complexity of the mappings. The relationship between the canonical model and the external user model may result in increased or decreased mapping complexity compared to direct mappings thus affecting performance.

- **Overall provision of user model interoperability** - The selection of a canonical model or direct mapping approach also has a significant impact on the overall architecture of the interoperability approach. The canonical model approach inherently results in a mixed architecture that is a hybrid of both the mapping and standardisation approaches. This approach allows a shared representation of the domains to be provided to reduce the variability in the process but still maintain the flexibility and advanced interoperability provided by the mappings. As stated previously, the canonical model requires fewer mappings but the mappings can vary in complexity depending on the interoperability tasks. However, the canonical model is also a central point of failure in any approach. If an issue with the canonical model arises it can affect the translation of many interoperable user models. The canonical model mapping approach also adds an extra translation step to the interoperability process when compared to the direct mapping approach. This could potentially allow more inconsistencies to develop within the exchanged user information. However, well designed and extensively tested mappings would reduce the possibility of these inconsistencies developing.
- **User model refactoring** - Providing multi-system user model interoperability over a longer period of time may require the ability to refactor the user models. Using the direct mapping approach for multiple interoperable user models produces significantly more mappings. If the external user models require refactoring this can require the editing of more mappings than in an equivalent canonical model approach. However, the canonical model may also need refactoring to support different types of user models. This can be a result of an initial design that is too basic or which does not suit the required mapping tasks. In the canonical model approach, all the external user models are mapped to this common user model representation.

As a result, refactoring the canonical model structure will require changes to multiple existing mappings. This is a significant challenge in the long-term application of the canonical model approach. To support multi-system long-term user model interoperability requires mechanisms to support the long-term refactoring and evolution of the canonical model and the resulting editing of multiple mappings.

The interdependent relationships between the user models and the mappings result in complex challenges in both the direct and canonical model mapping approaches. These challenges are magnified significantly when attempting to provide long-term sustainable interoperability between multiple user models. Collections of existing systems, user models and mappings generated over long periods of time by multiple administrators would be required to produce findings that were reliable and relevant. This research is primarily concerned with the impact of the domain-aware approach on key aspects of mapping and translation between multiple heterogeneous user models. Examining the long-term evolution, refactoring and sustainable management of user model interoperability is a complex issue and not within the scope of the research in this thesis.

Overall, the canonical model approach has been selected to provide a consistent, shared user model representation for the mapping of each external system. The fundamental concept of the canonical model also inherently provides domain knowledge in complex tasks for user model interoperability. As a result, it is one of the key mechanisms for the delivery of the domain-aware approach. The design of the canonical model used in the FUSE approach is presented in the following section.

Design of a Canonical User Model

In the FUSE approach, the purpose of the interoperability is to provide translation between existing heterogeneous user models. As a result, the design of the canonical model is in the form of a hierarchical data model. Mappings are created to base elements to translate instance data between the various user models. This contrasts with other approaches, for example, mapping to an ontology [Kalfoglou and Schorlemmer, 2003]. An ontology represents concepts and their semantic relationships in graph-based structures. It can be used to reason about the entities within the ontology. However, reasoning about user models is not an objective of the research presented in this thesis. There is also significant additional complexity involved in mapping existing data models to an ontology for the purposes of translation. Investigating the use of ontologies in the

mapping process is a potential aspect for future work.

As stated in Section 3.4.1, education was selected as the application area for in-depth analysis of the domain-aware approach to user model interoperability. The design of the canonical model should reflect commonly used structures and terminology in the application area. As a result, IMS LIP, which is a widely used specification for learner information, was selected as the basis for the canonical model. IMS LIP contains a large number of elements that represent many characteristics of a learner. The top-level elements of the specification are shown in Figure 3.3. The top-level elements also have many sub-elements that describe more specific aspects of the learner.

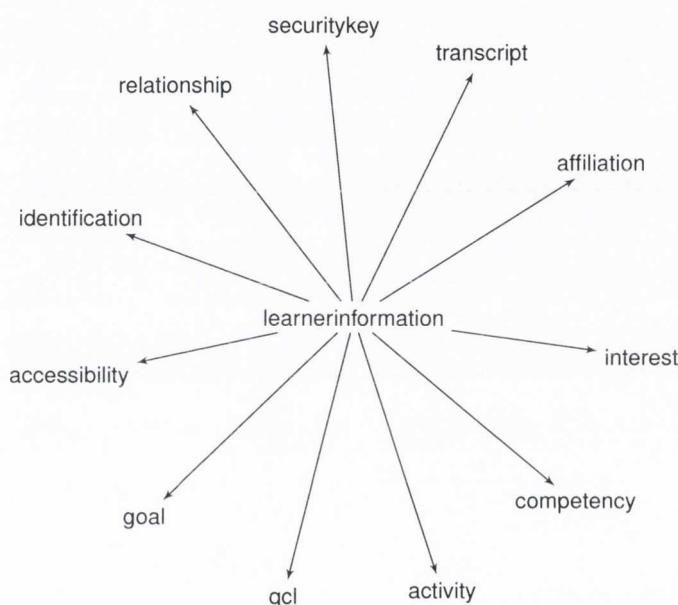


Figure 3.3: Top-level IMS LIP elements

As described in Section 3.4.1, a number of education-based user models were selected from learning management systems, user modelling systems and adaptive systems. From these user models, the multiple overlapping domains of user, education, databases and SQL were also chosen to represent the type of complex interoperability tasks that are commonly required.

To construct the canonical user model, relevant terms and structure from IMS LIP were selected as the basis of the canonical user model. These elements referred to general user identification and education-based assessments and competencies. These elements were modified as needed to support different types of learner information not commonly represented in IMS LIP, for example, user models from adaptive educational systems. In particular, the competency element has been edited to support more conventional adaptive learner modelling techniques such as the

overlay approach, which consists of domain concepts and associated knowledge values. Also, the assessment element was modified in order to exchange more detailed assessment information between educational systems.

The general user and education-based elements were then combined with subject domain elements representing databases and SQL. As IMS LIP is subject domain-neutral, the structure and terminology for these elements was retrieved from established online educational resources such as w3schools [W3schools, 2012]. The resulting canonical user model represents knowledge from multiple overlapping domains related to the selected user models. The canonical user model is applied in the development and evaluation of FUSE and is shown in Figure 3.4.

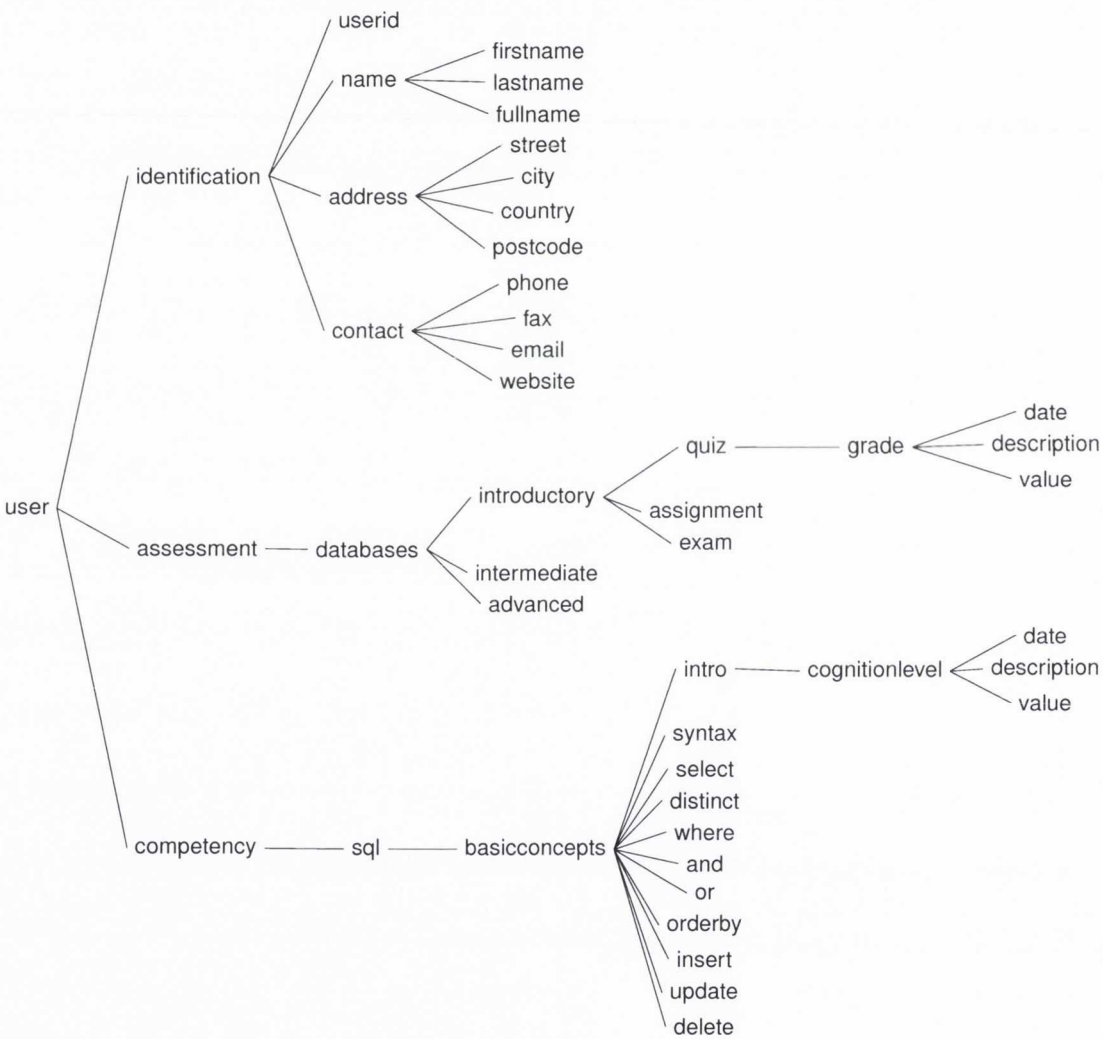


Figure 3.4: Canonical user model elements

Overall, a canonical user model was designed which could provide a consistent, shared user model representation for mapping-based interoperability. The canonical user model was also a key mechanism for the provision of the FUSE domain-aware approach. It incorporates overlapping knowledge from multiple domains which support the administrator in the complex mapping and translation tasks in user model interoperability. User model mapping types are the second domain-aware mechanism in the FUSE approach and are presented in the next section.

3.5.2 User Model Mapping Types

Many generic mapping types have been developed to overcome heterogeneity issues and are commonly used in existing mapping tools [Altova, 2012]. However, in the mapping tools there are few mapping types that focus on common mapping problems in specific application areas such as education. The development of user model mapping types is the second key mechanism to provide the FUSE domain-aware approach.

Generic Mapping Types

From the review of state of the art approaches and mapping tools and examining common user information representations, a set of required generic mapping types was identified. These generic mapping types are shown in Figure 3.5. Detailed examples of each of these mapping types are also presented in the next section in Table 3.3.

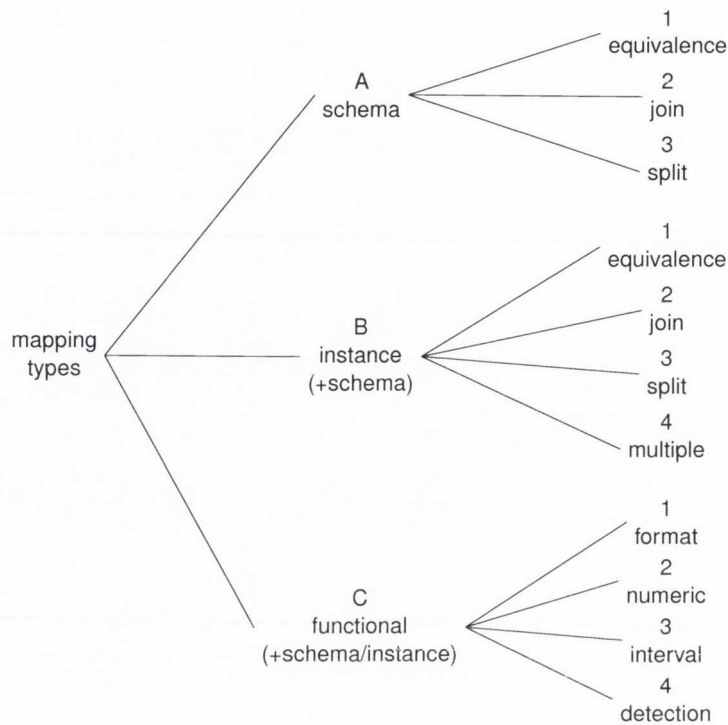


Figure 3.5: Generic mapping types

The generic mapping types are divided into categories relating to schema, instance and functional mappings. The generic mapping types have also been ordered in terms of increasing complexity which was determined by the number and difficulty of each step involved in defining a mapping. The main categories are:

- **Schema mapping types** are the most basic form of generic mapping type and are created between the equivalent schema elements of two user models. These mapping types can represent many-to-one, one-to-many or many-to-many relationships. For example, the schema mapping can be joined, where multiple schema elements from one user model are equivalent to one schema element in another user model, and split, where one schema element is equivalent to multiple schema elements.
- **Instance mapping types** allow more complex mapping of specific instance data from user model elements. Instance mappings can also be joined and split and multiple instance values can be defined as equivalent within a single mapping.
- **Functional mapping types** allow various manipulations of the instance data in a translation between user model schema elements. Functional mapping types include format, numeric, interval and detection. Format mapping types allow the conversion between data type

representations, for example, different date formats. Numeric mapping types allow mathematical manipulation of numerical data. Interval mapping types allow manipulation of data containing value ranges. Detection mapping types allow checks for the existence of particular data values.

The mapping types developed in this approach can be bi-directional or uni-directional. For bi-directional mappings this means that an inverse mapping type is assigned to each mapping type. For example, a schema split can also define the reverse schema join mapping. In some cases, a full bi-directional mapping may not always be possible. For example in education, the translation of a percentage grade to a letter grade will function correctly but an exact reverse translation is not possible. In these cases, a uni-directional mapping is defined or the reverse mapping type has an appropriate result encoded.

The identified generic mapping types are a representative selection that allow the mapping of the most common forms of data representations. They are equivalent to mapping types in many other generic mapping tools such as Altova Mapforce. The set of mapping types can be expanded further as more uncharacteristic data representations are required.

User Model Mapping Types

The generic mapping types can be used to construct mappings between user models. The administrator must construct the entire mapping each time, identifying the required values and have a detailed understanding of how it should function. However, the generic mapping types can be combined with domain-specific knowledge to provide a selection of user model mapping types that are specifically designed for heterogeneous user models in particular application areas. These user model mapping types can support an administrator in typically complex user model interoperability tasks across multiple overlapping domains. In a web-based system, the user model mapping types can be reused by the administrator or shared with other administrators to build a large collection of relevant mappings in an application area of interest.

As described in Section 3.4.1, various systems from the application area of education were examined for the type of user information that they maintain. Education is a rich source of user models and user information and is representative of other application areas. These user models were used to identify a selection of common user model mapping types in the overlapping domains of user, education, databases and SQL. The overlapping domains were chosen because of

the availability of existing user models and to provide the type of complex interoperability tasks that are commonly required in user models. Table 3.3 presents a representative sample of the type of user model mapping types that can be created including the generic mapping types that are used to form them. The examples are ranked in order of increasing complexity. This was determined by the complexity of the underlying generic mapping type and the level of domain knowledge in the user model mapping type.

Table 3.3: Examples of user model mapping types

No.	User Model Mapping Type	Example
1	equivalent first name (A1 schema equivalence)	firstname = forename
2	join first name and last name (A2 schema join)	firstname + lastname = fullname
3	split address and get country (A3 schema split)	address[street,city,country] = country
4	equivalent learning style (B1 instance equivalence)	type[reflector]/score[10] = value[10]
5	join SQL concepts (B2 instance join)	concept[SQL1] + concept[SQL2] = concept[SQLA]
6	split SQL concepts (B3 instance split)	concept[SQL1] = concept[SQLA] + concept[SQLB]
7	multiple equivalent user ids (B4 instance multiple)	userid[jsmith, rjones] = userid[06125,00242]
8	assessment date format (C1 functional format)	date[31/01/2012] = date[01-31-2012]
9	grades 0-20 to 0-100 (C2 functional numeric)	score[18] * 5 = result[90]
10	grades 0-20 to Pass/Fail (C3 functional interval)	score[11-20,0-10] = grade[Pass,Fail]
11	SQL concept to 0/1 score (C4 functional detection)	concept[] = score[0] or concept[SQL1] = score[1]

Examples 1, 2, 3 and 7 demonstrate user model mapping types for user identification information. Example 1 is an *equivalent first name (A1 schema equivalence)* user model mapping type and is used to translate the user's first name. Example 2 is a *join first name and last name* user model mapping type and is used to combine and translate the user's first and last names. Example 3 is a *split address and get country (A3 schema split)* user model mapping type and is used to separate and translate the country value of the user's address. Example 7 is a *multiple equivalent user ids (B4 instance multiple)* user model mapping type and is used to translate user identifiers based on

multiple specified equivalences.

Examples 4, 8, 9 and 10 demonstrate user model mapping types for assessments. Example 4 is an *equivalent learning style (B1 instance equivalence)* user model mapping type and is used to translate values for specific learning styles. Example 8 is an *assessment date format (C1 functional format)* user model mapping type and is used to translate between different representations of assessment dates. Example 9 is a *grades 0-20 to 0-100 (C2 functional numeric)* user model mapping type and is used to translate numeric grade values between a 0-20 grade scale to 0-100 grade scale. Example 10 is a *grades 0-20 to Pass/Fail (C3 functional interval)* user model mapping type and is used to translate a numeric grade to a text-based grade where 11-20 = Pass and 0-10 = Fail.

Examples 5, 6, 11 demonstrate user model mapping types for SQL competencies. Example 5 is a *join SQL concepts (B2 instance join)* user model mapping type and is used to combine and translate specific instances of SQL concepts. Example 6 is a *split SQL concepts (B3 instance split)* user model mapping type and is used to separate and translate specific instances of SQL concepts. Example 11 is an *SQL concept to 0/1 score (C4 functional detection)* user model mapping type and is used to detect a text-based SQL concept name and translate it to a numeric 0/1 competency value.

If an appropriate user model mapping type does not already exist it can be created using the generic mapping type and saved as a user model mapping type. This domain-specific mapping would then be available to be reused any time this type of translation was required in a user model. The administrator can search and select it from a list of relevant user model mapping types in the application area. Administrators can also inspect these selected mapping types and edit them if needed to support any differences in the user information.

Selection of User Model Mapping Types

The domain-aware processes and tools developed in this research support all of the identified generic and user model mapping types. However, a representative selection of user model mapping types is also required to allow an in-depth analysis of the key challenges involved and to provide a consistent structure for the development and evaluation of the approach. The selected user model mapping types are:

- *equivalent first name (A1 schema equivalence)*

- *split address and get country (A3 schema split)*
- *equivalent learning style (B1 instance equivalence)*
- *grades 0-20 to 0-100 (C2 functional numeric)*
- *grades 0-20 to Pass/Fail (C3 functional interval)*
- *SQL concept to 0/1 score (C4 functional detection)*

The user model mapping types were selected to provide the best representation of different user models, domains and generic mapping types. As described previously, education was selected as the application area for the user model mapping types as it is a rich source of user models. The selected user models use many common user modelling techniques, such as the overlay approach in adaptive systems [Brusilovsky and Millán, 2007], that would potentially be present in user models from other areas. A range of example user model mapping types was created for the overlapping domains of user, education, databases and SQL. These domains were selected as they provide the type of complex mapping issues that are likely to exist in many user models in various applications areas. Finally, the six user model mapping types were selected to represent a range of increasingly complex and domain-specific interoperability tasks. The selected user model mapping types were identified using an evidence-based approach and provide a set of examples that are representative of many user model interoperability tasks. The user model mapping types provide a consistent reference set that can be applied throughout the subsequent development and evaluation of the FUSE approach.

Overall, the application of user model mapping types has many benefits. Potentially, there would be a large amount of domain-specific mappings created for user models. These user model mapping types should be more relevant and easier to identify and implement by an administrator who is not an expert in mapping techniques. When combined with the canonical user model, this produces a domain-aware approach that can provide greater support for the complex domain-specific tasks required for user model interoperability. In the next section, the domain-aware approach is applied in the form of detailed FUSE processes for user model mapping and translation.

3.6 FUSE Processes

Two interdependent processes were developed which encompass the objectives and methods selected to provide user model interoperability. The first process describes the steps in the manual creation of user model mappings and the second process describes the steps in the automatic execution of user model translation. Both processes adopt the domain-aware approach and use the two mechanisms to provide domain knowledge in the interoperability tools. This domain knowledge is provided in the form of a canonical user model to which all mappings are created and mapping types that allow common user model mappings to be created. The processes are described in the following sections and are shown in Figure 3.6 and Figure 3.7. The highlighted segments of the processes indicate key areas where the domain-aware mechanisms are applied. The segments with dashed borders indicate an automatic step in the process and a continuous border indicates a manual step.

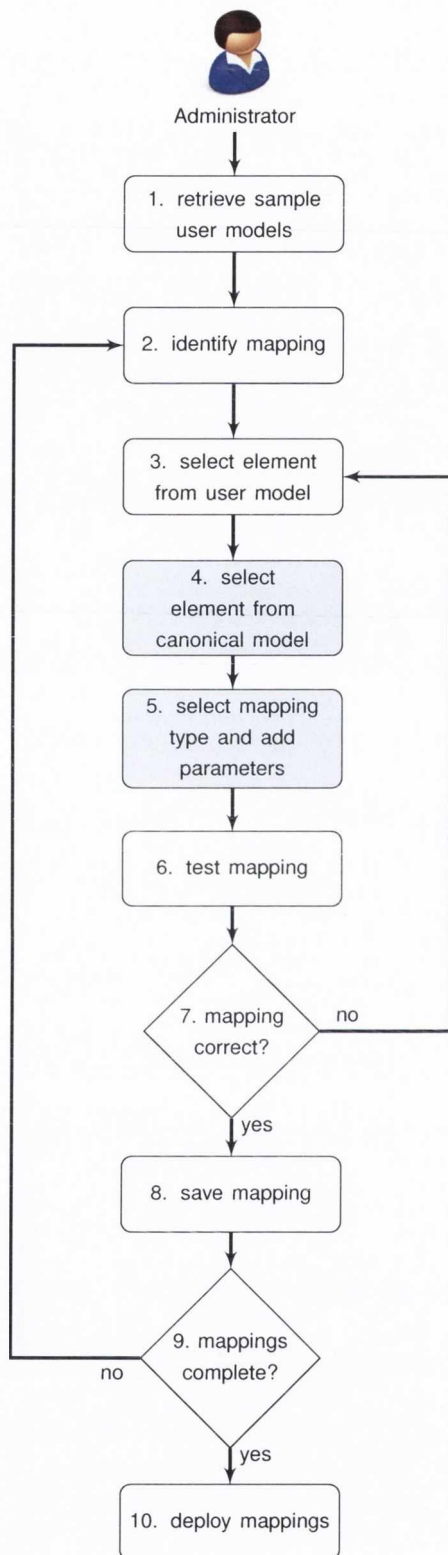


Figure 3.6: User model mapping process

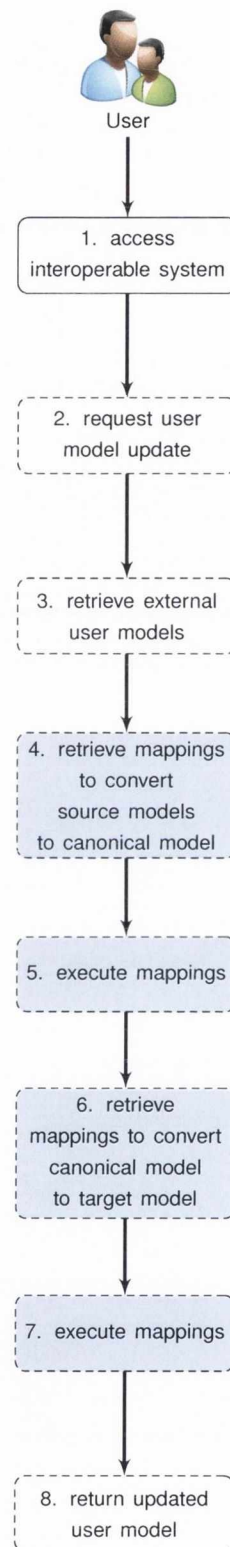


Figure 3.7: User model translation process

3.6.1 User Model Mapping

The user model mapping process is a pre-runtime, initialisation phase that is performed manually by an administrator with the assistance of a mapping tool. As shown in Figure 3.6, the user model mapping process contains ten steps:

1. Representative sample user models are gathered from the various systems adopting interoperability. These include both user model schemas and instances. The instances contain sample user information that can be used to test the mappings.
2. An administrator navigates and analyses the various sample user models. Following this, an appropriate mapping between the user models is identified.
3. The user model is navigated to the appropriate user model element and it is selected for use in the mapping.
4. As the canonical model mapping approach is being used in this process, the administrator navigates the canonical user model, identifies a matching element and selects it for use in the mapping.
5. To create the mapping, an appropriate mapping type is chosen. This could be a specialised user model mapping type or a generic mapping type. Any additional mapping parameters are specified for complex mapping types. The selected element from the user model is then added to the mapping.
6. The administrator decides to test the constructed mapping. It is validated by executing the mapping and using it to exchange sample user model instances.
7. If the administrator determines that the mapping is not valid they can reassess the mapping, returning to step 3 in the process to identify any adjustments that can be made to the mapping.
8. Once the administrator is satisfied that the mapping is valid and will perform the required translation of user information they can save it for deployment. If they determine that this mapping type will be useful in the future they can also make it available for reuse as a new user model mapping type.
9. The mapping process from step 2 to step 8 is repeated until all the required mappings are created from each user model to the canonical user model. This is determined by the administrator using a practical, continuous process of mapping creation and testing using sample user models.

10. When the administrator determines that all the required mappings have been created they can deploy the mappings for use in the automatic user model translation process.

As shown in Figure 3.6, steps 4 and 5 are highlighted. These indicate the key areas where the domain-aware mechanisms are provided. The canonical user model supports the administrator by providing a consistently visualised and centralised user model for the mapping process. The user model mapping types provide a set of customised mapping types specifically for the domains of the user models, thereby making it easier to create and manage the mapping process.

3.6.2 User Model Translation

When the mappings have been deployed and the interoperable systems have been set up to exchange their user models, the user model translation process can be conducted. This process can be run on-demand as the users access the different interoperable systems. Once the request for a user model update is made, the steps are conducted automatically, allowing the update of the user model within the target system. As shown in Figure 3.7, the user model translation process contains eight steps:

1. A user accesses one of the interoperable systems that are exchanging user information. The interoperable systems have been previously mapped by the administrator in the user model mapping process.
2. The interoperable target system makes a request for an update of its user model for that particular user, specifying which external source systems it wants to provide user information.
3. The required user model instances for that user are automatically retrieved from the specified external source systems.
4. To convert the source user models to the canonical user model, the mappings, which were created and deployed during the user model mapping process, are retrieved.
5. The retrieved mappings are executed to convert the source user models into the canonical user model representation.
6. To convert the canonical user model to the target model, the mappings, which were also created and deployed during the user model mapping process, are retrieved.
7. The retrieved mappings are executed to convert the canonical user model into the target user

model representation.

8. The updated target user model, which contains the additional user information translated from the source user models, is returned to the target system. The target system can then use this updated user model to provide an enhanced experience to the user.

As shown in Figure 3.7, steps 4 to 7 are highlighted which show the key areas where the domain-aware mechanisms are provided. The mappings from the source user models to the target user model are created using the canonical model approach as a common centralised mapping location. The mappings also include the user model mapping types which were specifically designed to map user models. This combination of the canonical user model and user model mapping types potentially provides methods for efficient performance in the automatic user model translation process.

Overall, both of the processes support all of the objectives and methods selected for this approach. They encompass both the manual user model mapping steps and the automatic user model translation steps required to provide user model interoperability between multiple existing heterogeneous user models. The domain-aware approach is a key part of both of these processes and provides knowledge of multiple overlapping domains to support the delivery of advanced forms of user model interoperability.

3.7 Summary

This chapter has presented FUSE: a domain-aware approach to provide user model interoperability. This approach provides greater support for the complex domain-specific tasks involved in the interoperability of heterogeneous user models. Initially, the most relevant key issues, methods and metrics for user model interoperability were selected from the state of the art review in Chapter 2. This defined the basis of the interoperability approach such as the use of mapping-based translation techniques and the use of XML and XQuery as the user model and mapping representation formats. Following this, the concept of domain-aware user model interoperability was defined and the application area of education was selected to allow more in-depth analysis. A selection of systems, user models and domains in education was also presented that could be used throughout the design, implementation and evaluation of the FUSE domain-aware approach.

Next, the FUSE domain-aware approach was presented in detail. Two mechanisms were developed to provide knowledge of multiple overlapping domains in user model interoperability. The canonical user model allows mappings to be created to a consistent and shared user model representation. The user model mapping types provide a set of customised mapping types specifically for the domains of the user models. Finally, two detailed processes to provide user model mapping and translation were described. The two domain-aware mechanisms are incorporated in key steps in the processes, providing the required domain-aware functionality. The two defined processes are implemented in the form of two tools; the FUSE Mapping Tool and the Translation Service, which are presented in the next chapters.

Chapter 4

Development and Evaluation of the FUSE Mapping Tool (Cycle 1)

4.1 Introduction

To implement and evaluate the FUSE domain-aware approach presented in Chapter 3, two main interoperability tools were developed: the FUSE Mapping Tool and the FUSE Translation Service. This novel combination of a domain-aware approach in both the user model mapping and translation processes provides greater support in the complex domain-specific tasks involved in user model interoperability. This chapter presents the first cycle of design, implementation and evaluation of version 1 of the FUSE Mapping Tool. The Mapping Tool supports an administrator in the graphical creation of relationships between heterogeneous user models. The second cycle, leading to version 2 of the Mapping Tool, is presented in Chapter 5. The development and evaluation of the other major tool, the FUSE Translation Service, is presented in Chapter 6.

The design methodology is outlined, followed by the architecture of interdependent components that provide the user model mapping process. Next, the implementation of version 1 of the Mapping Tool is presented including the various technologies used and the major iteration milestones of the user interface development. These iterations demonstrate the evolution of version 1 of the Mapping Tool during the course of the development. The implementation of the various user models and mapping types from Chapter 3 are also presented in detail. Finally, a preliminary user trial of version 1 of the tool is described which identified areas where improvements could be made during the second development cycle.

4.2 Development of the FUSE Mapping Tool

The FUSE Mapping Tool is a graphical tool for the creation and testing of mappings between heterogeneous user models. In contrast to existing mapping tools, it is a web-based tool which uses domain knowledge to support an administrator in complex domain-specific interoperability tasks. The Mapping Tool encompasses each of the steps in the user model mapping process described in Section 3.6.

4.2.1 Methodology

The Mapping Tool was developed by following a predefined development methodology. This methodology is based on the Usability Engineering Lifecycle [Nielsen, 1993] and can be seen in Figure 4.1.

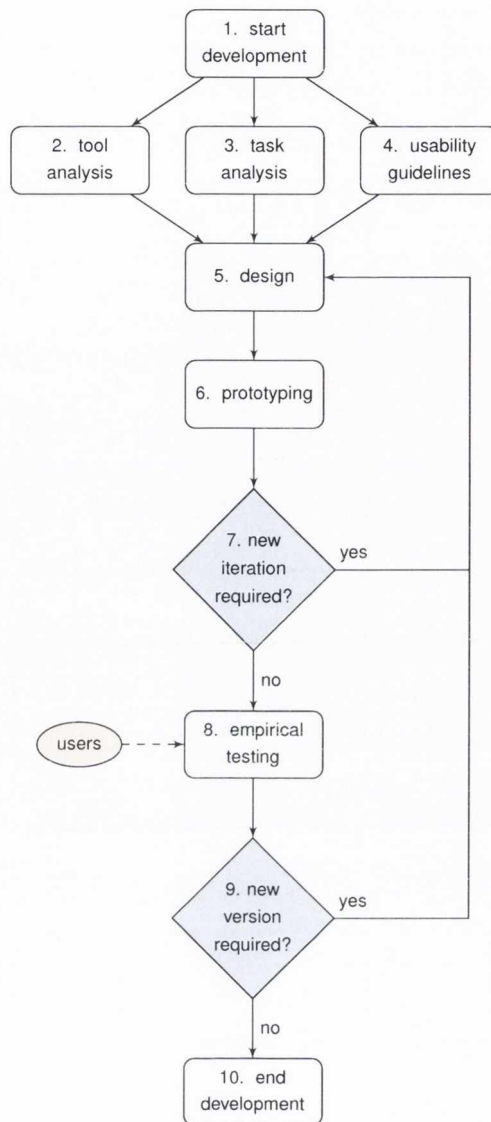


Figure 4.1: Mapping Tool development methodology

The steps in the development methodology are:

1. The FUSE approach to user model interoperability was defined and development was started on a mapping tool that would implement the FUSE user model mapping process and address the stated objectives of the research.
2. An analysis of related mapping tools was conducted as described in Section 2.2.1. This consisted of identifying and testing currently available commercial and research mapping tools. This analysis allowed the identification of features and techniques that are typically used within mapping tools and also identified existing weaknesses in these tools. Surveys of users of these mapping tools were reviewed to obtain additional information about current levels of

user satisfaction [Falconer et al., 2007].

3. The task analysis consisted of identifying key aspects of the mapping task in the context of user model interoperability. This step was completed through a combination of the mapping tool analysis, the state of the art review, and analysis of the selected user models. This allowed mapping tasks across a range of user models to be examined in detail and possible approaches in the development of a mapping tool to be clearly formulated.
4. Common usability guidelines were used to assist the development of a mapping tool. These were identified from the mapping tool analysis and in particular the commercial mapping tools such as Mapforce and Stylus Studio. Additional influential web design guidelines were also incorporated [Krug, 2006] [Nielsen, 1993]. These approaches provided the basis for the development of a tool that adhered to established web design principles.
5. The development of the user model mapping process, combined with the three preceding exploratory methods, led to the design of an initial architecture and graphical user interface for the FUSE Mapping Tool.
6. Following the design, appropriate techniques and technologies were selected and a prototype of the Mapping Tool architecture and graphical user interface was constructed to support a minimal, functional feature set.
7. The development of the Mapping Tool then followed an iterative process where three major design and implementation iterations were used to create a more advanced and refined tool. The iterations consisted of planned major increments to the functionality of the tool. The iterations were improved by the results of walkthroughs of the Mapping Tool functionality using the selected user models defined in Section 3.4.1.
8. When an iteration of the Mapping Tool was produced which addressed the major functional aspects of the user model mapping process, empirical testing with users was conducted. The first trial was exploratory and took the form of a semi-structured interview.
9. The results of this trial are presented in Section 4.3 and were used to generate version 2 of the Mapping Tool. Version 2 of the Mapping Tool was compared against an existing mapping tool in the main evaluation of this research. The development of version 2 of the Mapping Tool and the resulting evaluation is described in Chapter 5.
10. The results of the final empirical testing addressed the objectives of this research and

demonstrated the benefits of the FUSE Mapping Tool when compared to an existing mapping tool in terms of the key evaluation criteria; mapping accuracy, duration and usability. As a result, the development cycle of the Mapping Tool was completed.

Overall, this methodology proved to be a successful approach because it allowed the development of an initial tool based on the user model mapping process. Once this was in place, the tool was developed further based on feedback gathered during the user trial. This methodology delivered a tool that successfully addressed the research objectives, culminating in the findings of the main comparative evaluation against one of the current leading mapping tools. In the next sections, the architecture and implementation of the Mapping Tool are described in detail.

4.2.2 Architecture

The combination of the FUSE domain-aware approach in both the mapping and translation processes required the development of multiple interdependent components. This section describes the FUSE architecture, shown in Figure 4.2, from the perspective of the Mapping Tool and its dependent components such as the database of user models and mappings. The Translation Service which executes the mappings and performs the automatic user model translation process is also shown in the figure to represent the complete architecture of the FUSE approach, however, it not discussed in depth in this section. The Translation Service is described in detail in Chapter 6.

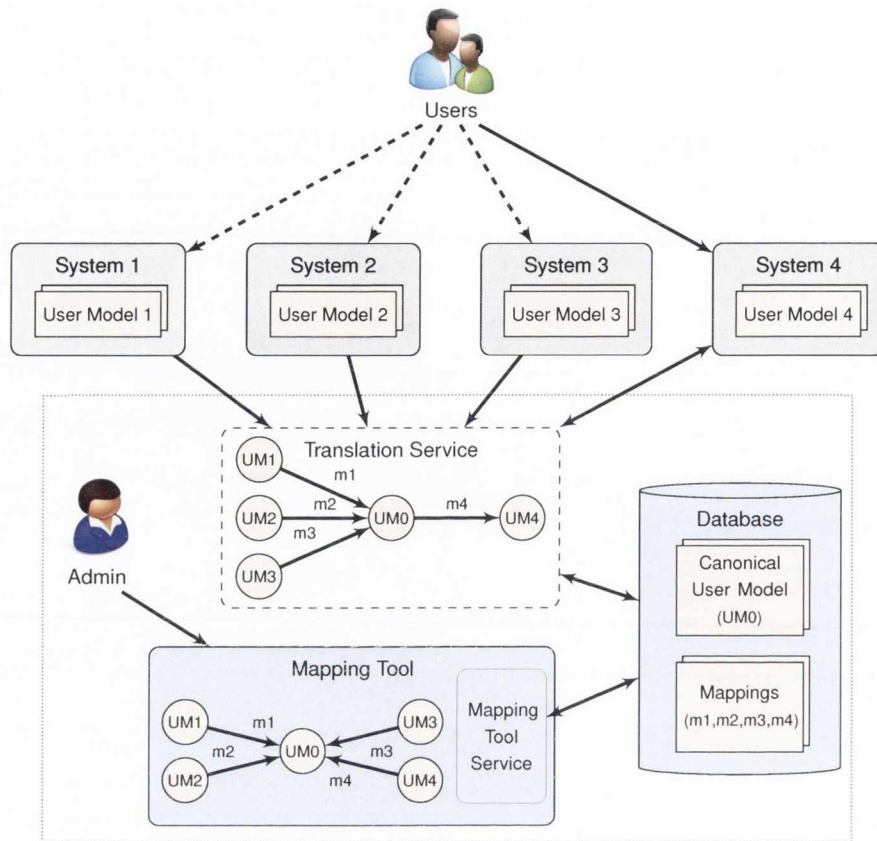


Figure 4.2: FUSE architecture

As shown in Figure 4.2, the Mapping Tool is a graphical tool for the creation of relationships between different types of user models. User model schemas and sample instances can be manually uploaded to the Mapping Tool or retrieved automatically from the source systems using web services (UM1, UM2, UM3, UM4). Mappings from elements in the user models to their equivalent elements in the canonical user model (UM0) are defined manually using the graphical user interface. The Mapping Tool can then automatically generate executable versions of the mappings (m1, m2, m3, m4).

The Mapping Tool supports the domain-aware approach through the provision of the two domain-specific mechanisms: the canonical user model and user model mapping types. The design of both of these mechanisms was described in detail in Chapter 3. In the Mapping Tool, the canonical model is used in the visual creation of the mappings. This is fundamentally different to existing mapping tools as it allows a clearer and more consistent visualisation, allowing the administrator to continually map to the same user model structure. The user model mapping types are also used in the visual creation of mappings. This approach allows the administrator to generate

and reuse specialised user model mappings to support domain-specific mapping tasks.

As shown in Figure 4.2, the Mapping Tool stores the various schemas, instances and mappings in the Database and can request them as needed depending on the current task being conducted in the tool. The Mapping Tool communicates with the Database using the Mapping Tool Service. This service allows various tasks to be completed such as saving and retrieving user models and executing mappings. The Mapping Tool Service along with the Database are designed to be server-side components whereas the Mapping Tool is designed to be a client-side component. Once the mappings have been completed in the Mapping Tool and deployed, they are also available from the Database to the Translation Service for use during the automatic user model translation process.

4.2.3 Implementation

As a web-based application, a combination of technologies were used to implement the Mapping Tool. The graphical user interface component of the tool was created using Adobe Flex [Adobe, 2012]. Flex is a software development kit for the development and deployment of cross-platform web applications based on the Adobe Flash platform. Flex combines the MXML and Actionscript languages. MXML is an XML-based user interface markup language that is primarily used to declaratively lay out the components of a user interface. Actionscript is an object-oriented language used primarily in the development of websites and web applications. These two languages compile to create a SWF file that runs client-side in the Flash player of the user's browser. A SWF file is an Adobe Flash file format used for animated multimedia and vector graphics.

The Mapping Tool stores its user models and mappings in the Database, and eXist [EXist, 2012] was chosen as the basis for this implementation. EXist is an open source database management system built on XML technology. EXist uses the XML-based query language XQuery to manipulate its data. As the mappings in this approach are also created using XQuery, the eXist database is the basis for the execution of the mappings and provision of the user model translation process.

For the client-side Mapping Tool to communicate with and perform actions using the server-side Database, a Mapping Tool Service was also implemented. The Mapping Tool Service is a server-side component which controls the exchange of user models and mappings between the

Mapping Tool and the Database. It also manages the execution of the user model mappings by using the eXist database's native support for XQuery. The Mapping Tool Service is constructed as a Java-based RESTful web service. The Mapping Tool communicates with the Mapping Tool Service using HTTP to exchange the user models and mappings.

As described in the development methodology, an iterative approach was adopted during the implementation of version 1 of the Mapping Tool. There were three major iterations of the Mapping Tool which consisted of planned major increments to the functionality of the tool. The iterations were also improved using walkthroughs of the Mapping Tool functionality. The three iterations are:

- **Iteration 1 of the Mapping Tool** consisted of a graphical tool with basic functionality to visualise and create mappings between heterogeneous user models. The architecture of iteration 1 consisted of a basic graphical tool only. This iteration did not include the Database to store user models and mappings or the Mapping Tool Service to interact with the Database. The user models and mappings were stored in the Mapping Tool to allow for the rapid evolution of the mapping type structures.

The graphical user interface of the first iteration of the Mapping Tool is shown in Figure 4.3. It was entirely text-based and was constructed using a basic set of pre-existing Flex components. To visualise the various user models in the user interface, they were represented using the common tree visualisation. The external user models were represented on the left side of the user interface and the canonical user model was represented on the right side of the user interface. This tree-based design was similar to those used in the various existing commercial and research mapping tools such as Altova Mapforce. To create mappings in the user interface basic mapping types could be selected from a list. Source and target element selections could be made from these user models and, in the central area of the user interface, a basic mapping could be constructed by specifying different parameters. There was no testing capability in this first iteration of the Mapping Tool. To test the mappings, they had to be manually executed using an XQuery processor and sample user model instances.

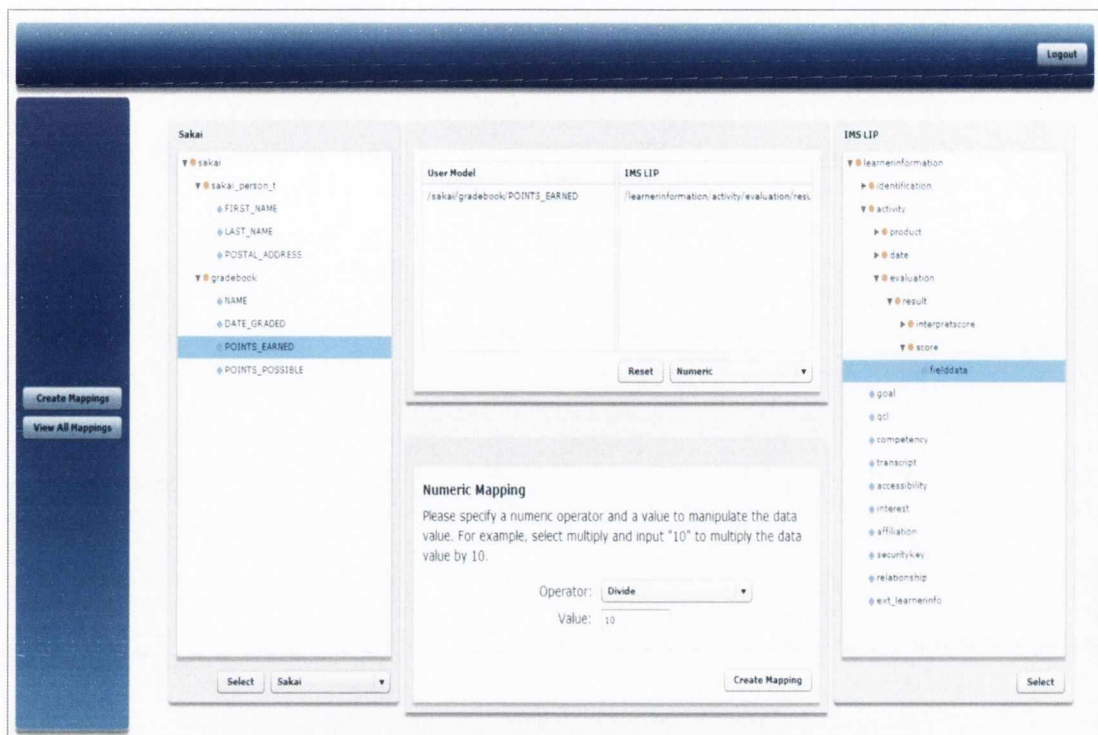


Figure 4.3: Iteration 1 of the Mapping Tool

- **Iteration 2 of the Mapping Tool** consisted of improved existing functionality and new functionality to test and maintain mappings between heterogeneous user models. The architecture of iteration 2 consisted of an extended graphical tool with additional components. This included the Database to store user models and mappings and the Mapping Tool Service to interact with the Database. These components allowed the user models and mappings to be managed across different administrator sessions. The Database also acted as an XQuery processor allowing the implementation of a mapping testing stage in the Mapping Tool. In this iteration, the evolution of the user model and mapping representations continued with the introduction of more complex mapping types such as instance mappings.

The graphical user interface of the second iteration of the Mapping Tool is shown in Figure 4.4. This iteration was also mainly text-based with the two tree visualisations of the user models aligned side-by-side. This was the first basic attempt to modify the typical design of mapping tools where mappings are created in a central area between two tree-based data models. The mapping creation area was moved to the right side of the user interface to accommodate the user model layout and drag and drop functionality was added to allow the selection of various user model elements. One of the key improvements in this iteration was the addition of a testing

stage where mappings could be verified using sample user model instances. This allowed administrators to rapidly test created mappings and ensure that they performed the correct functions. The addition of the Database also allowed user models and mappings to be saved and reused.

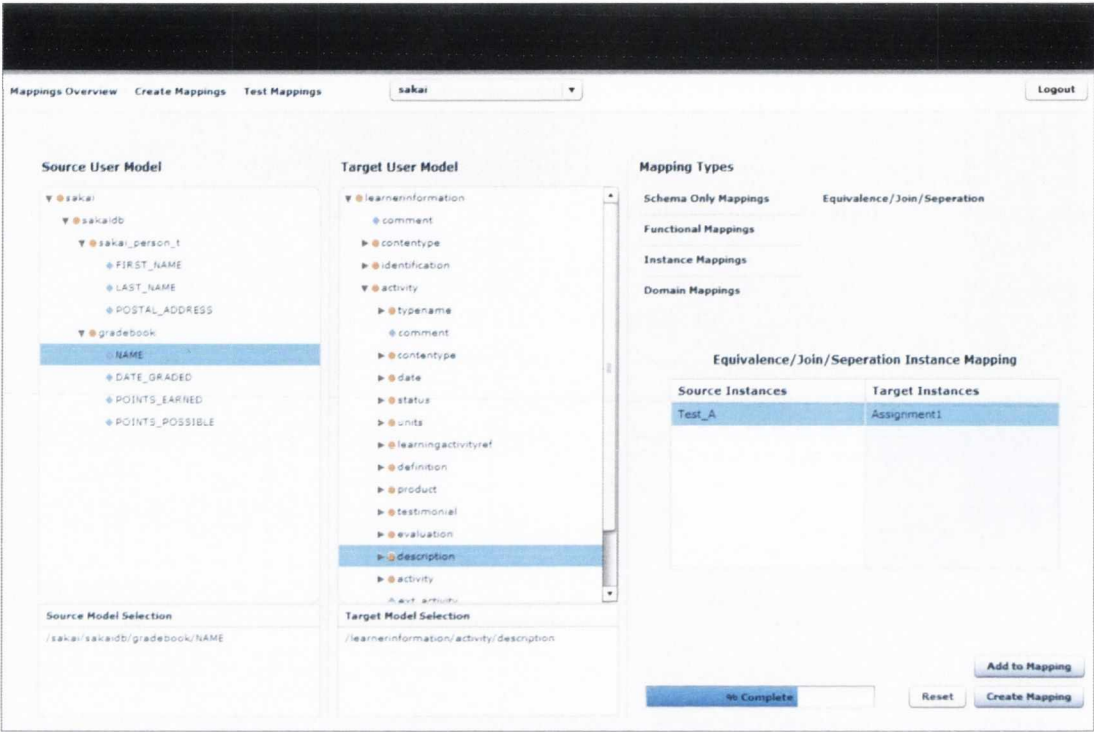


Figure 4.4: Iteration 2 of the Mapping Tool

- **Iteration 3 of the Mapping Tool** consisted of improved existing functionality and new functionality to graphically visualise the mapping creation between heterogeneous user models. The Database and Mapping Tool Service were refined to improve the exchange and management of the user models and mappings. As in previous iterations, the evolution of the user model and mapping representations continued. Existing mapping types were refined and more complex mapping types such as interval mappings were defined.

The graphical user interface of the third iteration of the Mapping Tool is shown in Figure 4.5. This iteration introduced a major change in the visualisation of the user model mapping process. While the external user models continued to be represented using a tree visualisation, a graph-based visualisation was implemented for the canonical model. The graph was separated into different categories of user model data that expanded left-to-right and could be navigated by using drag and drop and zoom functionality. To create mappings, elements from the external

user models were selected from the tree on the right side of the screen. An equivalent element was then identified in the canonical model graph structure and an appropriate mapping type was selected from a list. This created a mapping container that extended from the element on the canonical model graph. Various parameters could then be specified and the user model element could be dragged and dropped into the mapping container to create the mapping.

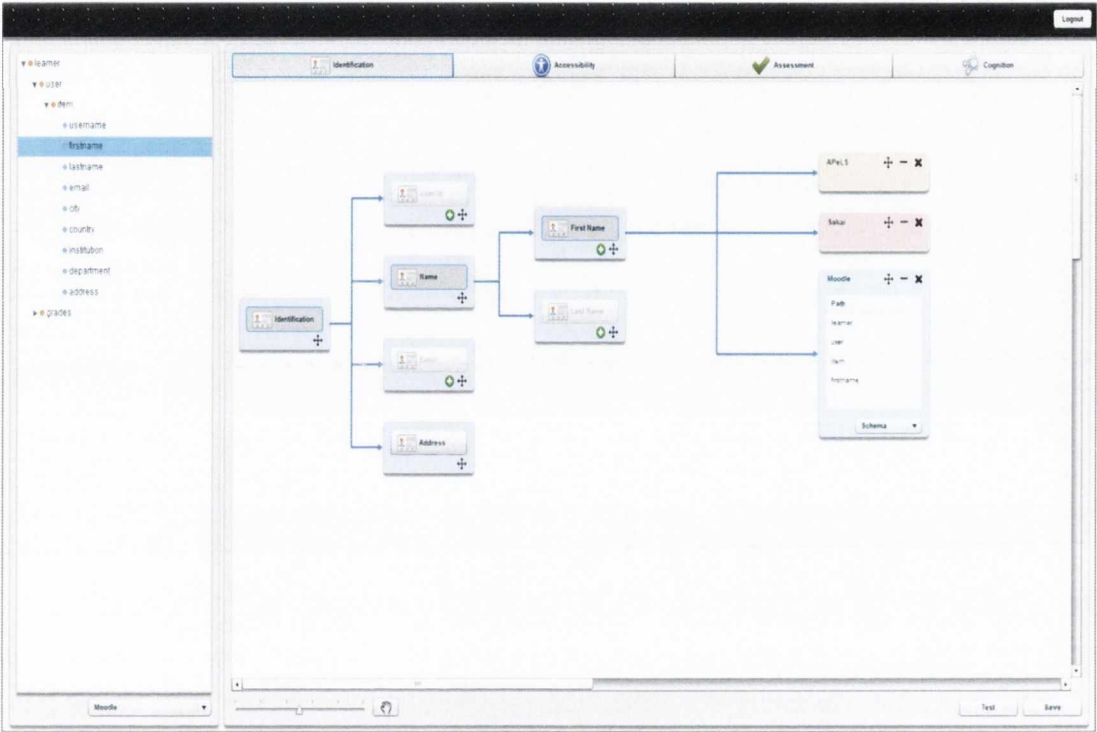


Figure 4.5: Iteration 3 of the Mapping Tool

The evolution of the Mapping Tool through these iterations has influenced the implementation of the main data models: the user models and the mapping types. In the following sections, the implementation of both the user models and mapping types is described.

User Models

Different functionality in the Mapping Tool, such as user model visualisation or mapping testing, imposes different requirements on the representation of the user models. The external user models that are being exchanged, such as those defined in Section 3.4.1, are represented in two forms. The first representation is used to visualise the user model in the Mapping Tool. This requires an attribute-based XML format that Flex uses as input to its visual tree component. The second

version of the user model is the sample instance which is used during the testing of the mappings. These are representative of the user models that are applied within the external systems and can contain instance values to test different aspects of the mappings. Examples of the external user models used in the Mapping Tool are presented in Appendix A.

The Mapping Tool uses an internal canonical user model, presented in Section 3.5.1, that acts as a common exchange representation for all external user models. Similar to the external user models, there are two different versions for visualisation purposes and exchange testing. The visualisation of the canonical user model is different from the visualisation of the external user models and is represented in the form of a hierarchical tree graph developed using Flex. As a result, the visual user model is currently represented in MXML, an XML-based user interface markup language in Flex. A second version of the canonical user model is maintained to handle the testing of user model exchange and can be seen in Appendix A. This version of the canonical user model is XML-based and similar in structure to the external user models described earlier. A canonical user model instance is maintained for each user as the central point for the translation between the source and target user models. This avoids any synchronisation issues when translation is being performed for multiple users simultaneously.

Overall, the user models used in the Mapping Tool are primarily XML-based. This contrasts with some of the surveyed state of the art approaches which use RDF [World Wide Web Consortium (W3C), 2012b]. FUSE was designed to support existing systems with heterogeneous forms of user information. RDF is not currently supported in many systems which are key targets for interoperability. For example in education, learning management systems have extensive support for web services and XML but little or no support for RDF. As a result, an approach based on XML was selected to provide more widespread applicability to existing systems and different types of user information.

Mapping Types

The final representation of the mapping types evolved through the three major iterations of the Mapping Tool. The key requirements in the definition of the mapping type representation were that it could support complex mappings, could be easily manageable in a web-based user interface, was easy to generate automatically and was easily executable as a translation language. To achieve all of these requirements, it was decided to represent the mappings using two

complementary representations; a high-level XML representation for visualisation and a low-level XQuery representation for executable translation.

- **XML representation** of mappings was used in the graphical user interface of the Mapping Tool. XML was selected because it is suitable to store, retrieve and manage mappings. Flex also has good support for working with XML. Figure 4.6 presents an example of the XML mapping representation. More detailed examples can also be found in Appendix B. The XML example represents two *equivalent first name (A1 schema equivalence)* mapping types. These mapping types demonstrate the canonical model approach where the source (UM1) and target (UM4) user models are both mapped to the canonical user model (UM0). This is achieved using XPath to specify the path to the selected element in each user model. The path can contain instance values to identify more specific sections of the user model. The mapping has a number of associated parameters which are inputs into the various mapping types, for example the character “*” is added when the administrator performs the multiplication of a numeric value. Metadata is also defined to support the management and visualisation of the mappings in the Mapping Tool. This includes the mapping id, group, type, name, description and display information. The mapping type is a simple string identifier which is subsequently used to change the XML mapping into an executable XQuery mapping.

```

<mappings>
  <mapping><uid>BFA4C409-125F-F61C-BB31-36012D8B233A</uid>
    <idnumber>1</idnumber>
    <group>1</group>
    <type>text_121</type>
    <name>Identification</name>
    <description>equivalent first name</description>
    <display x="339.35" y="765.6" color="16771286">max</display>
    <sources>
      <source um="UM1">/learner/general/name/forename</source>
    </sources>
    <targets>
      <target um="UM0">/user/identification/name/firstname</target>
    </targets>
    <operators>
      <operator_a/>
      <operator_b/>
    </operators>
    <values>
      <value_a/>
      <value_b/>
    </values>
  </mapping>
  <mapping><uid>6B3587D1-BEA6-7CC4-AD5D-36012D8B8BC0</uid>
    <idnumber>2</idnumber>
    <group>1</group>
    <type>text_121</type>
    <name>Identification</name>
    <description>equivalent first name</description>
    <display x="512.1" y="758.9" color="14086911">max</display>
    <sources>
      <source um="UM4">/learner/user/item1/firstname</source>
    </sources>
    <targets>
      <target um="UM0">/user/identification/name/firstname</target>
    </targets>
    <operators>
      <operator_a/>
      <operator_b/>
    </operators>
    <values>
      <value_a/>
      <value_b/>
    </values>
  </mapping>
</mappings>

```

Figure 4.6: Mapping example in XML representation

- **XQuery representation** of mappings was used to provide executable translation. Different approaches were examined, ultimately leading to two main XML-based options, XSLT and

XQuery. Both are widely used and extensive translation languages but XQuery was selected because it was less syntactically complex and, as a result, easier to generate automatically from the first XML-based mapping representation. Another important factor in the mapping format is the requirement for XML update functionality. The eXist database is used with the Mapping Tool and this supports an XQuery-based update of XML therefore providing a suitable environment for the mappings-based translation of XML user models. Figure 4.7 presents an example of the XQuery mapping representation. More detailed examples can also be found in Appendix B. The XQuery example consists of two parts; the mapping from the source user model to the canonical user model is executed first and then the mapping from the canonical user model to the target user model is executed. The XQuery representation can vary for different mapping types but typically takes the form of a series of if-else statements. Firstly, the mapping checks the source value and if it is empty, then the target value is replaced with an empty element. If the source value is not empty then the mapping uses the XML update mechanism to replace the target value with the source value. In more complex mappings, the parameters are also applied, for example, to multiply a numeric source value by ten the parameters “*” and “10” would be added to the end of the source path.

```

<mappings>
  <mapping>
    {
      if (doc($UM1)/learner/general/name/forename != '')
      then
        update replace
        doc($UM0)/user/identification/name/firstname
        with
        <firstname>{doc($UM1)/learner/general/name
        /forename/text()}</firstname>
      else
        update replace
        doc($UM0)/user/identification/name/firstname
        with
        <firstname/>
    }
  </mapping>
  <mapping>
    {
      if (doc($UM0)/user/identification/name/firstname != '')
      then
        update replace
        doc($UM4)/learner/user/item1/firstname
        with
        <firstname>{doc($UM0)/user/identification/name
        /firstname/text()}</firstname>
      else
        update replace
        doc($UM4)/learner/user/item1/firstname
        with
        <firstname/>
    }
  </mapping>
</mappings>

```

Figure 4.7: Mapping example in XQuery representation

A potential benefit of using the two-part mapping representation is that it provides the flexibility to add more executable translation formats in the future. Mappings can be represented using other languages such as SPARQL [World Wide Web Consortium (W3C), 2012d] or EDOAL [David et al., 2011]. These languages could be supported in the Mapping Tool without having to make any significant changes. To change the executable mapping representation would only require the addition of a new component that can change the XML mapping representation to the new format instead of XQuery.

4.3 Preliminary Analysis of Usability in User Model Mapping

4.3.1 Objectives and Methodology

Following the development of the third iteration of the Mapping Tool it was evaluated in a user trial. The objective of this initial trial was to **allow participants to use the tool and capture qualitative feedback about modifications that could be made to improve the user experience.** The user trial consisted of semi-structured interviews with each participant after they had used the tool to complete a selection of mapping tasks. The participant responses about different aspects of the tool were categorised and encoded to allow analysis of the results and plan further development of the tool. A second user trial focusing on the qualitative and quantitative evaluation of version 2 of the Mapping Tool is presented in Chapter 5. The evaluation was conducted with seven participants who were each given the same basic training in the Mapping Tool. The participants were selected from a pool of computer science postgraduate students, research staff and academics in Trinity College Dublin. The participants used the tool to complete mapping tasks between existing heterogeneous user models. Each mapping task consisted of multiple stages and required many interactions with different aspects of the tool. The main sections of the trial were:

1. To start the trial, the functionality of the tool was demonstrated and the participants were allowed to use the tool to complete some sample tasks.
2. Following this, the participants were required to complete five mapping tasks of increasing complexity between four heterogeneous user models. The selection of user models presented in Section 3.4.1 was used to provide the source and target user models in this trial. The user models are also shown in Table 4.1.

Table 4.1: Trial user models

User Model	System
UM0	FUSE Canonical Model
UM1	APeLS
UM2	CUMULATE
UM3	Sakai
UM4	Moodle

The mapping tasks were created using the mapping types selected in Section 3.5.2 and represent common user model interoperability tasks between user models in the application area of

education. The selected mapping tasks represent the identification, creation and verification of executable mappings between existing systems. Each of these tasks required the participants to perform multiple steps including examining the user models, identifying the appropriate section of the source and target user models, selecting the appropriate mapping type, and constructing and testing the mapping. To analyse different types of mappings, the tasks were selected in terms of increasing complexity and the level of domain knowledge required.

3. When the mapping tasks were completed, the participants took part in a semi-structured interview process to gather feedback about various key aspects of the tool. The interview results were assembled into a list of issues and recommendations which were then used to identify the most relevant modifications to the tool. The modifications were implemented in the second version of the Mapping Tool which is described in detail in Chapter 5.

Mapping Tasks

The five mapping tasks that the participants completed during the course of the trial are:

1. Map firstname in UM1 to its equivalent in UM4.

This mapping task represents the translation of the first name of the user. The user model mapping type *equivalent first name (A1 schema equivalence)* is used to translate the user's first name from the source user model 1 to the canonical user model. The same user model mapping type, *equivalent first name (A1 schema equivalence)*, is subsequently used to map from the canonical user model to the target user model 4.

2. Map the appropriate part of address from UM3 to country from UM4.

This mapping task represents the translation of the country value of the user's address. The user model mapping type *split address and get country (A3 schema split)* is used to split and translate the country value of the user's address from the source user model 3 to the canonical user model. The basic user model mapping type, *equivalent address (A1 schema equivalence)*, is used to map from the canonical user model to the target user model 4.

3. Map the reflector score from "honey and mumford learning style scores" in UM1 to its equivalent in UM4.

This mapping task represents the translation of a learning style score. The user model mapping type *equivalent learning style (B1 instance equivalence)* is used to translate a score for a

specific learning style in the source user model 1 to the canonical user model. The user model mapping type, *equivalent learning style (A1 schema equivalence)*, is subsequently used to map from the canonical user model to the target user model 4.

4. Map the final exam score for course “Database Systems 2” (maximum score of 20) in UM3 to the grade for course “Advanced Database Systems” (maximum score of 100) in UM4.

This mapping task represents the translation of a course assessment across different grading scales. The user model mapping type *grades 0-20 to 0-100 (C2 functional numeric)* is used to create the mapping from user model 3 to the canonical user model. The basic user model mapping type, *equivalent assessment (A1 schema equivalence)*, is subsequently used to map from the canonical user model to the target user model 4.

5. Map the knowledge value for concept “db.tables.create” in UM1 to the knowledge value of concept “TableCreation” (maximum score of 1) in UM2.

This mapping task represents the detection of a text-based competency and its translation to a numeric competency value. The user model mapping type *SQL concept to 0/1 score (C4 functional detection)* is used to create the mapping from user model 1 to the canonical user model. The basic user model mapping type, *equivalent competency (A1 schema equivalence)*, is used to map from the canonical user model to the target user model 2.

To complete the mapping tasks, the participants used the domain-aware approach in the Mapping Tool represented by the two domain-aware mechanisms; the canonical user model and user model mapping types. The participants were required to navigate and select elements from the canonical user model. The canonical user model represented multiple overlapping domains consisting of identification information from the user domain, assessment and competency information from the education domain, and databases and SQL information from the subject domains. To create the mappings, the participants were required to navigate, select and apply appropriate mapping types. In the Mapping Tool, multiple user model mapping types were provided which represented typical mappings in user models in the area of identification, assessment and competencies. These mapping types provided multiple potential combinations in each tool from which the participant had to select the required options and construct the mappings.

4.3.2 Semi-Structured Interviews

After participants had completed the mapping tasks, semi-structured interviews were conducted [Cohen et al., 2007]. The main questions used in these interviews can be seen in Appendix C and cover the key aspects of the Mapping Tool such as the canonical user model visualisation and the mapping creation process. The main objective of these interviews was to identify which aspects of the tool the participants liked and disliked, from the perspective of both general tool usability and task-based tool usability. The discussions were not limited to the predefined list of questions and many other related issues were also discussed.

The results of this trial are shown in Table 4.2 which presents a summary of the responses given by the participants. In order to analyse the issues and recommendations, the responses have been categorised into key areas of the Mapping Tool. The first general category includes responses about the overall tool and trial. The next four categories include responses relating to the source and target user models, the canonical user model, mapping creation, and mapping testing. The final category includes additional issues that were observed and noted during the course of the mapping tasks by the trial administrator. The number of participants that described an issue or recommendation is also listed.

Table 4.2: Semi-structured interview results

Category	Id	Usability Issue or Recommendation	No. of Users
General	A1	Provide more training in tool.	6
	A2	Add more examples and guidance within the tool.	2
	A3	Clarify and refine terminology.	2
	A4	Reduce number of clicks required to create mappings.	1
	A5	Add project-based management.	1
Source and target user models	B1	Make path and instance selection easier.	5
	B2	Automatically expand source tree nodes.	3
	B3	View instances from source tree.	2
	B4	Source model dropdown list not obvious.	1
	B5	Highlight path of tree selection and XML.	1
Canonical user model	C1	Make easier to understand the directionality of mappings.	2
	C2	Change knowledge “score” to “value”.	2
	C3	Allow editing of canonical model.	1
	C4	Create mappings to other higher levels of canonical model.	1
Mapping creation	D1	Improve resizing and icons of mapping container.	5
	D2	Make selection of the required mapping type easier.	3
	D3	Do not allow blank mappings.	2
	D4	Allow reuse of mappings (mapping templates).	2
	D5	Provide mapping summary/history.	2
	D6	Provide mapping activity/progress log.	1
	D7	Provide semi-automatic matching.	1
	D8	Automatically realign connectors to avoid overlap.	1
Mapping testing	E1	Add search feature in XML and tree.	5
	E2	Improve readability of XML user models.	4
	E3	Make easier to select source and targets.	2
	E4	Display only updated segment of XML user model.	2
	E5	Hyperlink XML changes to mappings.	1
	E6	Highlight target user model tab during testing.	1
	E7	Automatic preview of testing during mapping creation.	1
	E8	Select source and target during mapping creation.	1
Observed	F1	Reset user models when entering testing area.	6
	F2	Fix issue on instance selection below level of path.	6
	F3	Switch XML to current tree node automatically.	4
	F4	Alert when no updates have been achieved during testing.	4
	F5	Automatically scroll to updated XML element in testing.	3
	F6	Make extra screen space and zoom functionality more obvious.	3
	F7	Make tooltip functionality more obvious.	2
	F8	Improve readability of numeric operators.	2

As can be seen in Table 4.2 the most common general responses were the need for more training, examples, guidance and clearer terminology within the tool. In relation to the source and

target user models, the most common responses were the difficulty of selecting schema elements and instance values, and the automatic expansion of the user model tree visualisations. For the canonical user model, the most common responses referred to some difficulty understanding the shared canonical model approach and some of the terminology used in the model. In mapping creation, the participants wanted changes to the mapping containers and the selection of the mapping types. In mapping testing, the participants wanted search functionality and changes to the XML model visualisation. Finally, some observed issues included resetting the XML user models when changing between creation and testing areas and minor issues in instance selection.

4.4 Discussion

Version 1 of the Mapping Tool was evaluated with users to assess the current level of usability of the tool and to get feedback about any modifications that could be made to improve the user experience. The results have shown that there were some issues with usability and certain modifications could be made to improve the tool.

- **General** - participant responses were related to both the tool and the overall trial. In the tool, the participants wanted more guidance, examples and clearer terminology. These responses were largely due to the early prototype nature of the tool. The main trial-related issues consisted of the participants requesting more training in the tool and clearer terminology in the mapping tasks. These responses were due to the overall difficulty of the trial and the limited time participants had to use the tool. These issues would be addressed in subsequent versions of both the tool and the trial.
- **Source and target user models** - most participants found the drag and drop functionality for path and instance selection difficult to use. During the course of the discussions it was found that this section of the process did not perform as the participants expected. The functionality needed to be redesigned to adopt a more standardised approach. It was also suggested that the navigation could be improved by the automation of some features of the source tree visualisation. These improvements are important because they would reduce the workload of the administrator when examining the various sections of multiple user models for potentially equivalent mappings.
- **Canonical user model** - there were no significant issues encountered by the majority of

participants. However, some participants did have issues understanding the shared canonical user model approach and how that related to the directionality of the mapping process. Some participants expected to explicitly select the source and target user models during the mapping creation. However, in the canonical model approach all mappings are created to the canonical model. The ability to specify directionality (source and target user models) is performed in the testing area after the mappings have been created. This issue generally arises if participants have some prior experience with direct mapping approaches and generic mapping tools. In the Mapping Tool, modifications can be made in this area to emphasise and clarify the process of mapping to a single shared canonical user model.

- **Mapping creation** - participants wanted improvements to the usability of the mapping containers used to specify mappings and the process of selecting mapping types. These issues required relatively small functional and aesthetic modifications to the tool. Other issues that were mentioned that required more complex implementation were the addition of a mapping summary and the ability to reuse mappings. These features would greatly enhance the usability of the tool, particularly from the perspective of long-term management of user model interoperability by administrators.
- **Mapping testing** - the participants wanted to make the process simpler by adding a search feature to quickly identify particular sections of the user model schemas and instances. They also suggested that the legibility of the XML be improved through better visualisation. The majority of the requested improvements would enhance the capability of the mapping testing area. The testing area is a major feature of the tool and helps support the provision of practical user model interoperability. It is also a feature that is not commonplace in the related research approaches for user model interoperability.
- **Observed** - participants also encountered issues that were observed by the administrator of the experiment. In most cases, these were small issues that the participants were not aware of and could be removed easily. They mainly consisted of minor issues in the visualisations in mapping creation and testing areas such as the schema tree list and the XML.

Overall, the stated objective of the trial was to allow participants to use the tool and capture qualitative feedback about modifications that could be made to improve the user experience. This was achieved and the results of the user trial identified many usability issues that could be resolved in version 2 of the Mapping Tool. Version 2 will be used in the main evaluation of this research.

The Mapping Tool will be compared against a leading commercial mapping tool to analyse the impact of the domain-aware approach from the perspective of user model mapping.

4.5 Summary

This chapter detailed the first cycle of design, implementation and evaluation of the FUSE Mapping Tool. The Mapping Tool allows the manual creation of mappings between heterogeneous user models and implements the FUSE user model mapping process which was described in Chapter 3. The Mapping Tool functions in conjunction with the Translation Service, which implements the FUSE user model translation process, to form a novel domain-aware approach to user model interoperability.

In this chapter, the development of the FUSE Mapping Tool was presented initially. This consisted of the methodology that was employed throughout the course of its development. An architecture and graphical user interface for the Mapping Tool were designed based on the FUSE user model mapping process. The user interface was then implemented in an iterative development cycle. In this cycle, the user models and mappings types that describe how the user information is managed and translated were also defined and developed. Following the initial iterations of version 1 of the Mapping Tool, a preliminary user-based evaluation was conducted to assist in the development of version 2 of the tool. The user trial consisted of semi-structured interviews with participants after they had completed a selection of user model mapping tasks. The results of the interviews generated a list of usability issues and recommendations in key areas that were used to develop the second cycle of design, implementation and evaluation of the Mapping Tool. This second cycle is described in the next chapter.

Chapter 5

Development and Evaluation of the FUSE Mapping Tool (Cycle 2)

5.1 Introduction

This chapter presents the second cycle of design, implementation and evaluation of the Mapping Tool for user model interoperability. Following the preliminary analysis of the Mapping Tool in Chapter 4, a number of potential modifications were identified. Firstly, the modifications that were selected for implementation are described leading to the development of version 2 of the Mapping Tool. Next, a walkthrough of the main functionality is presented. The walkthrough uses the selected user models and mapping types to demonstrate how the tool performs the user model mapping process.

The evaluation of version 2 of the Mapping Tool which consisted of two experiments is then detailed. The first experiment was concerned with the accuracy and duration of the domain-aware mapping approach applied in the Mapping Tool. This is evaluated via a direct comparison with a generic approach to user model interoperability as applied in a commercial mapping tool, Altova Mapforce. The second experiment evaluates the relative usability of the Mapping Tool compared with Mapforce. These experiments assess the key aspects of the domain-aware approach applied in the Mapping Tool and compare them against the main state of the art approaches and tools in user model interoperability. Finally, the results are summarised and discussed to determine the key findings of the experiments.

5.2 Development of the FUSE Mapping Tool

5.2.1 Selected Modifications

Following on from the user trial of version 1 of the Mapping Tool in Chapter 4, a number of selections were made from the list of usability issues and recommendations that was generated. These selections were used to improve the overall level of usability and generate version 2 of the Mapping Tool. The modifications were selected based on their relevance to the research objectives. As can be seen in Table 5.1, 38 usability issues or recommendations were recorded and categorised during the semi-structured interview process which was conducted in the first user trial. From these issues and recommendations, 26 were selected and implemented in version 2 of the Mapping Tool. The remaining 12 were considered not to be relevant or outside the scope of the stated research objectives.

Table 5.1: Selected modifications in the Mapping Tool

Category	Id	Usability Issue or Recommendation	Applied
General	A1	Provide more training in tool.	Y
	A2	Add more examples and guidance within the tool.	Y
	A3	Clarify and refine terminology.	Y
	A4	Reduce number of clicks required to create mappings.	Y
	A5	Add project-based management.	N
Source and target user models	B1	Make path and instance selection easier.	Y
	B2	Automatically expand source tree nodes.	Y
	B3	View instances from source tree.	N
	B4	Source model dropdown list not obvious.	Y
	B5	Highlight path of tree selection and XML.	Y
Canonical user model	C1	Make easier to understand the directionality of mappings.	Y
	C2	Change knowledge “score” to “value”.	Y
	C3	Allow graphical editing of canonical model.	N
	C4	Create mappings to other higher levels of canonical model.	N
Mapping creation	D1	Improve resizing and icons of mapping container.	Y
	D2	Make selection of the required mapping type easier.	Y
	D3	Do not allow blank mappings.	Y
	D4	Allow reuse of mappings (mapping templates).	Y
	D5	Provide mapping summary/history.	Y
	D6	Provide mapping activity/progress log.	N
	D7	Provide semi-automatic matching.	N
	D8	Automatically realign connectors to avoid overlap.	N
Mapping testing	E1	Add search feature in XML and tree.	Y
	E2	Improve readability of XML user models.	Y
	E3	Make easier to select source and targets.	Y
	E4	Display only updated segment of XML user model.	N
	E5	Hyperlink XML changes to mappings.	N
	E6	Highlight target user model tab during testing.	Y
	E7	Automatic preview of testing during mapping creation.	N
	E8	Select source and target during mapping creation.	N
Observed	F1	Reset user models when entering testing area.	Y
	F2	Fix issue on instance selection below level of path.	Y
	F3	Switch XML to current tree node automatically.	Y
	F4	Alert when no updates have been achieved during testing.	N
	F5	Automatically scroll to updated XML element in testing.	Y
	F6	Make extra screen space and zoom functionality more obvious.	Y
	F7	Make tooltip functionality more obvious.	Y
	F8	Improve readability of numeric operators.	Y

The categories were defined based on key areas of the Mapping Tool. The first general category includes issues relating to the overall tool and experiment. The next four categories include issues

in the source and target user models, the canonical user model, mapping creation, and mapping testing. The final category includes observed issues that were noted during the course of the mapping tasks by the trial administrator. The following sections discuss the selection of the various issues and recommendations in each category:

- **General** - four of the five issues were addressed. This involved more training being incorporated in the design of new experiments (A1), more guidance (A2), and clearer terminology used throughout the tool (A3). Also, the number of clicks required to create mappings was reduced (A4). A recommendation to add a project-based management approach to the mapping process was not addressed. This would be a useful feature in a commercial-quality tool, allowing multiple large interoperability projects to be maintained (A5). However, in this case it is not relevant to the research objectives and not implemented in this version of the tool.
- **Source and target user models** - four of the five issues and recommendations were addressed. The selection of paths and instances was completely redesigned to make it more intuitive (B1). This also resulted in changes to the mapping containers where the values were dragged and dropped. The functionality of the user model tree visualisation was enhanced in a number of ways including the automatic expansion of tree nodes (B2). The drop-down list of user models was enlarged and made more prominent (B4). Finally, the path of the selected elements was also highlighted in the tree (B5). A recommendation to allow viewing of the user model instances when using the tree visualisation of the user model schema was not addressed (B3). This was not implemented in circumstances where it could potentially confuse the clear separation of the creation and testing areas in the tool. This issue could be examined in future versions of the tool.
- **Canonical user model** - two of the four issues raised were addressed. The canonical user model approach was redesigned to make the process easier to understand (C1). The separate categories of the canonical user model were merged into a single user model visualisation that could be navigated more easily using drag and drop and zooming functionality. The layout direction was also changed from a left-to-right approach to a top-to-bottom approach. This is a more common method to represent a large hierarchical tree-based model and made the navigation more intuitive. Small issues with terminology in the canonical model were also addressed by identifying more common terms used in existing standards such as IMS LIP (C2). Two recommendations were not implemented in this version of the tool. The first was the ability

to graphically edit the canonical model (C3). The canonical model can currently be modified by editing the underlying XML-based data model. Adding graphical support for this process would be useful in the long-term management of user model interoperability. The second recommendation addressed the ability to create mappings to levels above the base elements (C4). At present, the canonical model is in the form of a hierarchical data model where mappings are created to base elements to translate instance values. This could be modified to support mappings to levels above the base elements. This would potentially allow the mapping of different model structures such as graph-based ontologies. Neither of these aspects were relevant to the research objectives and would have required significant extensions of the tool. However, they are areas of interest for future work.

- **Mapping creation** - five of eight recommendations were implemented. These consisted of redesigning the layout and functionality of the mapping containers where the mappings are specified (D1). The process of creating mappings was made clearer, requiring fewer user actions and providing more user guidance (D2). Functionality was added to alert the user if newly created mappings were left blank or incomplete (D3). The selection of mapping types was significantly extended with functionality to save mappings as templates for future reuse (D4). A summary view of created mappings and templates was also added (D5). This allowed the user to generate lists of domain-specific mappings that could be reused in other interoperability tasks or by other users. The recommendations not implemented were a mapping activity log where administrators could review specific details of previous mappings they created (D6). While potentially informative for the user this was deemed a feature to support long-term use of the tool and therefore a potential aspect for future versions. The recommendation of semi-automatic matching is potentially very useful as a means to identify basic equivalence mappings automatically (D7). This is a key objective of the future work in the Mapping Tool. The automatic realignment of connectors would also be a useful feature that would make the visualisation slightly clearer however, this functionality was excluded because it would require significant implementation for minor usability improvements (D8).
- **Mapping testing** - four of the eight issues and recommendations raised by the participants were addressed. A search feature was added to both the tree-based schema view and the XML-based instance view of the user models (E1). This allows the user to easily identify the required sections of potentially large user models. The legibility of the XML models in the testing area was also improved (E2). The selection of the source and target models was redesigned (E3).

This was changed to a popup of checkboxes where the sources and target have to be selected to continue with the test. This resolves previous issues where users forgot to select or change the source and target models during testing because they were located at the side of the screen. Additional highlighting was also provided to immediately draw the user's attention to any changes in the sample user model instances during testing (E6). Issues not addressed include displaying only the updated segment of the XML user model (E4). This recommendation was infeasible as the inability to view the full user model could potentially lead the user to make errors during the mapping process. Three other issues relate to the closer integration of the mapping creation and testing areas. These issues are the hyperlinking of changes in the XML user models to their relevant mapping containers (E5), the automatic previewing of tests in the mapping creation area (E7), and the selection of source and target models during mapping creation (E8). These features are not relevant to the research objectives. However, the feasibility of providing more integration between the mapping creation and testing is a possible area of interest for future work.

- **Observed** - seven of eight features were implemented. Many of these were small issues which were quickly addressed such as resetting the instance user models when entering the testing area (F1) and the refinement of the instance selection process (F2). Improvements were made to emphasise the tab displaying the current XML user model in the testing area (F3) and also to automatically scroll to updated sections in the XML user model (F5). The drag and drop and zoom features were improved and made more prominent within the tool (F6). The tooltips that provided additional guidance were also improved to make them more useful (F7) and the legibility of the numeric operators was improved (F8), as previously they were too small and difficult to identify. The only recommendation that has not been implemented is the addition of alerts when no updates have been performed by the mappings during testing (F4). This would be useful but is not a key feature and will be implemented in the next version of the tool.

Overall, approximately two-thirds of the issues and recommendations identified in the user trial were implemented in version 2 of the Mapping Tool. Issues were addressed in each of the main categories, improving aspects of the general user interface, the source and target user models, the canonical user model, the mapping creation and the mapping testing. The selection of the issues which were addressed was based on the relevancy to the research objectives. The implemented modifications led to a significantly improved tool that provided a better representation of the FUSE approach to user model interoperability. A detailed step-by-step walkthrough of the functionality

of version 2 of the Mapping Tool is presented in the next section.

5.3 Mapping Tool Walkthrough

The following section presents a detailed walkthrough of the main functionality of version 2 of the Mapping Tool, shown in Figure 5.1. This walkthrough demonstrates the core novelty and effectiveness of the FUSE Mapping Tool: providing users with additional domain knowledge to support the creation of rich relationships between user models.

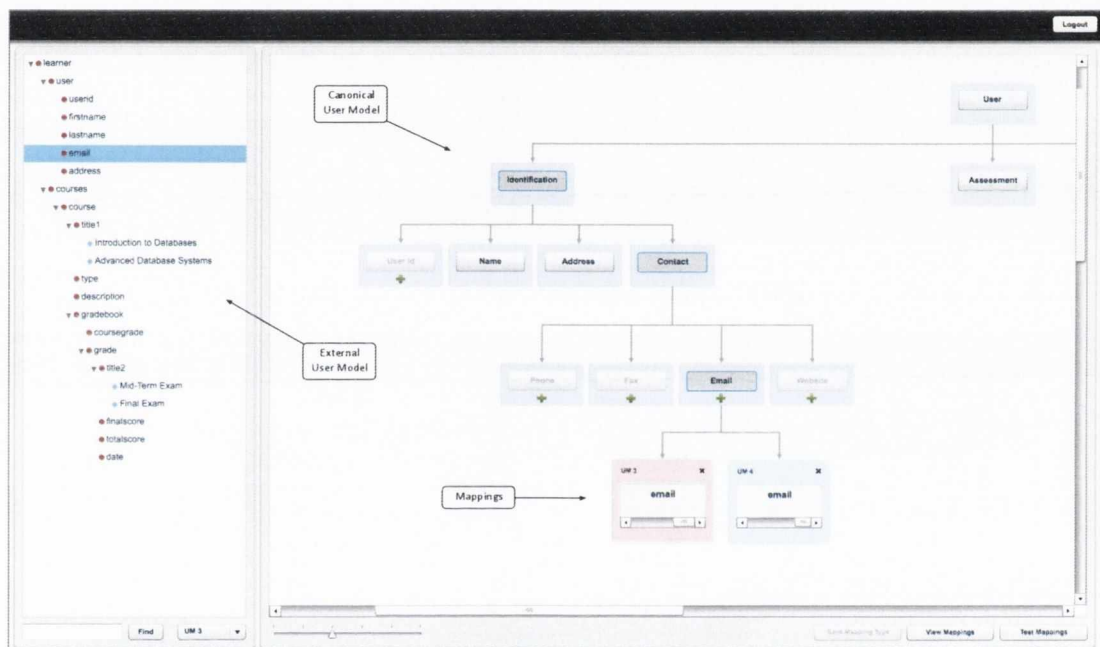


Figure 5.1: Version 2 of the Mapping Tool

The walkthrough uses the selected user models from Section 3.4.1 and the selected mapping examples from Section 3.5.2 to provide a representative demonstration of interoperability between multiple existing user models. The first basic example mapping describes each step of the process in detail. Following this, some more complex mappings are also summarised. To provide clarity and consistency in the walkthrough, the examples are one-to-one mappings between two user models. However, the system can also support more advanced many-to-many mappings between multiple user models.

5.3.1 Basic User Model Mapping Example

Mapping Example 1 - First Name

Mapping example 1 represents the translation of the first name of the user as shown in Figure 5.2. The user model mapping type *equivalent first name (AI schema equivalence)* is used to translate the user's first name from the source user model 1 to the canonical user model. The same user model mapping type, *equivalent first name (AI schema equivalence)*, is subsequently used to map from the canonical user model to the target user model 4.

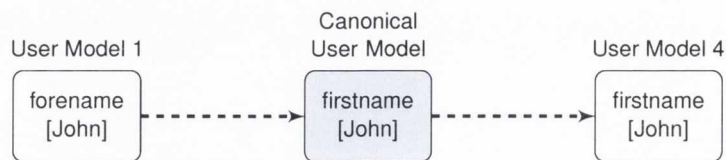


Figure 5.2: Mapping example 1

The steps involved in creating mapping example 1 in the Mapping Tool are:

1. User model 1 is imported into the Mapping Tool using XML-based schema and instance files.

2. User model 1 is navigated and the “forename” element is identified for translation (Figure 5.3).

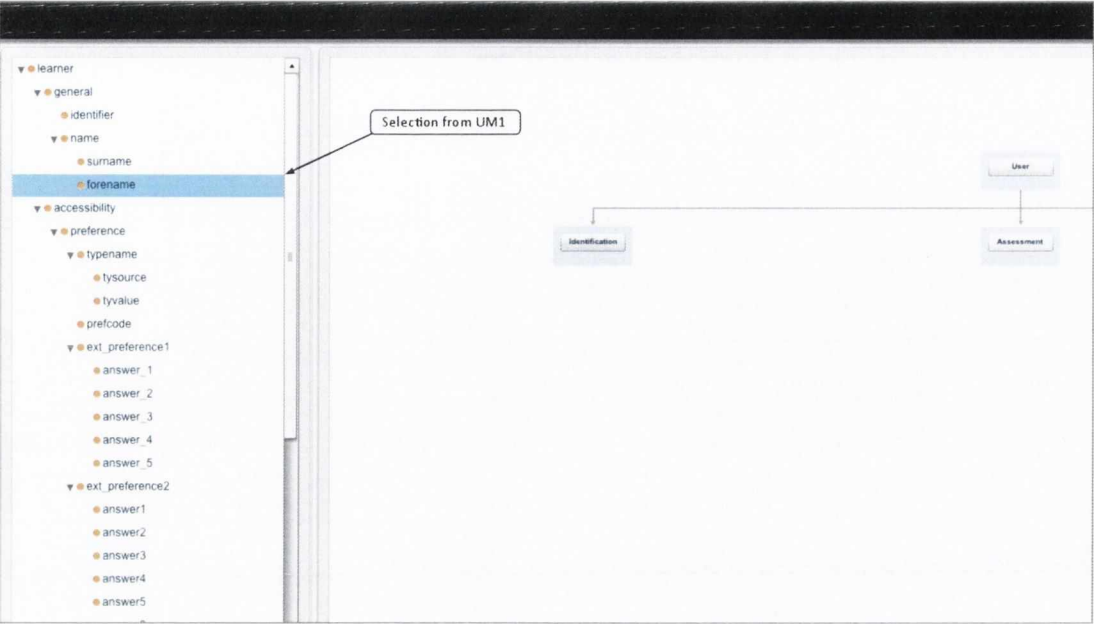


Figure 5.3: Navigation of user model 1 to “forename” element

3. The canonical user model is navigated and the equivalent “firstname” element is identified (Figure 5.4).

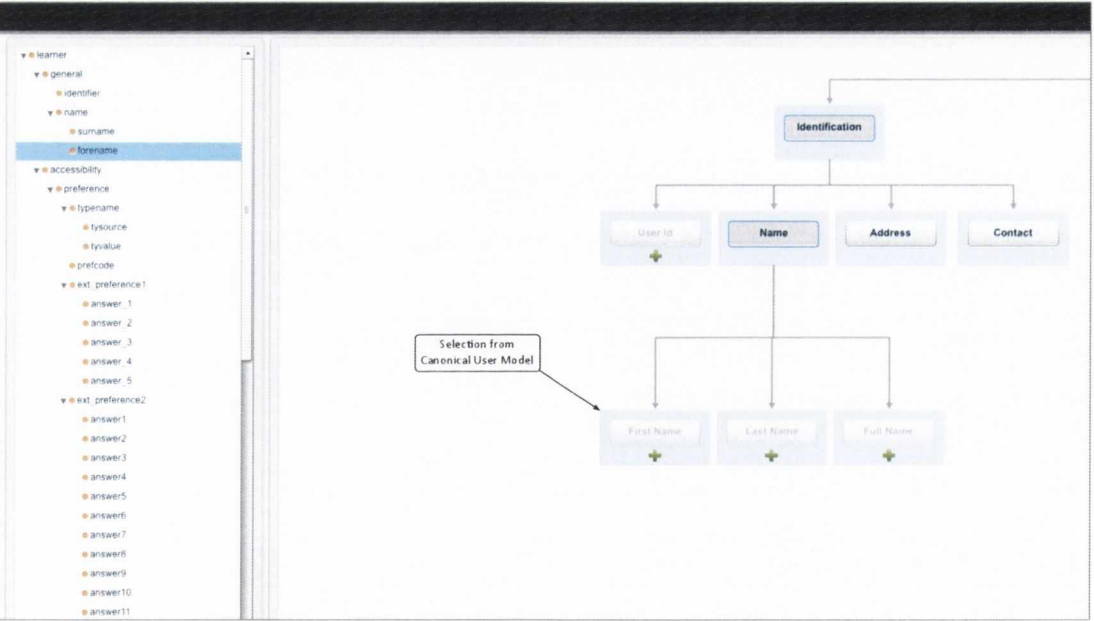


Figure 5.4: Navigation of canonical user model to “firstname” element

4. The user model mapping type *equivalent first name* is selected from the list of available mapping types (Figure 5.5).

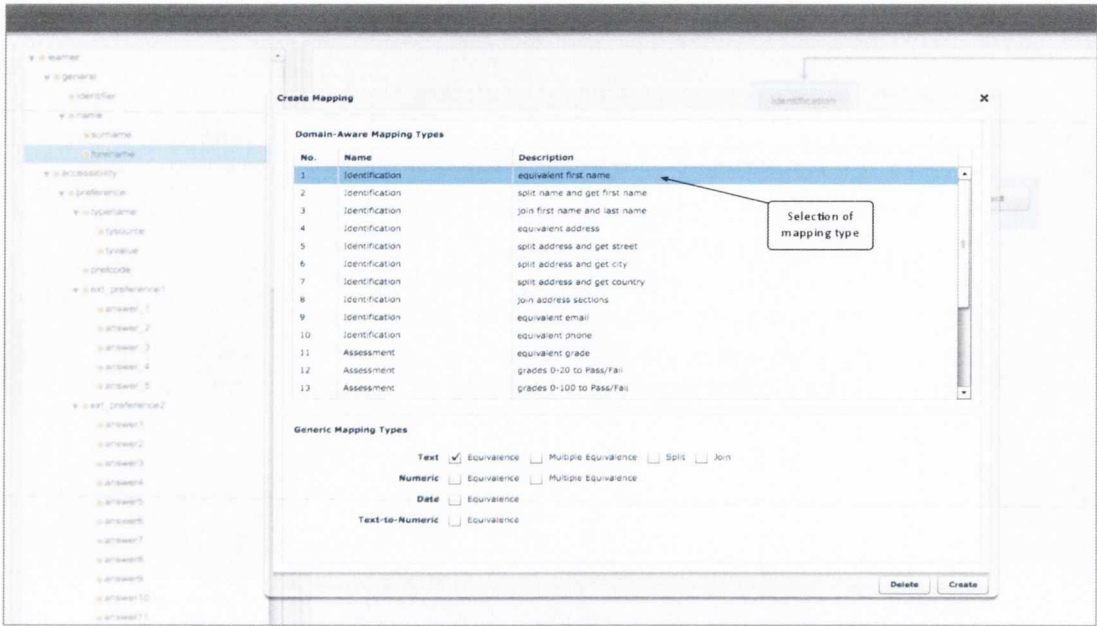


Figure 5.5: Selection of “equivalent first name” user model mapping type

5. The element "forename" is dragged and dropped from user model 1 to the newly created mapping type (Figure 5.6).

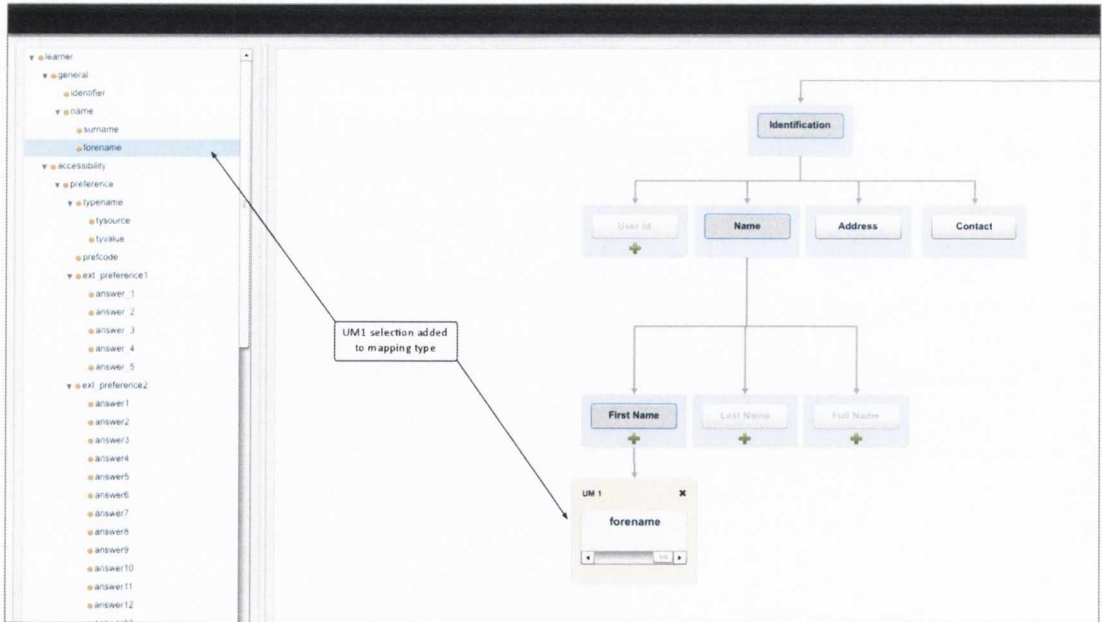


Figure 5.6: Selection of user model 1 “forename” element

6. Steps 1-5 are repeated for user model 4 and the equivalent “firstname” element is mapped to the canonical user model (Figure 5.7).

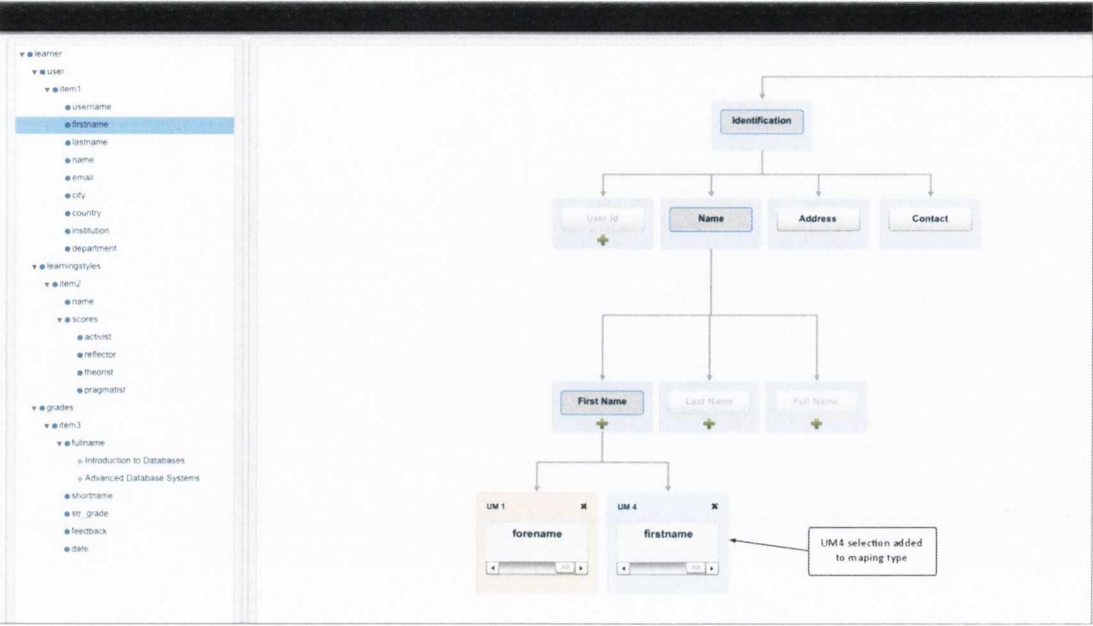


Figure 5.7: Steps 1-5 repeated to select “firstname” element from user model 4

7. The testing area is selected which is used to validate the mapping using sample user model instances (Figure 5.8).

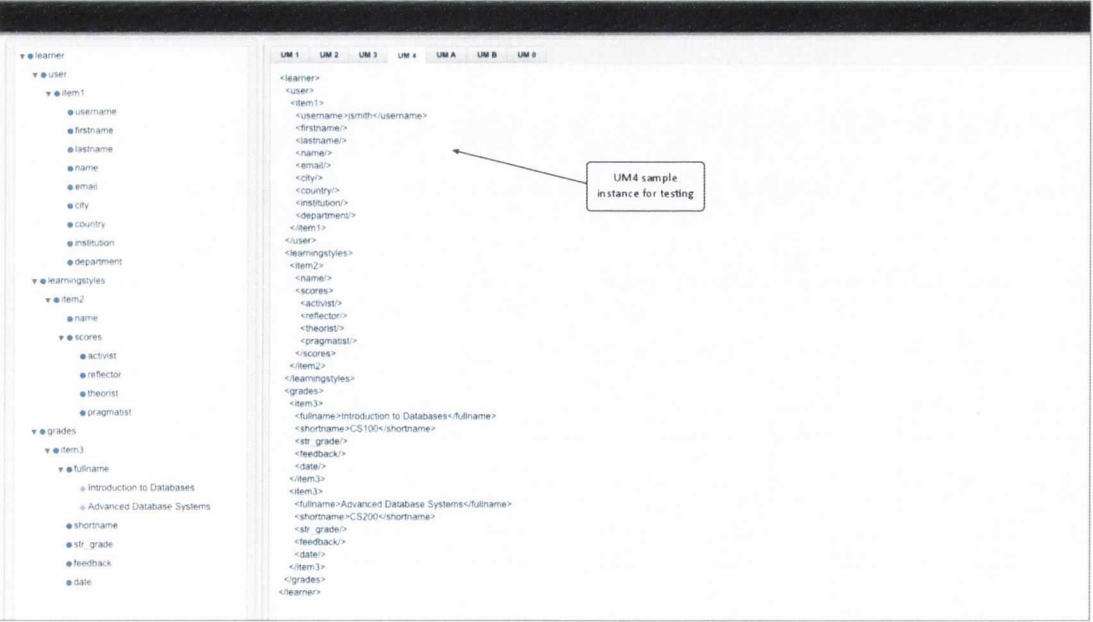


Figure 5.8: Selection of testing area to validate created mappings

8. User model 1 is selected as the source user model and user model 4 is selected as the target user model in the testing area (Figure 5.9).

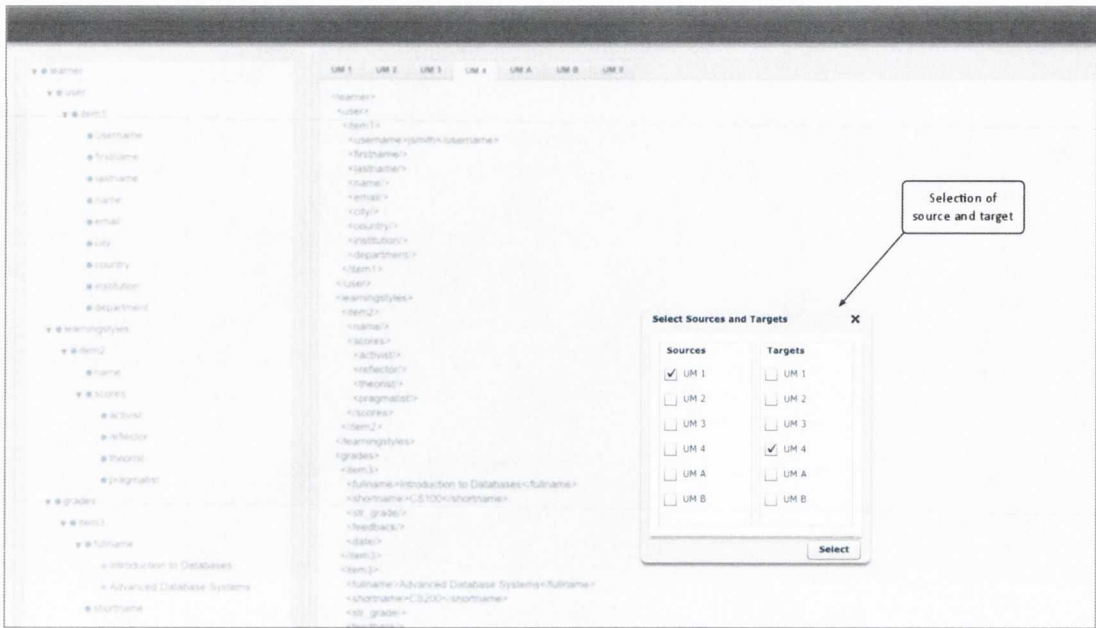


Figure 5.9: Selection of source and target for mapping testing

9. The translation is executed and the result is highlighted in user model 4 (Figure 5.10).

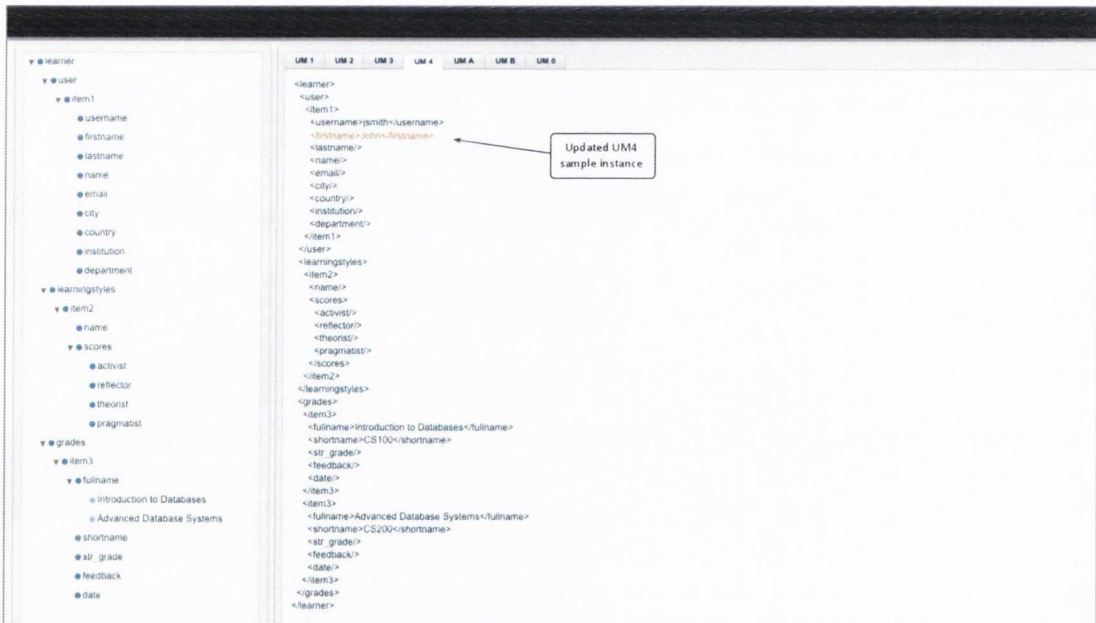


Figure 5.10: Result of mapping testing

10. The mapping is considered valid and deployed for use in the Translation Service.

5.3.2 More Complex Mapping Examples

Mapping example 1 demonstrated a basic translation of the user’s first name. However, providing interoperability between existing user models requires many different types of mappings that can vary greatly in complexity. As shown in Section 3.5.2, more complex mappings can be created to map specific instance data or perform different functional actions during the translation of data such as numeric manipulations and format conversions. Mappings can also be created to join and split multiple aspects of the user model data for translation. The Mapping Tool can support the creation and execution of the complex mapping types identified in Section 3.5.2. In this section, some additional examples are given which demonstrate how these more complex mappings are created in the Mapping Tool. As many of the steps are similar to mapping example 1, these examples are summarised to show the mapping visualisation in the mapping creation area rather than the full mapping process.

Mapping Example 2 - Address

Mapping example 2 represents the translation of the country value of the user’s address as shown in Figure 5.11.

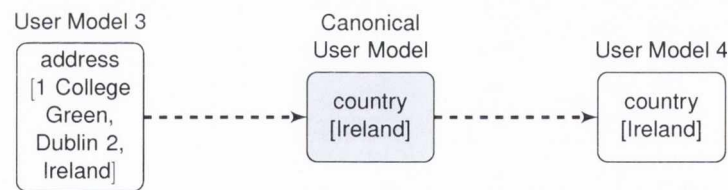


Figure 5.11: Mapping example 2

The key difference from mapping example 1 is in the selection of the mapping types and the requirement for additional parameters. In this case, the user model mapping type *split address and get country (A3 schema split)* is used to split and translate the country value of the user’s address from the source user model 3 to the canonical user model. In the resulting mapping container, the parameters “,” and “3” are used to extract the third segment of the address when it is split at each comma. The basic user model mapping type, *equivalent address (A1 schema equivalence)*, is subsequently used to map from the canonical user model to the target user model 4. The creation of both mappings to and from the canonical user model in the Mapping Tool is shown in Figure 5.12.

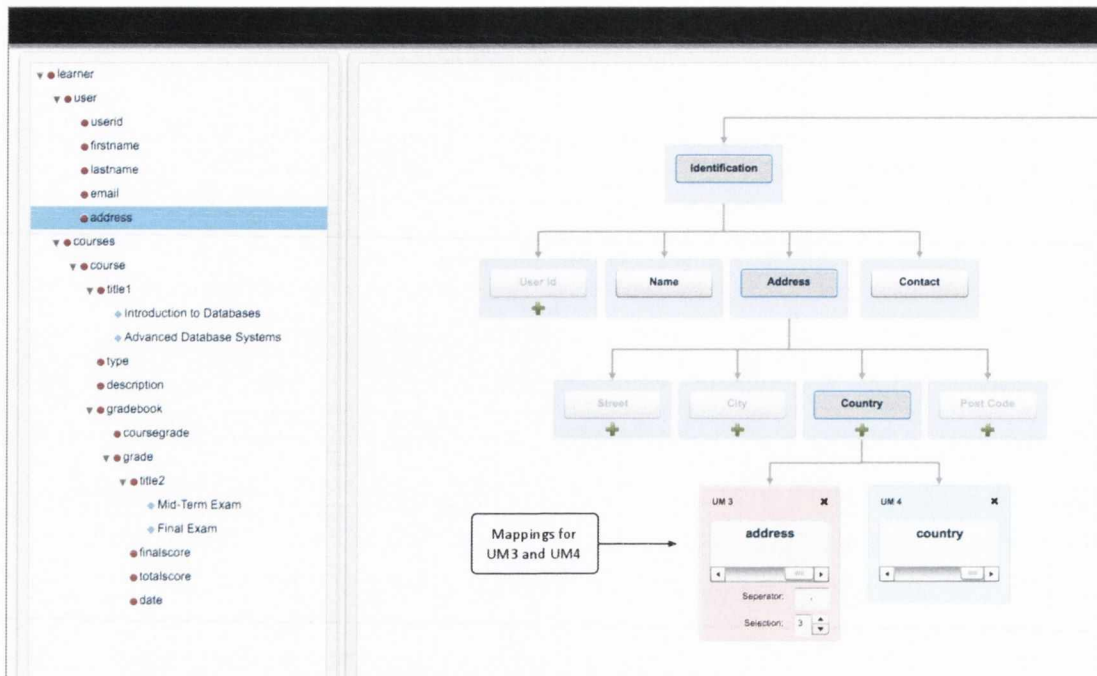


Figure 5.12: Mapping example 2 in the Mapping Tool

Mapping Example 3 - Assessment

Mapping example 3 represents the translation of a course assessment across different grading scales as shown in Figure 5.13.

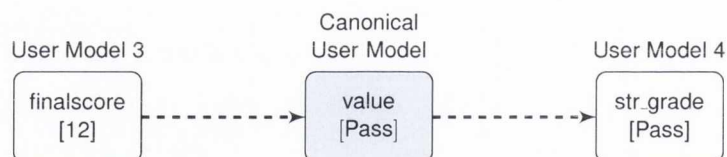


Figure 5.13: Mapping example 3

In this case, the user model mapping type *grades 0-20 to Pass/Fail (C3 functional interval)* is used to create the mapping from user model 3 to the canonical user model. In the resulting mapping container, the parameters “11-20” , “0-10”, “Pass” and “Fail” are used to convert the assessment value from a numeric grading scale to a text-based grading scale. The basic user model mapping type, *equivalent assessment (A1 schema equivalence)*, is subsequently used to map from the canonical user model to the target user model 4. The creation of the mappings to and from the canonical user model in the Mapping Tool is shown in Figure 5.14.

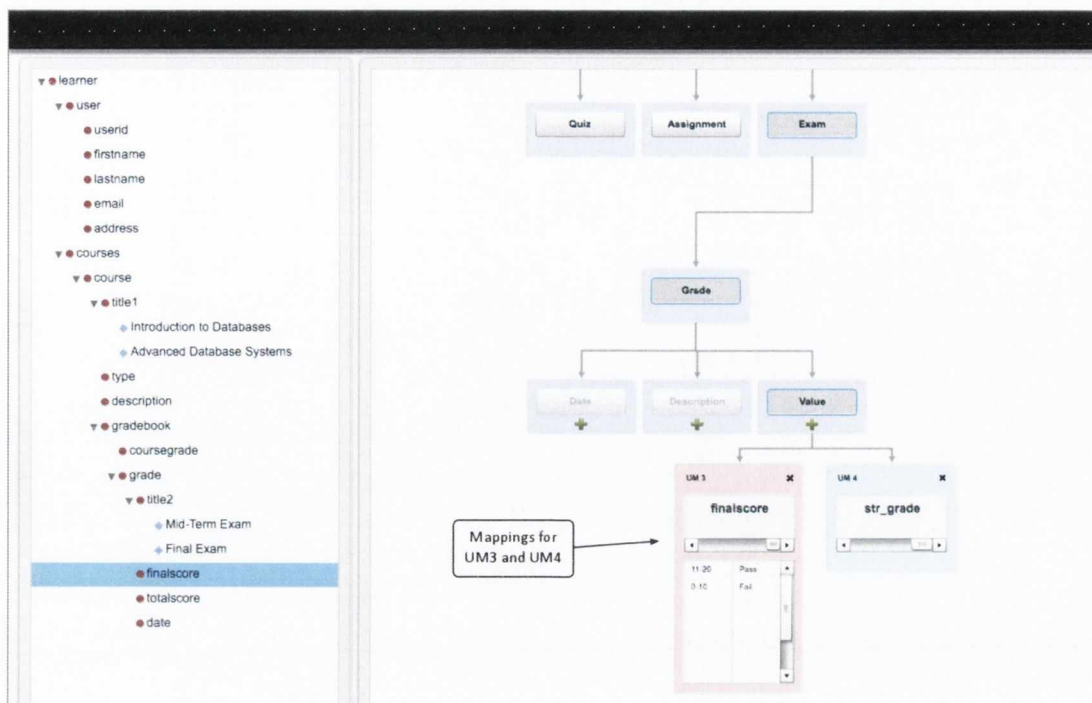


Figure 5.14: Mapping example 3 in the Mapping Tool

Mapping Example 4 - Competency

Mapping example 4 represents the detection of a text-based competency and its translation to a numeric competency value as shown in Figure 5.15.

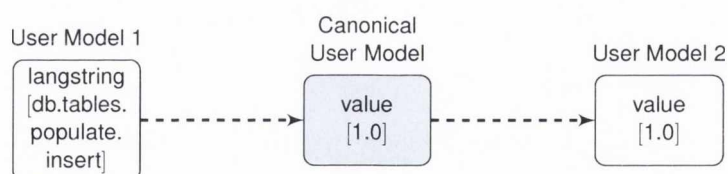


Figure 5.15: Mapping example 4

In this case, the user model mapping type *SQL concept to 0/1 score (C4 functional detection)* is used to create the mapping from user model 1 to the canonical user model. In the resulting mapping container, the parameters “0” and “1” are used to detect and convert the text-based competency to a numeric competency knowledge value. The basic user model mapping type, *equivalent competency (A1 schema equivalence)*, is subsequently used to map from the canonical user model to the target user model 2. The creation of both mappings to and from the canonical user model in the Mapping Tool is shown in Figure 5.16.

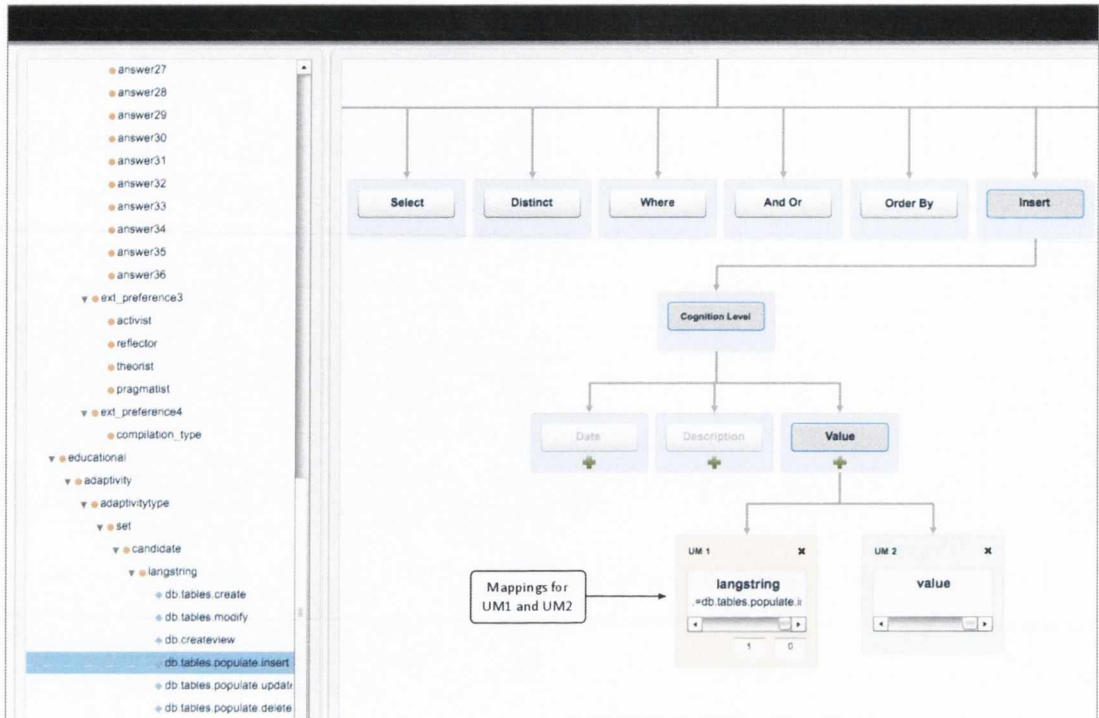


Figure 5.16: Mapping example 4 in the Mapping Tool

Overall, this section has presented a detailed walkthrough of the main functionality of version 2 of the Mapping Tool. The walkthrough demonstrated the ability of the tool to provide users with additional domain knowledge to support the creation of rich relationships between user models. In the following sections, the subsequent evaluation of various key aspects of version 2 of the Mapping Tool is presented.

5.4 Analysis of Accuracy and Duration in User Model Mapping

5.4.1 Objectives and Methodology

Version 2 of the Mapping Tool was evaluated in a user trial which consisted of two experiments. The objective of experiment 1 was to assess the mapping accuracy and duration of the tool. To achieve this, the following hypothesis was proposed:

- The alternative hypothesis (H_1) is that using the domain-aware Mapping Tool increases **accuracy** while maintaining equivalent **duration** times when creating mappings for user model interoperability.

- The null hypothesis (H_0) is that using the domain-aware Mapping Tool does not increase **accuracy** while maintaining equivalent **duration** times when creating mappings for user model interoperability.

The experiment consisted of a comparison between the FUSE Mapping Tool, which contains the domain-aware mechanisms, and the commercial mapping tool Altova Mapforce, in which a typical generic mapping approach was applied. Experiment participants were required to complete four extensive mapping tasks using both the Mapping Tool and Mapforce. The order of the tools was alternated between participants to account for any familiarity with the tasks during the experiment. The evaluation was conducted with 12 participants who were given the same basic training in both the mapping tools. The participants were selected from a pool of computer science postgraduate students, research staff and academics in Trinity College Dublin. This resulted in participants that had the basic skills required to conduct the technically challenging task of creating mappings between user models. A detailed analysis of the prior experience of the participants is described in Section 5.5.2.

Altova Mapforce was chosen to deploy a generic mapping approach in this experiment as it is one of the most advanced mapping tools available and one of the few tools which can create and execute the required complex mappings between user models. Its graphical interface was also independent of any translation language such as XQuery or XSLT and did not require the administrator to have specific skills to start mapping. Mapforce is a large commercial interoperability tool and a small selection of its many features were used in this experiment. Mapforce is a Windows-based desktop application in contrast with the Mapping Tool which is web-based.

User Models and Mapping Tasks

In the experiment, user models and mapping tasks were chosen to represent typical user model interoperability between multiple heterogeneous systems. The user models were selected in Section 3.4.1 and are also shown Table 5.2. The four selected source and target user models are from an adaptive SQL web course based on the APeLS system, a web-based user modelling system called CUMULATE, and two open source learning management systems: Sakai and Moodle.

Table 5.2: Experiment user models

User Model	System
UM0	FUSE Canonical Model
UM1	APeLS
UM2	CUMULATE
UM3	Sakai
UM4	Moodle

The user models contain overlapping information from user, educational and subject domains. In the user domain, information about the user such as identification is represented. In the educational domain, information such as competencies, assessments and courses are represented. The subject domains represent information such as particular topics in databases or SQL. The terms and information in the user models can also overlap across each of these domains. The multiple overlapping domains of user, education, databases and SQL were chosen to represent the type of complex interoperability tasks that are commonly required in user models.

The mapping tasks were created using the mapping types selected in Section 3.5.2 and represent common user model interoperability tasks between user models in the application area of education. The same mapping tasks were also presented in the Mapping Tool walkthrough in Section 5.3. The selected mapping tasks represent the identification, creation and verification of executable mappings between existing systems. Each of these tasks required the participants to perform multiple steps including examining the user models, identifying the appropriate section of the source and target user models, selecting the appropriate mapping type, and constructing and testing the mapping. To analyse the accuracy and duration across different types of mappings, the tasks were selected in terms of increasing complexity and the level of domain knowledge required. The mapping tasks are:

1. Map forename in UM1 to its equivalent in UM4.

This mapping task represents the translation of the first name of the user. The user model mapping type *equivalent first name (A1 schema equivalence)* is used to translate the user's first name from the source user model 1 to the canonical user model. The same user model mapping type, *equivalent first name (A1 schema equivalence)*, is subsequently used to map from the canonical user model to the target user model 4.

2. Map the country from address in UM3 to its equivalent in UM4.

This mapping task represents the translation of the country value of the user's address. The user model mapping type *split address and get country (A3 schema split)* is used to split and translate the country value of the user's address from user model 3 to the canonical user model. The basic user model mapping type, *equivalent address (A1 schema equivalence)*, is used to map from the canonical user model to the target user model 4.

3. Map the finalscores for "Final Exam" in "Advanced Database Systems" in UM3 to its equivalent in UM4. Use the following to convert the finalscores: 11-20 = Pass, 0-10 = Fail.

This mapping task represents the translation of a course assessment across different grading scales. The user model mapping type *grades 0-20 to Pass/Fail (C3 functional interval)* is used to create the mapping from user model 3 to the canonical user model. The basic user model mapping type, *equivalent assessment (A1 schema equivalence)*, is subsequently used to map from the canonical user model to the target user model 4.

4. Map the concept "db.tables.populate.insert" in UM1 to its equivalent numeric knowledge value in UM2. Use the following to convert the concept: concept exists = 0, concept does not exist = 1.

This mapping task represents the detection of a text-based competency and its translation to a numeric competency value. The user model mapping type *SQL concept to 0/1 score (C4 functional detection)* is used to create the mapping from user model 1 to the canonical user model. The basic user model mapping type, *equivalent competency (A1 schema equivalence)*, is used to map from the canonical user model to the target user model 2.

To complete the mapping tasks, the participants used the domain-aware approach in the Mapping Tool represented by the two domain-aware mechanisms; the canonical user model and user model mapping types. The participants were required to navigate and select elements from the canonical user model. The canonical user model represented multiple overlapping domains consisting of identification information from the user domain, assessment and competency information from the education domain, and databases and SQL information from the subject domains. To create the mappings, the participants were required to navigate, select and apply appropriate mapping types. In the Mapping Tool, 25 user model mapping types were provided which represented typical mappings in user models in the areas of identification, assessment and competencies. In Mapforce, the participants used a generic approach and mapped directly between the user models using a selection of 10 relevant generic mapping types to construct the mapping tasks. These

mapping types provided multiple potential combinations in each tool from which the participant had to select the required options and construct the mappings.

Accuracy and Duration Analysis

To analyse the accuracy of the participants’ mappings, a “gold standard” benchmark for each mapping task was created for both mapping tools as shown in Table 5.3.

Table 5.3: Gold standard mapping steps in the Mapping Tool and Altova Mapforce

		Mapping Tool		Altova Mapforce
T1	S1	Select firstname in UM0	S1	Select forename in UM1
	S2	Select mapping type	S2	Map to firstname in UM4
	S3	Map forename in UM1		
	S4	Select mapping type		
	S5	Map firstname in UM4		
T2	S1	Select country in UM0	S1	Select address in UM3
	S2	Select mapping type	S2	Tokenize at comma
	S3	Map address in UM3	S3	If position is 3 then return true else false
	S4	Select mapping type	S4	Filter (or if-else) if true
	S5	Map country in UM4	S5	Map to country in UM4
T3	S1	Select value path in UM0	S1	Select title1 from UM3
	S2	Select mapping type	S2	Select title2 from UM3
	S3	Map finalscores in UM3	S3	Select finalscores from UM3
	S4	Select “Final Exam” in UM3	S4	If title1 equals “Advanced Database Systems” then return true else return false
	S5	Select “Advanced Database Systems” in UM3	S5	If title2 equals “Final Exam” then return true else return false
	S6	Select mapping type	S6	If both true return true else return false
	S7	Map str_grade in UM4	S7	If true filter finalscores
			S8	If equal or greater than 11 return true else false
			S9	If true return Pass, else return Fail
			S10	Map to str_grade in UM4
T4	S1	Select value path in UM0	S1	Select langstring from UM1
	S2	Select mapping type	S2	If langstring equals “db.tables.populate.insert” then return true else return false
	S3	Map “db.tables.populate.insert” in UM1	S3	If true filter langstring
	S4	Select mapping type	S4	If node exists return true else false
	S5	Map value in UM2	S5	If true return 0, else return 1
			S6	Map to value in UM2

The gold standard mapping steps were defined using a practical test-based approach where the mappings were constructed in each tool and extensively tested to confirm that they function correctly. The mappings were then divided into individual steps of the mapping creation process such as selecting from the source and target user models and applying the mapping type. For each step of each task an accuracy score was given; 0 for incomplete or major errors, 0.5 for minor errors or 1 for completed steps. The results of these steps were then used to calculate a percentage accuracy score for each individual task and for the tasks overall. Often, the gold standard steps were the only possible solution for each mapping task. However, in some cases alternative steps were also possible. In the evaluation, participants were given equivalent accuracy scores for alternative steps.

To analyse the duration of the mapping tasks, the average time taken in both the Mapping Tool and Mapforce to perform each of the mapping tasks was recorded. Each task had a time limit to manage the overall duration of the experiment. The first and second tasks were limited to five minutes each (300 seconds) and the third and fourth tasks were limited to ten minutes each (600 seconds).

The experiment results for the accuracy and duration of the mapping tasks are presented in the following sections and discussed in Section 5.6. More detailed evaluation results can also be found in Appendix C.

5.4.2 Mapping Accuracy

In this section, the results of the mapping tasks in both the Mapping Tool and Mapforce are compared for accuracy. As can be seen in Figure 5.17, the Mapping Tool is more accurate across the majority of the mapping tasks. Task 1 had 100% accuracy in both tools. This demonstrates that users can perform basic equivalence mappings in both tools equally well. For task 2, there is a difference in accuracy of 33%, in task 3 it is 30% and in task 4 it is 20%. There was an overall difference in accuracy across all tasks of 29%.

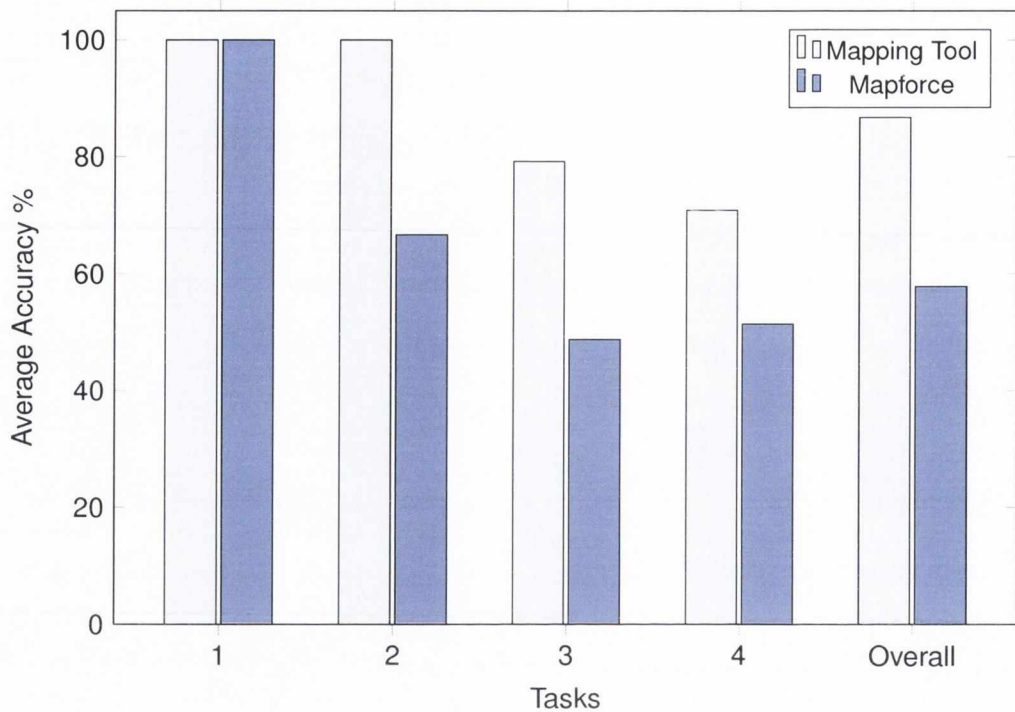


Figure 5.17: Average task accuracy results

Paired t-tests were carried out on the results and showed there is a statistically significant difference in accuracy across most of the tasks. Task 1 is fully completed in both tools so there is no difference in accuracy. In tasks 2 and 3 there is a statistically significant difference in accuracy ($p < 0.001$). In task 4, there is a statistically significant difference in accuracy but with a smaller degree of certainty ($p < 0.01$). There is also a statistically significant difference in overall accuracy ($p < 0.001$).

In the Mapping Tool, the results show that there is 100% accuracy for the first name mapping in task 1 and the address mapping in task 2. For the assessment mapping in task 3 there is a drop of 21% and for the competency mapping in task 4 there is a further drop of 8%. In Mapforce, there is 100% accuracy for the first name mapping in task 1. There is a drop of 33% for the address mapping in task 2. For the assessment mapping in task 3 there is a drop of 18% and for the competency mapping in task 4 there is a drop of 16%.

5.4.3 Mapping Duration

In this section, the duration results of the mapping tasks in both the Mapping Tool and Mapforce are presented. As can be seen in Figure 5.18, the first task is completed faster in Mapforce (47% of the allotted time in the Mapping Tool vs. 26% in Mapforce). Task 2 is completed faster in the Mapping Tool (68% of the allotted time in the Mapping Tool vs. 96% in Mapforce). Paired t-tests for tasks 1 and 2 show that the difference in duration between the tools is significant ($p < 0.01$).

Durations for task 3 (90% of the allotted time in the Mapping Tool vs. 89% in Mapforce) and task 4 (81% of the allotted time in the Mapping Tool vs. 90% in Mapforce) are much closer, indicating that the most complex mappings were difficult to fully complete in both tools. Paired t-tests for tasks 3 and 4 show that the difference in duration between the tools is not significant ($p > 0.05$). The overall time spent on all the mapping tasks in the Mapping Tool is 76% of the maximum allotted time compared to 80% in Mapforce. Paired t-tests show that this is not a statistically significant difference ($p > 0.05$).

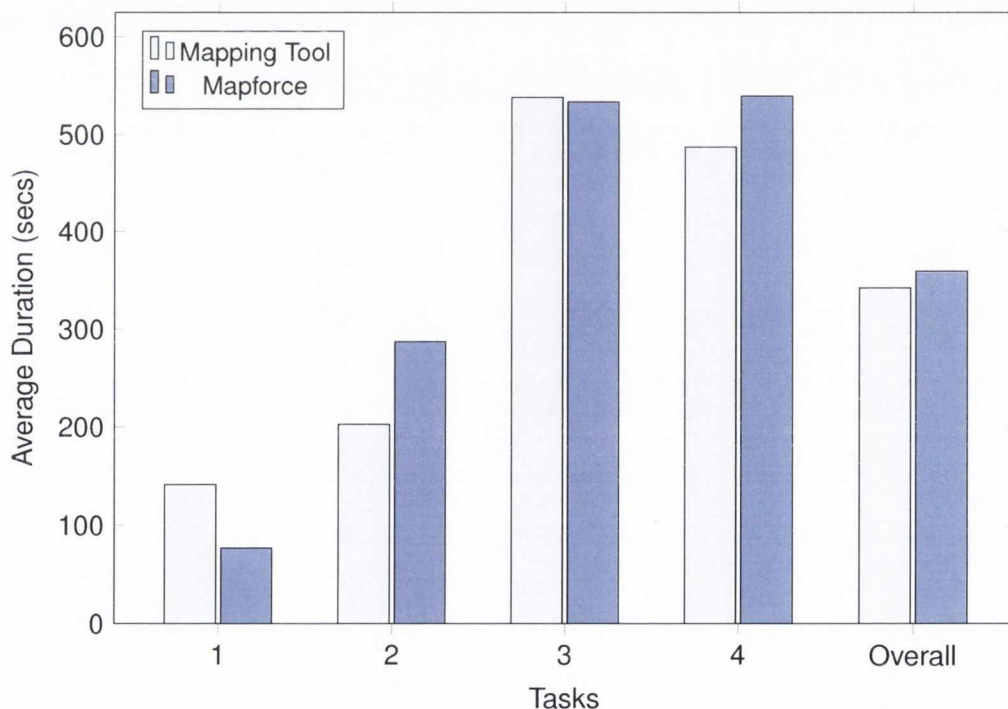


Figure 5.18: Average task duration results

5.5 Analysis of Usability in User Model Mapping

5.5.1 Objectives and Methodology

The objective of experiment 2 was to assess the usability of the FUSE Mapping Tool. To achieve this, the following hypothesis was proposed:

- The alternative hypothesis (H_1) is that the domain-aware Mapping Tool provides high quality **usability** that is better than an existing generic tool in the creation of mappings for user model interoperability.
- The null hypothesis (H_0) is that the domain-aware Mapping Tool does not provide high quality **usability** that is better than an existing generic tool in the creation of mappings for user model interoperability.

To test the defined hypothesis, a number of questionnaires were provided to the participants after using both tools in experiment 1. The results of the questionnaires combined with the results of experiment 1 provide an analysis of the overall effectiveness, efficiency and user satisfaction [International Organization for Standardization (ISO), 1998] of a domain-aware mapping tool in the mapping of user models. This was then compared and contrasted with the results for the generic mapping approach applied in Mapforce.

The results from experiment 1 demonstrated the first two aspects of the usability analysis: effectiveness and efficiency. The improved mapping accuracy in the Mapping Tool when compared to a generic approach in Mapforce demonstrated higher levels of effectiveness when completing the experiment mapping tasks. The approximate equivalence of the mapping duration in both the Mapping Tool and Mapforce demonstrated equivalent efficiency in the experiment mapping tasks.

To assess the user satisfaction, a series of questionnaires were given to the participants at different stages of the experiment. The first questionnaire was related to prior experience and was given to the participants at the start of the experiment. This questionnaire was used to identify the prior experience and relevant skills of the experiment participants. This information could assist in the analysis of the results of the other questionnaires. The second questionnaire was a System Usability Scale (SUS) study [Brooke, 1996] which results in a single numeric usability score for a system. The SUS study was followed by a questionnaire that contained a series of custom

task-based questions. Some questions required responses using a 5-point Likert scale with a section for open comments while other questions required only open comments. Both the SUS and task-based questionnaires were given to the participants immediately upon completion of the mapping tasks in each of the mapping tools. Finally, when the participants had used both tools they were presented with a questionnaire that directly compared key aspects of both tools. The participants were asked to choose their preferred approaches in both tools and also make some comments about their choice. The results of the experiment are presented in the following sections and discussed in Section 5.6. More detailed results can also be found in Appendix C.

5.5.2 Prior Experience Questionnaire

To gain some insight into the prior experience of the participants a questionnaire was given at the start of the experiment and is shown in Table 5.4. The participants were selected from a pool of computer science postgraduate students, research staff and academics in Trinity College Dublin. This resulted in participants that had the basic skills required to conduct the technically challenging task of creating mappings between user models. The participant responses for questions 1 to 7 were recorded using a 5-point Likert scale which consisted of; strongly disagree, disagree, neither agree or disagree, agree, strongly agree. Question 8 allowed the participants to provide open responses.

Table 5.4: Prior experience questionnaire

No.	Question
Q1	I have experience using XML.
Q2	I have experience using XPath.
Q3	I have experience using XQuery.
Q4	I have experience using SQL.
Q5	I have experience using user models.
Q6	I have experience creating mappings between different data models (e.g. databases, XML schemas, ontologies, etc).
Q7	I have experience creating mappings between different user models.
Q8	Please briefly describe any previous mapping tasks you have performed and state the mapping tool used.

The participant responses are summarised in Figure 5.19 and shown in detail in Table C.9.

They show that the participants generally had a high level of experience in XML which is a commonly used data representation language. They also had a high level of experience in the

query language SQL which is a subject domain of the experiment tasks. They had lower levels of experience in the XML-based translation languages XPath and XQuery which are the basis for the mappings in both the FUSE Mapping Tool and Mapforce. The participants also had lower levels of experience with user models. The participants did have some experience with performing mappings between data models although the lowest average score was associated with their experience of mapping user models. The open responses to the final question regarding their previous mapping tasks revealed that some participants had previously performed mapping between web services and others had some experience from conducting other mapping-based experiments. Two of the users also stated that they had some basic experience using Mapforce to map data models.

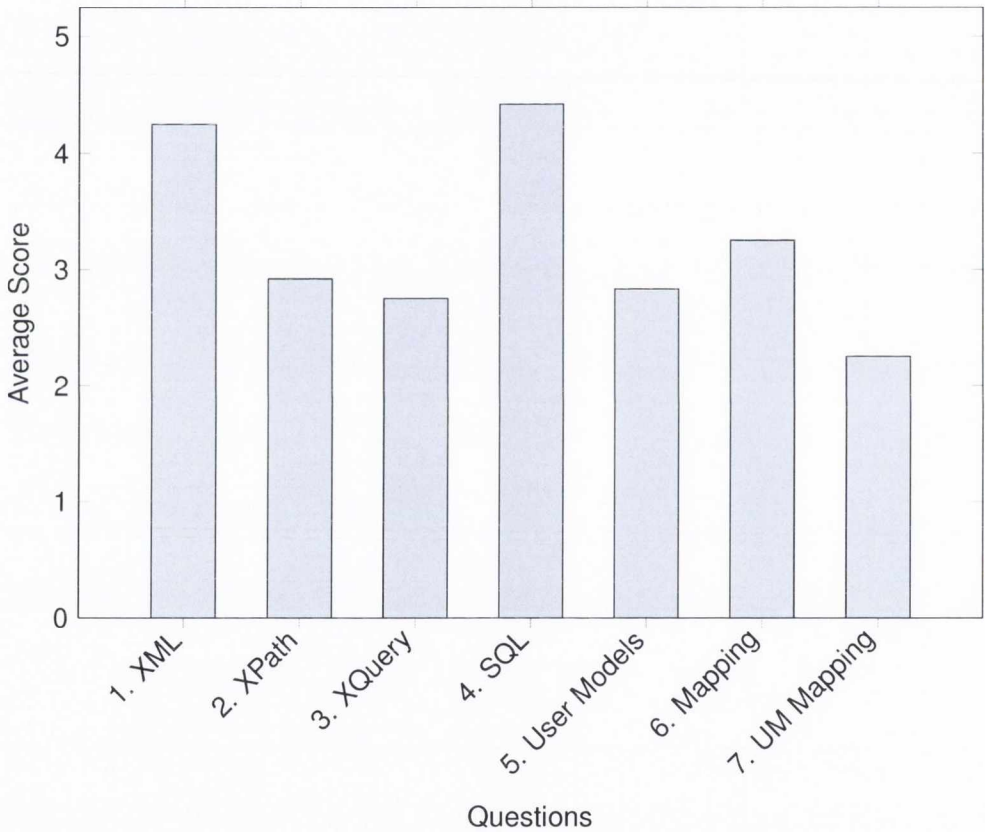


Figure 5.19: Average participant prior knowledge results

Overall, the participants had basic levels of experience in the general technologies and approaches used in the mapping tools. One of the key objectives of this research was to develop an approach that supported users who were not mapping and interoperability specialists but had a basic set of technical skills and who may be required to exchange user information between systems. Both

mapping tools were trialled to determine which provided greater support in the creation of user model mappings.

5.5.3 SUS Questionnaire

SUS is a widely used and reliable questionnaire tool for measuring the usability of a wide variety of products and services. It consists of ten questions that generate a single value that can be used to determine the usability of an individual system and to compare the usability of multiple systems. The questions in SUS are shown in Table 5.5.

Table 5.5: SUS questionnaire

No.	Question
Q1	I think that I would like to use this system frequently.
Q2	I found the system unnecessarily complex.
Q3	I thought the system was easy to use.
Q4	I think that I would need the support of a technical person to be able to use this system.
Q5	I found the various functions in this system were well integrated.
Q6	I thought there was too much inconsistency in this system.
Q7	I would imagine that most people would learn to use this system very quickly.
Q8	I found the system very cumbersome to use.
Q9	I felt very confident using the system.
Q10	I needed to learn a lot of things before I could get going with this system.

To determine the relative value of the SUS result an adjective rating scale can be applied as shown in Figure 5.20 [Bangor et al., 2009]. In this experiment, the Mapping Tool obtained an average SUS score of 72. On the adjective rating scale, 72 indicates a “good” level of usability. To compare the domain-aware approach employed in the Mapping Tool against a generic approach, a SUS score for Mapforce was also calculated. Mapforce obtained an average score of 50 that indicates “ok” usability on the adjective rating scale. The full participant responses to the SUS questionnaire can be seen in Table C.11.

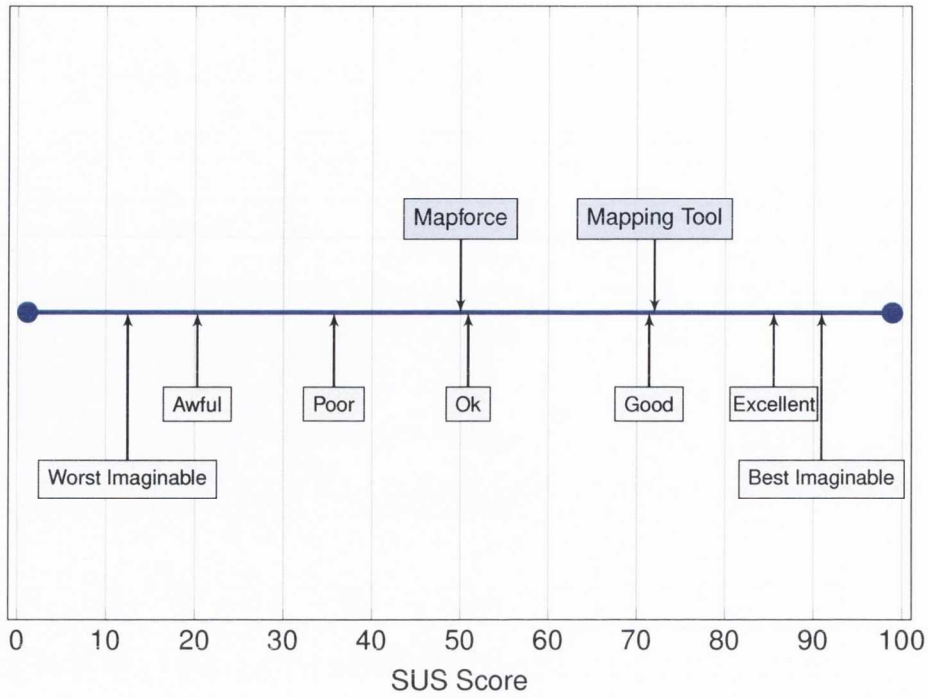


Figure 5.20: SUS results

5.5.4 Task Questionnaire

Following the completion of the tasks in each tool, the participants were also asked ten custom task-based questions as shown in Table 5.6.

Table 5.6: Task questionnaire

No.	Question
Q1	I found it easy to complete the mapping tasks.
Q2	I found the tool helpful in completing the mapping tasks.
Q3	I found it easy to navigate the user models using this tool.
Q4	I found it easy to identify mappings using this tool.
Q5	I found it easy to create mappings using this tool.
Q6	I found it easy to test the mappings using this tool.
Q7	What aspects of the mapping task did you find the most difficult?
Q8	What aspects of the tool did you like?
Q9	What aspects of the tool did you not like?
Q10	Are there any new tool features you think might assist you in the mapping process?

The responses to the first six questions use the 5-point Likert scale and are summarised in Figure 5.21. For question 1, the participants stated that it was easier to complete the tasks in the Mapping

Tool. For question 2, they found the Mapping Tool to be more helpful in completing the mapping tasks. For question 3, they found it easier to navigate the user models in the Mapping Tool. For each of these three questions, paired t-tests identified a statistically significant difference in the participant responses between both tools ($p < 0.01$). For question 4, the participants provided a more positive response to identifying mappings using the Mapping Tool. However, this result was found to be statistically insignificant ($p > 0.05$). For question 5, the participants stated that they found it easier to create mappings using the Mapping Tool. There was a statistically significant difference in the participant responses between both tools ($p < 0.01$). For question 6, they found it easier to test the mappings using the Mapping Tool. This was also found to be a statistically significant difference ($p < 0.05$).

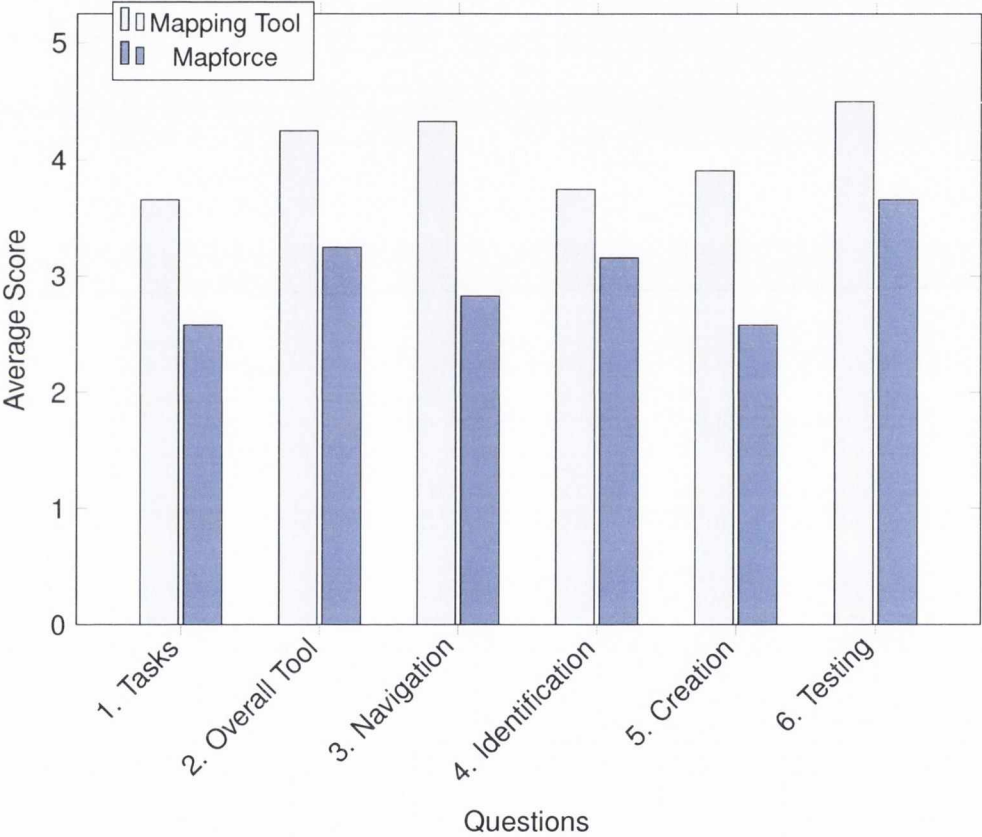


Figure 5.21: Average task-based user satisfaction results

To gain further understanding of how the participants viewed both tools four questions with open responses were also presented to the participants. The full responses are presented in Tables C.15, C.16, C.17 and C.18.

The participants were asked what aspects of the mapping task they found most difficult. In the

Mapping Tool, the two most common responses related to difficulty in the completion of the last two mapping tasks and unfamiliarity with the domain and terminology. In Mapforce, the most common responses were based primarily around the tool and its ability to create the mappings. The participants found the combination of mapping components difficult and the last two mapping tasks were considered much more challenging to complete.

The participants were asked which aspects of the tools they liked. For the Mapping Tool the most common responses were the general ease-of-use of the user interface and the ability to use user model mapping types. For Mapforce, the most common responses were the navigation and drag and drop functionality and the ability to create and link mapping types.

The participants were also asked what aspects of the tools they disliked. In this case, the most common responses for the Mapping Tool related to the layout, buttons and XML viewing in the testing area. For Mapforce, the most common responses were selecting mapping types and creating mappings and the cluttered user interface when there were multiple mappings between user models.

Finally, the participants were asked if there were any new tool features that they thought might assist them in the mapping process. For the Mapping Tool, the most common suggestions were based around greater integration of the mapping creation and testing processes, making it quicker and easier to test newly created mappings. In Mapforce, the most common responses were greater guidance for the user in the form of examples and suggestions for mappings. Also, visualisation changes were suggested to make it easier to navigate the user models.

5.5.5 Comparison Questionnaire

When participants had used both tools they were given a final comparison questionnaire. This questionnaire consisted of two questions which required the participants to compare and comment on their preferred tool and methods. In particular, the questions were focused on determining the participants' preferences regarding the two domain-aware mechanisms, the canonical user model and the user model mapping types. The questions were as follows:

1. Please state which tool's mapping approach you found easier to use and why?

Tool A (i.e. mapping all user models to the permanent user model)

Tool B (i.e. direct mapping between all user models)

2. Please state which tool you found it easier to create mappings in and why?

Tool A (i.e. the tool with mapping to the permanent user model)

Tool B (i.e. the tool with direct mapping between the user models)

The first question asked the participants if they preferred the direct mapping approach applied in Mapforce or the canonical model mapping approach applied in the Mapping Tool. 67% of the participants in this experiment stated that they found it easier to use the canonical model mapping approach applied in the Mapping Tool. The main comments were that the canonical model approach was clearer, more intuitive, more relevant and provided consistency across the tasks. 33% of participants in this experiment stated that they preferred the direct mapping approach applied in Mapforce. The main comments of these participants were that they found it more intuitive to map directly between the user models, rather than through the canonical model.

The second question asked the participants if they preferred using the generic mapping types in Mapforce or the user model mapping types in the Mapping Tool. 100% of participants in this experiment stated that they found it easier to create mappings in the Mapping Tool. The most common comments were that the user model mappings were easier to use, more intuitive and more relevant than the generic mapping types. The full responses provided by the participants are presented in Tables C.20 and C.21.

5.6 Discussion

The results provided by the two experiments have generated a number of findings about the FUSE domain-aware approach to user model mapping and its implementation in the form of the Mapping Tool. These findings have been made in comparison with a generic approach applied in one of the leading commercial mapping tools, Altova Mapforce. In the following sections, the results and hypotheses of each experiment are discussed.

Experiment 1 - Analysis of Accuracy and Duration in User Model Mapping

The first experiment demonstrates that for a set of representative mapping tasks, the domain-aware approach in the Mapping Tool provides better accuracy and achieves similar duration times when compared to the generic approach using Mapforce. This experiment provides indicative results based on a small user population and with selected user models, domains, and mapping tasks in

the application area of education.

Accuracy is the key requirement in mapping tasks and providing tools that make it easier for administrators to map accurately is important to achieve interoperability amongst diverse and distributed user information. In this experiment, the Mapping Tool is shown to be more accurate across the majority of the mapping tasks when compared with the generic approach applied in Mapforce. There is a consistent decrease in accuracy across both tools suggesting that the benefits of the FUSE approach apply across tasks of different complexity. However, the difference in accuracy does decrease as each task is performed. One possible explanation for this could be that the participants were becoming more familiar with the use and combination of the generic mapping types in Mapforce. The experiment results represent the mapping task accuracy within the specified time limits. It is possible that higher levels of accuracy could be achieved in both tools over longer periods of use.

The duration is also an important factor as mapping can be a complex process where poorly performing tools can cause significant delays. The domain-aware approach in the Mapping Tool maintains mapping duration times except in the case of the most basic equivalence mappings. The likely reason for the basic equivalence mapping task being faster in Mapforce was because no complex mapping components were required. The mapping could be created using a simple line connector between the equivalent parts of the user model. The Mapping Tool uses the canonical model mapping approach which requires two mappings to the canonical model for every one direct mapping in the generic approach. In both tools, there was a significant increase in duration from more simple mappings to more complex mappings. The more complex mappings took approximately three times longer than the more basic mappings. This suggests that mapping tools in general should provide more support in the creation of complex mappings.

The mapping accuracy and duration results of this experiment are also significant because the canonical model approach in the Mapping Tool requires larger numbers of mappings initially to complete the same interoperability tasks as Mapforce. The results of the experiment indicate that the initially larger number of mappings does not adversely affect the performance of the Mapping Tool. As the number of interoperable systems increases in the long-term, the canonical approach will require fewer mappings overall than the direct approach. The canonical model approach could potentially be more accurate and faster than the direct approach when providing multi-system user model interoperability. However, as described in Section 3.5.1, the canonical model presents additional issues in maintaining and refactoring those mappings over time. Issues affecting the

long-term management of user model interoperability are outside the scope of the research in this thesis and were not assessed in the evaluation. However, this is an area of interest for future work relating to the Mapping Tool.

The alternative hypothesis (H_1) of the experiment was that using the domain-aware Mapping Tool increases accuracy while maintaining equivalent duration times when creating mappings for user model interoperability. The null hypothesis (H_0) was that using the domain-aware Mapping Tool does not increase accuracy while maintaining equivalent duration times when creating mappings for user model interoperability. The experiment metrics have been evaluated using a comparison of the Mapping Tool with a generic approach applied using Altova Mapforce. Across a number of representative mapping tasks of increasing complexity, the accuracy of the domain-aware approach was shown to be significantly better than the generic approach. The duration of both approaches was measured and, despite differences in individual mapping tasks, the overall duration was similar in both tools. As a result, the null hypothesis (H_0) can be rejected and the alternative hypothesis (H_1) accepted, which states that using the domain-aware Mapping Tool increases accuracy while maintaining equivalent duration times when creating mappings for user model interoperability.

Experiment 2 - Analysis of Usability in User Model Mapping

In the second experiment, the domain-aware approach in the Mapping Tool demonstrates better usability than the generic approach in Mapforce for the mapping tasks. This experiment provides indicative results based on a small user population and with selected user models, domains, and mapping tasks in the application area of education.

Usability can be defined as the effectiveness, efficiency and user satisfaction of a tool when used to complete a specific task. In this case, the mapping accuracy and duration results from experiment 1 represent the effectiveness and efficiency of the Mapping Tool. The user satisfaction was then determined using standardised usability questionnaires such as SUS and custom task-based questionnaires. These questionnaires showed consistently higher levels of user satisfaction across a range of aspects when using the domain-aware Mapping Tool to complete user model mapping tasks. Combined with the higher levels of mapping accuracy and equivalent durations from experiment 1, this indicates the overall usability of the domain-aware approach in the Mapping Tool is better than the generic approach in Mapforce. It is possible that some of

the recorded improvement in usability was as a result of the general layout and appearance of the user interface. However, this is likely to be negligible as the most significant differences in the design of the Mapping Tool when compared to Mapforce result from the use of the domain-aware mechanisms.

While the mapping accuracy and duration analysis was focused on assessing the overall impact of the domain-aware approach on user model mapping, the usability study also gave some insights into the impact of the individual domain-aware mechanisms. The responses from the experiment participants clearly state that the user model mapping types were preferred to the generic mapping types although some users preferred the flexibility of linking different mapping types in Mapforce. The use of the canonical model mapping approach was also preferred but not as conclusively. One third of the participants in the experiment preferred the direct mapping approach used in Mapforce. This suggests that the canonical model approach may not be suitable for all users in all interoperability scenarios. In some cases, such as small scale interoperability with a limited number of mappings, a direct mapping approach may more beneficial. Providing an optional direct mapping approach for these types of interoperability scenarios could be a potential feature for future versions of the Mapping Tool.

The alternative hypothesis (H_1) of the experiment was that the domain-aware Mapping Tool provides high quality usability that is better than a existing generic tool in the creation of mappings for user model interoperability. The null hypothesis (H_0) was that the domain-aware Mapping Tool does not provide high quality usability that is better than an existing generic tool in the creation of mappings for user model interoperability. The experiment metrics have been evaluated using a comparison of the Mapping Tool with a generic approach applied using Altova Mapforce. Across a number of representative mapping tasks of increasing complexity, the usability of the domain-aware Mapping Tool was shown to be significantly better than the generic approach in Mapforce. The usability of the Mapping Tool was also proven to be of a high standard using standardised measures for usability assessment. As a result, the null hypothesis (H_0) can be rejected and the alternative hypothesis (H_1) accepted, which states that the domain-aware Mapping Tool provides high quality usability that is better than an existing generic tool in the creation of mappings for user model interoperability.

Overall, the results of these two experiments demonstrate that incorporating domain knowledge within tools and processes allows administrators to perform better in what is typically a complex and difficult task of mapping user models. These experiment results were established using a

selected group of existing user models and mappings which are representative of the types of interoperability tasks encountered in many different user models.

5.7 Summary

This chapter has presented the second cycle of design, implementation and evaluation of the Mapping Tool for user model interoperability. Following the initial development and evaluation of the Mapping Tool in Chapter 4, a number of modifications were made to the tool to improve usability. Version 2 of the tool provided the best representation of the FUSE user model mapping process defined in Chapter 3. The Mapping Tool encapsulated the domain-aware approach in the form of two mechanisms, the canonical user model and the user model mapping types. Version 2 of the Mapping Tool was described in a walkthrough of the main functionality which used an example from the selected user models to demonstrate the way in which the tool performs the user model mapping process.

Once version 2 of the Mapping Tool was developed, it was evaluated to assess the impact of the FUSE domain-aware approach on user model mapping. This evaluation took the form of two experiments which examined the main aspects of the Mapping Tool. The first experiment examined the Mapping Tool for its ability to support users in efficiently creating accurate mappings between user models. The second experiment examined the Mapping Tool for its overall usability in the creation of user model mappings. In both of these experiments, the results from the Mapping Tool were compared to the results for a commercial mapping tool, Altova Mapforce. In both cases, the Mapping Tool was demonstrated as being more accurate, more usable and equally as efficient as Mapforce when completing user model mapping tasks. Overall, the FUSE domain-aware approach has demonstrated improvement in the user model mapping process when directly compared with a typical generic mapping approach.

Chapter 6

Development and Evaluation of the FUSE Translation Service

6.1 Introduction

The development and evaluation of the first main tool in the FUSE approach, the Mapping Tool, was described in Chapters 4 and 5. In this chapter, the design, implementation and evaluation of the second main tool, the FUSE Translation Service, is presented. The Translation Service supports the exchange of information between heterogeneous user models by executing the mappings created by the Mapping Tool.

To provide the user model translation process, the design of an architecture of interdependent components is described. Following this, the implementation of the Translation Service is presented including the various technologies used and the major stages of development. The first stage developed an initial prototype to support exchange between selected user models. The second stage was concerned with providing more expansive testing and evaluation of the tool. Next, a walkthrough of the Translation Service describes each step in the process of user model translation. Finally, the evaluation of the prototype tool is described which assessed its current level of performance. This is conducted across a number of key aspects at both the schema and instance levels.

6.2 Development of the FUSE Translation Service

The FUSE Translation Service is a tool for the execution of mappings and the translation of user information between heterogeneous user models. It is a web-based tool that encompasses each of the steps in the user model translation process that were described in Section 3.6. It is a novel tool that integrates closely with the Mapping Tool and, as a result, applies both domain-aware mechanisms: the canonical user model and user model mapping types, in the translation of heterogeneous user model information. In the next sections, the architecture and implementation of the Translation Service is presented.

6.2.1 Architecture

This section describes the FUSE architecture, shown in Figure 6.1, from the perspective of the Translation Service and its dependent components such as the database of user models and mappings. The Mapping Tool which allows an administrator to perform the manual user model mapping process is also shown in the figure to represent the complete architecture of the FUSE approach. The Mapping Tool was described in detail in Chapters 4 and 5.

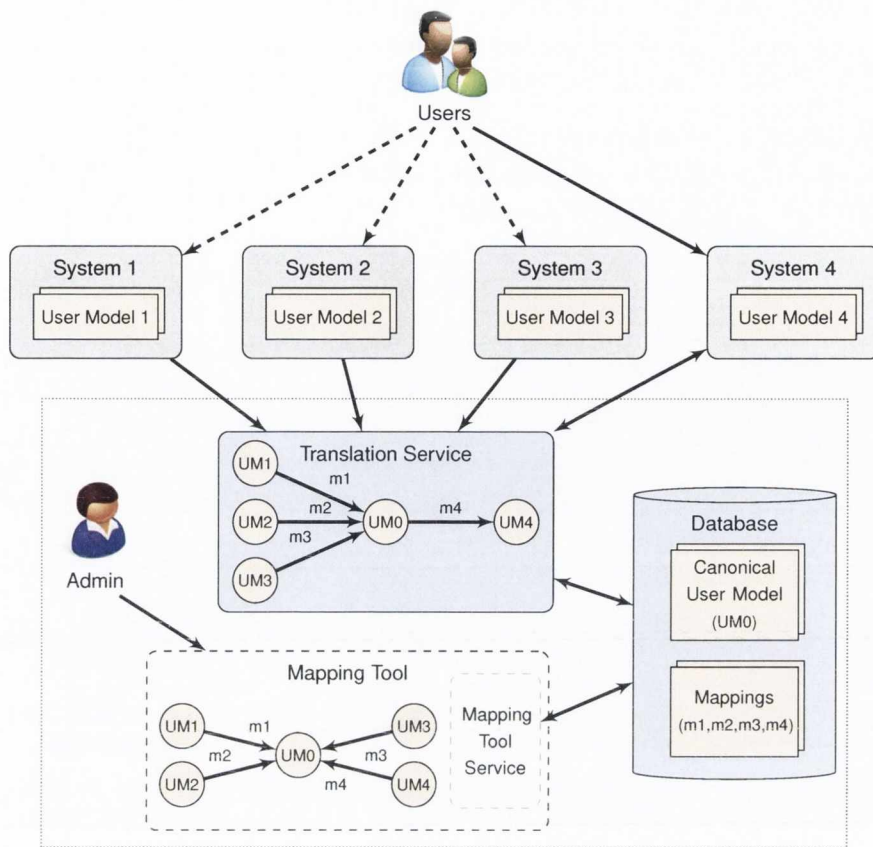


Figure 6.1: FUSE architecture

When the administrator-led user model mapping process has been completed for a number of systems, the user model translation process can be performed to automatically exchange user information between those systems. The central point for this exchange is the Translation Service. This service handles the management of the user models and translates between the various user model representations. The user model instances are not stored locally in the Translation Service and after the translation has been performed they are returned to the individual interoperable systems.

As shown in Figure 6.1, the Translation Service is a tool to automatically translate user model instances between external systems (UM1, UM2, UM3, UM4). The Translation Service is a server-side component and has access to the mappings that were created using the Mapping Tool and stored in the Database (m1, m2, m3, m4). These mappings represent executable relationships between the external user model elements and their equivalent elements in the canonical user model (UM0). The Translation Service retrieves and automatically executes these mappings to transform user information between the heterogeneous user model representations.

The Translation Service supports the domain-aware approach through the provision of the two domain-specific mechanisms: the canonical user model and the user model mapping types. The canonical user model approach adds an additional step to the translation process but ultimately reduces the number of mappings required in multi-system interoperability. The user model mapping types are created to map user model information in specific domains. Their use can reduce the overall number of mapping types required in the translation process.

As shown in Figure 6.1, the Translation Service temporarily stores the various user model instances in the Database. It then executes the mappings which have been stored there by the Mapping Tool. This updates the required target user model instance which can then be returned to the requesting system. The Translation Service communicates with the Database directly as they are both server-side components.

6.2.2 Implementation

The Translation Service was implemented using a combination of technologies. It consists of a RESTful web service and is constructed using Java-based technologies. The Translation Service provides an interface that accepts HTTP requests for user model updates from the various interoperable systems. It also sends HTTP requests to other systems to retrieve their user models.

The Translation Service manages the user models and mappings stored in the Database by the Mapping Tool. EXist [EXist, 2012] was chosen as the basis for this database implementation. EXist is an open source database management system built on XML technology. EXist uses the XML-based query language XQuery [World Wide Web Consortium (W3C), 2012e] to manipulate its data. As the mappings in this approach are also created using XQuery, the processor in the eXist database is the basis for the execution of the user model translation process. The XQuery processor executes the mappings to perform translations between the individual instances of the user models.

To allow the Translation Service to manage the exchange of user models, there are some requirements for the systems that are sharing their user information. The external systems should incorporate web services to expose user model instances for exchange, and update the system with newly received user model instances. If a system does not provide a web service that can export user models then a plugin is required which provides this functionality. Typically, the same web

service or plugin can be reused in any instantiation of the system and a number have already been developed in this research for systems such as Sakai and Moodle.

The development of the Translation Service was closely related to the Mapping Tool and, as a result, evolved during the course of the implementation. In particular, as the user models and mapping types progressed during the development of the Mapping Tool, the Translation Service and the Database also evolved to support these changes. The development of the user models and mapping types is discussed in detail as part of the Mapping Tool implementation in Section 4.2.3. The implementation of the Translation Service had two major stages:

- **Stage 1 of the Translation Service** involved the development of an initial prototype to support exchange between the selected user models defined in Section 3.4.1. The Translation Service was developed to exchange user information between four different systems in the area of database and SQL education; Sakai [Sakai, 2012], Moodle [Moodle, 2012], CUMULATE [Brusilovsky et al., 2005] and APeLS [Conlan and Wade, 2004]. A number of different mapping types were created and different types of user information were shared from Sakai, Moodle and CUMULATE and merged and translated into an APeLS-based SQL web course, which is shown in Figure 6.2. This prototype was used to assess whether the current Translation Service could exchange user model information effectively and efficiently. It also allowed testing of the applicability of the domain-aware mechanisms using existing systems.

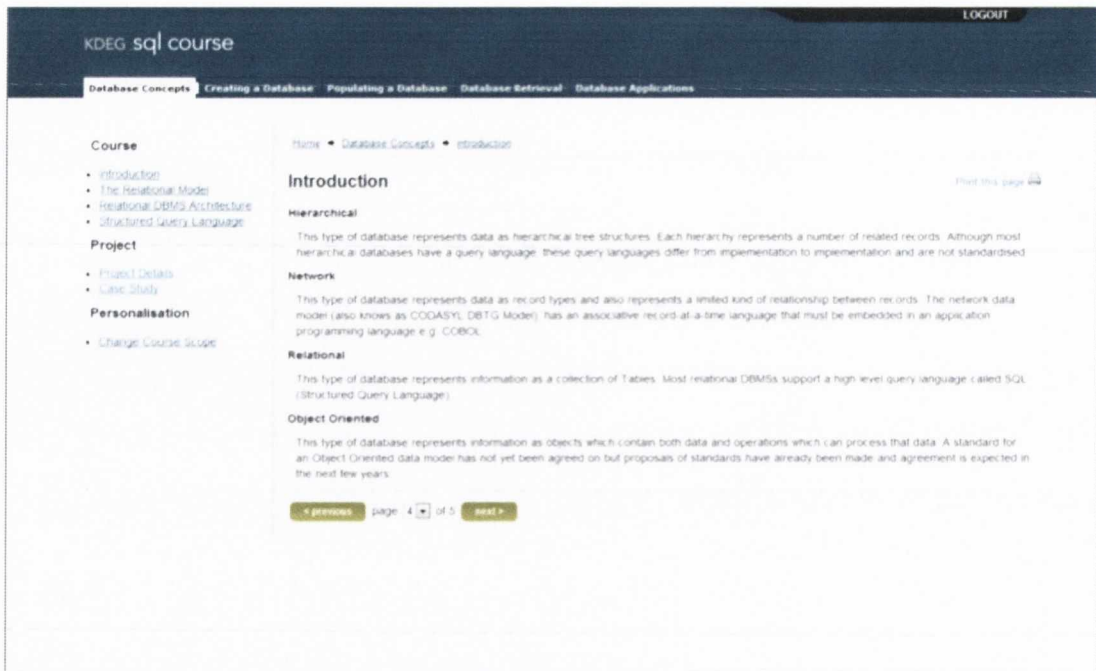


Figure 6.2: APeLS-based SQL web course

- **Stage 2 of the Translation Service** was implemented to allow for more expansive testing and evaluation. To achieve this, a set of services were developed specifically for the performance testing of the Translation Service. The Test Services were developed using HTML and Javascript and provided testing of various key aspects of the user model translation process. A version of the Test Service is shown in Figure 6.3. The Test Service allowed multiple exchange requests to be made to the Translation Service using any type of specified user model. The required parameters of the translation process could also be specified allowing aspects such as the source and target user models to be defined. The use of multiple distributed Test Services was used to replicate multi-system user model interoperability. It is envisaged that the Test Services could potentially be combined with an extended Mapping Tool and used to provide a single comprehensive service for the creation, testing, execution and management of long-term user model interoperability between multiple systems.

Parameters

authentication

ad012fe25e69434e39c3d3

source_ids

///9093ab9582742c24a615db6cd47914

target_id

baac5f9d-b168-4490-9ab1-05a6d3d5a109

map_id

f334e676-033d-4e39-8b89-e2748b7e354f

opt_id

pull

user_id

6a13b087-1d95-4a01-85ce-467a701eeb86

query

Update

Save

Reset

User Model

<learner>

<user>

<userid>jamith</userid>

<firstname>John</firstname>

<lastname>Smith</lastname>

<email>jamith@test.com</email>

<address>1 College Green, Dublin 2, Ireland</address>

</user>

<courses>

<course>

<title>Introduction to Databases</title>

<type>course</type>

<description>Introduction to Databases</description>

<gradebook>

<coursegrade>9</coursegrade>

<grade>

<title2>Mid-Term Exam</title2>

<finalscore>14.0</finalscore>

<totalscore>20.0</totalscore>

<date>29/10/2010</date>

</grade>

<grade>

<title2>Final Exam</title2>

<finalscore>14.0</finalscore>

<totalscore>20.0</totalscore>

<date>29/06/2011</date>

</grade>

</gradebook>

</course>

<course>

<title>Advanced Database Systems</title>

<type>course</type>

<description>Advanced Database Systems</description>

<gradebook>

<coursegrade>C</coursegrade>

Figure 6.3: Test Service for the Translation Service

6.3 Translation Service Walkthrough

The following section presents a detailed walkthrough of the main functionality of the Translation Service and demonstrates its ability to implement the FUSE user model translation process. The walkthrough uses the selected user models from Section 3.4.1 and the selected mapping tasks from Section 3.5.2 to demonstrate the interoperability of user information between multiple existing user models. An example of four combined mapping tasks between these user models is presented. Following this, the steps involved in executing these mappings and translating the relevant user information are described.

6.3.1 User Model Translation Example

The example used in the walkthrough represents a combination of four different mapping tasks as shown in Figure 6.4.

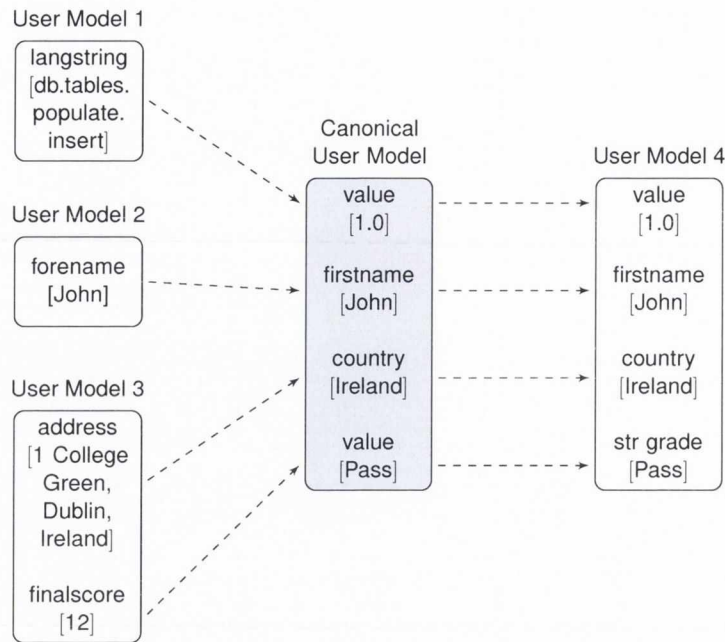


Figure 6.4: Translation example 1

The selected mapping tasks were identified in Section 3.5.2. They are:

1. The translation of the first name of the user. The user model mapping type *equivalent first name (A1 schema equivalence)* is used to translate the user's first name from the source user model 2 to the canonical user model. The same user model mapping type, *equivalent first name (A1 schema equivalence)*, is subsequently used to map from the canonical user model to the target user model 4.
2. The translation of the country value of the user's address. The user model mapping type *split address and get country (A3 schema split)* is used to split and translate the country value of the user's address from the source user model 3 to the canonical user model. The basic user model mapping type, *equivalent address (A1 schema equivalence)*, is used to map from the canonical user model to the target user model 4.
3. The translation of a course assessment across different grading scales. The user model mapping type *grades 0-20 to Pass/Fail (C3 functional interval)* is used to create the mapping from user model 3 to the canonical user model. The basic user model mapping type, *equivalent assessment (A1 schema equivalence)*, is subsequently used to map from the canonical user model to the target user model 4.
4. The detection of a text-based competency and its translation to a numeric competency value.

The user model mapping type *SQL concept to 0/1 score (C4 functional detection)* is used to create the mapping from user model 1 to the canonical user model. The basic user model mapping type, *equivalent competency (AI schema equivalence)*, is used to map from the canonical user model to the target user model 4.

The steps involved in executing the mapping tasks in the translation example and exchanging the user information using the Translation Service are shown in Figure 6.5.

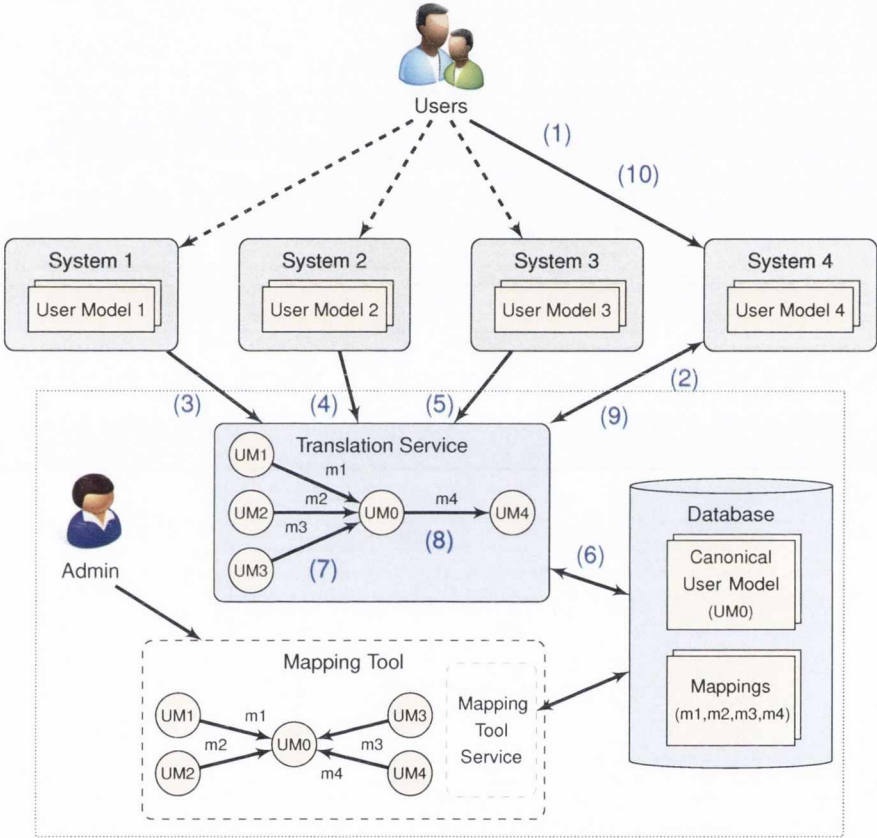


Figure 6.5: User model translation steps in the Translation Service

The steps are:

1. An end-user accesses system 4 to use its features based around SQL education.
2. System 4 sends a HTTP request to the Translation Service providing the end-user's current instance of user model 4.
3. The Translation Service retrieves the end-user's SQL competency "db.tables.populate.insert" from user model 1 in system 1.

4. The Translation Service retrieves the end-user's "forename" from user model 2 in system 2.
5. The Translation Service retrieves the end-user's "address" and their database assessment "finalscore" from user model 3 in system 3.
6. The relevant mappings (which were created using the Mapping Tool) and canonical user model are retrieved from the Database.
7. The relevant mappings (m1, m2, m3) are executed to translate the retrieved instances of user model 1, 2 and 3 into the instance of the canonical user model.
8. The relevant mappings (m4) are executed to translate the instance of the canonical user model into the retrieved instance of user model 4.
9. The updated instance of user model 4 is returned to system 4 as the response of the HTTP request and replaces the local instance for that end-user.
10. The end-user interacts with system 4 which is using the updated version of user model 4 to provide the most relevant user information.

In the following sections, the evaluation of the Translation Service is described. The evaluation consists of an analysis of user model translation performance.

6.4 Analysis of Performance in User Model Translation

6.4.1 Objectives and Methodology

Experiments 1 and 2 presented in Chapter 5 were concerned with the assessment of the FUSE Mapping Tool and its ability to support the manual user model mapping process. The objective of experiment 3 is to assess the FUSE Translation Service for its ability to execute the mappings and perform the automatic user model translation process. To achieve this, the following hypothesis was proposed:

- The alternative hypothesis (H_1) is that the domain-aware Translation Service provides efficient **performance** in the automatic translation of multiple heterogeneous user models.
- The null hypothesis (H_0) is that the domain-aware Translation Service does not provide efficient **performance** in the automatic translation of multiple heterogeneous user models.

To test the defined hypothesis, a number of different aspects of the Translation Service were examined. These aspects relate to user model interoperability at both the schema and instance levels. At the schema level, various aspects were analysed for their impact on the efficiency of the automatic user model translation process in the Translation Service. They are:

- **Number and complexity of mappings between user models** relates to the interoperability of larger and more complex user models which results in the use of increasing numbers of mappings and more complex mapping types.
- **Number of interoperable user models** relates to adding more heterogeneous user models to multi-model exchange scenarios which also results in more user models being exchanged.
- **Number of elements within the user models** relates to the number of elements within the interoperable user models which results in larger models being exchanged.

At the instance-level, one of the most important aspects of a runtime user model interoperability service is the **number of concurrent requests for user model instances** it can process. To be useful in practical settings, the interoperability service must be able to support translation between systems with many users. To test the impact on performance when there are large numbers of concurrent requests for user model updates, the Translation Service was load tested using the Apache HTTP server benchmarking tool, ab, [Apache Software Foundation, 2012] which allowed a specified number of concurrent requests to be sent to the Translation Service.

For the experiment hypothesis, the key value to measure the performance of the Translation Service was the response time when updating a sample user model. To conduct the experiment accurate timings were recorded and compared to appropriate web system performance recommendations to draw conclusions about the overall performance of the Translation Service. The durations were measured in milliseconds and results were calculated using the mean of 100 requests. The experimental setup consisted of the Translation Service and various Test Services deployed as source and target sample systems. As described previously, the Test Services used in the exchange were custom-built systems that could vary different aspects of the requests, for example, the number of source and target models and the size of the user models. The exchange was performed using the RESTful web service interface of the Translation Service.

6.4.2 User Model Schemas

In this section, the results of performance tests which focused on the user model schemas are presented. The results are then discussed in detail in Section 6.5.

Figure 6.6 shows the results for the exchange of user information between one source and one target as the number of concurrently executed mappings between the source user model and canonical user model increases from 0 to 100. Figure 6.6 also shows this increase in the number of mappings using mapping types of increasing complexity. Four different mapping types selected in Section 3.5.2 are applied. They are: Type 1 - *equivalent first name (A1 schema equivalence)*, Type 2 - *split address and get country (A3 schema split)*, Type 3 - *grades 0-20 to Pass/Fail (C3 functional interval)*, and Type 4 - *SQL concept to 0/1 score (C4 functional detection)*.

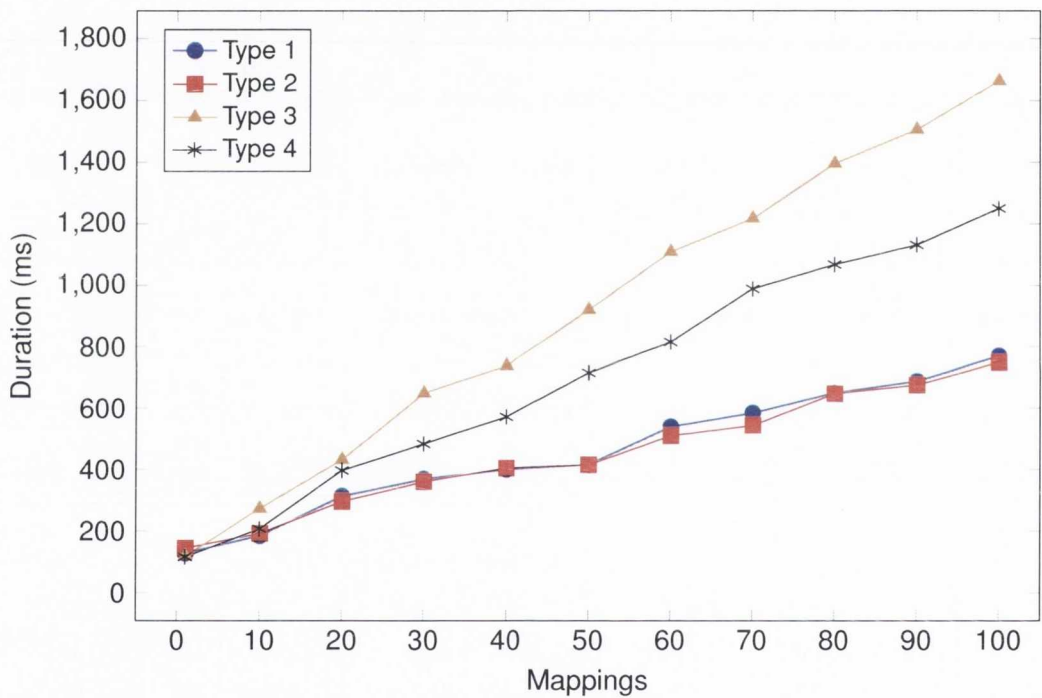


Figure 6.6: Mapping complexity and quantity

Figure 6.7 shows the results for the concurrent exchange of user information between an increasing number of sources and one target. This demonstrates the length of time required to exchange user information between interoperable systems and how the addition of new interoperable systems affects the performance independently of different user models and mapping types. In these translation results, there is one basic mapping type, *equivalent first name (A1 schema equivalence)*, from each source user model to the canonical user model and from the canonical user model

to the target user model. The number of source user models is then increased from 0 to 10 to assess the performance of the translation. Another aspect shown in Figure 6.7 is the time taken to exchange from the source user models to the canonical model. This reduces the number of executed mappings by half and gives an indication of what the performance of a direct mapping approach would be if the same interoperability scenario was applied.

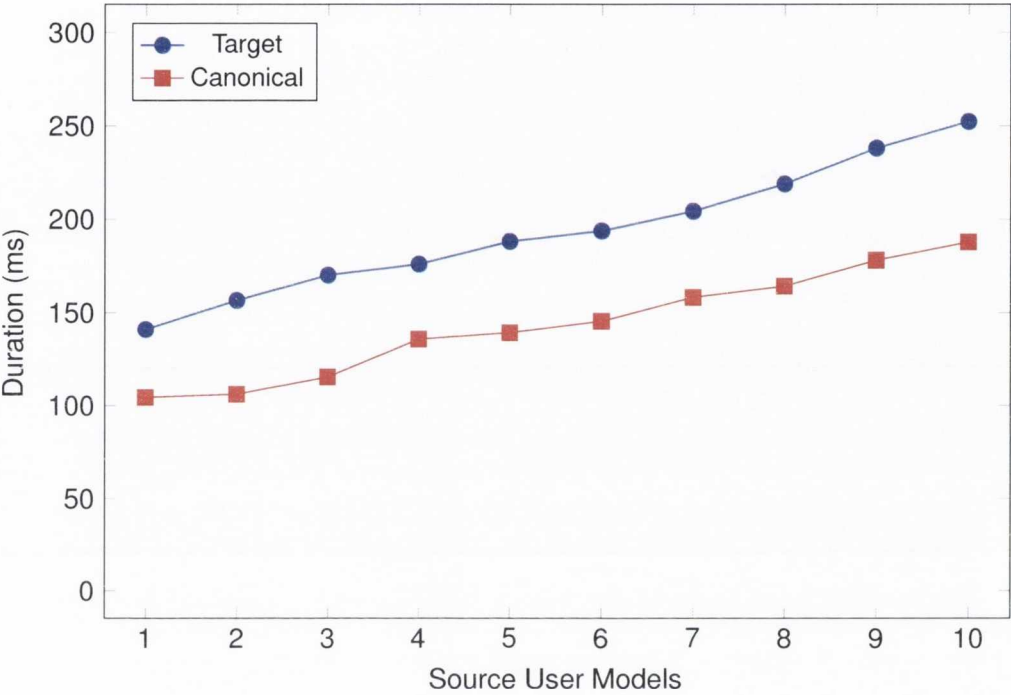


Figure 6.7: User model quantity

Finally, Figure 6.8 shows the results of exchange between one source and one target as the number of elements in the source user model increases. In this case, one basic mapping type, Type 1 - *equivalent first name (AI schema equivalence)*, is created from the source user model to the canonical user model and from the canonical user model to the target user model. The time taken to execute this translation was then measured as the number of additional elements within the source user model was increased from 0 to 10000. The results were recorded in increments of 1000. A large number of elements was required to demonstrate a consistent increase in the response times. These results demonstrate how the increasing size of the source user model affects the retrieval of the source element and the subsequent exchange of user information.

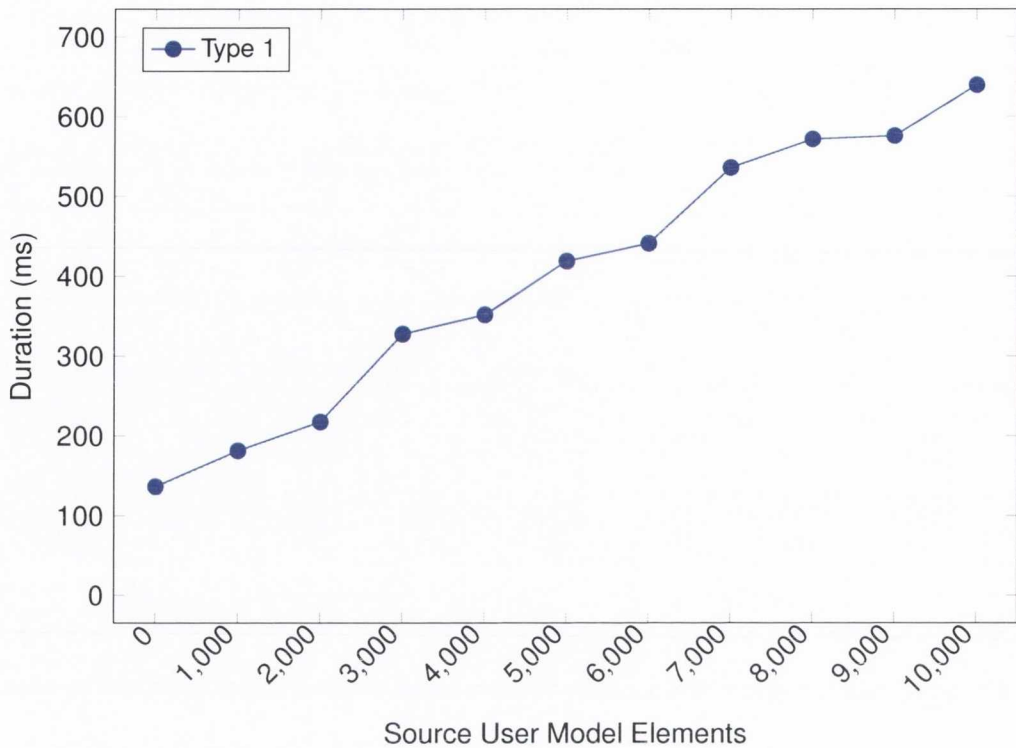


Figure 6.8: User model size

6.4.3 User Model Instances

In this section, the results of a performance test which focused on the user model instances are presented. The results are discussed in detail in Section 6.5. This performance test was completed with five source user models. One basic mapping type, *equivalent first name (AI schema equivalence)*, was created from each source user model to the canonical user model and from the canonical user model to the target user model. Due to the REST-based architecture of the Translation Service and limitations of the ab benchmarking tool, the test was performed using multiple concurrent instances for one user rather than multiple concurrent instances for varying users. In this case, the tool activity is similar and the results are indicative of the type of load experienced with multiple varying users. The results can be seen in Figure 6.9 which shows the response times for 1, 25, 50, 75 and 100 concurrent requests for user model updates. For 1 request the response time is 169.33ms which is consistent with the previous performance tests. For 25 concurrent requests, 100% of the requests are processed in 1152.95ms. For 50 concurrent requests, 100% of the requests are processed in 2682.08ms. For 75 concurrent requests, 100% of the requests are processed in 4515.80ms. For 100 concurrent requests, 100% of the requests are

processed in 7080.20ms. Figure 6.10 shows the same results where the duration is graphed against the number of concurrent requests to demonstrate the increase in execution time.

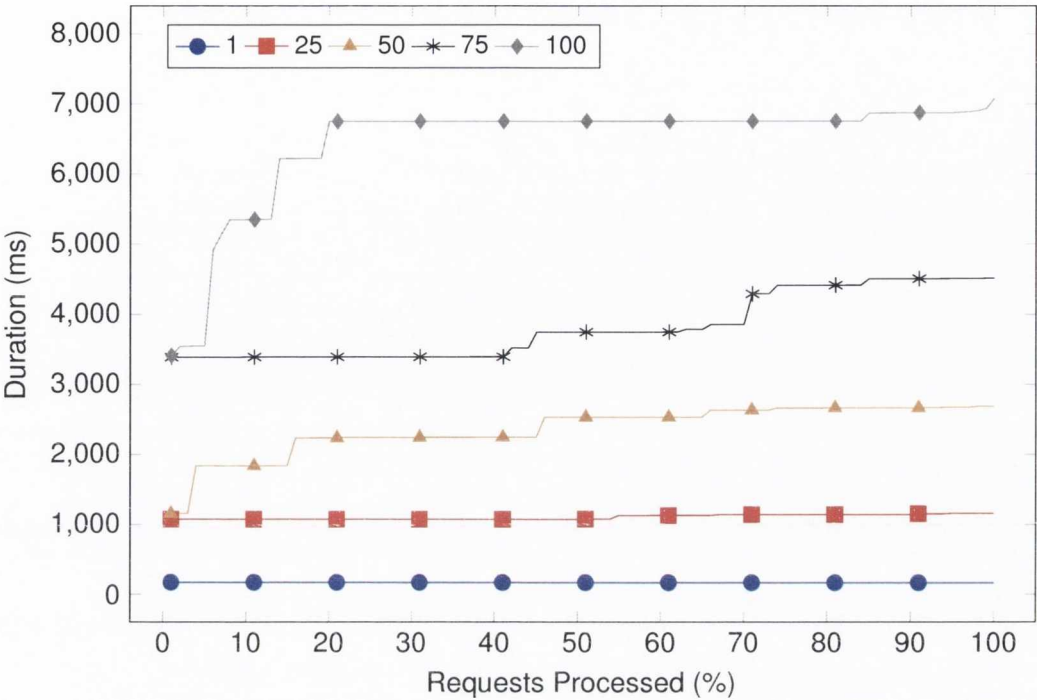


Figure 6.9: User model instance quantity 1

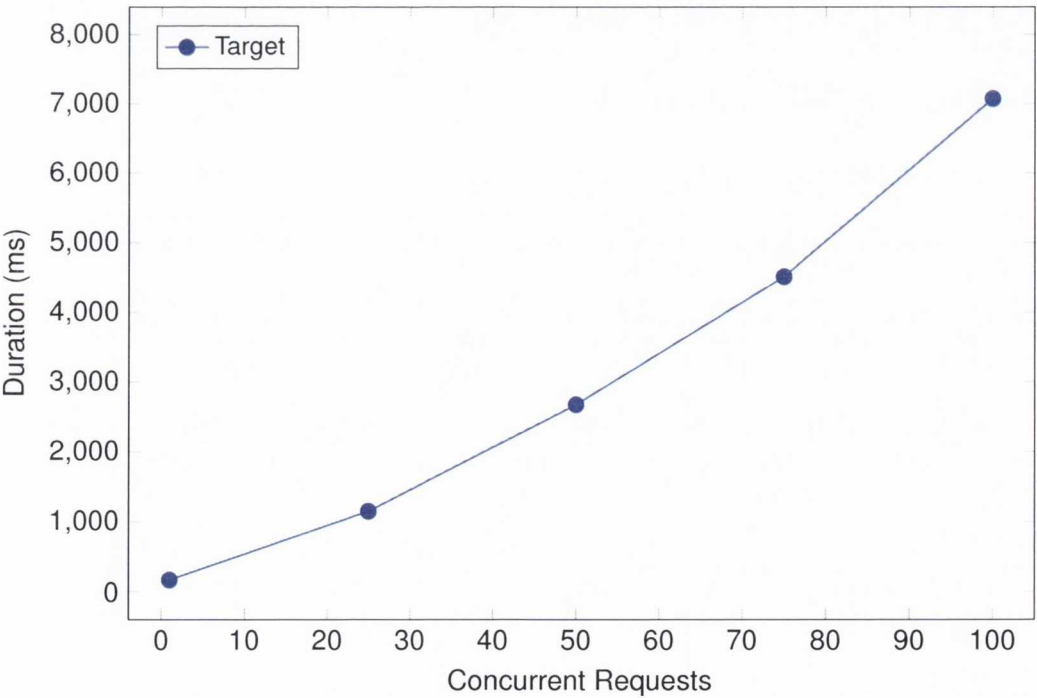


Figure 6.10: User model instance quantity 2

The results presented in the previous sections have provided a detailed description of the performance of the Translation Service at both the schema and instance levels. These results are analysed in more depth in the discussion of the evaluation in the next section.

6.5 Discussion

Experiment 3 was concerned with the analysis of the performance of the Translation Service during the user model translation process. The experiment consisted of recording the response time of the Translation Service as variations were made at both the schema and instance levels of the user models. To analyse the results of this experiment, some common performance guidelines were used. Nielsen [Nielsen, 1993] describes three important limits when discussing response times.

- 0.1 seconds is the limit for having the user feel that the system is reacting instantaneously.
- 1 second is the limit for a user's flow of thought to remain uninterrupted, even though the user will notice the delay.
- 10 seconds is the limit for keeping a user's attention focused on the dialogue.

The results of experiment 3 have generated a number of findings about the performance of the Translation Service. In the following sections, the individual results and the overall hypothesis of the experiment are discussed.

- **Number and complexity of mappings between user models** - The first schema level results, shown in Figure 6.6, indicate that the number and type of mappings has a significant impact on the performance of the translation process. From the perspective of overall response times of the Translation Service, both the first and second mapping types are under the 1 second limit for usability up to 100 mappings. For the third mapping type, up to 50 mappings can be processed in under 1 second and for the fourth mapping type approximately 70 mappings can be processed in under 1 second. The results indicate that increasing numbers of both the *A1 schema equivalence* mapping type and the *A3 schema split* mapping type affect performance in similar ways. Each new mapping of either type adds approximately 6ms to the exchange performance. For the third mapping type, *C3 functional interval*, the increase in performance is approximately 15ms for each new mapping. For the fourth mapping type, *C4 functional detection*, the increase in performance is approximately 11ms for each new mapping. The

results have shown that the first and second mapping types are similar despite the more complex tokenisation action being performed in the second mapping type. The third mapping type has the largest effect on performance despite the higher complexity of the fourth mapping type. The performance effects are likely due to the composition and execution of the XQuery in the eXist database. Further analysis would be required to determine whether there is an individual cause such as the use of a particular XQuery function or whether it is a cumulative effect across multiple aspects of the process. However, the result does suggest that improving the efficiency of the individual mapping representations in XQuery could have an impact on the overall performance of the Translation Service.

- **Number of interoperable user models** - The second schema level results, shown in Figure 6.7, indicate that the number of user models also affects the performance but that it is not as significant a factor as the number of mappings. The response time for the exchange between one source user model and one target user model is 141ms. The addition of each new interoperable source user model from 1 to 10 added approximately 10-15ms to the exchange performance. The response time for the exchange of 10 user models is 253ms, which is under the 1 second usability limit. The rate of increase in performance was also similar when the exchange between multiple source user models and the canonical model was considered independently. There was an approximate difference of between 35-50ms for these results and the results to the target user model. The canonical user model results can be contrasted with the previous results examining mapping complexity and quantity. The previous results indicated that the execution of 10 mappings between a single source user model and the canonical user model took approximately 148ms. In contrast, the result for 10 mappings between 10 separate source user models and the canonical user model is approximately 188ms. Overall, there is a difference of approximately 40ms between the two approaches. This indicates that the number of interoperable user models does not have as significant an impact on performance as the number of mappings between the user models. To analyse this aspect of the Translation Service in more detail, a larger number of interoperable user models would be required. This is a challenge due to the requirement to configure individual Test Services to replicate the functionality of multiple interoperable systems. Addressing this issue could be a potential area of interest for future work.
- **Number of elements within the user models** - The third schema level results, shown in Figure 6.8, indicate that the number of elements in the user models also has a relatively small impact

on the performance. The response time for a very large source user model containing 10000 elements is 640ms which is under the 1 second usability limit. For every 100 new elements in a user model, approximately 5ms is added to the translation performance. These results show that the size of the user models will not have a significant impact on the performance of the translation except in the case of extremely large user models. This could potentially become an issue if dealing with large volumes of event type user and usage data. The size of this unprocessed user data could expand significantly as the user actions are tracked over time and this could potentially impact on the user model translation performance. Filtering and managing the translation of this type of user data is an area of interest for future work.

- **Number of concurrent requests for user model instances** - The instance level results, shown in Figure 6.9 and Figure 6.10, indicate that for up to 25 concurrent requests the Translation Service can maintain a response time of approximately 1 second. This response time would not affect the flow of thought of users of a system which is requesting user model updates. 100% of 50 concurrent requests are performed in under 3 seconds, 100% of 75 concurrent requests are performed in under 5 seconds, and 100% of 100 concurrent requests are performed in under 7 seconds. These results are under the 10 second limit for keeping a user's attention focused within a system. However, the performance results indicate that the current version of the Translation Service may not be suitable in large-scale application scenarios. For example, providing real-time user model updates with large numbers of concurrent users may not be fully possible without some degradation of user experience. One alternative approach would be to request updates of the user models at fewer intervals. For example, update requests could be made only at the start and end of a user's session in the target system. It is also envisaged that significant improvements could be made to these results by improving the efficiency of Translation Service implementation and adding new features such as caching and multiple parallel XQuery processors. More advanced hardware and processing capabilities could also provide a significant improvement in translation performance.

The alternative hypothesis (H_1) of the experiment was that the domain-aware Translation Service provides efficient performance in the automatic translation of multiple heterogeneous user models. The null hypothesis (H_0) was that the domain-aware Translation Service does not provide efficient performance in the automatic translation of multiple heterogeneous user models. The results of the experiment have demonstrated that at the schema level the Translation Service can support the interoperability of large numbers of user models containing many elements. The Translation

Service can also support large numbers of mappings although as the number and complexity of the mappings increases, the performance levels will be affected. The results of the experiment have also demonstrated that the Translation Service can support concurrency of user requests. Increases in the number of requests will also affect the response time of the Translation Service. Overall, although the efficiency of the Translation Service could be improved for large-scale deployments, the prototype version developed for this research can provide efficient user model translation. As a result, the null hypothesis (H_0) can be rejected and the alternative hypothesis (H_1) accepted which states that the domain-aware Translation Service provides efficient performance in the automatic translation of multiple heterogeneous user models.

6.6 Summary

This chapter has presented the design, implementation and evaluation of the Translation Service for user model interoperability. This tool provided the best representation of the FUSE user model translation process defined in Chapter 3. The FUSE process encapsulated a domain-aware approach to user model translation in the form of two domain-aware mechanisms, the canonical user model and the user model mapping types, both of which are primary features of the Translation Service. Firstly, the architecture of the Translation Service was presented and the main aspects of its implementation were described. Next, a walkthrough of the main functionality of the Translation Service was detailed which used an example from the selected user models to demonstrate how the tool is used to perform the FUSE user model translation process.

Once the Translation Service was developed, it was evaluated to assess the impact of the FUSE process on user model translation. This evaluation took the form of an experiment which examined the Translation Service for its ability to provide efficiency in mapping execution and user model translation. The performance of the Translation Service was examined from a number of different aspects at both the schema and instance levels. The results of the experiment have demonstrated that the Translation Service reaches acceptable levels of performance but would need to be developed further to support large-scale user model interoperability.

Chapter 7

Evaluation of the FUSE Domain-Aware Approach

7.1 Introduction

The development and evaluation of the two main components of the FUSE approach, the Mapping Tool and the Translation Service, has been described in detail in the previous chapters. The evaluation presented in Chapter 5 was concerned with the assessment of the FUSE Mapping Tool and its ability to support the manual user model mapping process. The evaluation presented in Chapter 6 was concerned with the assessment of the FUSE Translation Service and its ability to execute the automatic user model translation process.

In this chapter, an evaluation is presented which compares the overall FUSE approach, consisting of both the Translation Service and the Mapping Tool, with the existing state of the art approaches to user model interoperability. The state of the art approaches were reviewed in Chapter 2 and compared using a set of key issues for user model interoperability [Carmagnola et al., 2011]. The FUSE approach is assessed using these key issues and compared with the previously surveyed approaches. This allowed a consistent and systematic evaluation of the overall FUSE approach within the context of the state of the art approaches to user model interoperability.

7.2 Comparison with Surveyed Approaches

This evaluation is concerned with comparing the FUSE approach to the state of the art approaches using the six key issues for user model interoperability presented in Chapter 2. As described in Section 3.3, key issue 7 which refers to privacy and key issue 8 which refers to pre and post interoperability task were considered outside of the scope of the research in this thesis and omitted from the analysis. The methods used in the FUSE approach to address the key issues 1 to 6 are discussed in the following sections in relation to the surveyed approaches:

1. **Interoperability task** describes the intended purpose of the interoperability system and the main function that will be performed. Generally, approaches have either facilitated the exchange of user models or provided a centralised user modelling/adaptation service that stores and maintains user information for individual systems.

The selection of an appropriate method depends mainly on the intended purpose of the interoperability system. The interoperability task of the FUSE approach is to exchange user models. The majority of other surveyed approaches also perform this function. The approaches that provide a user modelling/adaptation service are Personis, CUMULATE and GUMF. These approaches are designed to provide centralised management of user modelling across various systems.

2. **Interoperability architecture** specifies the overall architectural design that is used to exchange the user models. This key issue is closely related to the interoperability task and the main designs are the centralised, decentralised and mixed architectures.

There are advantages and disadvantages associated with each architectural design. The centralised approaches such as Personis, CUMULATE and GUMF, provide common representations and locations which make it easier to manage multiple user models. However, the centralised management inhibits flexibility and wider applicability in the exchange of user models.

Decentralised approaches such as the UUCM and GUMO/UserML provide more flexible interoperability where different combinations of systems can exchange user models directly. However, they rely on the standardisation of the user models to support exchange which inhibits more widespread use. Other decentralised approaches such as Carmagnola and Dimitrova's approach and Cena's approach attempt to provide automatic translation of user information.

However, these approaches rely on the user data in different systems being similar and highly structured to allow matches to be determined semi or fully automatically. These approaches are not easily applicable to existing systems and user models which are likely to contain high levels of data heterogeneity.

The mixed architecture is adopted in the FUSE approach and also by GUC and Berkovsky's approach. Mixed approaches provide more flexibility and applicability to existing user models by providing translation between different representations. A common centralised component reduces the complexity in the management of the user model translation. The common user model and translation approaches are discussed in more detail in the related key issue: the representation of data exchange.

3. **Protocols and languages for communication** describes the methods used to exchange the user models between systems.

HTTP and web services are the most commonly employed methods by the surveyed approaches to exchange the user models. The most common representation format for user models amongst the surveyed approaches is RDF. Some other XML-based languages have been developed which provide specialised user model structure and syntax such as UserML.

In the FUSE approach, RESTful web services and HTTP are the main basis for the exchange of user models. The emergence of common web-based exchange methods across the majority of approaches is a positive aspect for the wider provision of user model interoperability. The RESTful web service method is suitable for the exchange of XML user models and is widely supported in many web-based systems.

For the user model representation, the FUSE approach uses XML. This contrasts with some of the surveyed approaches which use RDF. FUSE was designed to support existing systems with heterogeneous forms of user information. RDF is currently not supported in many systems which were key targets for interoperability. For example in education, learning management systems have extensive support for web services and XML but little or no support for RDF. As a result, an approach based on XML was selected to provide more widespread applicability to existing systems and different types of user information.

4. **Representation of data exchange** describes how the user models are represented during the exchange process from a syntactic and semantic perspective. It describes the two main methods used to represent the exchanged user data; the common user model or translation approach.

Approaches such as Personis, GUMO/UserML, UUCM, and Dolog and Schafer's approach rely on a common representation of the user model and do not provide translations between representations. While this approach promotes standardisation of user models, it does not support existing heterogeneous user models.

Approaches such as CUMULATE, GUMF, GUC, Carmagnola and Dimitrova's approach, Cena's approach, and Berkovsky's approach provide translation between different user model representations. They also typically incorporate a common user model that provides a shared user model representation across systems. To specify the translations, many of the approaches require mappings between user models. These can be defined either by manual mapping or by employing one of the current generic mapping tools available such as COMA++. These tools are often generic and difficult to use for this complex and specialised task [Falconer et al., 2007].

The representation of the data exchange in the FUSE approach is in the form of a canonical user model combined with a translation approach. A mappings-based approach was selected to provide the translation process. These methods provide a flexible approach that can be applied to different existing systems and user model representations. This key issue is also where the FUSE approach provides a significant difference compared to the surveyed approaches.

The FUSE approach provides support in the complex domain-specific tasks inherent in user model interoperability. This is provided through the incorporation of the two domain-aware mechanisms: the canonical user model and the user model mapping types. These mechanisms provide domain knowledge that can support the complexity of providing user model interoperability. The FUSE Mapping Tool and Translation Service have been specifically designed and developed to incorporate these domain-aware mechanisms. This contrasts with many of the surveyed approaches which do not provide custom tools or techniques to support domain-specific user model interoperability tasks.

The component tools of the FUSE domain-aware approach have been evaluated in a number of experiments examining the accuracy, duration and usability of user model mapping and the performance of user model translation. The results of the user model mapping experiments have indicated that the FUSE domain-aware approach improves accuracy and usability while maintaining efficient duration times in mapping tasks compared to a leading commercial mapping tool. The results of the user model translation experiment indicates that the FUSE

approach supports efficient translation performance across a range of key aspects of the process.

5. **Exchanged data type** specifies the types of data that can be exchanged by the interoperability system. Examples of data types include user, usage, environment, domain, inferred, reasoning and social data.

The exchanged data types are largely dependent on the key issue: the representation of data exchange. As the surveyed approaches use various methods for the representation of data exchange they vary in the data types that they support. For example, UUCM and GUMO/UserML attempt to provide standardised representations of the user models. These approaches restrict the supported data types to those which have been explicitly defined in the standardised model. These data types are then difficult to modify to support different data types. Translation approaches with common models, such as CUMULATE and Berkovsky's approach generally support a wider range of data types. The information is translated and a shared user model representation is used which can be expanded or modified as needed to support different data types. Mapping-based translation approaches, such as GUMF and GUC have the potential to support the majority of the data types because new mappings can be created to support new data forms. These types of approaches can be restricted by the associated common user model representations.

In the case of the FUSE approach, a common canonical model and mapping-based translation method was selected. As a result, this provides flexibility in the support of the majority of different types of user data. The canonical model can also be extended as needed to support different forms of data. In the evaluation, the FUSE approach was primarily applied using forms of user and domain-specific data in the application area of education.

6. **Integration of exchanged data** describes the incorporation of the exchanged data into the systems. In general, there are two high level approaches; the exchanged user data is not merged into the systems' existing user data but is retrieved as needed and the exchanged user data is merged into existing user data.

In the majority of the related approaches there is no merging provided into existing user data. Many of the approaches do not consider this as being relevant to the core exchange process. They view it as the responsibility of the interoperable system to incorporate the exchanged data as needed [Carmagnola et al., 2011]. Two approaches which do support the integration of exchanged data are Dolog and Schafer's approach and Carmagnola and Dimitrova's approach

which attempt to resolve conflicts and merge exchanged user data.

The FUSE approach provides merging of exchanged user data into the interoperable systems. This functionality is a result of the mapping-based approach and selected technologies. The mappings are manually specified, thereby allowing an administrator to identify and resolve potential conflicts in exchanged user data. The administrator can also simulate the potential results of mappings by testing with sample use model instances. The selection of XML and XQuery for user model and mapping representations also allows update mechanisms to be used which merge exchanged user data into existing user models. Providing integration of exchanged data is a key benefit when providing interoperability between existing heterogeneous systems.

To summarise the analysis, Table 2.1 has been updated to include the FUSE approach. Table 7.1 shows the updated comparison of user model interoperability approaches for the key issues 1 to 6.

Table 7.1: Comparison of FUSE with surveyed approaches

Approach	Interoperability task	Interoperability architecture	Protocols and languages for communication	Representation of data exchange	Exchanged data type	Integration of exchanged data
FUSE [Walsh et al., 2012] [Walsh et al., 2011a]	Exchange user models	Mixed	REST XML	Common user model and translation	User, usage, environment, domain, inferred, social	Merging of data
Personis [Kay et al., 2002] [Assad et al., 2007]	User modelling service	Centralised	HTTP JSON	Common user model	User, environment, inferred	No merge of data
CUMULATE [Brusilovsky et al., 2005] [Yudelton et al., 2007]	User modelling service	Centralised	HTTP XML	Common user model and translation	User, usage, domain, inferred	No merge of data
GUMF [Abel et al., 2009]	User modelling service	Centralised	REST/SOAP RDF	Common user model and translation	User, usage, environment, domain, inferred	No merge of data
UUCM [Mehta et al., 2005]	Exchange user models	Decentralised	HTTP CSCP	Common user model	User, usage, environment, social	No merge of data
GUMO/UserML [Heckmann and Krueger, 2003] [Heckmann et al., 2005]	Exchange user models	Decentralised	HTTP UserML, OWL	Common user model	User, usage, environment, inferred	No merge of data

Approach	Interoperability task	Interoperability architecture	Protocols and languages for communication	Representation of data exchange	Exchanged data type	Integration of exchanged data
Carmagnola and Dimitrova [Carmagnola and Dimitrova, 2008]	Exchange user models	Decentralised	HTTP RDF, RDFS	Translation	User, usage, domain, inferred, reasoning	Merging of data
Cena [Cena, 2011]	Exchange user models	Decentralised	SOAP RDF, RDFS	Common user model and translation	User, domain	No merge of data
Dolog and Schafer [Dolog and Schafer, 2005]	Exchange user models	Centralised or Mixed	SOAP RDF, RDFS	Common user model	User, usage, environment, domain, inferred	Merging of data
GUC [Van Der Sluijs and Houben, 2006]	Exchange user models	Mixed	HTTP OWL	Common user model and translation	User, usage, environment, inferred	No merge of data
Berkovsky [Berkovsky et al., 2008]	Exchange user models	Mixed	n/a	Common user model and translation	User, usage, environment, inferred, social	No merge of data

7.3 Discussion

This evaluation was concerned with the comparison of the overall FUSE approach, consisting of both the Translation Service and Mapping Tool, with the current state of the art approaches for user model interoperability. The state of the art approaches use many varied methods for each key issue and these were examined in detail in Chapter 2. In the FUSE approach, methods were selected to address the key issues and ultimately provide a comprehensive approach to user model interoperability.

As shown in the comparison, the FUSE approach provides a high standard method for each key issue addressed. The rationale for the selected methods was based on the findings of the extensive state of the art review. FUSE provides a mapping-based approach which uses a canonical user model as a common representation. This mixed architecture method combines the flexibility and complex interoperability provided by the mappings with the structure and consistency of the shared user model. The selection of a mapping-based approach allows the mapping of the majority of user data types. The selection of REST web services and XML as the communication methods provides interoperability between many existing systems. The selection of XQuery and XML as the mapping and user model representation languages also supports the merging of data into existing XML-based user models. This results in a comprehensive overall approach which compared to the surveyed approaches provides high standard methods across the key issues 1 to 6 of user model interoperability.

The FUSE approach provides its most significant contributions in key issue 4: the representation of data exchange. In this case, FUSE provides a domain-aware approach consisting of custom tools for both user model mapping and translation. Both tools support the two domain-aware mechanisms: the canonical user model and user model mapping types. These mechanisms provide support for the complex domain-specific nature of user model interoperability tasks. Previous experiment results have indicated that this approach can improve accuracy and usability while maintaining duration times in specific user model mapping tasks. This comparison was performed against a state of the art commercial mapping tool, Altova Mapforce. Mapforce is representative of the type of tool used to provide mappings in existing approaches to user model interoperability. Previous experiment results have also indicated that the FUSE approach can provide efficient performance in user model translation.

Overall, FUSE provides a comprehensive approach across the selected key issues for user model

interoperability when compared to existing state of art approaches. It also provides indications of significant improvements in one of the key issues: the representation of data exchange.

7.4 Summary

This chapter has presented a detailed and systematic comparison of the overall FUSE approach against the current state of the art approaches to user model interoperability. Through the analysis of the key issues and how they have been applied in the state of the art approaches, the FUSE approach was categorised and assessed using the same criteria. The results have shown that the FUSE domain-aware approach provides comprehensive user model interoperability in the key issues 1 to 6 through the combination of the Mapping Tool and Translation Service. The previous experiments have also indicated that the FUSE domain-aware approach and tools provide improvements in accuracy and usability when compared with an existing approach and tool while also maintaining efficient durations and performance.

Chapter 8

Conclusions

8.1 Research Question and Objectives

This thesis has presented the FUSE domain-aware approach to user model interoperability. The central question of this thesis was:

How does the domain-aware approach developed in FUSE affect the accuracy, duration and usability of user model mapping, the performance of user model translation, and the overall provision of user model interoperability?

Three objectives were derived to address the research question:

1. **Conduct a review of the state of the art in user model interoperability to determine current key issues and methods.** This objective was exploratory and was addressed in a number of key review stages.

Descriptions of common terminology, concepts and methods were detailed initially. The lack of widespread adoption of standards in many areas has led to the development of mapping approaches to provide interoperability between different data models. A review of common mapping tools demonstrated similarities in user interface design, with the generic tools mainly adopting tree-based visualisations of data models with line connections between mapped elements. The field of user modelling was also reviewed. This thesis was primarily focused on the interoperability of user models although extending to other user representations such as event-based user data is an objective for future work. Eight key issues for user model interoperability as defined by Carmagnola [Carmagnola et al., 2011] were then presented and

explored. The key issues cover a range of topics and demonstrate the broad and complex nature of user model interoperability.

A systematic review and comparison of the most influential user model interoperability approaches was conducted using the identified key issues. The field of user model interoperability is diverse and incorporates many different centralised, decentralised or mixed architectural approaches. The emergence of common communication and representation methods is a positive aspect in the field, as many approaches now adopt web services and XML-based representations. The approaches based on automatic mediation require more structured user model representations and generally cannot provide complex exchange of existing user data. The manual mediation approaches rely on existing generic tools that do not provide support for the domain-specific complexity of user model interoperability tasks.

A review of evaluation metrics from the surveyed approaches and additional metrics identified from the key issues was also conducted. It found there were many systems and frameworks proposed but there was limited evaluation conducted in the approaches. The evaluation performed was usually related to the overall performance of the system or the improvement in the interoperable user models after an exchange has been performed. A number of other metrics were identified based on the key issues that could be used to assess more specific aspects of approaches.

2. **Design and develop the FUSE domain-aware approach to user model interoperability and implement it in the form of mapping and translation tools.** This objective was addressed in a number of key design and implementation stages.

A number of key issues, methods and metrics were selected based on the findings of the state of the art review. As a result, a mapping-based approach was adopted to provide flexibility and complex interoperability. This was combined with a canonical model approach that provides a common user model representation for mapping. This mixed architectural approach provided a comprehensive combination of functionality for the provision of interoperability between existing heterogeneous user models.

The domain-aware approach was developed to address the lack of support for domain-specific user model interoperability tasks. This approach provides knowledge of multiple overlapping domains in the interoperability processes and tools. To develop the approach in-depth, the main concepts were refined in the specific application area of education. Although the approach

was designed to be applicable to any application area, education was selected as there is a rich collection of existing educational services which contain representative user models across multiple domains. A selection of user models and domains were chosen to allow the development of a functional approach that could be applied to existing interoperability issues.

The FUSE approach was developed to apply the selected methods and implement domain-aware user model interoperability. FUSE consisted of two domain-aware mechanisms: a canonical user model and user model mapping types. The canonical model has a fundamental impact on the application of the approach and was selected for the number of advantages it provides such as consistency and management of mappings. It also allows consistent domain knowledge to be used in the user model interoperability. However, it provides a number of disadvantages such as being a single point of failure, where modifications can impact many user models. The user model mapping types provide mappings that are created specifically to map within particular domains. They also provide domain knowledge and support to an administrator in the selection and application of complex mapping types. The FUSE approach and resulting mechanisms provide domain-aware user model interoperability without restricting the tool to particular domains. The mechanisms are designed to be extendable and applicable in other domains.

The FUSE approach was applied in two detailed processes; manual user model mapping and automatic user model translation. The processes were then developed in the form of two tools; a Mapping Tool for user model mapping and a Translation Service for user model translation. The Mapping Tool is a web-based tool that contrasts with existing mapping tools by incorporating the two domain-aware mechanisms which support the administrator through the provision of domain knowledge. The canonical model is used to provide a richer graphical visualisation in the user interface and support the identification and creation of user model mappings. The user model mapping types support the user in creating complex mapping for user models. The Translation Service also applies the two domain-aware mechanisms which fundamentally affect the user model translation process. In particular, the canonical model adds an additional translation stage to the process and requires more mappings initially than a direct mapping approach. However, in the long-term it will require fewer mappings as the number of interoperable user models increases.

3. **Evaluate the tools to determine the effect on accuracy, duration and usability of user model mapping, the performance of user model translation, and the overall provision of**

user model interoperability. This objective was addressed in a number of key evaluation stages.

To evaluate accuracy, duration and usability in user model mapping, the Mapping Tool was applied in two user trials. In the trials, participants used the Mapping Tool to complete a number of representative interoperability tasks. The first trial was exploratory and provided user feedback that was used to improve the tool. In the second trial, the participants completed the tasks in a leading commercial mapping tool, Altova Mapforce. This allowed for a comparison of the domain-aware Mapping Tool against a generic and direct mapping approach applied in Mapforce. In experiment 1, mapping duration was measured and mapping accuracy was calculated using a comparison with “gold standard” benchmark mappings. The results show a statistically significant improvement in accuracy in the Mapping Tool compared to Mapforce for the experiment tasks. The durations are also shown as equivalent despite the additional mapping stage required by the canonical user model approach. In experiment 2, the usability of the tools was assessed. Questionnaires were provided to participants to measure user satisfaction. When combined with the accuracy and duration results from experiment 1, they allowed an analysis of the overall usability of the domain-aware Mapping Tool compared to the generic and direct approach in Mapforce. The Mapping Tool had significantly higher levels of user satisfaction across a range of key aspects. Feedback from the participants generally stated that the two domain-aware mechanisms were preferred to the generic approach. However, one-third of participants preferred the direct mapping approach to the canonical model mapping approach. This suggests that the canonical model may not be suitable for all interoperability tasks or for all users.

To evaluate performance in user model translation, the Translation Service was applied in experiment 3. The Translation Service performance was measured from various aspects relating to user models and mappings at the schema and instance levels. The results were compared with common performance guidelines for web-based systems. At the schema level, it was demonstrated that the number and complexity of mappings has a significant impact on the performance of the translation. In this case, more syntactically efficient XQuery mappings could provide notable improvements in performance. The impact of the number of interoperable user models was also assessed. Results of this performance assessment indicated that the number of user models also affects the performance but it is not as significant a factor as the number of mappings. Larger numbers of interoperable user models would be required to

analyse this aspect in more detail and this is a potential area for future work. The third schema level results demonstrated that the number of elements in the user models has a relatively small impact on the performance. At the instance level, it was demonstrated that the Translation Service could support multiple concurrent requests for user model translation. However, when providing multi-system interoperability with large numbers of users using the current version of the Translation Service, dynamic real-time translation may not be fully possible without some degradation of user experience. It is envisaged that significant improvements could be made by enhancing the efficiency of the Translation Service implementation and adding new features such as caching and multiple parallel XQuery processors.

Finally, the overall FUSE approach for user model interoperability was evaluated using a comparison with existing approaches. The FUSE approach, consisting of both the Mapping Tool and Translation Service, was systematically compared to the state of the art approaches reviewed in Chapter 2. This comparison was performed using the identified key issues for user model interoperability. The FUSE approach provides a comprehensive form of user model interoperability when compared to the other surveyed approaches. It provides a mapping-based approach which uses a canonical user model as a common representation. This mixed architecture method combines the flexibility and complex interoperability provided by the mappings with the structure and consistency of the shared user model. FUSE also provides custom tools for both user model mapping and translation. Both tools support the two domain-aware mechanisms: the canonical user model and user model mapping types. These mechanisms provide support for the complex domain-specific nature of user model interoperability tasks.

Overall, each of the three objectives was achieved, ultimately addressing different aspects of the stated research question. In the next section, the key contributions of this thesis are presented.

8.2 Contribution

The main contribution of this thesis is that **for user model interoperability tasks in the application area of education, the FUSE domain-aware approach and tools demonstrate improved accuracy and usability for user model mapping in multiple overlapping domains while also maintaining mapping duration times.** This contribution of the research was

supported by the development of the Mapping Tool for administrators to manually resolve heterogeneity between user models. This tool contrasts with the state of the art in graphical mapping tools by incorporating two domain-aware mechanisms; a canonical user model and user model mapping types, in a web-based tool. These mechanisms provide knowledge of multiple overlapping domains and allow administrators to create increasingly specialised user model mappings to a consistent and detailed canonical user model. Experiment results indicate that the accuracy and usability of the Mapping Tool was superior when compared with a mapping tool using currently established approaches. The duration of the mapping tasks was similar in both tools despite the additional requirements of the domain-aware approach. These indicative evaluations were conducted with multiple small user populations, in the application area of education, and with selected user models, domains, and mapping tasks.

The first minor contribution of this thesis is that **the FUSE domain-aware approach and tools provide efficient performance across multiple key aspects of user model translation**. This contribution of the research was supported by the development of the Translation Service which provides automatic exchange of user models between multiple heterogeneous web systems. The two domain-aware mechanisms; the canonical user model and user model mapping types, are also applied in this tool to support the execution of specialised user model mappings to a consistent, shared canonical user model. Experiment results indicate efficient translation performance across multiple key aspects relating to user models and mappings at the schema and instance levels.

The second minor contribution of this thesis is that **the overall FUSE domain-aware approach and tools provide a comprehensive form of user model interoperability when compared to existing state of the art approaches**. This contribution of the research was supported by a systematic functional analysis of the overall FUSE approach, consisting of the Mapping Tool and the Translation Service. Results indicate that the FUSE approach provides a comprehensive form of user model interoperability when compared with a selection of existing state of the art approaches.

Evidence of the contributions to the state of the art comes in the form of the following publications:

1. **Walsh E.**, O'Connor A., and Wade V., "Evaluation of a Domain-Aware Approach to User Model Interoperability", 23rd ACM Conference on Hypertext and Social Media, June 2012, Milwaukee, Wisconsin, USA. [Walsh et al., 2012]

2. **Walsh E.**, O'Connor A., and Wade V., "Supporting Learner Model Exchange In Educational Web Systems" 7th International Conference on Web Information Systems and Technologies, May 2011, Noordwijkerhout, The Netherlands. [Walsh et al., 2011a]
3. **Walsh E.**, and Wade V., "Lowering the Barriers to User Model Interoperability" Ubiquitous User Modeling Workshop at the Conference on User Modeling, Adaptation, Personalization, June 2009, Trento, Italy. [Walsh and Wade, 2009]
4. **Walsh E.**, Dagger D., and Wade V., "Supporting "Personalisation for All" through Federated User Model Exchange Services (FUMES)" Towards User Modelling and Adaptive Systems for All Workshop at User Modelling 07, June 2007, Corfu, Greece. [Walsh et al., 2007]
5. **Walsh E.**, Rafter R., Conlan O., and Wade V., "Dynamic Contextual Usage Metadata for Learning Resource Reuse in Adaptive Environments" 12th IEEE International Conference on Information Reuse and Integration, August 2011, Las Vegas, USA. [Walsh et al., 2011b]
6. O'Keeffe I., Staikopoulos A., Rafter R., **Walsh E.**, Yousuf B., Conlan O., and Wade V., "Personalized Activity Based eLearning", 12th International Conference on Knowledge Management and Knowledge Technologies (iKnow), September 2012, Graz, Austria. [O'Keeffe et al., 2012]
7. Staikopoulos A., O'Keeffe I., Rafter R., **Walsh E.**, Yousuf B., Conlan O., and Wade V., "AMASE: A framework for composing adaptive and Personalised Learning Activities on the Web", The 11th International Conference on Web-based Learning, September 2012, Sinaia, Romania. [Staikopoulos et al., 2012]
8. Dagger D., O'Connor A., Lawless S., **Walsh E.**, and Wade V., "Service Oriented eLearning Platforms: From Monolithic Systems to Flexible Services" IEEE Internet Computing Special Issue on Distance Learning , 2007. [Dagger et al., 2007a]
9. Dagger D., Rogers C., Wade V., Gaffney C., Armstrong K., Fitzmaurice B., Gill M., and **Walsh E.**, "Adaptive Simulations for Communication Skills Training in Healthcare" Personalisation for E-Health Workshop at User Modelling 07, June 2007, Corfu, Greece. [Dagger et al., 2007b]

Publications 1-4 present key aspects and findings of the FUSE domain-aware approach presented in this thesis. Publications 5-9 present related collaborative research that was conducted. It is intended that more evaluation results and other collaborative research will be published in appropriate venues in the future.

8.3 Future Work

There are four main areas of the FUSE domain-aware approach and tools that are of interest for future work. They are:

1. **Automatic matching** in the creation of mappings is an area that could be addressed in future versions of the FUSE approach. There is significant related research in this area and many algorithms and tools have been developed that provide automatic matching of data models. To provide this functionality in the Mapping Tool, one of these matching tools, S-Match [Giunchiglia et al., 2004], has been selected. S-Match is currently one of the most popular and highly cited matching tools available and is the most appropriate in terms of its technical representation of matches. This tool will be used to provide automatic matches that the administrator can then verify as mappings. It could also potentially be used to provide suggestions for mappings to the administrator. Matching tools such as S-Match are unlikely to automatically identify the complex mappings that are possible in the Mapping Tool. However, there are often many basic mappings required that could be identified and mapped automatically and, as a result, reduce the overall workload of the administrator.
2. **Semantic web technologies** are the second area that could be addressed in future versions of the FUSE approach. Progress is being made in the movement towards a semantic web and this has led to the development of new standards such as RDF and OWL for representing ontological concepts and their explicit semantics. There is significant related research that examines the issues inherent in the semantic interoperability of ontologies. Similarly to the FUSE approach, much of this work is focused on the use of mappings. Mappings have been represented using many formats such as SWRL and new representation formats have been developed such as EDOAL. In the FUSE tools, supporting these technologies and the different approaches they require will be a key challenge for future development. However, the existing tools were designed to be adaptable and extensible to support this type of potential evolution. For example, XML is used as the data model representation in the FUSE tools and both RDF and OWL are XML-based languages. Additionally, XQuery could continue to be applied as the mapping representation language to map RDF and OWL models or an extension to another related semantic web language could be applied.
3. **Additional application areas** is a third area of interest for future work in the FUSE approach. In-depth development and evaluation of the FUSE approach has been conducted using the

representative application area of education. However, the approach can also be applied to support other application areas and domains. It would require the identification of appropriate systems, user models and domains to develop practical interoperability tasks. It would not require any extensions to the FUSE tools but new domain knowledge would have to be provided in the form of an extended canonical model and user model mapping types. An expected future application area for the FUSE approach would be social data in particular domains from online sources such as Facebook. Another potential extension would be to support other forms of user data not within the scope of this thesis. In particular, analysing event-based user and usage data could potentially provide very rich forms of user models. Exchanging this data is becoming increasingly important in order to gain knowledge of the users actions and personalise systems accordingly. This type of data can expand rapidly and this brings additional challenges in the provision of interoperability. Finally, the FUSE approach also has applicability beyond user models as an interoperability system. It can be applied to provide interoperability in other forms, for example, mapping content metadata between systems. This will also bring different types of data models and new interoperability challenges. As a result, new mapping types and canonical models will be required to support these approaches but potentially FUSE could be expanded into a domain-aware approach for interoperability in many fields.

4. **Long-term management of user model interoperability** is a fourth area of interest for future work. The research in this thesis is fundamentally focused on the core technical, syntactic and semantic challenges in overcoming user model heterogeneity. However, there are a number of additional challenges in the long-term management and evolution of user model interoperability. Managing the expansion and refactoring of the user models and mappings over time can result in scalability and synchronisation issues. The canonical model approach used in FUSE can provide a number of advantages and disadvantages within this area, which could be a source of future work. To achieve this type of analysis, would require complex studies with multiple participants and user models conducted over extended periods of time. Another aspect of long-term management in user model interoperability is the provision of reuse and collaboration. One of the key benefits of a web-based approach is the ability to provide significantly more sharing and reuse of interoperability components between administrators. Multiple administrators may also need to collaborate to bring together the required expertise and experience for complex interoperability projects. In the FUSE tools, initial support for reuse has been provided in the user model mapping types that can be created, saved and reused

by the administrators. Future development of the tools will potentially focus on the provision of project-based interoperability with teams collaboratively creating mappings and canonical models. These various components can then be saved for further use and shared with team members or potentially to a wider public group of administrators to generate a large repository of mappings and canonical models.

Overall, these are four main areas of potential future work in the FUSE approach. The FUSE approach is suitable for continued evolution and could potentially be used to form the basis of new research questions and objectives.

8.4 Final Remarks

Exchanging user information between multiple sources can potentially bring many benefits that enhance the user experience in software applications. Richer and more dynamic user models can be constructed allowing more tailoring of content and services to the specific needs of each individual user. However, providing effective interoperability is a complex challenge as there can be significant heterogeneity between user models. As user model interoperability provides more expansive and detailed user models, it results in user models that contain information from multiple overlapping domains. Current generic approaches and tools offer insufficient support for the complex domain-specific nature of many user model interoperability tasks.

This thesis has presented the FUSE domain-aware approach to user model interoperability. This novel approach combines mapping and translation processes with two domain-aware mechanisms: a canonical user model and user model mapping types. These mechanisms provide domain knowledge to support complex user model interoperability tasks. The evaluation of the FUSE approach has demonstrated that it improves the accuracy of user model mapping while maintaining efficient duration times. It has also been shown to perform user model translation efficiently and to provide a comprehensive overall interoperability approach when compared to the state of the art approaches. Overall, FUSE is not a complete solution to the complex problem of providing user model interoperability. However, it does demonstrate that more specialised processes and tools can deliver improvements and are required to support the more widespread provision of user model interoperability.

Bibliography

- [Abel et al., 2009] Abel, F., Heckmann, D., Herder, E., and Hidders, J. (2009). A framework for flexible user profile mashups. In *Workshop on Adaptation and Personalization for Web 2.0, International Conference on User Modeling, Adaptation and Personalization - UMAP 2009*.
- [Adobe, 2012] Adobe (2012). Adobe Flex. <http://www.adobe.com/ie/products/flex.html>.
- [Altova, 2012] Altova (2012). Altova Mapforce. <http://www.altova.com/mapforce.html>.
- [Apache Software Foundation, 2012] Apache Software Foundation (2012). ab - Apache HTTP server benchmarking tool. <http://httpd.apache.org/docs/2.2/programs/ab.html>.
- [Aroyo et al., 2006] Aroyo, L., Dolog, P., Naeve, A., Nilsson, M., and Wild, F. (2006). Interoperability in Personalized Adaptive Learning. *Educational Technology & Society*, 9(2):4–18.
- [Assad et al., 2007] Assad, M., Carmichael, D., Kay, J., and Kummerfeld, B. (2007). PersonisAD: Distributed, active, scrutable model framework for context-aware services. In *Pervasive Computing: Proceedings of the 5th International Conference - PERVASIVE 2007*, volume 4480 of *Lecture Notes in Computer Science*, pages 55–72. Springer.
- [Asuncion and Van Sinderen, 2010] Asuncion, C. H. and Van Sinderen, M. J. (2010). Pragmatic Interoperability: A Systematic Review of Published Definitions. In *Enterprise Architecture, Integration and Interoperability*, volume 326 of *IFIP Advances in Information and Communication Technology*, pages 164–175. Springer.
- [Aumuellner et al., 2005] Aumuellner, D., Do, H.-H., Massmann, S., and Rahm, E. (2005). Schema and ontology matching with COMA++. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data - SIGMOD '05*. ACM.
- [Babbie, 2010] Babbie, E. (2010). *The Practice of Social Research*. Wadsworth.

- [Bangor et al., 2009] Bangor, A., Staff, T., Kortum, P., and Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4(3):114–123.
- [Berkovsky et al., 2007] Berkovsky, S., Eytani, Y., Kuflik, T., and Ricci, F. (2007). Enhancing privacy and preserving accuracy of a distributed collaborative filtering. In *Proceedings of the 2007 ACM conference on Recommender systems - RecSys '07*. ACM.
- [Berkovsky et al., 2009] Berkovsky, S., Heckmann, D., and Kuflik, T. (2009). Addressing Challenges of Ubiquitous User Modeling: Between Mediation and Semantic Integration. In *Advances in Ubiquitous User Modelling*, volume 5830 of *Lecture Notes in Computer Science*, pages 1–19. Springer.
- [Berkovsky et al., 2008] Berkovsky, S., Kuflik, T., and Ricci, F. (2008). Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction*, 18(3):245–286.
- [Bernstein et al., 2006] Bernstein, P. A., Melnik, S., and Churchill, J. E. (2006). Incremental schema matching. In *Proceedings of the 32nd International Conference on Very Large Databases - VLDB 2006*, volume 06 of *VLDB '06*, pages 1167–1170. VLDB Endowment.
- [Broekstra and Kampman, 2003] Broekstra, J. and Kampman, A. (2003). SeRQL: A Second Generation RDF Query Language. In *SWAD-Europe Workshop on Semantic Web Storage and Retrieval*.
- [Brooke, 1996] Brooke, J. (1996). SUS-A quick and dirty usability scale. In *Usability Evaluation In Industry*. Taylor & Francis.
- [Brooks et al., 2004] Brooks, C., Winter, M., Greer, J., McCalla, G., and Mike, W. (2004). The Massive User Modelling System (MUMS). In *7th International Conference on Intelligent Tutoring Systems - ITS 2004*, volume 3220 of *Lecture Notes in Computer Science*, pages 73–112. Springer.
- [Brusilovsky and Millán, 2007] Brusilovsky, P. and Millán, E. (2007). User Models for Adaptive Hypermedia and Adaptive Educational Systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*, pages 3–53. Springer.

- [Brusilovsky et al., 2005] Brusilovsky, P., Sosnovsky, S., and Shcherbinina, O. (2005). User Modeling in a Distributed E-Learning Architecture. In *Proceedings of the 10th International Conference on User Modeling - UM 2005*, volume 3538 of *Lecture Notes in Computer Science*. Springer.
- [Carmagnola and Cena, 2009] Carmagnola, F. and Cena, F. (2009). User identification for cross-system personalisation. *Information Sciences*, 179(1-2):16–32.
- [Carmagnola et al., 2011] Carmagnola, F., Cena, F., and Gena, C. (2011). User model interoperability: a survey. *User Modeling and User-Adapted Interaction*, 21(3):285–331.
- [Carmagnola and Dimitrova, 2008] Carmagnola, F. and Dimitrova, V. (2008). An Evidence-Based Approach to Handle Semantic Heterogeneity in Interoperable Distributed User Models. In *Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems - AH 2008*, volume 5149 of *Lecture Notes in Computer Science*, pages 73–82. Springer.
- [Cena, 2011] Cena, F. (2011). Integrating web service and semantic dialogue model for user models interoperability on the web. *Journal of Intelligent Information Systems*, 36(2):131–166.
- [Cohen et al., 2007] Cohen, L., Manion, L., and Morrison, K. (2007). *Research Methods in Education*. Routledge.
- [Conlan and Wade, 2004] Conlan, O. and Wade, V. (2004). Evaluation of APeLS - An Adaptive eLearning Service Based on the Multi-Model, Metadata-Driven Approach. In *Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems - AH2004*, volume 3137 of *Lecture Notes in Computer Science*, pages 504–518. Springer.
- [Dagger et al., 2007a] Dagger, D., O'Connor, A., Lawless, S., Walsh, E., and Wade, V. (2007a). Service-oriented e-learning platforms - From monolithic systems to flexible services. *IEEE Internet Computing*, 11(3):28–35.
- [Dagger et al., 2007b] Dagger, D., Rogers, C., Wade, V., Gaffney, C., Armstrong, K., Gill, M., and Walsh, E. (2007b). Adaptive Simulations for Communication Skills Training in Healthcare. In *Workshop on Personalisation for E-Health, 11th International Conference on User Modeling - UM 2007*.
- [David et al., 2011] David, J., Euzenat, J., Scharffe, F., and Trojahn Dos Santos, C. (2011). The Alignment API 4.0. *Semantic Web*, 2(1):3–10.

- [De Bra et al., 2003] 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 14th ACM Conference on Hypertext and Hypermedia - HT 2003*. ACM.
- [Dolog and Schafer, 2005] Dolog, P. and Schafer, M. (2005). A Framework for Browsing, Manipulating and Maintaining Interoperable Learner Profiles. In *Proceedings of the 10th International Conference on User Modeling - UM 2005*, volume 3538 of *Lecture Notes in Computer Science*. Springer.
- [EXist, 2012] EXist (2012). eXist Open Source XML Database. <http://exist-db.org/>.
- [Falconer and Storey, 2007] Falconer, S. and Storey, M.-A. (2007). A cognitive support framework for ontology mapping. In *The Semantic Web: Proceedings of the 6th International Semantic Web Conference - ISWC 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 114–127. Springer.
- [Falconer, 2009] Falconer, S. M. (2009). *Cognitive support for semi-automatic ontology mapping*. PhD thesis, University of Victoria, Canada.
- [Falconer et al., 2007] Falconer, S. M., Noy, N., and Storey, M.-a. (2007). Ontology mapping-a user survey. In *Workshop on Ontology Matching, 6th International Semantic Web Conference - ISWC 2007*.
- [Fellbaum, 1998] Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press.
- [Fu et al., 2011] Fu, B., Brennan, R., and O’Sullivan, D. (2011). Using Pseudo Feedback to Improve Cross-Lingual Ontology Mapping. In *The Semantic Web - Research and Applications: Proceedings of the 8th Extended Semantic Web Conference - ESWC 2011*, volume 6643 of *Lecture Notes in Computer Science*, pages 336–351. Springer.
- [Giunchiglia et al., 2004] Giunchiglia, F., Shvaiko, P., and Yatskevich, M. (2004). S-Match: an algorithm and an implementation of semantic matching. In *The Semantic Web - Research and Applications: Proceedings of the 1st European Semantic Web Symposium - ESWS 2004*, volume 3053 of *Lecture Notes in Computer Science*, pages 61–75. Springer.
- [GRAPPLE, 2012] GRAPPLE (2012). GRAPPLE. <http://www.grapple-project.org/>.
- [Haslhofer, 2008] Haslhofer, B. (2008). *A Web-based Mapping Technique for Establishing Metadata Interoperability*. PhD thesis, University of Vienna, Austria.

- [Heckmann, 2005] Heckmann, D. (2005). *Ubiquitous User Modeling*. PhD thesis, Saarland University, Germany.
- [Heckmann and Krueger, 2003] Heckmann, D. and Krueger, A. (2003). A user modeling markup language (UserML) for ubiquitous computing. In *Proceedings of the 9th International Conference on User Modeling - UM 2003*, volume 2702 of *Lecture Notes in Computer Science*, pages 148–152. Springer.
- [Heckmann et al., 2005] Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., and von Wilamowitz-Moellendorff, M. (2005). GUMO-the general user model ontology. In *Proceedings of the 10th International Conference on User Modeling - UM 2005*, volume 3538 of *Lecture Notes in Computer Science*, pages 428–432. Springer.
- [Horvitz et al., 1998] Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. (1998). The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence - UAI 1998*, pages 256–265. Morgan Kaufmann.
- [IEEE, 2012] IEEE (2012). IEEE Public and Private Information Specification. <http://ltsc.ieee.org/>.
- [IMC (UK) Learning Ltd, 2010] IMC (UK) Learning Ltd (2010). Learning Management Systems - are organisations making the most of them? Technical report, IMC (UK) Learning Ltd.
- [IMS Global Learning Consortium, 2012] IMS Global Learning Consortium (2012). IMS Learner Information Package Specification. <http://www.imsglobal.org/profiles/>.
- [International Organization for Standardization (ISO), 1998] International Organization for Standardization (ISO) (1998). Ergonomic requirements for office work with visual display terminals. Part 11: Guidance on usability. Technical report, International Organization for Standardization.
- [International Organization for Standardization (ISO), 2012] International Organization for Standardization (ISO) (2012). ISO. <http://www.iso.org/>.
- [Internet Engineering Task Force, 2012] Internet Engineering Task Force (2012). LDAP Schema. <http://tools.ietf.org/html/rfc4519>.

- [Kalfoglou and Schorlemmer, 2003] Kalfoglou, Y. and Schorlemmer, M. (2003). Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(01):1–31.
- [Kavcic, 2004] Kavcic, A. (2004). Fuzzy user modeling for adaptation in educational hypermedia. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(4):439–449.
- [Kay et al., 2002] Kay, J., Kummerfeld, B., and Lauder, P. (2002). Personis: a server for user models. In *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems - AH 2002*, volume 2347 of *Lecture Notes in Computer Science*, pages 203–212. Springer.
- [Kobsa, 2007] Kobsa, A. (2007). Privacy-enhanced personalization. *Communications of the ACM*, 50(8):24–33.
- [Kobsa and Fink, 2006] Kobsa, A. and Fink, J. (2006). An LDAP-based user modeling server and its evaluation. *User Modeling and User-Adapted Interaction*, 16(2):129–169.
- [Krug, 2006] Krug, S. (2006). *Don't Make Me Think!: A Common Sense Approach to Web Usability*. New Riders.
- [Martin, 2012] Martin, L. (2012). CedarCrestone 2011-2012 HR Systems Survey Highlights. Technical report, CedarCrestone.
- [McCalla et al., 2000] McCalla, G., Vassileva, J., and Greer, J. (2000). Active learner modelling. In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems - ITS 2000*, volume 1839 of *Lecture Notes in Computer Science*, pages 53–62. Springer.
- [Mehta, 2007] Mehta, B. (2007). Learning from what others know: Privacy preserving cross system personalization. In *Proceedings of the 11th International Conference on User Modeling - UM 2007*, volume 4511 of *Lecture Notes in Computer Science*, pages 57–66. Springer.
- [Mehta, 2008] Mehta, B. (2008). *Cross System Personalization: Enabling personalization across multiple systems*. PhD thesis, Univesity of Duisberg, Germany.
- [Mehta et al., 2005] Mehta, B., Niederee, C., Stewart, A., Degemmis, M., Lops, P., and Semeraro, G. (2005). Ontologically-Enriched Unified User Modeling for Cross-System Personalization. In *Proceedings of the 10th International Conference on User Modeling - UM 2005*, volume 3538 of *Lecture Notes in Computer Science*. Springer.

- [MIT, 2012] MIT (2012). MIT OpenCourseWare. <http://ocw.mit.edu/index.htm>.
- [Moodle, 2012] Moodle (2012). Moodle. <http://moodle.org/>.
- [Nielsen, 1993] Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann.
- [Noy and Musen, 2002] Noy, N. F. and Musen, M. A. (2002). Evaluating Ontology-Mapping Tools: Requirements and Experience. In *OntoWeb-SIG3 Workshop, 13th International Conference on Knowledge Engineering and Knowledge Management - EKAW 2002*.
- [Noy and Musen, 2003] Noy, N. F. and Musen, M. A. (2003). The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024.
- [Nurseitov et al., 2009] Nurseitov, N., Paulson, M., Reynolds, R., and Izurieta, C. (2009). Comparison of JSON and XML Data Interchange Formats: A Case Study. In *Proceedings of the 22nd International Conference on Computer Applications in Industry and Engineering - CAINE 2009*, pages 157–162. ISCA.
- [Ohlsson, 1994] Ohlsson, S. (1994). Constraint-Based Student Modeling. In *Student Modelling: The Key to Individualized Knowledge-Based Instruction*, pages 167–189. Springer.
- [O’Keeffe et al., 2012] O’Keeffe, I., Staikopoulos, A., Rafter, R., Walsh, E., Yousuf, B., Conlan, O., and Wade, V. (2012). Personalized Activity Based eLearning. In *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies - i-KNOW 2012*. ACM.
- [Open Solutions for Education, 2012] Open Solutions for Education (2012). OpenSIS. <http://opensis.com/>.
- [O’Sullivan et al., 2007] O’Sullivan, D., Wade, V., and Lewis, D. (2007). Understanding as we roam. *IEEE Internet Computing*, 11(2):26–33.
- [Progress Software, 2012] Progress Software (2012). Stylus Studio XQuery Mapper. http://www.stylusstudio.com/xquery_mapper.html.
- [Sakai, 2012] Sakai (2012). Sakai. <http://sakaiproject.org/>.
- [Sheth, 1999] Sheth, A. P. (1999). Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. In *Interoperating Geographic Information Systems*, pages 5–30. Springer.

- [Sheth and Larson, 1990] Sheth, A. P. and Larson, J. (1990). Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–236.
- [Shvaiko and Euzenat, 2012] Shvaiko, P. and Euzenat, J. (2012). Ontology matching: state of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*.
- [Staikopoulos et al., 2012] Staikopoulos, A., O’Keeffe, I., Rafter, R., Walsh, E., Yousef, B., Conlan, O., and Wade, V. (2012). AMASE: A framework for composing adaptive and Personalised Learning Activities on the Web. In *Advances in Web-Based Learning: Proceedings of the 11th International Conference on Web-based Learning - ICWL 2012*, volume 7558 of *Lecture Notes in Computer Science*, pages 190–199. Springer.
- [Van Der Sluijs and Houben, 2006] Van Der Sluijs, K. and Houben, G.-J. (2006). A generic component for exchanging user models between web-based systems. *International Journal of Continuing Engineering Education and Life-Long Learning*, 16(1):64–76.
- [VanLehn, 1988] VanLehn, K. (1988). Student modeling. In *Foundations of Intelligent Tutoring Systems*, pages 55–78. Lawrence Erlbaum Associates.
- [W3schools, 2012] W3schools (2012). w3schools. <http://www.w3schools.com/>.
- [Walsh et al., 2007] Walsh, E., Dagger, D., and Wade, V. (2007). Supporting Personalisation for All through Federated User Modelling Exchange Services (FUMES). In *Workshop on Towards User Modelling and Adaptive Systems for All, 11th International Conference on User Modeling - UM 2007*.
- [Walsh et al., 2011a] Walsh, E., O’Connor, A., and Wade, V. (2011a). Supporting Learner Model Exchange in Educational Web Systems. In *Proceedings of the 7th International Conference on Web Information Systems and Technologies - WEBIST 2011*. INSTICC.
- [Walsh et al., 2012] Walsh, E., O’Connor, A., and Wade, V. (2012). Evaluation of a Domain-Aware Approach to User Model Interoperability. In *Proceedings of the 23rd ACM Conference on Hypertext and Social Media - HT 2012*, pages 197–206. ACM.
- [Walsh et al., 2011b] Walsh, E., Rafter, R., Conlan, O., and Wade, V. (2011b). Dynamic contextual usage metadata for learning resource reuse in adaptive environments. In *Proceedings of the IEEE International Conference on Information Reuse and Integration - IEEE IRI 2011*, pages 474–479. IEEE.

- [Walsh and Wade, 2009] Walsh, E. and Wade, V. (2009). Lowering the Barriers to User Model Interoperability. In *Workshop on Ubiquitous User Modelling, 17th International Conference on User Modeling, Adaptation, and Personalization - UMAP 2009*.
- [Wegner, 1996] Wegner, P. (1996). Interoperability. *ACM Computing Surveys*, 28(1):285–287.
- [World Wide Web Consortium (W3C), 2012a] World Wide Web Consortium (W3C) (2012a). OWL Web Ontology Language. <http://www.w3.org/TR/owl-features/>.
- [World Wide Web Consortium (W3C), 2012b] World Wide Web Consortium (W3C) (2012b). Resource Description Framework (RDF). <http://www.w3.org/RDF/>.
- [World Wide Web Consortium (W3C), 2012c] World Wide Web Consortium (W3C) (2012c). Semantic Web Rule Language (SWRL). <http://www.w3.org/Submission/SWRL/>.
- [World Wide Web Consortium (W3C), 2012d] World Wide Web Consortium (W3C) (2012d). SPARQL Query Language. <http://www.w3.org/TR/rdf-sparql-query/>.
- [World Wide Web Consortium (W3C), 2012e] World Wide Web Consortium (W3C) (2012e). XQuery 1.0: An XML Query Language. <http://www.w3.org/TR/xquery/>.
- [World Wide Web Consortium (W3C), 2012f] World Wide Web Consortium (W3C) (2012f). XSL Transformations (XSLT) Version 2.0. <http://www.w3.org/TR/xslt20/>.
- [Yudelson et al., 2007] Yudelson, M., Brusilovsky, P., and Zadorozhny, V. (2007). A user modeling server for contemporary adaptive hypermedia: An evaluation of the push approach to evidence propagation. In *Proceedings of the 11th International Conference on User Modeling - UM 2007*, volume 4511 of *Lecture Notes in Computer Science*, pages 27–36. Springer.

Glossary

Domain A group of related terms that represent a particular field of knowledge.

Domain-Aware Approach The incorporation of exchangeable knowledge from one or more domains into generic tools and processes to support complex interoperability tasks.

Domain-Aware Mechanism A system component that provides domain knowledge.

Domain-Aware User Model Interoperability The incorporation of exchangeable knowledge from one or more domains into generic tools and processes to support the exchange of distributed user data across applications.

Interoperability The ability of two or more software components to cooperate despite differences in language, interface, and execution platform.

User Model Interoperability The process of exchanging distributed user data across applications.

User Model Mapping The process of creating relationships between equivalent elements and values in heterogeneous user models.

User Model Translation The process of transforming heterogeneous user models between different representations.

Appendix A

User Model Examples

Figure A.1: Example of user model 1

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <learner>
3    <general>
4      <identifier>jsmith</identifier>
5      <name>
6        <surname>Smith</surname>
7        <forename>John</forename>
8      </name>
9    </general>
10   <accessibility>
11     <preference>
12       <typename>
13         <tysource sourcetype="custom"/>
14         <tyvalue>scope questionnaire choices</tyvalue>
15       </typename>
16       <prefcode/>
17       <ext_preferencel>
18         <answer1>a</answer1>
19         <answer2>a</answer2>
20         <answer3>a</answer3>
21         <answer4>a</answer4>
22         <answer5>a</answer5>
23       </ext_preferencel>
24     </preference>
25     <preference>
26       <typename>
27         <tysource sourcetype="custom"/>
28         <tyvalue>learning style questionnaire choices</tyvalue>
29       </typename>
30       <prefcode/>
31       <ext_preference2>
32         <answer1>0b</answer1>
33         <answer2>0b</answer2>
34         <answer3>1a</answer3>
35         <answer4>1a</answer4>
36         <answer5>1a</answer5>
37         <answer6>1a</answer6>
38         <answer7>1a</answer7>
39         <answer8>1a</answer8>
40         <answer9>1a</answer9>
41         <answer10>1a</answer10>
42         <answer11>1a</answer11>
43         <answer12>1a</answer12>
44         <answer13>1a</answer13>
45         <answer14>1a</answer14>
46         <answer15>1a</answer15>
47         <answer16>1a</answer16>
```

```

48         <answer17>1a</answer17>
49         <answer18>1a</answer18>
50         <answer19>1a</answer19>
51         <answer20>1a</answer20>
52         <answer21>0b</answer21>
53         <answer22>1a</answer22>
54         <answer23>1a</answer23>
55         <answer24>0b</answer24>
56         <answer25>0b</answer25>
57         <answer26>1a</answer26>
58         <answer27>1a</answer27>
59         <answer28>1a</answer28>
60         <answer29>1a</answer29>
61         <answer30>1a</answer30>
62         <answer31>0b</answer31>
63         <answer32>1a</answer32>
64         <answer33>1a</answer33>
65         <answer34>1a</answer34>
66         <answer35>1a</answer35>
67     </ext_preference2>
68 </preference>
69 <preference>
70     <typename>
71         <tysource sourcetype="custom"/>
72         <tyvalue>honey and mumford learning style scores</tyvalue>
73     </typename>
74     <prefcode/>
75     <ext_preference3>
76         <activist>80</activist>
77         <reflector>40</reflector>
78         <theorist>40</theorist>
79         <pragmatist>50</pragmatist>
80     </ext_preference3>
81 </preference>
82 <preference>
83     <typename>
84         <tysource sourcetype="custom"/>
85         <tyvalue>learning style compilation type</tyvalue>
86     </typename>
87     <prefcode/>
88     <ext_preference4>
89         <compilation_type/>
90     </ext_preference4>
91 </preference>
92 </accessibility>
93 <educational>
94     <adaptivity>
95         <adaptivitytype name="competencies.required">
96             <set type="ALL">
97                 <candidate>
98                     <langstring lang="en">db.concepts.introduction</langstring>
99                 </candidate>
100                 <candidate>
101                     <langstring lang="en">db.concepts.relational_model</langstring>
102                 </candidate>
103                 <candidate>
104                     <langstring lang="en">db.concepts.rdbms</langstring>
105                 </candidate>
106                 <candidate>
107                     <langstring lang="en">db.concepts.sql.syntax</langstring>
108                 </candidate>
109                 <candidate>
110                     <langstring lang="en">db.tables.constraints</langstring>
111                 </candidate>
112                 <candidate>
113                     <langstring lang="en">db.tables.data_types</langstring>
114                 </candidate>
115                 <candidate>
116                     <langstring lang="en">db.tables.create</langstring>
117                 </candidate>

```



```

118         <candidate>
119             <langstring lang="en">db.tables.modify</langstring>
120         </candidate>
121         <candidate>
122             <langstring lang="en">db.views</langstring>
123         </candidate>
124         <candidate>
125             <langstring lang="en">db.createview</langstring>
126         </candidate>
127         <candidate>
128             <langstring lang="en">db.tables.populate</langstring>
129         </candidate>
130         <candidate>
131             <langstring lang="en">db.tables.populate.insert</langstring>
132         </candidate>
133         <candidate>
134             <langstring lang="en">db.tables.populate.update</langstring>
135         </candidate>
136         <candidate>
137             <langstring lang="en">db.tables.populate.delete</langstring>
138         </candidate>
139         <candidate>
140             <langstring lang="en">db.retrieval.simple.select</langstring>
141         </candidate>
142         <candidate>
143             <langstring lang="en">db.retrieval.select</langstring>
144         </candidate>
145         <candidate>
146             <langstring lang="en">db.retrieval.select.operators</langstring>
147         </candidate>
148         <candidate>
149             <langstring lang="en">db.retrieval.clauses.where</langstring>
150         </candidate>
151         <candidate>
152             <langstring lang="en">db.retrieval.clauses.where.search_conditions
153             </langstring>
154         </candidate>
155         <candidate>
156             <langstring lang="en">db.retrieval.joins</langstring>
157         </candidate>
158         <candidate>
159             <langstring lang="en">db.retrieval.simple.joins</langstring>
160         </candidate>
161         <candidate>
162             <langstring lang="en">db.retrieval.aggregate_functions</langstring>
163         </candidate>
164         <candidate>
165             <langstring lang="en">db.examples.knowledge_tree</langstring>
166         </candidate>
167         <candidate>
168             <langstring lang="en">db.applications.embedded_sql</langstring>
169         </candidate>
170         <candidate>
171             <langstring lang="en">db.applications.embedded_sql.structure
172             </langstring>
173         </candidate>
174     </set>
175 </adaptivitytype>
176 </adaptivity>
177 </educational>
178 </learner>

```

Figure A.2: Example of user model 2

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <report>
3     <user>jsmith</user>
4     <user-app-hash>2010-03-05 14:03:36.968</user-app-hash>
5     <user-hash>2010-08-10 17:51:58.437</user-hash>

```

```

6      <concepts>
7          <concept>
8              <name1>Insertion</name1>
9              <cog_levels>
10                 <cog_level>
11                     <name2>knowledge</name2>
12                     <value>0.124</value>
13                 </cog_level>
14                 <cog_level>
15                     <name2>comprehension</name2>
16                     <value>0.478</value>
17                 </cog_level>
18                 <cog_level>
19                     <name2>application</name2>
20                     <value>0.958</value>
21                 </cog_level>
22                 <cog_level>
23                     <name2>synthesis</name2>
24                     <value>0.345</value>
25                 </cog_level>
26             </cog_levels>
27         </concept>
28         <concept>
29             <name1>Update</name1>
30             <cog_levels>
31                 <cog_level>
32                     <name2>knowledge</name2>
33                     <value>0.0</value>
34                 </cog_level>
35                 <cog_level>
36                     <name2>comprehension</name2>
37                     <value>0.0</value>
38                 </cog_level>
39                 <cog_level>
40                     <name2>application</name2>
41                     <value>0.0</value>
42                 </cog_level>
43                 <cog_level>
44                     <name2>synthesis</name2>
45                     <value>0.0</value>
46                 </cog_level>
47             </cog_levels>
48         </concept>
49         <concept>
50             <name1>Deletion</name1>
51             <cog_levels>
52                 <cog_level>
53                     <name2>knowledge</name2>
54                     <value>0.0</value>
55                 </cog_level>
56                 <cog_level>
57                     <name2>comprehension</name2>
58                     <value>0.0</value>
59                 </cog_level>
60                 <cog_level>
61                     <name2>application</name2>
62                     <value>0.0</value>
63                 </cog_level>
64                 <cog_level>
65                     <name2>synthesis</name2>
66                     <value>0.0</value>
67                 </cog_level>
68             </cog_levels>
69         </concept>
70         <concept>
71             <name1>Join</name1>
72             <cog_levels>
73                 <cog_level>
74                     <name2>knowledge</name2>
75                     <value>0.0</value>

```

```

76         </cog_level>
77     <cog_level>
78         <name2>comprehension</name2>
79         <value>0.0</value>
80     </cog_level>
81     <cog_level>
82         <name2>application</name2>
83         <value>0.0</value>
84     </cog_level>
85     <cog_level>
86         <name2>synthesis</name2>
87         <value>0.0</value>
88     </cog_level>
89 </cog_levels>
90 </concept>
91 <concept>
92     <name1>TableCreation</name1>
93     <cog_levels>
94         <cog_level>
95             <name2>knowledge</name2>
96             <value>0.0</value>
97         </cog_level>
98         <cog_level>
99             <name2>comprehension</name2>
100             <value>0.0</value>
101         </cog_level>
102         <cog_level>
103             <name2>application</name2>
104             <value>0.0</value>
105         </cog_level>
106         <cog_level>
107             <name2>synthesis</name2>
108             <value>0.0</value>
109         </cog_level>
110     </cog_levels>
111 </concept>
112 <concept>
113     <name1>TableAlteration</name1>
114     <cog_levels>
115         <cog_level>
116             <name2>knowledge</name2>
117             <value>0.0</value>
118         </cog_level>
119         <cog_level>
120             <name2>comprehension</name2>
121             <value>0.0</value>
122         </cog_level>
123         <cog_level>
124             <name2>application</name2>
125             <value>0.0</value>
126         </cog_level>
127         <cog_level>
128             <name2>synthesis</name2>
129             <value>0.0</value>
130         </cog_level>
131     </cog_levels>
132 </concept>
133 <concept>
134     <name1>ViewCreation</name1>
135     <cog_levels>
136         <cog_level>
137             <name2>knowledge</name2>
138             <value>0.0</value>
139         </cog_level>
140         <cog_level>
141             <name2>comprehension</name2>
142             <value>0.0</value>
143         </cog_level>
144         <cog_level>
145             <name2>application</name2>

```

```

146         <value>0.0</value>
147     </cog_level>
148     <cog_level>
149         <name2>synthesis</name2>
150         <value>0.0</value>
151     </cog_level>
152 </cog_levels>
153 </concept>
154 </concepts>
155 </report>

```

Figure A.3: Example of user model 3

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <learner><user><userid>jsmith</userid>
3      <firstname>John</firstname>
4      <lastname>Smith</lastname>
5      <email>jsmith@test.com</email>
6      <address>1 College Green,Dublin 2,Ireland</address>
7  </user>
8  <courses><course><title1>Introduction to Databases</title1>
9      <type>course</type>
10     <description>Introduction to Databases</description>
11     <gradebook><coursegrade>B</coursegrade>
12         <grade><title2>Mid-Term Assignment</title2>
13             <finalscore>16.0</finalscore>
14             <totalscore>20.0</totalscore>
15             <date>29/10/2010</date>
16         </grade>
17         <grade><title2>Final Exam</title2>
18             <finalscore>14.0</finalscore>
19             <totalscore>20.0</totalscore>
20             <date>29/06/2011</date>
21         </grade>
22     </gradebook>
23 </course>
24 <course><title1>Advanced Database Systems</title1>
25     <type>course</type>
26     <description>Advanced Database Systems</description>
27     <gradebook><coursegrade>C</coursegrade>
28         <grade><title2>Mid-Term Assignment</title2>
29             <finalscore>12.0</finalscore>
30             <totalscore>20.0</totalscore>
31             <date>20/08/2010</date>
32         </grade>
33         <grade><title2>Final Exam</title2>
34             <finalscore>13.0</finalscore>
35             <totalscore>20.0</totalscore>
36             <date>05/06/2011</date>
37         </grade>
38     </gradebook>
39 </course>
40 </courses>
41 </learner>

```

Figure A.4: Example of user model 4

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <learner>
3      <user>
4          <item1>
5              <username>jsmith</username>
6              <firstname/>
7              <lastname/>
8              <name/>
9              <email/>
10             <city/>
11             <country/>
12             <institution/>

```



```

13         <department/>
14     </item1>
15 </user>
16 <learningstyles>
17     <item2>
18         <name/>
19         <scores>
20             <activist/>
21             <reflector/>
22             <theorist/>
23             <pragmatist/>
24         </scores>
25     </item2>
26 </learningstyles>
27 <grades>
28     <item3>
29         <fullname>Introduction to Databases</fullname>
30         <shortname>CS100</shortname>
31         <str_grade/>
32         <feedback/>
33         <date/>
34     </item3>
35     <item3>
36         <fullname>Advanced Database Systems</fullname>
37         <shortname>CS200</shortname>
38         <str_grade/>
39         <feedback/>
40         <date/>
41     </item3>
42 </grades>
43 </learner>

```

Figure A.5: Example of canonical user model

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <user>
3     <identification>
4         <userid/>
5         <name>
6             <firstname/>
7             <lastname/>
8             <fullname/>
9         </name>
10        <email/>
11        <address>
12            <street/>
13            <city/>
14            <country/>
15            <postcode/>
16        </address>
17        <contact>
18            <phone/>
19            <fax/>
20            <email/>
21            <web/>
22        </contact>
23    </identification>
24    <competency>
25        <sql>
26            <basicconcepts>
27                <intro>
28                    <cognitionlevel>
29                        <date/>
30                        <description/>
31                        <value/>
32                    </cognitionlevel>
33                </intro>
34                <syntax>
35                    <cognitionlevel>

```

```

36         <date/>
37         <description/>
38         <value/>
39     </cognitionlevel>
40 </syntax>
41 <select>
42     <cognitionlevel>
43         <date/>
44         <description/>
45         <value/>
46     </cognitionlevel>
47 </select>
48 <distinct>
49     <cognitionlevel>
50         <date/>
51         <description/>
52         <value/>
53     </cognitionlevel>
54 </distinct>
55 <where>
56     <cognitionlevel>
57         <date/>
58         <description/>
59         <value/>
60     </cognitionlevel>
61 </where>
62 <andor>
63     <cognitionlevel>
64         <date/>
65         <description/>
66         <value/>
67     </cognitionlevel>
68 </andor>
69 <orderby>
70     <cognitionlevel>
71         <date/>
72         <description/>
73         <value/>
74     </cognitionlevel>
75 </orderby>
76 <insert>
77     <cognitionlevel>
78         <date/>
79         <description/>
80         <value/>
81     </cognitionlevel>
82 </insert>
83 <update>
84     <cognitionlevel>
85         <date/>
86         <description/>
87         <value/>
88     </cognitionlevel>
89 </update>
90 <delete>
91     <cognitionlevel>
92         <date/>
93         <description/>
94         <value/>
95     </cognitionlevel>
96 </delete>
97 </basicconcepts>
98 </sql>
99 </competency>
100 <assessment>
101     <databases>
102         <introductory>
103             <exam>
104                 <grade>
105                     <date/>

```

```

106         <description/>
107         <value/>
108     </grade>
109 </exam>
110 <assignment>
111     <grade>
112         <date/>
113         <description/>
114         <value/>
115     </grade>
116 </assignment>
117 <quiz>
118     <grade>
119         <date/>
120         <description/>
121         <value/>
122     </grade>
123 </quiz>
124 </introductory>
125 <intermediate>
126     <exam>
127         <grade>
128             <date/>
129             <description/>
130             <value/>
131         </grade>
132     </exam>
133     <assignment>
134         <grade>
135             <date/>
136             <description/>
137             <value/>
138         </grade>
139     </assignment>
140     <quiz>
141         <grade>
142             <date/>
143             <description/>
144             <value/>
145         </grade>
146     </quiz>
147 </intermediate>
148 <advanced>
149     <exam>
150         <grade>
151             <date/>
152             <description/>
153             <value/>
154         </grade>
155     </exam>
156     <assignment>
157         <grade>
158             <date/>
159             <description/>
160             <value/>
161         </grade>
162     </assignment>
163     <quiz>
164         <grade>
165             <date/>
166             <description/>
167             <value/>
168         </grade>
169     </quiz>
170 </advanced>
171 </databases>
172 </assessment>
173 </user>

```


Appendix B

Mapping Examples

Figure B.1: Example of XML mappings

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 < mappings>< mapping>< uid>BFA4C409-125F-F61C-BB31-36012D8B233A</uid>
3   < idnumber>1</idnumber>
4   < group>1</group>
5   < type>text_l21</type>
6   < name>Identification</name>
7   < description>equivalent first name</description>
8   < display x="339.35" y="765.6" color="16771286">max</display>
9   < sources>< source um="UM_1">/learner/general/name/forename</source>
10  </sources>
11  < targets>< target um="UM_0">/user/identification/name/firstname</target>
12  </targets>
13  < operators>< operator_a/>
14    < operator_b/>
15  </operators>
16  < values>< value_a/>
17    < value_b/>
18  </values>
19 </mapping>
20 < mapping>< uid>6B3587D1-BEA6-7CC4-AD5D-36012D8B8BC0</uid>
21   < idnumber>2</idnumber>
22   < group>1</group>
23   < type>text_l21</type>
24   < name>Identification</name>
25   < description>equivalent first name</description>
26   < display x="512.1" y="758.9" color="14086911">max</display>
27   < sources>< source um="UM_4">/learner/user/item1/firstname</source>
28  </sources>
29  < targets>< target um="UM_0">/user/identification/name/firstname</target>
30  </targets>
31  < operators>< operator_a/>
32    < operator_b/>
33  </operators>
34  < values>< value_a/>
35    < value_b/>
36  </values>
37 </mapping>
38 < mapping>< uid>7E9FDC61-D312-B217-08DE-36012D8C61B7</uid>
39   < idnumber>3</idnumber>
40   < group>1</group>
41   < type>text_l2m</type>
42   < name>Identification</name>
43   < description>split address and get country</description>
44   < display x="724.55" y="696.05" color="16766678">max</display>
45   < sources>< source um="UM_3">/learner/user/address</source>
46  </sources>
47  < targets>< target um="UM_0">/user/identification/address/country</target>
```

```

48     </targets>
49     <operators><operator_a>3</operator_a>
50     <operator_b/>
51     </operators>
52     <values><value_a>,</value_a>
53     <value_b/>
54     </values>
55 </mapping>
56 <mapping><uid>70AE2FA0-81E7-4828-55FE-36012D8CF7B6</uid>
57     <idnumber>4</idnumber>
58     <group>1</group>
59     <type>text_121</type>
60     <name>Identification</name>
61     <description>equivalent address</description>
62     <display x="932.8" y="697.95" color="14086911">max</display>
63     <sources><source um="UM_4">/learner/user/item1/country</source>
64     </sources>
65     <targets><target um="UM_0">/user/identification/address/country</target>
66     </targets>
67     <operators><operator_a/>
68     <operator_b/>
69     </operators>
70     <values><value_a/>
71     <value_b/>
72     </values>
73 </mapping>
74 <mapping><uid>88DF911A-A0BD-FB24-C8D6-36012D8DBD09</uid>
75     <idnumber>5</idnumber>
76     <group>3</group>
77     <type>numeric_m2m</type>
78     <name>Assessment</name>
79     <description>grades 0-20 to Pass/Fail</description>
80     <display x="1621.15" y="1123.05" color="16766678">max</display>
81     <sources><source um="UM_3">/learner/courses/course[title='Advanced_Database_Systems']
82     /gradebook/grade[title2='Final_Exam']/finalscore</source>
83     </sources>
84     <targets><target um="UM_0">/user/assessment/databases/advanced/exam/grade/value</target>
85     </targets>
86     <operators><operator_a/>
87     <operator_b/>
88     </operators>
89     <values><value_a>[11-20:Good][0-10:Bad]</value_a>
90     <value_b/>
91     </values>
92 </mapping>
93 <mapping><uid>78B9A002-CBD8-612D-97EC-36012D8C87B0</uid>
94     <idnumber>6</idnumber>
95     <group>3</group>
96     <type>text_121</type>
97     <name>Assessment</name>
98     <description>equivalent assessment</description>
99     <display x="1404" y="1156.05" color="14086911">max</display>
100    <sources><source um="UM_4">/learner/grades/item3[fullname='Advanced_Database_Systems']
101    /str_grade</source>
102    </sources>
103    <targets><target um="UM_0">/user/assessment/databases/advanced/exam/grade/value</target>
104    </targets>
105    <operators><operator_a/>
106    <operator_b/>
107    </operators>
108    <values><value_a/>
109    <value_b/>
110    </values>
111 </mapping>
112 <mapping><uid>F25DB2C3-3B7E-00C6-28CB-36012D8EC24B</uid>
113     <idnumber>7</idnumber>
114     <group>4</group>
115     <type>text_numeric_121</type>
116     <name>Competency</name>
117     <description>SQL concept to 0/1 score</description>

```

```

118 <display x="1990.15" y="1112.55" color="16771286">max</display>
119 <sources><source um="UM_1">/learner/educational/adaptivity/adaptivitytype/set/candidate
120 /langstring[.='db.tables.populate.insert']</source>
121 </sources>
122 <targets><target um="UM_0">/user/competency/sql/basicconcepts/insert/cognitionlevel
123 /value</target>
124 </targets>
125 <operators><operator_a>=</operator_a>
126 <operator_b>=</operator_b>
127 </operators>
128 <values><value_a>0</value_a>
129 <value_b>1</value_b>
130 </values>
131 </mapping>
132 <mapping><uid>7984BE04-EF4E-695B-E523-36012D8EF6CC</uid>
133 <idnumber>8</idnumber>
134 <group>4</group>
135 <type>numeric_121</type>
136 <name>Competency</name>
137 <description>equivalent competency</description>
138 <display x="2183.65" y="1108.85" color="14090200">max</display>
139 <sources><source um="UM_2">/report/concepts/concept[name1='Insertion']/cog_levels
140 /cog_level[name2='knowledge']/value</source>
141 </sources>
142 <targets><target um="UM_0">/user/competency/sql/basicconcepts/insert/cognitionlevel
143 /value</target>
144 </targets>
145 <operators><operator_a>*</operator_a>
146 <operator_b>div</operator_b>
147 </operators>
148 <values><value_a>1</value_a>
149 <value_b></value_b>
150 </values>
151 </mapping>
152 </mappings>

```

Figure B.2: Example of source to canonical XQuery mappings

```

1 import module namespace customfunctions='fuse' at 'xmldb:exist:///db/fuse/mapping_tool
2 /modules/custom_functions.xq';
3 declare variable $um1 := '/mapping_tool/testing/admin1/instances/um1.xml';
4 declare variable $um3 := '/mapping_tool/testing/admin1/instances/um3.xml';
5 declare variable $um0 := '/mapping_tool/testing/admin1/instances/um0.xml';
6 <mappings>
7 {
8   if (doc($um1)/learner/general/name/forename != '')
9   then
10    update replace
11    doc($um0)/user/identification/name/firstname
12    with
13    <firstname>{doc($um1)/learner/general/name/forename/text()}</firstname>
14   else
15    update replace
16    doc($um0)/user/identification/name/firstname
17    with
18    <firstname/>
19  }
20  {
21    if (doc($um3)/learner/user/address != '')
22    then
23    let $sep := tokenize(doc($um3)/learner/user/address/text(),',')
24    return
25    update replace
26    doc($um0)/user/identification/address/country
27    with
28    <country>{$sep[3]}</country>
29   else
30    update replace
31    doc($um0)/user/identification/address/country

```

```

32   with
33   <country/>
34   }
35   {
36   if (doc($um3)/learner/courses/course[title1='Advanced_Database_Systems']
37   /gradebook/grade[title2='Final_Exam']/finalscore >= 11 and doc($um3)/learner
38   /courses/course[title1='Advanced_Database_Systems']/gradebook/grade[title2='Final_Exam']
39   /finalscore <= 20)
40   then((
41   update delete
42   doc($um0)/user/assessment/databases/advanced/exam/grade/value
43   ),(
44   update insert
45   <value>Pass</value>
46   into doc($um0)/user/assessment/databases/advanced/exam/grade
47   ))
48   else
49   if (doc($um3)/learner/courses/course[title1='Advanced_Database_Systems']/gradebook
50   /grade[title2='Final_Exam']/finalscore >= 0 and doc($um3)/learner/courses
51   /course[title1='Advanced_Database_Systems']/gradebook/grade[title2='Final_Exam']
52   /finalscore <= 10)
53   then((
54   update delete
55   doc($um0)/user/assessment/databases/advanced/exam/grade/value
56   ),(
57   update insert
58   <value>Fail</value>
59   into doc($um0)/user/assessment/databases/advanced/exam/grade
60   ))
61   else
62   ((
63   update delete
64   doc($um0)/user/assessment/databases/advanced/exam/grade/value
65   ),(
66   update insert
67   <value/>
68   into doc($um0)/user/assessment/databases/advanced/exam/grade
69   ))
70   }
71   {
72   if (doc($um1)/learner/educational/adaptivity/adaptivitytype/set/candidate
73   /langstring[.='db.tables.populate.insert'] != '')
74   then((
75   update delete
76   doc($um0)/user/competency/sql/basicconcepts/insert/cognitionlevel/value
77   ),(
78   update insert
79   <value>0</value>
80   into doc($um0)/user/competency/sql/basicconcepts/insert/cognitionlevel
81   ))
82   else((
83   update delete
84   doc($um0)/user/competency/sql/basicconcepts/insert/cognitionlevel/value
85   ),(
86   update insert
87   <value>1</value>
88   into doc($um0)/user/competency/sql/basicconcepts/insert/cognitionlevel
89   ))
90   }
91 </mappings>

```

Figure B.3: Example of canonical to target XQuery mappings

```

1 import module namespace customfunctions='fuse' at 'xmldb:exist:///db/fuse/mapping_tool
2 /modules/custom_functions.xq';
3 declare variable $um2 := '/mapping_tool/testing/admin1/instances/um2.xml';
4 declare variable $um4 := '/mapping_tool/testing/admin1/instances/um4.xml';
5 declare variable $um0 := '/mapping_tool/testing/admin1/instances/um0.xml';
6 <mappings>

```



```

7      {
8      if (doc($um0)/user/identification/name/firstname != '')
9      then
10     update replace
11     doc($um4)/learner/user/item1/firstname
12     with
13     <firstname>{doc($um0)/user/identification/name/firstname/text()}</firstname>
14     else
15     update replace
16     doc($um4)/learner/user/item1/firstname
17     with
18     <firstname/>
19     }
20     {
21     if (doc($um0)/user/identification/address/country != '')
22     then
23     update replace
24     doc($um4)/learner/user/item1/country
25     with
26     <country>{doc($um0)/user/identification/address/country/text()}</country>
27     else
28     update replace
29     doc($um4)/learner/user/item1/country
30     with
31     <country/>
32     }
33     {
34     if (doc($um0)/user/assessment/databases/advanced/exam/grade/value != '')
35     then
36     update replace
37     doc($um4)/learner/grades/item3[fullname='Advanced_Database_Systems']/str_grade
38     with
39     <str_grade>{doc($um0)/user/assessment/databases/advanced/exam/grade/value/text()}
40     </str_grade>
41     else
42     update replace
43     doc($um4)/learner/grades/item3[fullname='Advanced_Database_Systems']/str_grade
44     with
45     <str_grade/>
46     }
47     {
48     if (doc($um0)/user/competency/sql/basicconcepts/insert/cognitionlevel/value != '')
49     then
50     update replace
51     doc($um2)/report/concepts/concept[name1='Insertion']/cog_levels
52     /cog_level[name2='knowledge']/value
53     with
54     <value>{doc($um0)/user/competency/sql/basicconcepts/insert/cognitionlevel
55     /value/text()}</value>
56     else
57     update replace
58     doc($um2)/report/concepts/concept[name1='Insertion']/cog_levels
59     /cog_level[name2='knowledge']/value
60     with
61     <value/>
62     }
63 </mappings>

```


Appendix C

Evaluation Results

Table C.1: Semi-structured interview

- Q1. What is your general opinion of the user interface?
- Q2. Are there any aspects of the user interface that you liked?
- Q3. Are there any aspects of the user interface that you did not like and would like to see changed or removed?
- Q4. Are there any new features you think might assist you in the mapping process?
- Q5. Did you have any difficulty navigating the user interface?
- Q6. Did you have any difficulty interpreting any of the menus or buttons?
- Q7. Did you have any difficulty completing any of the mapping tasks?
- Q8. What aspect of the mapping process did you find the most difficult?
- Q9. Did you have any difficulty navigating and finding the required data in the source models?
- Q10. Did you have any difficulty navigating the canonical model and creating mappings?
- Q11. Did you have any difficulty testing the mappings?
- Q12. Any other comments?

Table C.2: Mapping accuracy and duration for task 1

User	S1	S2	S3	Mapping Tool		T1	Time (sec)	Time (%)
				S4	S5			
User 1	1.00	1.00	1.00	1.00	1.00	100.00	150.00	50.00
User 2	1.00	1.00	1.00	1.00	1.00	100.00	149.00	49.67
User 3	1.00	1.00	1.00	1.00	1.00	100.00	83.00	27.67
User 4	1.00	1.00	1.00	1.00	1.00	100.00	87.00	29.00
User 5	1.00	1.00	1.00	1.00	1.00	100.00	115.00	38.33
User 6	1.00	1.00	1.00	1.00	1.00	100.00	192.00	64.00
User 7	1.00	1.00	1.00	1.00	1.00	100.00	190.00	63.33
User 8	1.00	1.00	1.00	1.00	1.00	100.00	163.00	54.33
User 9	1.00	1.00	1.00	1.00	1.00	100.00	136.00	45.33
User 10	1.00	1.00	1.00	1.00	1.00	100.00	161.00	53.67
User 11	1.00	1.00	1.00	1.00	1.00	100.00	91.00	30.33
User 12	1.00	1.00	1.00	1.00	1.00	100.00	178.00	59.33
	100.00	100.00	100.00	100.00	100.00	100.00	141.25	47.08

User	S1	S2	Mapforce		T1	Time (sec)	Time (%)
User 1	1.00	1.00			100.00	109.00	36.33
User 2	1.00	1.00			100.00	64.00	21.33
User 3	1.00	1.00			100.00	80.00	26.67
User 4	1.00	1.00			100.00	94.00	31.33
User 5	1.00	1.00			100.00	65.00	21.67
User 6	1.00	1.00			100.00	142.00	47.33
User 7	1.00	1.00			100.00	109.00	36.33
User 8	1.00	1.00			100.00	34.00	11.33
User 9	1.00	1.00			100.00	99.00	33.00
User 10	1.00	1.00			100.00	61.00	20.33
User 11	1.00	1.00			100.00	33.00	11.00
User 12	1.00	1.00			100.00	34.00	11.33
	100.00	100.00			100.00	77.00	25.67

No Errors 1.00
Minor Errors 0.50
Major Errors 0.00

Table C.3: Mapping accuracy and duration for task 2

User	Mapping Tool						Time (secs)	Time (%)
	S1	S2	S3	S4	S5	T2		
User 1	1.00	1.00	1.00	1.00	1.00	100.00	171.00	57.00
User 2	1.00	1.00	1.00	1.00	1.00	100.00	300.00	100.00
User 3	1.00	1.00	1.00	1.00	1.00	100.00	93.00	31.00
User 4	1.00	1.00	1.00	1.00	1.00	100.00	192.00	64.00
User 5	1.00	1.00	1.00	1.00	1.00	100.00	290.00	96.67
User 6	1.00	1.00	1.00	1.00	1.00	100.00	292.00	97.33
User 7	1.00	1.00	1.00	1.00	1.00	100.00	246.00	82.00
User 8	1.00	1.00	1.00	1.00	1.00	100.00	174.00	58.00
User 9	1.00	1.00	1.00	1.00	1.00	100.00	174.00	58.00
User 10	1.00	1.00	1.00	1.00	1.00	100.00	144.00	48.00
User 11	1.00	1.00	1.00	1.00	1.00	100.00	276.00	92.00
User 12	1.00	1.00	1.00	1.00	1.00	100.00	83.00	27.67
	100.00	100.00	100.00	100.00	100.00	100.00	202.92	67.64

User	Mapforce						Time (sec)	Time (%)
	S1	S2	S3	S4	S5	T2		
User 1	1.00	0.00	0.00	0.00	1.00	40.00	300.00	100.00
User 2	1.00	1.00	0.00	0.00	1.00	60.00	272.00	90.67
User 3	1.00	1.00	0.00	0.00	1.00	60.00	300.00	100.00
User 4	1.00	1.00	1.00	1.00	1.00	100.00	228.00	76.00
User 5	1.00	1.00	1.00	0.00	1.00	80.00	300.00	100.00
User 6	1.00	1.00	0.00	0.00	1.00	60.00	300.00	100.00
User 7	1.00	1.00	0.00	0.00	1.00	60.00	300.00	100.00
User 8	1.00	1.00	0.00	0.00	1.00	60.00	300.00	100.00
User 9	1.00	1.00	0.00	0.00	1.00	60.00	300.00	100.00
User 10	1.00	1.00	1.00	1.00	1.00	100.00	246.00	82.00
User 11	1.00	1.00	0.00	0.00	1.00	60.00	300.00	100.00
User 12	1.00	1.00	0.00	0.00	1.00	60.00	300.00	100.00
	100.00	91.67	25.00	16.67	100.00	66.67	287.17	95.72

No Errors 1.00
Minor Errors 0.50
Major Errors 0.00

Table C.4: Mapping accuracy and duration for task 3

User	Mapping Tool							T3	Time (secs)	Time (%)
	S1	S2	S3	S4	S5	S6	S7			
User 1	1.00	0.00	1.00	0.00	0.00	1.00	1.00	57.14	600.00	100.00
User 2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	100.00	500.00	83.33
User 3	0.50	1.00	1.00	0.00	1.00	1.00	1.00	78.57	555.00	92.50
User 4	1.00	1.00	1.00	1.00	1.00	0.00	1.00	85.71	600.00	100.00
User 5	1.00	1.00	1.00	0.00	0.00	1.00	1.00	71.43	510.00	85.00
User 6	1.00	1.00	0.00	1.00	0.00	1.00	1.00	71.43	600.00	100.00
User 7	1.00	1.00	1.00	0.00	1.00	0.00	1.00	71.43	600.00	100.00
User 8	1.00	0.50	1.00	1.00	0.00	0.50	1.00	71.43	600.00	100.00
User 9	1.00	0.50	1.00	1.00	1.00	0.50	1.00	85.71	600.00	100.00
User 10	1.00	1.00	1.00	0.00	1.00	1.00	1.00	85.71	341.00	56.83
User 11	1.00	1.00	1.00	1.00	1.00	0.00	1.00	85.71	600.00	100.00
User 12	1.00	1.00	1.00	1.00	0.00	1.00	1.00	85.71	350.00	58.33
	95.83	83.33	91.67	58.33	58.33	66.67	100.00	79.17	538.00	89.67

User	Mapforce						
	S1	S2	S3	S4	S5	S6	S7
User 1	0.00	0.00	1.00	0.00	0.00	0.00	0.00
User 2	0.00	0.00	1.00	0.00	0.00	0.00	0.00
User 3	0.00	1.00	1.00	0.00	1.00	0.00	1.00
User 4	1.00	1.00	1.00	1.00	1.00	0.00	1.00
User 5	0.00	0.00	1.00	0.00	0.00	0.00	0.00
User 6	1.00	0.00	1.00	1.00	0.00	0.00	0.00
User 7	1.00	0.00	0.00	1.00	0.00	0.00	0.00
User 8	1.00	0.00	1.00	0.00	0.00	0.00	1.00
User 9	0.00	1.00	1.00	1.00	1.00	1.00	1.00
User 10	0.00	0.00	1.00	0.00	0.00	0.00	0.00
User 11	1.00	0.00	0.00	1.00	0.00	0.00	1.00
User 12	1.00	1.00	1.00	1.00	1.00	1.00	0.00
	50.00	33.33	83.33	50.00	33.33	16.67	41.67

User	Mapforce			T3	Time (sec)	Time (%)
	S8	S9	S10			
User 1	0.00	0.00	1.00	20.00	540.00	90.00
User 2	1.00	0.50	1.00	35.00	448.00	74.67
User 3	1.00	0.50	1.00	65.00	600.00	100.00
User 4	0.00	0.00	1.00	70.00	600.00	100.00
User 5	1.00	0.50	1.00	35.00	600.00	100.00
User 6	1.00	0.00	0.00	40.00	600.00	100.00
User 7	1.00	0.00	1.00	40.00	600.00	100.00
User 8	0.00	0.00	1.00	40.00	400.00	66.67
User 9	0.00	0.00	1.00	70.00	600.00	100.00
User 10	1.00	1.00	1.00	40.00	339.00	56.50
User 11	0.00	0.00	1.00	40.00	600.00	100.00
User 12	1.00	1.00	1.00	90.00	472.00	78.67
	58.33	29.17	91.67	48.75	533.25	88.88

No Errors 1.00 Minor Errors 0.50 Major Errors 0.00

Table C.5: Mapping accuracy and duration for task 4

User	Mapping Tool					T4	Time (secs)	Time (%)
	S1	S2	S3	S4	S5			
User 1	1.00	0.00	1.00	0.00	0.00	40.00	512.00	85.33
User 2	1.00	1.00	1.00	1.00	0.50	90.00	534.00	89.00
User 3	0.50	0.50	1.00	0.50	1.00	70.00	402.00	67.00
User 4	1.00	1.00	1.00	1.00	1.00	100.00	450.00	75.00
User 5	0.50	0.50	1.00	0.50	0.00	50.00	472.00	78.67
User 6	0.00	0.00	0.00	1.00	1.00	40.00	600.00	100.00
User 7	1.00	1.00	1.00	0.00	0.00	60.00	600.00	100.00
User 8	0.00	0.50	1.00	0.50	0.00	40.00	410.00	68.33
User 9	1.00	1.00	1.00	1.00	1.00	100.00	492.00	82.00
User 10	1.00	1.00	1.00	1.00	1.00	100.00	503.00	83.83
User 11	0.00	1.00	1.00	0.00	1.00	60.00	410.00	68.33
User 12	1.00	1.00	1.00	1.00	1.00	100.00	463.00	77.17
	66.67	70.83	91.67	62.50	62.50	70.83	487.33	81.22

User	Mapforce						T4	Time (secs)	Time (%)
	S1	S2	S3	S4	S5	S6			
User 1	0.00	0.00	0.00	0.00	0.00	1.00	16.67	564.00	94.00
User 2	1.00	1.00	1.00	0.00	0.00	1.00	66.67	421.00	70.17
User 3	0.00	1.00	0.00	0.00	0.50	1.00	41.67	600.00	100.00
User 4	1.00	1.00	1.00	0.00	0.00	1.00	66.67	435.00	72.50
User 5	1.00	1.00	0.00	0.00	0.50	1.00	58.33	600.00	100.00
User 6	1.00	0.00	0.00	1.00	0.50	1.00	58.33	600.00	100.00
User 7	1.00	1.00	0.00	0.00	1.00	1.00	66.67	600.00	100.00
User 8	0.00	0.00	1.00	0.00	0.00	0.00	16.67	482.00	80.33
User 9	1.00	1.00	0.00	0.00	1.00	1.00	66.67	600.00	100.00
User 10	0.00	0.00	0.00	1.00	1.00	1.00	50.00	370.00	61.67
User 11	0.00	1.00	0.00	0.00	0.00	1.00	33.33	600.00	100.00
User 12	1.00	1.00	0.00	0.50	1.00	1.00	75.00	600.00	100.00
	58.33	66.67	25.00	20.83	45.83	91.67	51.39	539.33	89.89

No Errors 1.00
Minor Errors 0.50
Major Errors 0.00

Table C.6: Mapping accuracy and duration for tasks 1-4

User	Mapping Tool				
	Mean Accuracy	Time (secs)	Mean Time (secs)	Time (mins)	Time (%)
User 1	72.73	1433.00	358.25	23.88	79.61
User 2	97.73	1483.00	370.75	24.72	82.39
User 3	86.36	1133.00	283.25	18.88	62.94
User 4	95.45	1329.00	332.25	22.15	73.83
User 5	79.55	1387.00	346.75	23.12	77.06
User 6	77.27	1684.00	421.00	28.07	93.56
User 7	81.82	1636.00	409.00	27.27	90.89
User 8	77.27	1347.00	336.75	22.45	74.83
User 9	95.45	1402.00	350.50	23.37	77.89
User 10	95.45	1149.00	287.25	19.15	63.83
User 11	86.36	1377.00	344.25	22.95	76.50
User 12	95.45	1074.00	268.50	17.90	59.67
	86.74	1369.50	342.38	22.83	76.08

User	Mapforce				
	Mean Accuracy	Time (secs)	Mean Time (secs)	Time (mins)	Time (%)
User 1	30.43	1513.00	378.25	25.22	84.06
User 2	54.35	1205.00	301.25	20.08	66.94
User 3	60.87	1580.00	395.00	26.33	87.78
User 4	78.26	1357.00	339.25	22.62	75.39
User 5	56.52	1565.00	391.25	26.08	86.94
User 6	54.35	1642.00	410.50	27.37	91.22
User 7	56.52	1609.00	402.25	26.82	89.39
User 8	43.48	1216.00	304.00	20.27	67.56
User 9	69.57	1599.00	399.75	26.65	88.83
User 10	60.87	1016.00	254.00	16.93	56.44
User 11	47.83	1533.00	383.25	25.55	85.17
User 12	80.43	1406.00	351.50	23.43	78.11
	57.79	1436.75	359.19	23.95	79.82

Table C.7: Paired t-tests for accuracy and duration

User	Difference	Accuracy	
		Paired t-tests	
User 1	42.29249012	Mean	28.95256917
User 2	43.37944664	Standard Deviation	9.39883368
User 3	25.49407115	Standard Error	2.71320958
User 4	17.19367589	Test Statistic	10.67096674
User 5	23.02371542	Degrees of Freedom	11
User 6	22.92490119	p-value	0.00000038
User 7	25.29644269		
User 8	33.79446640	T1 p-value	
User 9	25.88932806	T2 p-value	0.00004401
User 10	34.58498024	T3 p-value	0.00012475
User 11	38.53754941	T4 p-value	0.00630730
User 12	15.01976285		

User	Difference	Duration	
		Paired t-tests	
User 1	80.00	Mean	67.25000000
User 2	-278.00	Standard Deviation	207.23334454
User 3	447.00	Standard Error	59.82311363
User 4	28.00	Test Statistic	1.12414744
User 5	178.00	Degrees of Freedom	11
User 6	-42.00	p-value	0.28488379
User 7	-27.00		
User 8	-131.00	T1 p-value	0.00050506
User 9	197.00	T2 p-value	0.00364406
User 10	-133.00	T3 p-value	0.84151181
User 11	156.00	T4 p-value	0.12344130
User 12	332.00		

Table C.8: Prior experience questionnaire

Q1. I have experience using XML.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q2. I have experience using XPath.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q3. I have experience using XQuery.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q4. I have experience using SQL.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q5. I have experience using user models.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q6. I have experience creating mappings between different data models (e.g. databases, XML schemas, ontologies, etc).				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q7. I have experience creating mappings between different user models.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q8. Please briefly describe any previous mapping tasks you have performed and state the mapping tool used.				

Table C.9: Participant responses to prior experience questionnaire

Users	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
User 1	D	SD	SD	A	N	D	SD	
User 2	A	N	N	A	D	N	D	
User 3	SA	A	A	A	A	N	N	
User 4	A	A	D	SA	SD	N	D	Some experience mapping web services using Mapforce
User 5	SA	A	A	A	N	A	D	Three tools in other experiments mapping sports terms, websites
User 6	SA	SA	A	SA	A	A	N	Mapforce a little bit, QVT Tool Visual Editor
User 7	A	D	A	A	N	N	A	
User 8	SA	A	A	SA	SA	SA	A	
User 9	A	A	N	SA	A	N	D	Object-Relational Mapping (OOP to DB)
User 10	A	D	D	A	D	A	D	
User 11	SA	SD	SD	A	D	SD	SD	
User 12	A	SD	SD	SA	SD	A	SD	Created a tool for mapping RDFa data in web pages and OWL ontologies

SA Strongly Agree
A Agree
N Neither Agree or Disagree
D Disagree
SD Strongly Disagree

Table C.10: SUS questionnaire

Q1. I think that I would like to use this system frequently.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q2. I found the system unnecessarily complex.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q3. I thought the system was easy to use.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q4. I think that I would need the support of a technical person to be able to use this system.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q5. I found the various functions in this system were well integrated.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q6. I thought there was too much inconsistency in this system.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q7. I would imagine that most people would learn to use this system very quickly.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q8. I found the system very cumbersome to use.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q9. I felt very confident using the system.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q10. I needed to learn a lot of things before I could get going with this system.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree

Table C.11: Participant responses to SUS questionnaire

Users	Mapping Tool										SUS
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	
User 1	N	D	N	D	A	SD	A	SD	D	D	70
User 2	SA	D	A	A	SA	D	A	SD	SA	D	80
User 3	A	D	N	N	A	D	A	D	N	N	65
User 4	A	D	A	A	A	D	N	D	N	D	65
User 5	A	D	A	D	A	D	A	SD	A	D	77.5
User 6	SA	D	A	D	A	D	A	D	A	D	77.5
User 7	N	D	A	N	A	D	A	D	A	D	70
User 8	N	D	A	N	A	SD	SA	SD	D	A	67.5
User 9	A	D	SA	N	SA	SD	A	D	A	N	77.5
User 10	SA	SD	SA	A	SA	SD	A	SD	SA	N	75
User 11	A	D	A	N	A	D	A	D	N	D	70
User 12	N	D	A	D	A	SD	N	D	N	D	70
Overall SUS Score:											72

Users	Mapforce										SUS
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	
User 1	D	A	D	A	A	D	D	A	SD	N	35
User 2	A	N	A	A	A	D	A	N	N	D	62.5
User 3	D	A	D	A	N	D	D	A	SD	SA	27.5
User 4	A	D	A	A	N	D	D	D	N	A	55
User 5	A	N	N	D	A	D	D	N	N	A	55
User 6	A	N	A	A	N	D	D	N	D	N	50
User 7	SA	D	A	D	A	SD	A	D	A	D	80
User 8	D	D	D	A	N	D	SD	D	SD	SA	35
User 9	N	A	D	A	A	D	N	N	D	A	42.5
User 10	A	SD	SA	N	A	D	N	D	N	A	67.5
User 11	D	A	D	SA	N	D	N	N	D	A	35
User 12	N	D	N	N	A	SD	D	N	A	N	60
Overall SUS Score:											50

SA Strongly Agree
A Agree
N Neither Agree or Disagree
D Disagree
SD Strongly Disagree

Table C.12: Task questionnaire

Q1. I found it easy to complete the mapping tasks.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q2. I found the tool helpful in completing the mapping tasks.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q3. I found it easy to navigate the user models using this tool.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q4. I found it easy to identify mappings using this tool.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q5. I found it easy to create mappings using this tool.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q6. I found it easy to test the mappings using this tool.				
Strongly Agree	Agree	Neither Agree or Disagree	Disagree	Strongly Disagree
Q7. What aspects of the mapping task did you find the most difficult?				
Q8. What aspects of the tool did you like?				
Q9. What aspects of the tool did you not like?				
Q10. Are there any new tool features you think might assist you in the mapping process?				

Table C.13: Participant responses to questions 1-6 of task questionnaire

Users	Mapping Tool					
	Q1	Q2	Q3	Q4	Q5	Q6
User 1	D	A	A	N	A	A
User 2	A	SA	SA	SA	A	SA
User 3	A	A	A	A	A	A
User 4	A	A	A	D	A	SA
User 5	N	A	SA	SA	A	SA
User 6	A	SA	A	A	A	SA
User 7	A	A	A	N	A	A
User 8	A	SA	SA	N	N	SA
User 9	A	SA	A	A	SA	SA
User 10	A	SA	SA	A	A	SA
User 11	N	D	A	A	N	N
User 12	A	A	A	A	A	A

Users	Mapforce					
	Q1	Q2	Q3	Q4	Q5	Q6
User 1	D	N	A	D	D	A
User 2	A	A	D	A	N	A
User 3	D	D	A	A	D	A
User 4	N	A	N	A	D	A
User 5	D	N	N	D	N	A
User 6	D	A	D	D	N	A
User 7	A	A	A	N	A	A
User 8	SD	D	SD	D	SD	D
User 9	D	N	A	A	D	N
User 10	A	A	SD	A	A	SA
User 11	D	D	N	A	D	D
User 12	N	A	N	N	N	A

SA Strongly Agree
A Agree
N Neither Agree or Disagree
D Disagree
SD Strongly Disagree

Table C.14: Paired t-tests for questions 1-6 of task questionnaire

User	Task Questionnaire					
	Q1 Diff	Q2 Diff	Q3 Diff	Q4 Diff	Q5 Diff	Q6 Diff
User 1	0	1	0	1	2	0
User 2	0	1	3	1	1	1
User 3	2	2	0	0	2	0
User 4	1	0	1	-2	2	1
User 5	1	1	2	3	1	1
User 6	2	1	2	2	1	1
User 7	0	0	0	0	0	0
User 8	3	3	4	1	2	3
User 9	2	2	0	0	3	2
User 10	0	1	4	0	0	0
User 11	1	0	1	0	1	1
User 12	1	0	1	1	1	0

Paired t-tests

Mean	1.08333333	1.00000000	1.50000000	0.58333333	1.33333333	0.83333333
Standard Deviation	0.99620492	0.95346259	1.50755672	1.24011241	0.88762536	0.93743687
Standard Error	0.28757959	0.27524094	0.43519414	0.35798962	0.25623537	0.27061471
Test Statistic	3.76707309	3.63318042	3.44673759	1.62946998	5.20354908	3.07940881
Degrees of Freedom	11	11	11	11	11	11
p-value	0.00311617	0.00393471	0.00545938	0.13148859	0.00029287	0.01048236

Table C.15: Participant responses to question 7 of task questionnaire

Mapping Tool	
Users	“What aspects of the mapping task did you find the most difficult?”
User 1	
User 2	To choose the correct type of mapping.
User 3	If the task/mapping has more complex conditions it seems hard to transfer them to the tool.
User 4	Mapping the XML nodes in the XML structure to the generic concepts provided by the tool.
User 5	I was not sure how to map the last task.
User 6	Task 3.
User 7	
User 8	More complex tasks. I was unsure of the correct filters to use to achieve my goals.
User 9	In general, lack of previous knowledge of the content and structure of the XML files makes it hard to identify the correct and logical mappings.
User 10	The concept of specifying the exact values to map, as opposed to all values.
User 11	I found the testing aspect the most difficult as I was unsure what exactly to look for.
User 12	The part that requires familiarity with terminology in this domain.
Mapforce	
Users	“What aspects of the mapping task did you find the most difficult?”
User 1	Mapping things that did not have an obvious equivalence i.e. mapping strings to ints or mapping booleans to integer ranges.
User 2	To compose different mapping components in order to achieve certain constraints.
User 3	Using multiple logical operators and using string tokenizers.
User 4	Checking existence of elements and their values before mapping them across.
User 5	A lot of functions which I found difficult to use.
User 6	Complex expressions.
User 7	Mapping components.
User 8	Anything beyond a simple mapping - unsure where to store variables as processing the complex mappings. Although it is nice to have a visual tool - the draggable components are basically a limited selection of code. I would prefer to code myself instead of be limited in this way.
User 9	Figuring out the tag structures and content.
User 10	Understanding the relationship between the nodes and elements.
User 11	Essentially any mapping beyond a simple linking of two equivalent concepts.
User 12	Combination of operators.

Table C.16: Participant responses to question 8 of task questionnaire

Mapping Tool	
“What aspects of the tool did you like?”	
User 1	
User 2	Use of graph and tree to display user model.
User 3	Easy to create mappings in a focused domain.
User 4	The filters were quick and easy to use and understand.
User 5	The interface was easy to use and I liked the functionality for selecting parts of text from the XML values.
User 6	Predefined mappings construction of mappings based on permanent model approach.
User 7	
User 8	Nice interface, intuitive and a lot nicer than having to generate code to accomplish these mapping tasks.
User 9	Good GUI, very straightforward, and clear. It was easy to use. Also, the part where I chose the functions/conditions from the list was very helpful and helped me choose the correct function easily and quickly.
User 10	The XML viewer, the common model and the specific mapping classes.
User 11	The graphical interface is very straightforward and easy to use. The labels for the mappings are very clear.
User 12	The GUI.
Mapforce	
“What aspects of the tool did you like?”	
User 1	It was easy enough to navigate.
User 2	A lot of types for mapping.
User 3	Graph was well designed and easy to navigate on.
User 4	Simple connections and logical operators.
User 5	Drag and drop of mappings is useful.
User 6	Visual components, drag and drop.
User 7	
User 8	Potential for visual tool.
User 9	The tooltips that indicate what the component does.
User 10	The hover over “tips” which explained the components. The testing feature and pop-up explanation (for example testing multiple outcomes).
User 11	The ability to link concepts using the mouse makes the interface somewhat intuitive.
User 12	The concept of wiring components together.

Table C.17: Participant responses to question 9 of task questionnaire

Mapping Tool	
“What aspects of the tool did you not like?”	
User 1	
User 2	
User 3	Maybe the view of the data should not be in testing but in a side tab.
User 4	Mapping the XML nodes in the XML structure to the generic concepts provided by the tool.
User 5	The go back to the user model button wasn't immediately obvious.
User 6	Improve navigation for nested elements.
User 7	
User 8	I would like if the tool provided more feedback - tooltips, invitations to create mappings.
User 9	I would like the tool to allow a tree navigation style wherever it is going to display the content of an XML file. In other words, whenever I see XML text in front of me I would like to be able to collapse and expand nodes.
User 10	Took a bit of time to get my head around the concept of the common model but once understood it made it a lot easier to model.
User 11	Moving the individual concepts in the interface can be a bit difficult.
User 12	Testing is not so responsive.
Mapforce	
“What aspects of the tool did you not like?”	
User 1	Cluttered interface when some mapping had already been done, not much feedback until the end.
User 2	To compose different mapping components in order to achieve certain constraints.
User 3	Logical concepts were too complicated to use. Not enough space.
User 4	Complex operators.
User 5	Explanations for functions could use more illustrative examples.
User 6	Element navigation, creating complex expressions visually complex.
User 7	
User 8	Limited programming-like nodes.
User 9	Having to figure out which components to use and how to stick them together. I am a programmer and I am very oriented towards writing code (in text) to do that, instead of using visual components. Also, the interface got so cluttered with lines and components as I created more mappings.
User 10	The representation of the XML models.
User 11	The graphical representation becomes cluttered quite quickly. Adding conditions to mappings can be quite difficult. Testing mappings can also be quite difficult.
User 12	

Table C.18: Participant responses to question 10 of task questionnaire

Mapping Tool	
“Are there any new tool features you think might assist you in the mapping process?”	
User 1	
User 2	Double click on the mapping type to select.
User 3	More dynamic showing of what effect my mappings have. Instead of clicking testing, more on the fly.
User 4	Possibly have a representation for each XML file where mappings can be similar to Mapforce.
User 5	
User 6	Selecting/viewing the mapping path using a tree based structure tree based view of permanent model.
User 7	
User 8	
User 9	
User 10	Some tooltips/mouse overs.
User 11	In the testing process some feedback as to potential sources of error could be useful.
User 12	Tooltips.
Mapforce	
“Are there any new tool features you think might assist you in the mapping process?”	
User 1	
User 2	
User 3	Suggestions and blending out things that cannot be done when clicking on a logical operator.
User 4	More operators.
User 5	
User 6	
User 7	
User 8	Better sample output and access to the user model data.
User 9	Maybe allow me to enter the mappings in a text-based manner (like programming code).
User 10	Better visualisation of the XML and also some example solutions (using each of the components).
User 11	A somewhat different graphical representation that doesn’t become cluttered quite so easily.
User 12	Examples of useful combinations.

Table C.19: Comparison questionnaire

- Q1. Please state which tool's mapping approach you found easier to use and why?
 Tool A (i.e. mapping all user models to the permanent user model)
 Tool B (i.e. direct mapping between all user models)
- Q2. Please state which tool you found it easier to create mappings in and why?
 Tool A (i.e. the tool with mapping to the permanent user model)
 Tool B (i.e. the tool with direct mapping between the user models)

Table C.20: Participant responses to question 1 of comparison questionnaire

“Please state which tool’s mapping approach you found easier to use and why? Tool A (i.e. mapping all user models to the permanent user model) Tool B (i.e. direct mapping between all user models)”		
User 1	Tool A	It was easier to map through the centralised model, I got less lost/confused by doing this step.
User 2	Tool B	Direct mapping is more intuitive for novice user. But a canonical model could be easier in managing mapping when the number of models is growing large.
User 3	Tool A	It was easy and more intuitive but again it depends on the domain and in a more open/complex domain Tool B may be more useful.
User 4	Tool B	With Mapforce I was confident of mapping the correct nodes and values.
User 5	Tool A	I found it more appropriate for the task at hand.
User 6	Tool A	Because the mappings is always based on a common model.
User 7	Tool A	
User 8	Tool A	Better to see an overview of the types of connections between the user models - a general model is beneficial and should help future groups of people working on mappings to agree on the same approach when mapping.
User 9	Tool A	
User 10	Tool A	I like the permanent user model but wonder who created it, was it automatic or is a domain expert required?
User 11	Tool B	The approach of directly mapping concepts from different models to each other is more intuitive than mapping two equivalent concepts to another on a different model.
User 12	Tool B	Wiring in Tool B seems more logical in my opinion if the user was familiar with combination of operators.

Tool A Mapping Tool
 Tool B Mapforce

Table C.21: Participant responses to question 2 of comparison questionnaire

“Please state which tool you found it easier to create mappings in and why? Tool A (i.e. the tool with mapping to the permanent user model) Tool B (i.e. the tool with direct mapping between the user models)”		
User 1	Tool A	Because it was easier to use the predefined mapping types.
User 2	Tool A	
User 3	Tool A	Easier to use and more intuitive.
User 4	Tool A	I preferred the mapping types provided by Tool A.
User 5	Tool A	It was more focused on creating user mappings and I was able to do the tasks with greater ease.
User 6	Tool A	Tool A because of the predefined mapping types and better navigation.
User 7	Tool A	Permanent user model because of its tree structure is more useful as while mapping you know you are at the right node.
User 8	Tool A	Better idea of where the data was and where it was going. Tool B was too much like a restricted visual representative of a programming language.
User 9	Tool A	
User 10	Tool A	It is easier to think about a common model which links two models than it is to directly link model A and B.
User 11	Tool A	Graphical interface in Tool A was easier to work with. Complex mappings were also easier to create using Tool A.
User 12	Tool A	In this case, Tool A comes with predefined operations which suit the tasks.

Tool A Mapping Tool
Tool B Mapforce

Table C.22: Number/complexity of mappings

Mappings	Type 1	Type 2	Type 3	Type 4
1	127.72	146.33	120.32	115.96
10	184.50	193.74	274.74	206.80
20	312.82	297.86	433.34	396.30
30	368.57	361.78	649.86	482.89
40	400.45	406.57	739.43	571.34
50	416.74	417.18	920.39	715.62
60	539.91	512.15	1109.75	813.91
70	585.06	545.77	1218.56	987.68
80	650.70	649.83	1397.00	1066.97
90	688.67	677.88	1507.27	1130.92
100	770.17	751.56	1664.94	1249.85

Table C.23: Number of exchanging user models

User Models	Target	Canonical
1	140.87	104.16
2	156.53	105.86
3	170.12	115.14
4	176.06	135.52
5	188.31	139.04
6	194.00	145.16
7	204.55	158.02
8	219.21	164.02
9	238.35	177.95
10	252.57	187.66

Table C.24: Size of user models

User Model Elements	Type 1
1	135.46
1000	180.47
2000	216.76
3000	326.61
4000	350.77
5000	418.05
6000	440.53
7000	535.66
8000	571.53
9000	575.72
10000	639.67

Table C.25: Percentage of concurrent user model requests processed

	1	25	50	75	100
Processed (%)	Time (ms)	Time (ms)	Time (ms)	Time (ms)	Time (ms)
1	169.33	1065.65	1145.20	3389.19	3410.81
2	169.33	1065.65	1149.41	3389.21	3538.94
3	169.33	1065.91	1149.41	3389.22	3545.75
4	169.33	1065.91	1829.45	3389.22	3549.41
5	169.33	1065.91	1829.45	3389.26	3549.47
6	169.33	1065.91	1829.58	3389.27	4917.73
7	169.33	1066.40	1829.58	3389.28	5140.07
8	169.33	1066.40	1830.28	3389.28	5346.68
9	169.33	1066.40	1830.28	3389.32	5347.36
10	169.33	1066.40	1830.37	3391.38	5348.38
11	169.33	1066.62	1830.37	3392.00	5348.50
12	169.33	1066.62	1831.52	3392.00	5350.78
13	169.33	1066.62	1831.52	3392.45	5350.85
14	169.33	1066.62	1831.53	3392.51	6218.10
15	169.33	1066.69	1831.53	3392.95	6219.48
16	169.33	1066.69	2229.18	3392.95	6219.71
17	169.33	1066.69	2229.18	3393.12	6220.34
18	169.33	1066.69	2229.26	3393.21	6220.65
19	169.33	1066.82	2229.26	3393.50	6221.43
20	169.33	1066.82	2229.50	3393.50	6751.00
21	169.33	1066.82	2229.50	3393.74	6751.16
22	169.33	1066.82	2237.55	3394.08	6751.50
23	169.33	1067.13	2237.55	3394.09	6751.77
24	169.33	1067.13	2237.70	3394.09	6751.94
25	169.33	1067.13	2237.70	3394.13	6752.10
26	169.33	1067.13	2237.78	3394.18	6752.22
27	169.33	1067.14	2237.78	3394.31	6752.23
28	169.33	1067.14	2238.14	3394.31	6752.29
29	169.33	1067.14	2238.14	3394.46	6752.36
30	169.33	1067.14	2238.27	3394.47	6752.50
31	169.33	1067.32	2238.27	3394.49	6752.58
32	169.33	1067.32	2238.55	3394.49	6752.72
33	169.33	1067.32	2238.55	3394.59	6752.79
34	169.33	1067.32	2238.90	3394.74	6752.84
35	169.33	1067.37	2238.90	3394.84	6752.89
36	169.33	1067.37	2238.99	3394.84	6752.92
37	169.33	1067.37	2238.99	3394.87	6752.94
38	169.33	1067.37	2239.39	3395.02	6752.99
39	169.33	1067.38	2239.39	3395.09	6753.01
40	169.33	1067.38	2239.42	3395.09	6753.02
41	169.33	1067.38	2239.42	3395.40	6753.05
42	169.33	1067.38	2239.44	3520.66	6753.16
43	169.33	1067.42	2239.44	3522.49	6753.19
44	169.33	1067.42	2239.74	3522.49	6753.29
45	169.33	1067.42	2239.74	3747.45	6753.31

46	169.33	1067.42	2525.30	3747.66	6753.33
47	169.33	1067.44	2525.30	3747.76	6753.36
48	169.33	1067.44	2525.84	3747.76	6753.49
49	169.33	1067.44	2525.84	3747.96	6753.51
50	169.33	1067.44	2526.03	3748.02	6753.58
51	169.33	1072.68	2526.03	3748.12	6753.61
52	169.33	1072.68	2526.08	3748.12	6753.61
53	169.33	1072.68	2526.08	3748.45	6753.63
54	169.33	1072.68	2526.13	3748.80	6753.66
55	169.33	1122.25	2526.13	3748.88	6753.72
56	169.33	1122.25	2526.18	3748.88	6753.74
57	169.33	1122.25	2526.18	3748.94	6753.89
58	169.33	1122.25	2526.21	3749.12	6753.99
59	169.33	1123.75	2526.21	3749.55	6754.01
60	169.33	1123.75	2526.30	3749.55	6754.12
61	169.33	1123.75	2526.30	3749.94	6754.19
62	169.33	1123.75	2526.69	3750.01	6754.25
63	169.33	1124.18	2526.69	3787.82	6754.49
64	169.33	1124.18	2527.17	3787.82	6754.58
65	169.33	1124.18	2527.17	3788.26	6754.61
66	169.33	1124.18	2627.62	3857.15	6754.61
67	169.33	1133.69	2627.62	3857.40	6754.69
68	169.33	1133.69	2628.24	3857.40	6754.76
69	169.33	1133.69	2628.24	3858.25	6754.87
70	169.33	1133.69	2629.07	3860.09	6754.88
71	169.33	1134.30	2629.07	4294.17	6754.93
72	169.33	1134.30	2629.58	4294.17	6755.02
73	169.33	1134.30	2629.58	4295.58	6755.06
74	169.33	1134.30	2657.97	4415.80	6755.27
75	169.33	1134.55	2657.97	4416.00	6755.45
76	169.33	1134.55	2658.35	4416.00	6755.47
77	169.33	1134.55	2658.35	4416.01	6755.75
78	169.33	1134.55	2658.39	4416.41	6755.84
79	169.33	1134.71	2658.39	4416.68	6755.88
80	169.33	1134.71	2658.69	4416.68	6755.98
81	169.33	1134.71	2658.69	4417.09	6756.01
82	169.33	1134.71	2658.82	4417.58	6756.32
83	169.33	1135.32	2658.82	4418.09	6756.33
84	169.33	1135.32	2659.08	4418.09	6756.50
85	169.33	1135.32	2659.08	4507.99	6872.83
86	169.33	1135.32	2659.20	4508.27	6873.03
87	169.33	1136.29	2659.20	4508.44	6873.92
88	169.33	1136.29	2659.24	4508.44	6873.98
89	169.33	1136.29	2659.24	4508.51	6875.21
90	169.33	1136.29	2659.62	4508.93	6875.32
91	169.33	1147.26	2659.62	4509.08	6875.41
92	169.33	1147.26	2665.75	4509.08	6875.46
93	169.33	1147.26	2665.75	4509.12	6875.79
94	169.33	1147.26	2667.93	4510.62	6876.62
95	169.33	1152.95	2667.93	4510.74	6876.98
96	169.33	1152.95	2670.06	4510.74	6885.02
97	169.33	1152.95	2670.06	4510.83	6890.42
98	169.33	1152.95	2682.08	4510.86	6904.25
99	169.33	1152.95	2682.08	4515.80	6935.96
100	169.33	1152.95	2682.08	4515.80	7080.20