

The OISIN framework: Ontology Interoperability in support of Semantic Interoperability

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

Declan O'Sullivan
Knowledge and Data Engineering Group,
Department of Computer Science,
Trinity College,
Dublin

2006

Declaration

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

Declan O'Sullivan

March 2006

Permission to lend or copy

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

Declan O'Sullivan

March 2006

ACKNOWLEDGEMENTS

I would like to thank my supervisor Vincent Wade for agreeing to supervise my research, for his unflagging support and insightful contributions. I would also like to thank Jane Grimson for first nurturing my interest in research so many years ago. My thanks to Dave Lewis, Owen Conlan and all in the KDEG research group for the great collaboration and friendship. In addition, thanks to Elaine Hanley for the proof reading/feedback and James Reilly for his help with my statistical analysis.

I would like to especially thank Diane Birnie for her absolute belief, constant encouragement, unwavering support and the many sacrifices. A special word of thanks to my brother Ciaran for his unfailing confidence, and continued interest in this research.

Finally, thanks are due to my two sons, Oisín and Eoin, for being wonderful reminders that there is more to life than research!

This thesis is dedicated to the memory of Kay, who indirectly made it all possible.

ABSTRACT

The promise of ontologies is in the sharing of an understanding of a domain that can be communicated between people and application systems (Fensel 2003). However, different ontologies arise due to the natural human diversity involved in modelling a domain. In other words, ontologies are normally created from a particular perspective. If the applications of an individual or company that use an ontology, need to interact with applications of other individuals or organisations that use different ontologies, there will be a need to map between the multiple ontologies either at inter-personal, intra-organisational or inter-organisational levels (Hameed et al. 2004). It is envisaged that such ontology mapping will be beneficial in a wide range of scenarios that require support for semantic interoperability, including: web service composition, distributed information querying, e-commerce catalogue matching, and personalisation of information. However, in order for the use of such mappings to gain widespread acceptance, a practical ontology mapping lifecycle needs to be put in place. In addition, given the potential importance of such ontology mappings to organisations, there is a need for such mappings to be created as part of some engineered process rather than as a result of *ad hoc* activities. This thesis identifies the activities, information artefacts, heuristics and guidance required to support a practical ontology mapping process, and examines how they can be coordinated and sequenced in the ontology mapping process. The thesis proposes the OISIN framework comprising an ontology mapping process supported by software tools. The framework: supports a full ontology mapping lifecycle; is extensible and adaptable; and supports the creation of “relevant mappings”, ranging from supporting a user’s determination of mappings to an application’s determination of mappings. The thesis details the specification of the ontology mapping process, describes the design and implementation of the supporting software tools, and presents the evaluation of the process and tools.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	III
ABSTRACT	IV
TABLE OF CONTENTS	V
TABLE OF FIGURES.....	IX
TABLE OF TABLES.....	XII
ABBREVIATIONS	XIV
1 INTRODUCTION	1
1.1 MOTIVATION	1
1.2 OBJECTIVE AND GOALS	4
1.3 CONTRIBUTION	5
1.4 TECHNICAL OVERVIEW	6
1.4.1 <i>Technical Approach</i>	6
1.4.2 <i>Thesis Overview</i>	7
2 BACKGROUND – SEMANTIC INTEROPERABILITY	9
3 STATE OF THE ART – ONTOLOGY MAPPING	15
3.1 “VISITING LECTURER” SCENARIO	15
3.2 OVERVIEW OF ONTOLOGY MAPPING	17
3.2.1 <i>Characterising Ontology Mismatches</i>	18
3.2.2 <i>Mapping discovery</i>	21
3.2.3 <i>Mapping Documentation and Interpretation</i>	28
3.3 KEY REQUIREMENTS	30
3.3.1 <i>Key Framework Challenges</i>	30
3.3.2 <i>Key Framework Requirements</i>	32
3.4 SELECTED STATE OF THE ART SYSTEMS	34
3.4.1 <i>MAFRA</i>	35
3.4.2 <i>COMA++</i>	39
3.4.3 <i>INRIA ontology alignment format</i>	42
3.4.4 <i>Protégé and PROMPT/Anchor</i>	44
3.5 KEY REQUIREMENTS AND SELECTED STATE OF THE ART SYSTEMS	47
3.6 IMPLEMENTATION TECHNOLOGIES	50
4 DESIGN	52
4.1 GENESIS OF THE PHASES.....	52
4.2 THE OISIN ONTOLOGY MAPPING PROCESS	55

4.2.1	<i>Characterisation Phase Activities</i>	58
4.2.2	<i>Mapping Phase Activities</i>	65
4.2.3	<i>Execution Phase Activities</i>	68
4.2.4	<i>Management Phase Activities</i>	71
5	TOOL IMPLEMENTATION	73
5.1	THE OISIN APPLICATIONS	73
5.1.1	<i>Information Model</i>	78
5.1.2	<i>Implementation technologies</i>	78
5.1.3	<i>Discover Ontologies – OisinDO application</i>	79
5.1.4	<i>Characterise Modelling – part of OisinIO application</i>	82
5.1.5	<i>Characterise Semantics – part of OisinIO application</i>	88
5.1.6	<i>Decide to Match, Select and Apply Match Algorithms – OisinMO application</i>	90
5.1.7	<i>Characterise Amenability – OisinSummarise application</i>	98
5.1.8	<i>Decide to Map – OisinDTM Application(s)</i>	99
5.1.9	<i>Identify Committed Mappings – OisinAC application(s)</i>	107
5.1.10	<i>Undertake Commitment Analysis – OisinMA application(s)</i>	110
5.1.11	<i>Interpret Mappings – OisinSMU service</i>	117
5.2	GRAPHICAL USER INTERFACE WALK THROUGH FOR ANCHOR CAPTURE.....	120
5.2.1	<i>OisinAC-semi-v1 walk through and workflow</i>	120
5.2.2	<i>OisinAC-semi-v2</i>	126
6	EVALUATION	128
6.1	INTRODUCTION	128
6.2	DECIDE TO MAP ACTIVITY EXPERIMENT	131
6.2.1	<i>Overview</i>	131
6.2.2	<i>Decisions based on examining Ontology directly</i>	133
6.2.3	<i>Decisions based on OISIN generated information</i>	137
6.2.4	<i>Analysis of Experiment Results</i>	147
6.3	COMMITTED MAPPING ACTIVITIES EXPERIMENT	151
6.3.1	<i>Overview</i>	152
6.3.2	<i>Mappings with support from Protégé/PROMPT tool</i>	155
6.3.3	<i>Difficulties Questionnaire</i>	157
6.3.4	<i>Mappings using the OISIN tools</i>	159
6.3.5	<i>Mappings Analysis</i>	161
6.3.6	<i>OISIN tools usability analysis</i>	163
6.3.7	<i>Discussion and Findings</i>	169
7	TRIAL AND RELATED WORK.....	172
7.1	INTRODUCTION	172
7.2	INTERPRET MAPPING ACTIVITY TRIAL	172

7.2.1	Overview.....	172
7.2.2	Applications.....	173
7.2.3	Trial Setup.....	177
7.2.4	Results.....	178
7.3	COMPARISON OF OISIN TO RELATED WORK.....	183
8	CONCLUSIONS	186
8.1	OBJECTIVES.....	186
8.2	CONTRIBUTION	193
8.3	FUTURE WORK.....	195
8.3.1	Extended OisIN (EOIN).....	195
8.3.2	Management Phase Activities	196
8.3.3	New application areas	196
8.4	FINAL REMARKS	197
	BIBLIOGRAPHY	198
	GLOSSARY	214
	APPENDICES.....	215
	APPENDIX A. TECHNOLOGIES.....	A-1
1.	IPSI-XQ.....	A-1
2.	Jena	A-3
3.	Wordnet/JWNL	A-4
4.	SWOOGLE.....	A-6
	APPENDIX B. UML SPECIFICATION OF OISIN ONTOLOGY MAPPING PROCESS	B-1
1.	Use Case Diagram	B-1
2.	Activity Diagrams.....	B-4
3.	Extensible Information Model.....	B-12
	APPENDIX C. XML DTDS.....	C-1
1.	ANC.dtd – Anchors/Committed Mappings	C-1
2.	CAN.dtd – Canonical Format.....	C-1
3.	CMB.dtd – Candidate Matches	C-1
4.	DIM.dtd – Dimensions Analysis.....	C-2
5.	DOM.dtd – Quality Analysis	C-2
6.	INT.dtd - Candidate Matches plus Anchors	C-3
7.	LANG.dtd – Language Analysis	C-3
8.	LEX.dtd – Term Analysis	C-3
9.	MA.dtd – Mapping Analysis	C-4
10.	MERLIN.dtd – WordNet analysis detail	C-5

11. <i>WNET.dtd – WordNet analysis statistics</i>	C-5
APPENDIX D. DECIDE TO MAP EXPERIMENT	D-1
1. <i>Decision Sheet</i>	D-1
2. <i>Excel Spreadsheet for AB Mapping</i>	D-2
3. <i>Decision Guidance Sheet</i>	D-6
4. <i>Task Questionnaire</i>	D-7
5. <i>MINITAB analysis output</i>	D-8
APPENDIX E. MAPPING EXPERIMENT	E-1
1. <i>First Task Questionnaire</i>	E-1
2. <i>Second Task Questionnaire</i>	E-3
APPENDIX F. SMU TRIAL	F-1
1. <i>Example XML instance document based on Ontology “C”</i>	F-1
2. <i>Example Queries from Group 3</i>	F-8

TABLE OF FIGURES

Figure 3-1: Partial is-a hierarchies of scenario ontologies	16
Figure 3.2: Classification of Ontology Mismatches (Klein 2001).....	19
Figure 3-3: Classification of schema-based matching approaches (Shvaiko and Euzenat 2004).....	23
Figure 3-4: Overview of MAFRA conceptual framework (Maedche et al. 2002).....	36
Figure 3-5: MAFRA plug-in architecture (Silva and Rocha 2003)	38
Figure 3-6: Overview of COMA++ Architecture (Aumueeller et al. 2005)	40
Figure 3-7: Overview of iterations in COMA++ Execution Engine (Aumueeller et al. 2005)	40
Figure 3-8: PROMPT algorithm overview	45
Figure 3-9: Example of anchor paths	45
Figure 4-1: Summary UML Activity diagram for OISIN process.....	57
Figure 4-2: UML Class Diagram for extensible information model of OISIN	58
Figure 5-1: UML package diagram for OISIN applications.....	74
Figure 5-2: OISIN applications/process relationship	75
Figure 5-3: Overview of tools for first implementation deployment	77
Figure 5-4: UML Sequence Diagram between Oisin applications/IPSI interface/Information Model	79
Figure 5-5: UML class diagram for ontology entity	80
Figure 5-6: UML Class diagram for Modelling Analysis	83
Figure 5-7: Snippet of Detect Pattern XQuery.....	84
Figure 5-8: Snippet of Ontology Shape XQuery.....	86
Figure 5-9: UML Class Diagram for Semantic Analysis	89
Figure 5-10: UML Class Diagram for Candidate Matches.....	91
Figure 5-11: Snippet of Term Match XQuery	94
Figure 5-12: Example of XML for a class match.....	95
Figure 5-13: Example of XML for a property match	96
Figure 5-14: Example page 1 of OisinDTM excel spreadsheet	100
Figure 5-15: Example page 2 of OisinDTM excel spreadsheet	101
Figure 5-16: Example page 3 of OisinDTM excel spreadsheet	102
Figure 5-17: Example page 4 of OisinDTM excel spreadsheet	103

Figure 5-18: Snippet of OisinDTM-auto XQuery.....	105
Figure 5-19: UML Class Diagram for OisinAC-semi-v1 application.....	108
Figure 5-20: Snippet of XQuery to retrieve all anchor paths.....	112
Figure 5-21: Example of ontology structural mismatch.....	112
Figure 5-22: Snippet of XQuery to implement anchor analysis	114
Figure 5-23: UML Class Diagram for Committed Analysis	115
Figure 5-24: Example OisinMA window presenting Mapping Analysis	116
Figure 5-25: Snippet of OisinSMU application XQuery.....	119
Figure 5-26: Main window of OisinAC-semi application.....	120
Figure 5-27: Overview of toolbar of OisinAC-semi	121
Figure 5-28: Overview of Anchor Capture Panel	122
Figure 5-29: Example of Mapping Details window.....	123
Figure 5-30: Anchor Capture Workflow	125
Figure 5-31: OisinAC-semi-v2 screenshot	126
Figure 6-1: Chart of Results for Protégé supported decisions	136
Figure 6-2: Chart of Results for spreadsheet supported decisions	141
Figure 6-3: Chart of Results for guidance supported decisions	144
Figure 6-4: Overall comparison of decisions.....	147
Figure 6-5: ANOVA analysis	149
Figure 6-6: Experiment mapping DTD	155
Figure 6-7: Analysis of Incorrect Mappings created.....	161
Figure 6-8: Analysis of Gold Mappings missed	162
Figure 6-9: Analysis of performance of groups in undertaking ontology mapping...	163
Figure 6-10: Opinions on OISIN Mapping Tools usability.....	167
Figure 7-1: XQuery used to invoke OisinSMU	177
Figure 7-2: Breakdown of Mapping Information kept for Class related terms	180
Figure 7-3: Breakdown of Mapping Information kept for Property related terms	180
Figure 7-4: Overall success rate	181
Figure A-1: Overview of IPSI-XQ Execution	A-2
Figure A-2: Jena model decomposition.....	A-4
Figure A-3: SWD Discovery phase of SWOOGLE architecture.....	A-7
Figure B-1: Use Case Diagram	B-2
Figure B-2: Activity Diagram for Characterisation Phase	B-5
Figure B-3: Activity Diagram for Mapping Phase.....	B-8

Figure B-4: Activity Diagram for Execution Phase	B-10
Figure B-5: Activity Diagram for Management Phase.....	B-11
Figure B-6: UML Class Diagram of Entities	B-12
Figure B-7: UML Class Diagram for Behaviour Governance Entities	B-14

TABLE OF TABLES

Table 3-1: Comparison of Selected SOA systems wrt key requirements.....	48
Table 5-1: XQuery functions developed.....	78
Table 6-1: Overview of Experiment Ontologies.....	131
Table 6-2: Gold-standard mapping decisions	132
Table 6-3: Summary of Decision Sheets for Protégé supported Students 1 to 5	134
Table 6-4: Summary of Decision Sheets for Protégé supported Students 6 to 10	135
Table 6-5: Overview of OISIN generated Excel spreadsheet	138
Table 6-6: Summary of Decision Sheets for spreadsheet supported Students 1 to 5.....	139
Table 6-7: Summary of Decision Sheets for spreadsheet supported students 6 to 10	140
Table 6-8: Decisions made with guidance sheet	143
Table 6-9: Opinions on Confidence in decisions and Difficulty of decision making	145
Table 6-10: Opinions on guidance	145
Table 6-11: Opinions on unhelpful guidance or information	146
Table 6-12: Suggested improvements to guidance and information.....	147
Table 6-13: Gold-standard mappings for AB and CD ontology pairs.....	154
Table 6-14: AB ontology mappings created manually by experiment groups	156
Table 6-15: CD ontology mappings created manually by experiment groups	157
Table 6-16: Summary of mapping task difficulties experienced by experiment groups	158
Table 6-17: Group comments about usefulness of Protégé/Anchor-PROMPT	159
Table 6-18: AB ontology mappings created by experiment groups using OISIN	160
Table 6-19: CD ontology mappings created by experiment groups using OISIN	160
Table 6-20: Opinions on usefulness of OISIN tools for ontology mapping.....	164
Table 6-21: Opinions on whether OISIN alleviated difficulties	165
Table 6-22: Opinions on OISIN versus Protégé/Anchor-PROMPT	166
Table 6-23: Opinions for what parts of mapping task OISIN tools useful	168
Table 6-24: Hours spent on each mapping task	169
Table 7-1: Terms extracted from Student XML instance documents	174
Table 7-2: Summary of Student Applications.....	175
Table 7-3: Query terms grouped by application	176

Table 7-4: Analysis of Terms with respect to Gold-standard Mapping Information.	179
Table 7-5: Comparison of OISIN with selected state of art systems	183
Table A-1: Jena accessor methods	A-3
Table B-1: Actors/Roles for OISIN.....	B-1
Table B-2: Activity Descriptions for Characterisation Phase.....	B-7
Table B-3: Activity Descriptions for Mapping Phase	B-9
Table B-4: Activity Descriptions for Execution Phase	B-10
Table B-5: Activity Descriptions for Management Phase	B-11
Table B-6: Description of Signals used in Activity Diagrams.....	B-17
Table B-7: Description of Events used in Activity Diagrams	B-17

ABBREVIATIONS

API	Application Programmers Interface
CORBA	Common Object Request Broker Architecture
DAML	DARPA Agent Markup Language
DCOM	Distributed Component Object Model
DOM	Document Object Model
DTD	Document Type Definition
EOIN	Extended OisIN
ER	Entity Relationship
eTOM	electronic Telecoms Operation Map
HTML	Hypertext Markup Language
IDL	Interface Definition Language
JDBC	Java Database Connectivity
JWNL	Java WordNet Library
ODBC	Object Database Connectivity
ODP	Open Distributed Processing
OIL	Ontology Inference Layer
OISIN	Ontology Interoperability for Semantic INteroperability
OMG	Open Management Group
OWL	Web Ontology Language
OWL-S	Web Services Ontology Language
P2P	Peer to Peer
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RMI	Remote Method Invocation
RUP	Rational Unified Process
SAX	Simple API for XML
SQL	Structured Query Language
SWRL	Semantic Web Rule Language
TMF	Telecommunications Management Forum
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
WSMO	Web Services Modelling Ontology
XML	eXtensible Markup Language
XSLT	XML Stylesheet Transformation

1 INTRODUCTION

1.1 Motivation

The rapid rise in the usage and sophistication of web content and services has led to the desire to make such content and services more discoverable and usable by applications at runtime. A key problem in meeting this desire is how to expose the semantics (meaning) of the web content or web service interfaces in a manner that is widely interpretable by applications. Significant progress has been made in addressing this problem over the last number of years, with proposals to express the semantics of information and web service interfaces using languages based on XML, the eXtensible Markup Language (XML 2000). However a more difficult problem remains arising from the different modelling of semantics. This problem relates to how to achieve mappings between the different semantic models used in the semantic descriptions. Different semantic models arise due to the natural human diversity involved in modelling a domain. Semantic models are normally created from a particular perspective and for different purposes. The motivation for this thesis was to identify a practical process for the creation of such semantic model mappings to support semantic interoperability between applications.

Early attempts to achieve the exposure of semantics through the use of the XML based languages Resource Description Framework (RDF 2004) and Resource Description Framework Schema (RDFS 2004) have had some success. However these approaches have been limited in their basic expressivity and in their ability to express mappings between different semantic models. To overcome these deficiencies, ontologies are increasingly being used as the means to express semantics.

An ontology is a formal, explicit specification of a shared conceptualization (Gruber 1993). A **conceptualization** in this definition, refers to an abstract model of how people think about things in the world, usually from a particular perspective. An **explicit specification** means that the concepts and relationships in the abstract model are assigned explicit names and definitions. The name is a word (hereafter referred to as term) and the definition is a specification of the meaning of the concept or relation.

An ontology can range from the simple notion of a taxonomy (knowledge with minimal parent/child structure) to a thesaurus (words and synonyms) to a conceptual model (with more complex knowledge) to a logical theory (with very rich, complex, consistent and meaningful knowledge) (Daconta et al. 2003).

Ontological engineering (borrowing from formal ontology and logic in philosophy and formal semantics in linguistics) addresses the problem of encoding declarative knowledge in a modular, reusable fashion by creating both foundational knowledge theories and domain theories. (Obrst and Liu 2003).

In recent years, the application of ontologies to the web has been encapsulated in the vision of the “semantic web”, where it is envisaged that ontologies can be used to improve existing web based applications or enable new uses of the internet (Berners-Lee 2001). Example use cases (OWL 2004a) range from navigating and searching web portals, through to enabling ubiquitous computing by supporting interoperability under “un-choreographed” conditions.

The promise of ontologies is in the sharing of an understanding of a domain that can be communicated between people and application systems (Fensel 2003). However, as seen in the Gruber definition, ontologies are defined from a particular **perspective**.

Whether it is a personal ontology, designed to support individual's needs and preferences, or an ontology created by a particular company to reflect that company's views on their industry, such ontologies will have biases and necessarily subjective features. If that individual or company needs to exchange information and knowledge with other individuals or organisations, there will be a need to map between multiple ontologies either at inter-personal, intra-organisational or inter-organisational levels (Hameed et al. 2004).

A typical example of where this difference in perspective is causing a problem is in the area of web service sequencing or composition. Recently languages such as the Web Services Ontology Language OWL-S (OWL-S 2004) and Web Service Modelling Ontology WSMO (WSMO 2005) have been proposed. These languages attempt to annotate web services with more semantic information so that they can be more easily used and discovered. Their introduction is in response to the deficiencies with respect

to semantic expressiveness of the Web Services Description Language WSDL (WSDL 2001), in terms of interface definition, and the Universal Description, Discovery and Integration UDDI service (UDDI 2003) in terms of service discovery. Thus the annotation of web services through the use of ontologies represents progress on the issue. Realistically however, an annotation of any one web service will take place from a particular perspective and using a particular selection of ontologies. Thus there is still a need to reconcile these different perspectives so that combined use of the ontologies can be achieved when composing or sequencing web services drawn from several sources together. Ontology mappings are seen as the way in which such reconciliation and combination can be enabled.

Other example use cases for ontology mapping could include (Euzenat et al. 2004): ontology driven data querying, where ontology mappings help decompose queries and combine responses; E-commerce catalogue matching, where mappings could allow for comparison of supply and demand catalogues; Personal Information Delivery, where mappings between a personal ontology, representing interests, and information service ontologies, representing available information streams, could be used to personalise information delivery.

Given the potential importance of such ontology mappings to organisations in areas such as web service composition, E-commerce catalogue matching and ontology based data integration, the question arises as to how to generate these mappings. Naturally organisations would prefer such mappings to be created as part of some engineered process rather than as a result of *ad hoc* activities.

However none of the current solutions provide practical support for an organisation for the entire lifecycle of ontology mapping. The reason for this could be that agreement on what is a full ontology mapping lifecycle has not yet emerged. In this thesis however, we define the scope of the ontology mapping lifecycle as: a **characterisation phase** encompassing characterisation activities which discover ontologies for mapping, analyse the difficulties that may be involved in undertaking a mapping, and generate the candidate matches between the ontologies; a **mapping phase** covering the mapping identification itself; an **execution phase** where mappings are rendered into different formats and are interpreted to support semantic interoperability between two

applications; and a **management phase** responsible for the ongoing maintenance and management of the mappings once deployed. The majority of state of the art systems concentrate on the generation of matches and mapping identification aspects alone (Noy 2004), although for most systems these two tasks are tightly bound together. In this thesis a clear distinction is maintained between candidate match generation and mapping determination. The reason for this is that with the dynamic and *ad hoc* nature of some of the expected use cases, there is an increasing need for applications themselves to determine the mappings that are relevant to them based on direct examination of the candidate match information. In contrast, current ontology mapping systems are focused on the identification of mappings from the perspective of an expert user or (rarely) an expert application.

1.2 Objective and Goals

This thesis identifies the activities, information artefacts, heuristics and guidance required to support a practical ontology mapping process, and examines how they can be coordinated and sequenced in the ontology mapping process.

In order to investigate this research question, three objectives were derived:

1. Research and identify models and processes that support ontology mapping;
2. Develop an integrated set of process and software tools to support practical ontology mapping;
3. Evaluate the use and benefit of the process and the software tools developed to generate ontology mappings, and demonstrate the usage of the generated mapping information artefacts at runtime to support semantic interoperability between applications.

This thesis proposes a framework comprising an ontology mapping process supported by software tools. The process consists of activities, information artefacts, heuristics and guidance. The framework is called OISIN (standing for **O**ntology **I**nteroperability for **S**emantic **I**nteroperability). Taking a framework approach enabled the desired practical characteristics of ontology mapping to be translated into requirements upon the framework.

It should be noted that the emphasis is placed on examining what are the key activities and software tools needed for the ontology mapping lifecycle, and not on any one activity. In particular, this thesis does not propose new ontology schema matching algorithms.

1.3 Contribution

The first contribution of the thesis is the development of a technology-neutral specification, expressed using the Unified Modelling Language - UML (UML 2004), for the ontology mapping process. The process supports a full ontology mapping lifecycle and features characterisation activities that identify how amenable two ontologies are for mapping. Such characterisation activities are currently lacking in the state of the art. In addition, the process supports a continuum of deployment strategies from all activities being undertaken at design time right through to all activities being undertaken at runtime. Furthermore, this process can be specialised to many different use cases and situations within an organisation. Finally, the same process can be used in use cases where the determination of what constitutes a mapping is to be undertaken by an expert user, or where the determination of what constitutes a mapping is to be undertaken by an application itself.

A second contribution is the finding arising from the evaluation experiments that inspection of information about ontology characteristics is as useful, and in some cases more useful, in estimating the difficulty involved in ontology mapping, than the current informal practice of direct human inspection of the ontologies (that is “by hand”). Direct examination of the ontologies by users can lead to inconsistent analysis of the difficulty involved in ontology mapping. In contrast, the automatic generation of ontology characteristic information lends itself to more consistent analysis. In addition, the nature of the characteristic information is such that the effort involved in the examination of the information is constant, irrespective of the size of the ontologies, whereas the effort involved in the explicit examination of the ontologies increases in direct proportion to the size of the ontology.

1.4 Technical Overview

This section presents the technical approach taken in the investigation of the thesis and provides an overview of how the thesis report is structured.

1.4.1 Technical Approach

An initial study of the state of the art of systems in the areas of ontology alignment and ontology integration was conducted, with particular focus on the role ontology mapping had to play and the approaches used to address ontology mapping. This literature review was supplemented by a case study based approach to investigating the practical aspects of ontology mapping. This involved the identification of two ontologies with overlapping domains, their manual mapping and the use of the mapping information identified to drive the runtime exchange of information between two systems. The literature and case based investigations helped in the formation of an understanding of the issues involved in the ontology mapping area including the strengths and the deficiencies of current approaches.

Based on the state of the art analysis and experience gained through the case study, an ontology mapping process and supporting software tools were designed. A framework approach was adopted. This allowed for some of the desired practical characteristics of the ontology mapping process to be placed as requirements upon the framework. For example, flexible deployment of the process is achieved through the specification of the ontology mapping process in UML and through careful design of the process activities. This specification has been designed to be extensible. Eight software tools were developed to support the ontology mapping process. These software tools were implemented using a combination of eXtensible Markup Language (XML) based technologies, Java and third party libraries.

Two experiments and a trial were designed in order to evaluate various aspects of the ontology mapping process and software tools that had been implemented. For these experiments, five ontology pairs were identified that exhibited varying degrees of overlap. Detailed mappings for two of the ontology pairs were generated. The experiments involved participation from final year computer science undergraduates and a M.Sc. dissertation class. The trial used input generated by the M.Sc. dissertation

class. The results from the experiments were analysed using Microsoft Excel and MINITAB software. The analysis supports the claim that the use of the OISIN framework brings important benefits in mapping between ontologies.

1.4.2 Thesis Overview

Chapter Two provides background on what is semantic interoperability and how it differs from system, syntactic, and structural interoperability. The chapter also presents the role of ontologies in representing semantics.

Chapter Three first overviews a scenario and two ontologies that will be used for example purposes in chapters three and four. A state of the art review in ontology mapping is then described. The chapter then presents the key framework requirements that need to be addressed in order to support a practical ontology mapping process. A selected set of ontology mapping systems are described and reviewed with respect to the framework requirements. Finally an overview of the technologies used in the implementation is provided, with a description of these technologies being presented in Appendix A.

Chapter Four presents the design of the ontology mapping process. A summary of the process is provided including descriptions of the overall activity diagram, presentation of the activities and information artefacts. Each activity is described in relation to the motivation for the activity and the difficulties in implementing the activity. The detailed UML specification for the process is included in Appendix B.

Chapter Five describes the implementation of the automatic and semi-automatic software tools that have been developed for use in the ontology mapping process. The Document Type Definitions (DTDs) for the XML information artefacts used in the implementation is included in Appendix C.

Chapter Six presents how the thesis was evaluated. In particular, each of the two experiments that were undertaken is described, results are presented and findings discussed. Supporting information for these experiments are presented in Appendices D and E.

Chapter Seven describes the trial that was undertaken to demonstrate the use at runtime of the mapping information generated by the ontology mapping process. The results and findings from the trial are discussed. Appendix F includes supporting information. This chapter also presents a comparison of the proposed framework with related work.

Chapter Eight presents conclusions, contributions and suggestions for future work.

2 BACKGROUND – SEMANTIC INTEROPERABILITY

This chapter investigates the definition of semantic interoperability and how it is different from system, syntactic, and structural interoperability. The chapter also describes the role of ontologies in representing semantics.

Interoperability has been a basic requirement for modern information systems since the mid-1970s and progressively different levels of interoperability have been addressed. These different levels of interoperability are necessary to reflect different levels of heterogeneity in computer systems that have arisen due to natural technological differences of hardware, operating systems, communication systems and application software. Sheth provides an excellent classification of the heterogeneity involved and the consequent levels of interoperability (Sheth 1999). In this classification, differences in hardware (instruction sets, data representation and so on) and differences in operating systems (file systems, inter-process communication mechanisms etc.) are termed **Platform Heterogeneity**. Differences of basic implementation of information services such as databases (different data models, concurrency control capabilities etc.) are termed **Information System Heterogeneity**. Sheth then introduces the term **System Heterogeneity** to cover both Information System and Platform heterogeneity. Three levels of Information System Heterogeneity are also proposed. **Syntactic Heterogeneity** concerns the differences in data representation or format. **Structural Heterogeneity** corresponds to the differences in the modelling of the information from a schema/structure perspective. **Semantic Heterogeneity** covers the differences in **semantic interpretation** of information. The meaning of semantic interpretation and how semantics are represented are addressed later in this section.

Techniques for overcoming these System, Syntactic, Structural and Semantic differences in information systems have been studied over the years as the scale of computer deployment has grown and as the requirement for interoperation between systems within and external to an organisation has increased.

Late 1970s to late 1980s

System interoperability was the first focus in the late 1970s and early 1980s, when the scope of interoperability spanned a handful of interconnected computers and databases, and the communication infrastructures were typically proprietary (e.g. IBM's Systems Network Architecture (SNA)) implementing a two-tier architecture. Research into how information system heterogeneity could be overcome was primarily driven by projects in the distributed, multi-database and federated database areas (Oszu and Valduriez 1999). During this time both a clearer understanding of how to handle data modelled using different database data models (e.g. network, object oriented and relational), and techniques for schema integration emerged. In addition techniques related to how data across different systems could be managed (e.g. distributed locking and two phase commit protocol) were introduced.

Late 1980s to late 1990s – Database and Middleware Communities

From the late 1980s to the late 1990s, the scope of interoperability for an organisation widened to tens of systems on a Local Area Network (LAN) and towards the end of this period to encompass systems on the internet. The three-tier client server architecture also became popular. System interoperability was further progressed through the acceptance of TCP/IP¹ as a standard for networking, the emergence of *de facto* distributed computing standards in support of application interoperability (i.e. Remote Method Invocation (RMI), Common Object Request Broker Architecture (CORBA) and Distributed Component Object Model (DCOM)) and the adoption of common database connectivity APIs (e.g. Object Database Connectivity (ODBC) and Java Database Connectivity (JDBC)). The emphasis in this period shifted to achieving syntactic and structural interoperability. In terms of syntax, the Hypertext Markup Language (HTML) was confirmed as the standard for web page representation and Moving Pictures Experts Group (MPEG) for pixel-level representation of image information. In terms of promoting interoperability of how information is structured, ANSI's Structured Query Language (SQL) and the Entity Relationship (ER) model supported the exchange of schema information in relational databases, Object Management Group (OMG) UML supported interchange of design information in object systems and OMG's Interface Definition Language (IDL) for the interchange of

¹ Transmission Control Protocol/Internet Protocol

information about how application interfaces were represented. In addition, with the requirement for storage of other kinds of data (e.g. images, audio, video) the use of metadata became more prevalent.

Metadata is usually defined as “data about data”. It can capture information that does or does not depend on the content of the asset, called respectively **content-dependent metadata** (e.g. size of a document) and **content-independent metadata** (e.g. modification date of a document). **Content-descriptive metadata** can describe the contents of an asset (e.g. describing the flavour of a wine in association with a picture of the wine bottle). **Domain-independent metadata** can capture information that is present independent of a domain (e.g. HTML). Finally **domain-specific metadata** is described in a manner specific to a domain of information (e.g. population size from the census domain).

Such metadata has been used to support the integration of information systems. In the distributed database environment the ideas of a System Catalog and Auxiliary Databases were introduced. The System Catalog contained metadata about an individual database and an Auxiliary Database was used to represent metadata related to the differences between databases (e.g. type conversion information). In the middleware community the Open Distributed Processing group’s ODP Trader (ODP 1993) was introduced to contain metadata about what distributed application services were deployed in a network. This Trader acted like a “yellow pages” server allowing humans or software systems to search for applications that would fulfil expressed service characteristics or needs. Some elements of this functionality were incorporated into the Interface Repository service, when it was introduced by the CORBA community to store metadata about application interfaces distributed across a CORBA based network.

Although the above initiatives by the database and middleware communities provided important contributions for the resolution of syntactic and structural interoperability during this period, the issue of semantic interoperability was not resolved.

Late 1980s to late 1990s – Artificial Intelligence Community

In parallel, in the 1980s up to the late 1990s, the Artificial Intelligence community endeavoured to progress the encoding of declarative (that is non-procedural, human confirmable) domain knowledge in a modular and reusable fashion through the creation of ontologies (Obsrt and Liu 2003).

Normally as humans we assign symbols to things/concepts. The same symbol can invoke in our mind different things/concepts. Depending on the mental model/context used, the interpretation of that symbol will result in a different concept/meaning being invoked. This symbol/model/concept relationship is known as the “meaning triangle” (Ogden and Richards 1946). Thus the idea behind an ontology² is to capture in a computer interpretable manner a part of our mental models about specific domains such that when a symbol and a reference to an ontology that contains that symbol is presented to an application, the application can interpret the intended meaning of that symbol. *Semantic interpretation is the mapping between some structured subset of data and a model of some set of objects in a domain with respect to the intended meaning of those objects and the relationships between those objects* (Daconta et al. 2003). Thus the goal of semantic interoperability is to ensure that consistent semantic interpretation of the same information is achieved across systems.

The need for a new sub-discipline in knowledge representation focusing on ontology engineering was partially in response to the deficiencies in mainstream knowledge based techniques such as expert systems that relied on knowledge being represented at the same level with little scope for reuse. As discussed in chapter one, ontological engineering addresses the problem of encoding declarative knowledge in a modular, reusable fashion by creating both foundational knowledge theories and domain theories. (Obrst and Liu 2003).

Great advances were made in the development of ontological engineering techniques during this period (Corcho and Gomez-Perez 2000). In addition ontological approaches that could aid information integration were demonstrated (Wache et al. 2001). However, very little focus was placed on how to represent ontologies in a neutral form

² For a more formal definition of an ontology, please refer to section 1.1

that would enable their exchange between systems. An example of one such initiative is DARPA's Knowledge Sharing Effort (Neches et al. 1991).

Late 1990s to date

During 1996, XML (XML 2000) was introduced by the World Wide Web Consortium (W3C). XML quickly established itself as a popular way to overcome syntactic and structural interoperability problems between parties that agreed on using a set of tags, XML schemas (XMLSchema 2004) or XML DTDs (XML 2000).

The database community has quickly embraced XML with the introduction of new languages for querying and updating XML data (that is XQuery (XQuery 2005) and XUpdate (XUpdate 2000) respectively). In addition, relational databases have been XML-enabled to store/retrieve XML data and databases specifically tailored for handling XML data (so called "native XML databases") have begun to emerge. Similarly, the middleware community have started to represent interfaces to web services using XML by means of WSDL.

To date such basic XML technologies have been demonstrated to be successful in well understood/static information exchange environments. However, the use of basic XML technologies are inappropriate for systems that want to interchange information in dynamic environments due to their innate inability to represent semantics (Cui et al. 2002).

In parallel, the Artificial Intelligence community also recognised the potential of XML for representing ontologies in neutral form and several proposals emerged, including the DARPA Agent Markup Language (DAML) (Bechofer et al. 2000), the Ontology Inference Layer (OIL) (Horrocks et al. 2000), and DAML+OIL (vanHarmelen et al. 2001). Each of these built upon the Resource Description Framework (RDF) that was introduced in 1998 by W3C as a way to represent metadata with XML (RDF 2004). More recently the Web Ontology Language called OWL (OWL 2004b) has been standardised by the W3C as a means to represent ontologies in XML, building upon the previous research initiatives mentioned.

Thus XML is providing one point of convergence between the database, middleware and Artificial Intelligence communities. The opportunity exists, for semantic interoperability of distributed and diverse data based applications and web service based applications, to be addressed through the use of ontologies and XML technologies.

However, before the use of ontologies to enable semantic interoperability becomes a mainstream technology, a myriad of issues still need to be addressed. These include (Chetbotko et al. 2004, Benjamin et al. 2002):

- engineering solutions for the development of ontologies, including provision of editing and collaboration tools;
- ontology mapping, alignment and merging;
- annotation of information resources using ontologies;
- ontology based information retrieval.

The key issue investigated in this thesis is that of ontology mapping. Chapter three provides an overview of the area and examines the state of the art.

3 STATE OF THE ART – ONTOLOGY MAPPING

As discussed in chapter two, Semantic Interoperability is not a new problem. The potential of ontologies is for the encoding of an understanding of a domain that can be exchanged between people and application systems. However ontologies are defined from a particular perspective, leading to multiple ontologies. As seen in chapter one, there are several use cases where multiple ontologies need to be accessed from several applications. Ontology mappings could provide a common platform from which multiple ontologies could be accessed to allow for the achievement of consistent semantic interpretation across multiple diverse applications, thereby supporting semantic interoperability.

This chapter presents the state of the art in ontology mapping. Section 3.1 introduces a “visiting lecturer” scenario and two ontologies that will be used as examples throughout chapters three and four. Section 3.2 provides an overview of the main research areas in ontology mapping of relevance to this thesis. Section 3.3 describes the key requirements that need to be addressed by a framework in order to make ontology mapping practical. Section 3.4 describes a representative selection of systems and reviews them in the light of the requirements. Section 3.5 provides a summary of how the selected state of the art systems address the requirements. Finally, Section 3.6 provides an overview of the technologies chosen to underpin the implementation described in chapter five.

3.1 *“Visiting Lecturer” scenario*

In this section a scenario is introduced that will be used throughout chapters three and four. The “visiting lecturer” scenario will be used to illustrate the kind of differences that arise in the modelling of ontologies. The scenario is for a “visiting lecturer” to a university for a six month term. The aim would be to enable the visiting lecturer’s personal software (which has been configured for the home university) to interoperate seamlessly with the visited university’s information services such that the lecturer can interact with the environment as if he/she was still in his/her own university. For example, the visiting lecturer will want to book lecture rooms, to get access to budget controlled resources such as video conferencing and so on without the need to recode

or rework his/her computing environment. This requires that a mapping is created between the visiting lecturer's ontology representing his/her concepts and the ontology of the university. For this scenario the ontology used for the visiting lecturer is that created by the University of Aberdeen (Aberdeen 2003) and the university ontology is represented by the ontology created by the University of Manchester (Manchester 2003). Screenshots from the ontology browsing tool Protégé (cf. section 3.4.4) of the is-a³ hierarchy of each ontology is presented side by side in Figure 3-1. This is only a partial view (due to space considerations) but it shows the main classes and super/sub class relationships that will be referenced in the following sections.

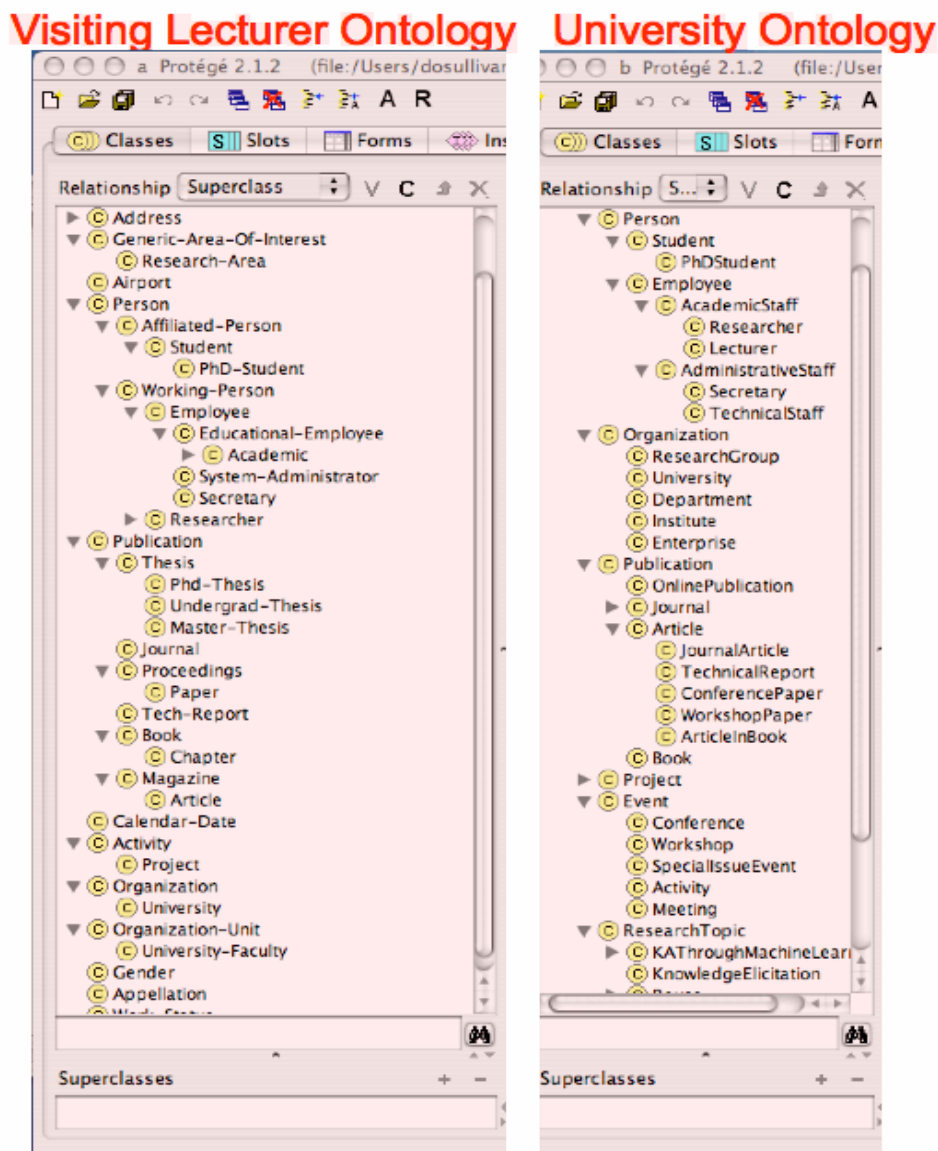


Figure 3-1: Partial is-a hierarchies of scenario ontologies

³ “is-a” represents the subclass/superclass relationship

3.2 Overview of Ontology Mapping

In order to generate ontology mappings, various matching techniques need to be applied to the ontologies. As discussed by Shvaiko and Euzenat (Shvaiko and Euzenat 2004), it is generally accepted that although there is a difference between ontology matching and general schema-matching⁴, the techniques developed for each can be mutually beneficial. Schema-matching is a general problem in many applications today, such as system integration, e-business, data warehousing and the semantic web. For this discussion a *schema* is defined to be a *set of elements* connected by some *structure* (Rahm and Bernstein 2001). Although the element formats, expressiveness and structure of the schemas may differ in each of these domains (relational schema, XML schema, OWL and so on), the motivation behind the matching process is the same, that is, to discover correspondences between elements in each of the schemas and document these correspondences using a *mapping expression* (Rahm and Bernstein 2001). This mapping expression can range over simple equivalences and complex correspondences. An example of a simple correspondence in the scenario is where `Paper` in the visiting lecturer ontology could be considered equivalent to `ConferencePaper` in the university ontology. An example of a complex correspondence would be that the `has-page-numbers` property of the visiting lecturer ontology is equivalent to the `lastPage` property *minus the* `firstPage` property of the university ontology.

The most comprehensive survey of ontology mapping systems over recent years has been developed by Kalfoglou and Schorlemmer who have surveyed the ontology mapping area under the headings: frameworks, methods and tools, translators, mediators, techniques, experience reports, theoretical frameworks, surveys, and examples (Kalfoglou and Schorlemmer 2003). Kalfoglou and Schorlemmer also clarify the terminology that claims to be relevant in the literature (alignment, merging, articulation, fusion, integration, and so on). The view of ontology mapping in this thesis aligns with the Kalfoglou and Schorlemmer view. In particular, ontology mapping is defined as *the task of relating the vocabulary of two ontologies that share the same domain of discourse in such a way that the structure of ontological signatures*

⁴ Two substantial differences are that: (a) ontology data models are richer in the number of primitives used and in complexity; (b) in database schemas the semantics of information are not explicitly exposed.

and their intended interpretations are respected (Kalfoglou and Schorlemmer 2003). The authors proceed to show that ontology mapping, as well as being an activity in its own right, can also be considered a sub activity of both ontology merging/integration and ontology alignment activities. Ontology merging/integration is an activity where a new ontology is created from one or more ontologies. Ontology alignment is where two or more ontologies are altered to bring them mutually consistent and coherent.

The state of the art in this section is presented under headings which align with the core of the ontology mapping lifecycle introduced in chapter one, that is: Characterising Ontology Mismatches (section 3.2.1); Mapping Discovery (section 3.2.2); Mapping Documentation and Interpretation (section 3.2.3).

3.2.1 Characterising Ontology Mismatches

Mismatches between ontologies are the key type of problems that hinder the combined use of independently developed ontologies (Klein 2001). Klein builds upon previous work (Kitakami et al. 1996, Visser et al. 1997, Wiederhold 1998, Grosso et al. 1998, Bowers and Delacambre 2000) in order to construct a broad classification for ontology mismatches shown in Figure 3-2.

Two levels of mismatches are distinguished by Klein: language and ontology. The “language level” distinguishes mismatches of the language primitives that are used to specify the ontology. This includes Syntax (e.g. OWL, RDFS), Logical Representation (e.g. differences in how “disjointness” is expressed in a language); Semantics of Primitives (e.g. “=” and “equalTo”) and Language Expressivity (e.g. some languages do not allow a way to express negation).

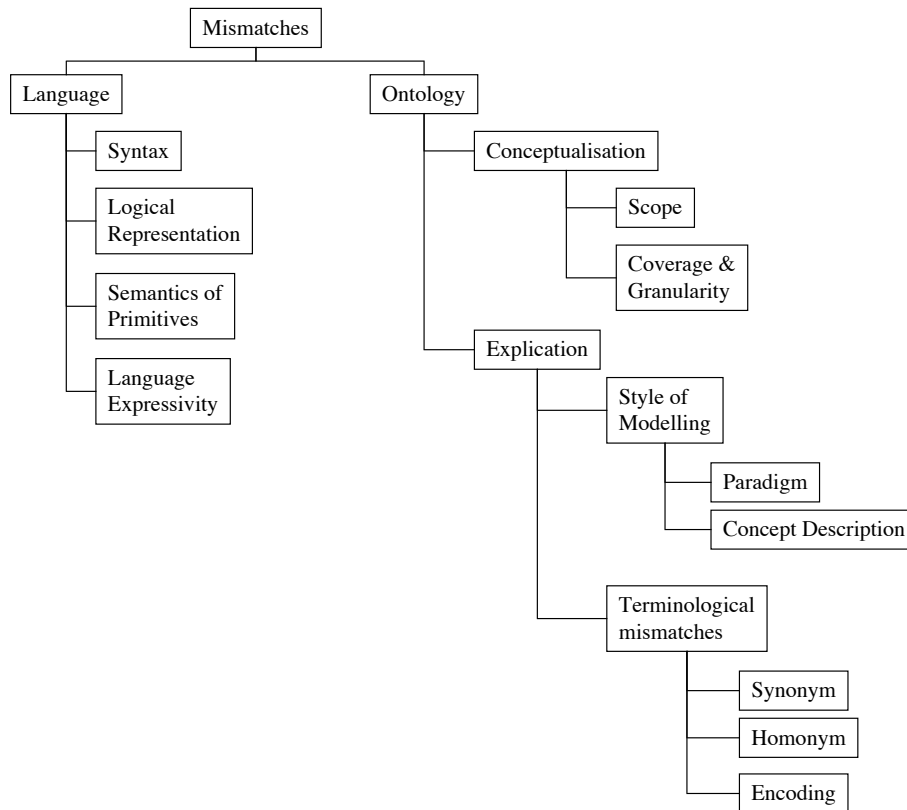


Figure 3.2: Classification of Ontology Mismatches (Klein 2001)

The “ontology level” distinguishes the differences in the way the domain of ontologies are modelled and is subdivided into conceptualisation and explication mismatches. “Conceptualisation Mismatches” relate to the difference in the way a domain is interpreted and is subdivided into Scope and Model coverage and granularity. A Scope mismatch is where two classes may seem to represent the same concept but are modelled differently. In the scenario for example, the contact details of the `PhdStudent` class in the university is modelled using properties `email`, `phone`, `fax` whereas in the ontology used by the visiting lecturer the contact details of a `PhdStudent` is simply a `has-room-number` property. A model coverage and granularity mismatch is where ontologies may appear to be modelling part of the same domain but at a different level of depth and detail. In the scenario, in the university ontology a `ResearchTopic` is decomposed into a large number of subclasses (e.g. `Knowledge Elicitation`, `Knowledge Management` etc.), whereas in the ontology used by the visiting lecturer all research topics are modeled as instances of a single `ResearchArea` class. “Explication Mismatches” represent a difference in the way the conceptualisation is specified, covering style of modelling, terminological and encoding mismatches. Style of modelling mismatches include Paradigm (e.g. how time

is represented) and Concept description mismatches. An example of a common concept description mismatch is where classes are modelled differently in an is-a hierarchy. In our scenario, a `Secretary` is modeled as a subclass of `AdministrativeStaff/Employee/Person` in the university ontology and as a subclass of `Employee/Working-Person/Person` in the visiting lecturer's ontology. Terminological mismatches cover Synonyms (same concept different name) and Homonyms (different concept same name). An example of a synonym in the scenario is `ConferencePaper` in the university ontology and `Paper` in the visiting lecturer ontology. An example of a homonym in the scenario relates to the name `Activity`. In the university ontology an activity it is a type of `Event`, whereas in the visiting lecturer ontology it has a broader meaning as can be seen by its subclass `Project`. Finally encoding mismatches occur where values are in different formats (e.g. European versus US dates). An example in the scenario would be where the `year` property of a publication in the university ontology is represented as an integer and the `has-publication-date` property in the visiting lecturer ontology is an instance of a calendar date object.

As recognised by Klein, the application of such classifications for ontologies is non-trivial and relies on human expertise to recognise ontology mismatches. For this reason, the incorporation of an activity to characterise ontology mismatches into ontology mapping systems has been rare. An exception is the KRAFT system (Gray et al. 1997) where an activity is explicitly incorporated in the system to apply an ontology mismatch classification (Visser et al. 1997) to the two ontologies. In the KRAFT system this is a manual activity. It is stated that the purpose of the activity is the early identification of the complexity that will be involved in deriving mappings. However details of how the various mismatches contribute to a complexity calculation are not described in KRAFT published literature.

Analysis

The KRAFT system provides an indication of the benefit of doing some pre-processing in order to determine the potential complexity involved in mapping. However it assumes that an exhaustive characterisation is undertaken that identifies all the mismatches between the ontologies before the mapping phase begins. As can be seen by the classification framework itself, this is a non-trivial exercise especially when

manually applied. In addition, it would make more sense if the identifier of the mismatch problem would propose a mapping solution at the same time given the detailed nature of some of the mismatch classifiers. Furthermore, it is clear that a mapping complexity calculation should not just be based on the mismatches between the ontologies but rather should also take account of other factors such as ontology quality.

In contrast, in the proposed OISIN framework the focus in the early parts of an ontology mapping lifecycle is on generally characterising how amenable ontologies are for mapping and characterising the potential matches, rather than characterising the ontology mismatches involved.

In summary, whereas there exists the means to express ontology mismatches, there remains the need for research into methods that will enable the semi-automatic or automatic characterisation of ontologies amenability for mapping.

3.2.2 Mapping discovery

Most research effort over the last number of years has concentrated on the identification of candidate matches of elements in schemas, which can then be analysed in order to determine what mappings should be expressed. For this reason, in this section the state of the art is discussed in two parts, match generation and derivation of mappings.

Match Generation

Rahm and Bernstein have introduced a taxonomy that allows the classification of techniques and approaches that have been used in schema-matching (Rahm and Bernstein 2001). Briefly the main classification criteria for individual matchers is described here:

- “Schema-based only” matchers only consider schema information, not instance data.
 - Element level matching determines the matching elements in the other schema based on linguistic approaches (e.g. lexical matching) and/or constraint based approaches (e.g. cardinality matches or type/domain

matches). Thus based on linguistic approaches, in the scenario `Article` in the visiting lecturer ontology would have candidate matches of `JournalArticle`, `ArticleinBook` and `Article` in the university ontology. Based on constraint approaches any properties that have *integer* as a data type in one ontology would have a candidate match with properties with *integer* as the data type in the other ontology.

- Structure level matchers attempt to match combinations of elements that appear together in a structure (e.g. columns grouped together in a table or classes related via an is-a hierarchy). For example in the scenario, `Person/Student/PhDStudent` of university ontology would structurally match with `Person/Affiliated-Person/Student/PhD-Student` since the linguistically matched elements appear in the same is-a order.
- Instance/contents level matchers examine the instances/contents of the actual data of the two systems in order to deduce possible correspondences. In particular, information retrieval techniques (such as Latent Semantic Indexing (Dumais et al. 1996) or co-active intelligence (Truran et al. 2005)) popular in the classification of information on the web are candidate techniques in this class of matchers.
- Hybrid matchers integrate several individual matchers based on a fixed workflow to determine match candidates. For example a common pattern would be to start with a linguistic matcher and then pass candidate matches through constraint and/or structural matchers for further analysis. More flexibly, composite matchers combine the results of independently executed individual matchers in a workflow that is determined by the user.

Shvaiko and Euzenat have recently expanded on the “Schema-based only” aspect of the taxonomy in order to reflect ontology matching research developments in the years since the Rahm and Bernstein classification (Shvaiko and Euzenat 2004). “Schema-based only” matching techniques are particularly relevant for this thesis, as this is the approach adopted in the proposed framework. In particular the taxonomy of Rahm and Bernstein has been altered in the light of techniques such as semantic model matching developed specifically for ontological schemas (Giunchiglia and Shvaiko 2003). Semantic model matching copes with semantic relationships between concepts. For

example, in the scenario each of the Organisation subclasses of the university ontology would all match to the Organisation class of the visiting lecturer through a “is more specific than” relationship.

Shvaiko and Euzenat propose a three layer classification (see Figure 3-3) for “schema-based only” matching techniques comprising of: upper layer, middle layer and lower layer.

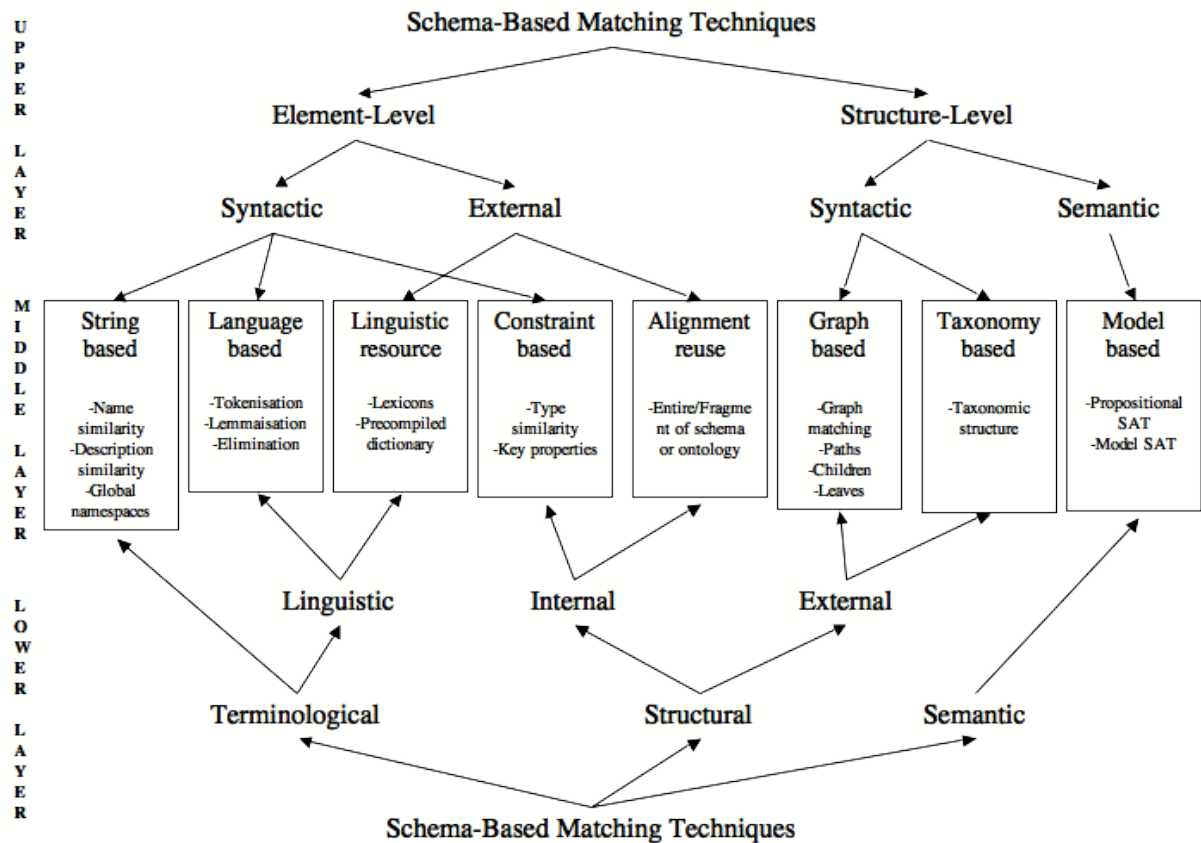


Figure 3-3: Classification of schema-based matching approaches (Shvaiko and Euzenat 2004)

The upper layer classifies the matching approaches according to the granularity of match and the way the matching techniques interpret the input information. It first uses the Rahm and Bernstein high level classification of element and structure levels and then introduces a classification of syntactic, external and semantic. Syntactic techniques interpret input just based on what is provided. External techniques exploit auxiliary domain knowledge such as expert input or online thesauri. Semantic techniques use some form of formal semantics to analyse the input and justify their results. An example of formal semantics would be model-theoretic semantics. The

middle layer presents classes of matching techniques based on how the input is interpreted. The lower layer is concerned with classifying the type of data considered by a particular matching algorithm. Terminological matching algorithms consider string data, structural algorithms consider structure data and semantic algorithms considers semantic data.

Particular focus is placed in this section on the middle layer, as this provides a useful way to overview the state of the art in matching techniques. The matching techniques at the middle layer are:

1. String based techniques (e.g. prefix, suffix, edit distance and n-gram tests) match names and name descriptions of ontology entities; In the scenario for example, a suffix test would identify `telephone` and `phone` as a match because the first string ends with the second one.
2. Language based techniques are based on Natural Language Processing techniques such as tokenisation (i.e. where names of entities are parsed into tokens), lemmatisation (i.e. finding base forms of names), and elimination (i.e. words that do not add meaning are discarded). In the scenario for example, `PhDStudent` and `Phd-Student` would match due to matching of significant tokens of `Phd` and `Student`.
3. Constraint based techniques concern internal constraints applied to the definition of entities such as types, cardinalities of attributes and keys. In the scenario, the `studentnumber` property would have a candidate match of every property in the other ontology that had type *integer* and had a cardinality of *exactly one*. This would match most identifier properties such as `employee_id`.
4. Linguistic resource techniques are those where lexicons or thesauri are used to match words based on linguistic relations between them (e.g. synonym). For example, `Journal` of the university ontology would have a synonym match with `Magazine` of the visiting lecturer ontology.
5. Alignment reuse techniques are those which exploit matches previously produced. For example, the GLUE system (Doan et al. 2002) uses machine learning techniques to find mappings by employing a text classification method called Naïve Bayes as a content learner, a name learner and a metalearner that combines the predictions from both. Another example is CAIMAN (Lacher

and Groh 2001) where text classification is used to measure the probability that two concepts (one from a repository ontology and one from a user's personal view of documents of interest) are corresponding. In OMEN (Mitra et al. 2004) a Bayesian network is constructed using meta-rules that capture the influence of the ontology structure and semantics of the ontology relations. The network is then trained using pre-existing mappings.

6. Graph based techniques consider the inputs as labelled graphs and undertake a similarity comparison. The intuition is that if two nodes from the inputs are similar then their neighbours will be similar too. For example, the Artemis (Castano et al. 2001) system calculates the name, structural and global affinity coefficients of two schemas by exploiting a common thesaurus.
7. Taxonomy based techniques consider the inputs as graphs and in particular exploits the is-a relationships. For example, the Anchor-PROMPT algorithm (Noy and Musen 2001) takes the two ontologies and a set of anchor-pairs of related terms and then analyses the ontologies limited by the anchors in order to determine the terms that frequently appear in similar positions on similar paths. Another example is the Naïve Ontology Mapping (NOM) system (Ehrig and Sure 2004) that exploits information about super/sub concepts and super/sub properties of an ontology.
8. Model based algorithms handle the input based on semantic interpretation typically through propositional satisfiability (SAT) or description logics (DL) techniques. For example, S-Match (Giunchiglia et al. 2004) takes two graphs as input and returns semantic relations (equivalence, more general, less general, mismatch, overlapping) between corresponding nodes as output.

It should be noted that the matching technique classification assigned to the example systems mentioned in the preceding discussion are not exclusive.

Typically most ontology mapping systems will adopt some form of string/language/constraint/linguistic based techniques (from classifications 1 to 4) as a base, and then incorporate further techniques (from classifications 5 to 8) in addition. Increasingly, an architecture is adopted where a number of individual matchers are executed and some aggregation of the results take place in order to determine the mappings. Example hybrid systems based on a fixed combination of matchers are:

APFEL (deBruijn et al. 2005a), QOM (Ehrig and Staab 2004), and CUPID (Madhavan et al. 2001). COMA (Do and Rahm 2002) is an example of a composite system based on a user specified combination of matchers.

Analysis

In the proposed OISIN framework, it is assumed that in the future a number of different matchers can be selected for application to a particular ontology mapping set, based on the characteristics of the ontologies. This assumes that the metadata necessary to characterise the applicability of match algorithms in different situations will be possible to document. However, whereas a lot of excellent matching algorithms have emerged over the last number of years, the implementations of these have typically been tied to particular system architectures. It is assumed that this situation will change and that the availability of independent matchers will be enabled through the adoption of a common format for exchange of matching results. Such a common format may be based on a format like the INRIA format discussed later in section 3.4.3. However, a key challenge that would still remain would be how to characterise the applicability of individual matchers for different situations. Even in papers describing composite systems where the user is enabled to choose which matcher to apply, no detail has been provided as to what metadata annotates these matchers to support the user in making their decision. In addition most evaluations of the matchers concentrate primarily on the functional aspects of the systems (such as accuracy of matches) rather than evaluation of applicability to different use cases.

Derivation of mappings

Fully automatic generation of mappings from ontology match information is generally considered impractical as yet (Klein 2001, Kalfoglou and Schorlemmer 2003, Noy 2004). This consideration arises from the lack of certainty involved with the automatic matching process, primarily due to the heuristic nature of most matching algorithms. One recent effort called QOM has shown how ontology mapping could be done at runtime in a reasonable time frame but at the cost of accuracy of the mappings (Ehrig and Staab 2004). The approach taken is to use heuristics to discard some of the candidate matches in order to speed up the efficiency of evaluating the candidate matches with respect to identification of mappings. For now however, it is expected

that semi-automatic techniques for creating mappings from ontology matching information will continue to dominate (Uschold and Gruninger 2004).

Mapping detection has primarily relied to date on a human user with some graphical user interface examining the matching information that has been generated. In systems such as COMA++ (cf. section 3.4.2) and SWOOP (Kalyanpur et al. 2004) the focus has been placed on providing graphical support for the presentation of matching information and point and click creation of mappings. In addition COMA++ provides the opportunity for the user to browse mappings created from other mapping process sessions. However, typically such systems do not provide explicit user support for identifying mappings. In other systems such as Protégé the user is led through decisions for mappings based on the iterative algorithm Anchor-PROMPT (cf. section 3.4.4). More recently, the idea of capturing and documenting ontology mapping patterns has been proposed (deBruijn et al. 2005b). The idea is to guide the developer of an ontology mapping in the construction of ontology mappings using patterns of mappings that are common. The pattern template includes Name, Problem, Context and Solution headings similar to software development patterns. A number of such patterns have been identified. For example the pattern SubClass mapping describes the problem where a class in one ontology is a subclass of a class in a second ontology but there is no way of expressing the additional properties of the subclass. The solution suggested in the pattern *is to establish a unidirectional mapping from a more specific class in one ontology to a broader class in another ontology. The relation is broadened to allow class expressions in addition to merely class names* (deBruijn et al. 2005b). The pattern approach has potential to be encoded in the graphical support provided to a user, as well as being encoded directly in an automated tool. Mapping discovery could also be achieved through the sharing/learning of mappings from peers. Some initial research on this aspect with respect to a peer-to-peer framework for the sharing of SWRL - Semantic Web Rule Language (SWRL 2004) expressed mappings has demonstrated that this method of discovery holds promise (Conroy 2005).

Analysis

The majority of systems to date have provided a graphical user interface approach, presenting the candidate matches to the user and allowing the user to express the mappings in some way. Although various presentations of the candidate matches have

been used (from the use of icons, through the use of graphs), no literature could be found on the effectiveness of the different presentation types. This underlines the common assumption that exists that a good presentation in itself is sufficient to enable users to identify mappings. Although this thesis makes the same assumption, it is done with the full realisation that further research into this area is required in order to better support the real world deployment of ontology mapping systems. In a similar way there is an assumption behind the Protégé's Anchor-Prompt approach and the deBruijn et al. approach on ontology patterns, that leading the user in the detection of mappings from candidate matches is what is required. However, again no user studies are available to support this position as yet. Thus further research is also required into what kind of mapping detection support would be desirable.

3.2.3 Mapping Documentation and Interpretation

Mappings between elements in schemas are usually expressed as pairs of related entities in some mapping format that is normally output as a separate document. The advantage of a separate document for the mappings is that mappings can be managed independently of the ontologies.

All the systems discussed in section 3.2.2 output mappings in a proprietary format typically aligned with the technology used by the matching system. This is one reason why direct comparison of ontology mapping tools has been a difficult exercise (Noy and Musen 2002). For example, the OntoMerge system (Dou et al. 2002) uses bridging axioms written in first order logic language to express the translation rules between the concepts in the ontologies, and then runs a theorem prover optimised for ontology translation over the ontologies and the axioms. Another example is the MAFRA system (Maedche et al. 2002) that includes a formal representation to specify the mappings. The formalism that is used to describe the Semantic Bridges is based on an ontology specified in DAML+OIL, called the Semantic Bridging Ontology (SBO).

Increasingly the need for an open mapping format is being recognised and proposals have begun to emerge (Euzenat 2004a, Euzenat 2005a, deBruijn et al. 2005a, deBruijn et al. 2005b). For example, XML based formats to enable comparison of the output of a variety of matching tools were developed for the I3CON contest (I3CON 2004) and

EON contest (EON 2004). In order to participate, the entrant systems needed to adapt their output to a given mapping format. Systems from Lockheed Martin, AT&T, Teknowledge, INRIA and University of Karlsruhe took part in the I3CON contest. Systems from Stanford University Fujitsu, INRIA, University of Montreal and University of Karlsruhe took part in the EON contest. Experience from these contests proved positive (Euzenat 2004b) and led to the development of the INRIA ontology alignment format (Euzenat 2005a), which is discussed further in Section 3.4.3. The format can also be rendered into different formats (SWRL, OWL etc.) for the purposes of interpretation. In contrast deBruijn et al. have proposed a generic mapping language that must be grounded in a declarative logical language and thus requires a reasoner. Initial groundings to OWL (Description Logic-based language) and WSML-Flight (a Logic Programming-based language) have been developed.

Analysis

It is useful that the research community has begun over the last year to address the issue of a common way to specify the results of matching algorithms and/or mapping systems. Unfortunately it is too early to determine whether one of the two prominent contenders (that is from INRIA and from deBruijn et al.) will emerge as the basis of a standard format or whether another will be proposed. The advantage of the INRIA format is that it can be used for representing results of match algorithms and results of mappings, which can be rendered into different mapping languages. The advantage of the deBruijn et al. format is that it has a formal basis. What is clear is that further and wider evaluation of the formats are required and that several issues remain to be addressed. One such issue is the manner in which strength/similarity/confidence in a match or mapping should be expressed. This is particularly important when combining the results of matchers from different vendors together or when sharing mappings between systems. Another issue that needs to be explored is whether a match or a mapping can be annotated with information that indicates whether or not the match/mapping is valid for particular application contexts.

In summary, the desire for a common format to express ontology matches/mappings in a manner that would be open to rendering into specific system or technology formats has only recently gained momentum.

3.3 Key Requirements

It was decided to adopt a framework approach to the development of the process and software tools required to support practical ontology mapping. The desire that ontology mapping should be practical, led to a number of challenges and requirements being placed on the framework. Section 3.3.1 describes the key framework challenges and Section 3.3.2 presents the derived requirements.

3.3.1 Key Framework Challenges

Although significant progress has been made in the research community in developing some of the technologies needed for mapping systems, the current challenges have to do with deploying such systems in the real world (Doan et al. 2004).

The first challenge is the provision of a **framework that supports all parts of the ontology mapping lifecycle** (Maedche et al. 2002). A framework consists of process and tools. Such a framework is necessary in order to enable organisations to situate the ontology mapping process within their organisational processes and to aid communication with contributors of solutions with respect to supporting the process. For example, the Unified Process (UP) is a framework defined to support the development of software systems (Jacobson et al. 1999). The artefacts produced and consumed in its process conform to the UML standards (UML 2004). This allows for tools from a variety of vendors to be used in different parts of its process. Another example can be seen in the telecommunications domain, where the availability of the eTOM reference framework (TMF 2005a) for management has provided a valuable reference framework for network management processes for telecom operators to use and for software vendors to use as a reference for implementation, as seen in a variety of success stories (TMF 2005b).

A second challenge is to make the **framework adaptable and extensible**. No two organisations are the same and even within the same organisation there may be different use cases for ontology mapping (Hameed 2004 et al., Euzenat 2004a). Thus the framework needs to be able to support the choice of which activities are to be

included. In addition, the framework needs to support the selection of whether each activity is manual, semi-automatic or automatic (Uschold and Gruninger 2004). Take for example the use cases of the Information Systems, Marketing and Training departments of an organisation. The Information Systems department might use ontologies in the definition of an enterprise data model to aid integration of systems. In this case, the characterisation activities may not exist (as there is no choice but to undertake the mapping) and the mapping and deployment activities are likely to be semi-automatic. The marketing department might use ontologies in the design of the organisation's website providing a portal to their products. In this case, the process mapping between the website ontology and the ontologies used by the product organisation is likely to be fully automatic. The Training department might want to leverage learning resources from the web for integration into corporate training system. In this case the characterisation activities are likely to be automatic, as there would be very precise criteria/policies of eligibility that can be encoded, but the mapping activities would need to be semi-automatic.

In addition, it should be possible to substitute tools and algorithms in the framework. Tools that support semi-automation of activities need to be substitutable as different users will choose different tools depending on the balance that is required between, for example, labour saving versus mapping accuracy (Uschold and Gruninger 2004). Algorithms will need to be substitutable to support deployment in different scenarios (Euzenat 2004a, Do and Rahm 2002). For example in a telecommunications operator organisation, a wide range of different matching tools may need to be available. A general purpose ontology matcher could support web site maintenance. However a task specific matcher would be required to undertake matching of network management information models as linguistic and structural matchers alone have been shown to be inadequate in providing useful candidate matches (Keeney et al. 2005).

A third challenge is to enable the **framework to create relevant mappings**. Just as an ontology is normally defined from some perspective, a mapping increasingly needs to be created from the relevant definer's perspective (whether human or application). Undertaking the entire mapping process at design time using current tools results in fixed mappings determined by some expert user and independent of the context of where the mapping is required. This is of course suited to the kind of organisational

scenarios already mentioned which are quite static in nature or involve the mapping of ontologies in well-defined or bounded areas. The need for more application relevant mappings can be seen in the areas of web service integration (Euzenat et al. 2004), autonomic management (Lewis et al. 2005), pervasive computing environments (Kong et al. 2004), and peer-to-peer knowledge networking (Ehrig et al. 2003). A use case from the SWAP project (Ehrig et al. 2003) provides an example motivational scenario. Here tools are provided in each peer that allows a user to easily create a personal ontology representing a view onto the local file system, emails or bookmarks. Queries are distributed across the peer-to-peer network looking for information on particular topics. The peer needs to create a relevant mapping dynamically between the ontology of the requester and the personal ontology in order to react and respond with information if necessary.

Finally, the majority of the mapping systems in the state of the art rely on ontology reasoners that are by no means widely deployed in organisations at present and whose performance as yet have not been proven (Guo et al. 2004). A part of this third challenge therefore is how to ensure that the determination of mappings by the applications can be achieved without the use of complex technology or APIs.

3.3.2 Key Framework Requirements

From the key framework challenges and a review of the state of the art, the following framework requirements have been derived.

The framework must support an ontology mapping lifecycle (deBruijn et al. 2005a, Maedche et al. 2002). This provides a means by which an organisation can understand the activities involved in ontology mapping and a means by which an organisation can engage in dialogue with potential developers of ontology mapping technology solutions.

The ontology mapping process of the framework must be specified in a manner that is technology-neutral. This allows the organisation to adapt the framework to different situations, deciding which activities are to be automated, semi-automated or human based. A consequence is that *all activities should be defined so that they are potentially*

automatable. This recognises however that automation may be at the expense of effectiveness (Ehrig and Staab 2004).

The framework should support both the full mapping of ontologies and the partial mapping of ontologies. This will allow the framework to support the tasks of ontology alignment, ontology merging/integration and combined use by applications of the ontologies at runtime (Klein 2001).

Any implementation of the framework should support substitution of algorithms and tools. This will allow the framework to be deployed in different situations and adapted to solve different problems. (Uschold and Gruninger 2004).

The framework should be able to support the determination of mappings independent of any one particular application and also support the determination of mappings by an application itself dependent on that particular application's perspective. In other words, mapping generation should be "relevant". On the one hand it may be required that mappings are determined by an expert user and independent of application usage (e.g. the tools described in section 3.2.2). On the other hand, the mapping may need to be generated from the perspective of a particular application (Firat et al. 2004, deBruijn et al. 2005a). *This gives rise to the requirement that the framework should have a clear separation between candidate match generation and mapping determination*. Candidate matches can thus be proposed using a combination of techniques suggested in section 3.2.2, with the determination of what is a mapping, given these candidate matches, being made either by an expert user or by the application itself, depending on the use case.

The framework should support some mechanism to reduce the uncertainty involved in determining mappings from a set of candidate matches. There is a degree of uncertainty in any automatic approach to matching two ontologies arising out of the syntactic representation of ontologies, similarity measure combination, and heuristic approaches to matching (Cross 2003). As a syntactic representation of an ontology cannot completely describe the semantics of different ontologies, automatic matching brings with it a degree of uncertainty (Miller et al. 2000). Different matchers have different ways of calculating similarity measures, thus there is natural uncertainty built

into the combinations. Finally, heuristic approaches by definition have uncertainty attached to them. Thus, a mechanism to reduce uncertainty is necessary in order to be confident about stating mappings. Where a human is involved in determining the mappings, the mechanism should reduce the uncertainty by guiding or drawing the attention of the human to potential mappings arising from the candidate matches (Doan et al. 2004). When an application is involved in determining the mapping itself, the mechanism should provide some reference or certainty points to aid the application in analysing the candidate matches (Campbell et al. 1995). Otherwise the search space becomes too large and confidence in choices becomes too low.

The implemented framework should have a low cost of ownership and have a low impact on the implementation of existing systems and applications. *The matching and mapping formats need to be widely interpretable and not dependent on a particular interpretation architecture.* For example, the format should not to be just dependent on ontology reasoning technologies (Euzenat 2005a).

However, the “holy grail” of fully automatic mapping is not yet achievable (Klein 2001, Noy 2004, Uschold and Guninger 2004) due to the lack of certainty arising out of the matching process and semi-automatic approaches may need to be supported in implementations for some time to come. *Where semi-automatic activities are involved, the framework should reduce the amount of effort required and the error rates involved in mapping (Uschold and Gruninger 2004) and enable the user to undertake the task in the context of the user’s organisational requirements (Doan et al. 2004).*

3.4 Selected State of the Art Systems

In this section, a selection of ontology mapping systems from the state of the art is described and comparisons drawn. The systems have been selected as representative of how the key framework challenges introduced in section 3.3.1 are being addressed, as there have been no systems to date that have addressed all the key framework challenges together. The name of the systems and the reasons for their selection follows:

- The MAFRA system (Maedche et al. 2002) is a good example of a framework, encompassing process and tools, that addresses the first key challenge of

supporting a large part of the ontology mapping lifecycle. This system continues to influence the design of mapping frameworks, such as the ontology mediation management framework of the IST SEKT project (Bruijn et al. 2005a).

- The COMA system (Do and Rahm 2002) is a typical database schema-matching approach that has been recently extended to mapping ontologies in a system called COMA++ (Aumüller et al. 2005). COMA/COMA++ provides a good example of the kind of features that can make an implementation framework adaptable and extensible. This addresses the second key framework challenge.
- The INRIA alignment format (Euzenat 2004a, Euzenat 2005a) is a recent development that contributes towards the achievement of the third key framework challenge related to the creation of *relevant* mappings. In order to generate a mapping based on the results of a number of matchers, a format for documenting matches is required. This format has been chosen for examination, as it is an open format that can be widely interpreted. The INRIA format was developed too late for use in the initial implementation described in this thesis but is under consideration as future work (cf. section 8.3).

In addition, the Protégé tool (Noy and Musen 2003) is described and analysed. Protégé was chosen for examination for three reasons. First, Protégé is the most widely available and used ontology-editing tool. The Anchor-PROMPT plug-in for ontology integration/mapping is provided with the tool. Second, Protégé has been used in our experimentation described in chapter five. Third, the Anchor-PROMPT algorithm (Noy and Musen 2001) inspired part of the design of the OISIN framework.

In the following subsections, each of the selected systems are described and analysed with respect to the requirements presented in section 3.3.2.

3.4.1 MAFRA

The main goal in MAFRA is to transform instances of the source ontology into instances of the target ontology. Semantic Bridges specify how to perform these transformations and are classified into concept bridges and property bridges. Concept

bridges define the transformations between source instances and target instances, whereas property bridges specify the transformations between source properties and target properties. MAFRA specifies an ontology for the specification of ontology mapping, called the Semantic Bridging Ontology (SBO). This is specified in the DAML+OIL ontology language. In the ontology there are classes to represent concepts, relations, attributes, semantic bridges (that defines correspondence between source and target ontologies), services (references resources to describe transformations), transformations (specifies a transformation procedure), rules (for specifying constraints on transformations) and conditions (preconditions for semantic bridge execution). There are also modelling primitives to allow for the aggregation of semantic bridges into a composition.

The MAFRA conceptual framework shown in Figure 3-4 consists of five horizontal and four vertical modules.

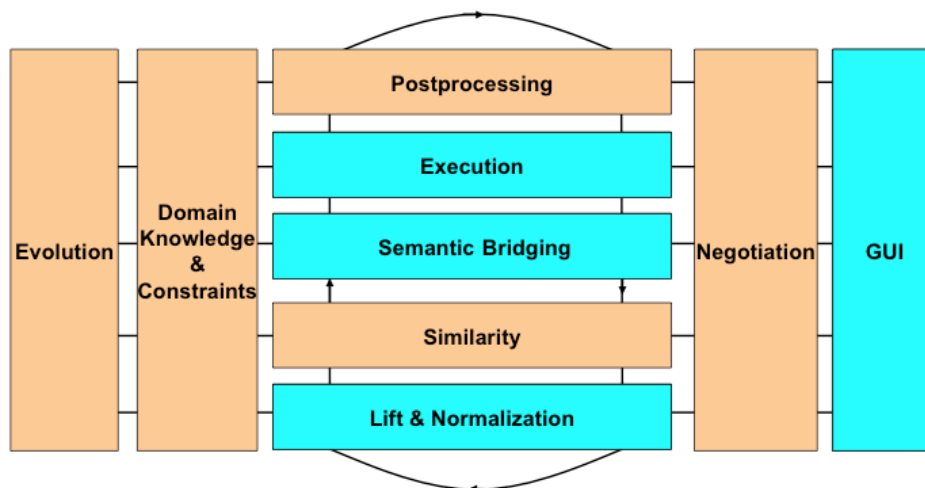


Figure 3-4: Overview of MAFRA conceptual framework (Maedche et al. 2002)

The horizontal modules correspond to five ontology mapping activities that have been defined by MAFRA, namely: Lift & Normalization; Similarity, Semantic Bridging, Execution and Postprocessing. The lift activity translates the ontologies into a canonical format. The normalization activity of the first phase reduces the lexical elements to a common set by tokenising the vocabulary, eliminating stop words and expanding acronyms. The similarity discovery phase uses WordNet (WordNet 2005) to assist in acquiring lexical similarity between terms in each ontology and then seeks similarity based on whether the term is used as an attribute or relationship property. The Semantic Bridging phase is broken down into five activities. First, candidate

bridging pairs for concepts are chosen automatically using heuristics and lexical relations based on similarities found in the previous phase. Second, property bridges are created semi-automatically. In the third activity, concepts that have no similarities discovered are examined to determine if similarities can be discovered through inference. The fourth activity refines the bridges discovered with the aim of improving quality. Finally, the domain expert associates transformation procedures with each bridge. In the execution phase the actual instances of the source and target ontologies are transformed by evaluating the semantic bridges. In the final post-processing phase, the results of execution are examined in order to check and improve the quality of the transformation.

The four vertical modules of MAFRA interact with the horizontal modules during the overall mapping process. The evolution module focuses on maintaining changes to the semantic bridges as the ontologies evolve. The emphasis is on changing the bridges based on changes to the ontologies only, rather than due to new mappings becoming available. The negotiation module aims to establish consensus on semantic bridges between two groups participating in the mapping process. The constraints and background knowledge module interfaces with third party resources (e.g. WordNet) to assist in similarity discovery. Finally the graphical user interface module attempts to ease the difficulties involved in the mapping process.

As shown in Figure 3-5, a service-oriented approach to the architecture of MAFRA is adopted in order to provide a modular, decentralised scheme where independent transformation modules are attached to the system's core modules of bridging, execution, negotiation and evolution (Silva and Rocha 2003). Several transformation services are available in the implementation described: Copy Instance, Copy Relation, Copy Attribute, Concatenate, Split, Attribute Table Translation.

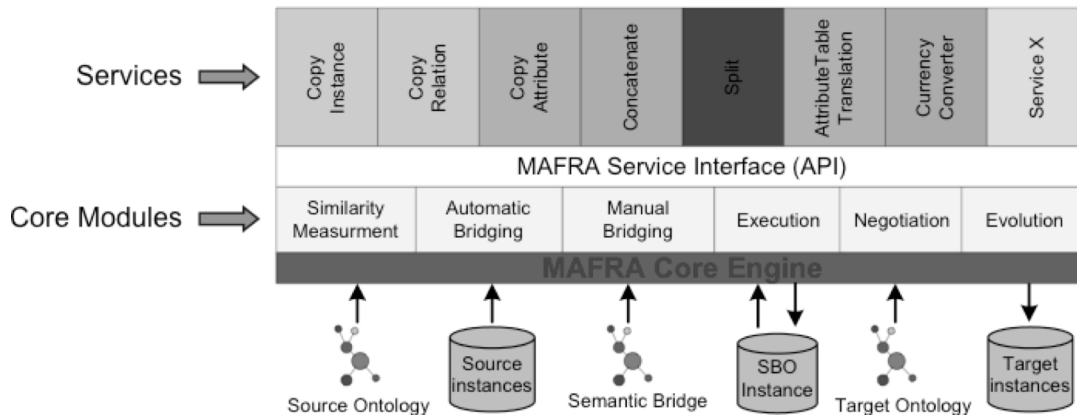


Figure 3-5: MAFRA plug-in architecture (Silva and Rocha 2003)

Analysis with respect to Requirements

The MAFRA conceptual framework includes a breakdown of activities. However, the activities do not support a full lifecycle for mapping as a number of activities are missing from the framework. For example, support for ontology discovery, support for assessing amenability of ontologies for mapping and support for sharing and integration of mappings. In addition, the mapping process is not described independently of the toolkit implementation that has been developed. For example, the execution phase just focuses on transforming instances of ontologies, to the exclusion of other mapping use cases such as merging or querying. The toolkit implementation has been made available as an open source project (Mafra 2004), and in theory this means that MAFRA could support substitution of tools or algorithms. However the API provided has not been designed with substitutability in mind. In addition the toolkit is very poorly documented, and there are a number of features missing that means that the toolkit is not immediately usable (e.g. the ability to create new bridges). The semantic bridges that are the main output of MAFRA can be exchanged easily. However, the ontology on which they are based is not easily extended to incorporate new relations or new kinds of mappings. Furthermore, the mapping format is not widely interpretable as it is custom-made for the MAFRA processing architecture. No information has been published showing how the system compares in terms of reduction of effort and/or error rates of users and/or accuracy of mappings.

3.4.2 COMA++

The original COMA system was designed to undertake schema-matching between elements of two schemas, supporting the mapping of relational database schemas and XML schemas (Do and Rahm 2002). COMA takes a composite match approach in that it combines the results of several independently executed match algorithms. *The match algorithms exploit different kinds of schema information such as names, data types, structural properties, auxiliary information or previous match results. Each matcher determines an intermediate match result consisting of a similarity value between 0 and 1 for each combination of schema elements* (Do and Rahm 2002). The combination of schema elements is achieved through two steps: first where each individual similarity value is aggregated into a combined value; and the second where a choice is made between match candidates for each schema element. New match algorithms can be included in the library and used in combination.

COMA++ (Aumüller et al. 2005) extends the original COMA system (for XML and database schema-matching) to include ontology matching, as well as adding new tool and matching features. New tool features include a graphical user interface, a repository for ontologies and mappings, and a mechanism to compose, merge, and compare different mappings. The new matching features allow a large match problem to be decomposed into smaller match problems and include the addition of ontology matchers and a mechanism to reuse existing match results. The COMA++ architecture is shown in Figure 3-6.

The Repository (MySQL relational database) stores various types of data related to match processing: imported schemas and ontologies; domain-specific taxonomies and synonym tables; definition and configuration of the matchers. The Schema Pool provides different functions to import/export schemas and ontologies to/from the internal directed graph format used.

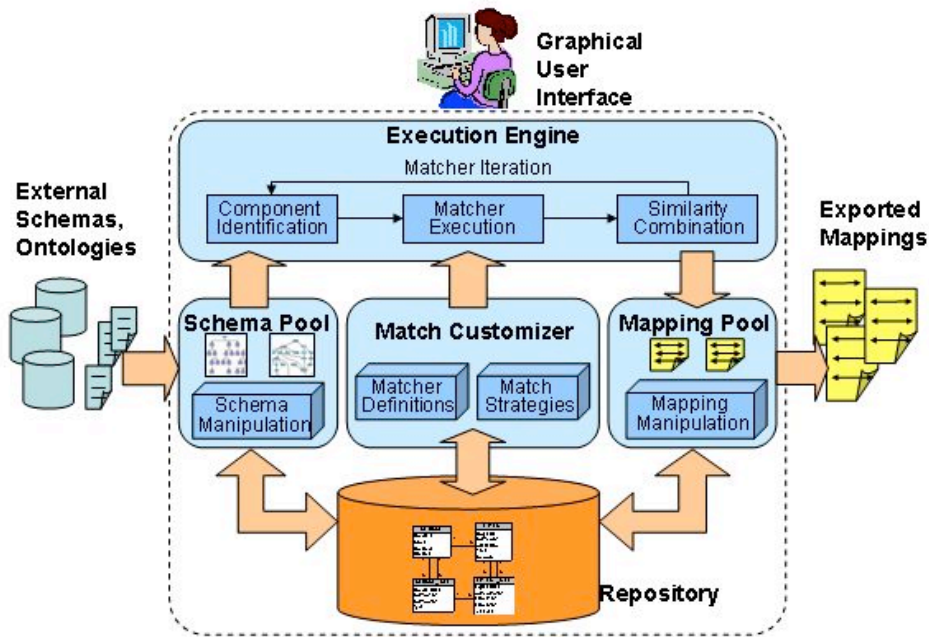


Figure 3-6: Overview of COMA++ Architecture (Aumueller et al. 2005)

As shown in Figure 3-7, the Execution Engine is performed in the form of match iterations consisting of: component identification that determines the relevant schema elements for matching; matching execution which applies multiple matchers from the Matcher Library of fifteen matchers to compute similarities; and similarity combination to combine the matcher similarities and generate the mappings between the components. Each iteration can be configured by the Match Customizer component.

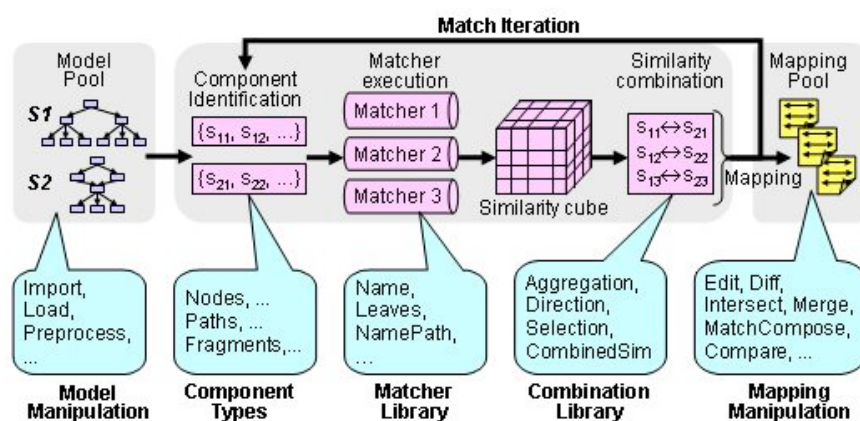


Figure 3-7: Overview of iterations in COMA++ Execution Engine (Aumueller et al. 2005)

Finally, the Mapping Pool stores and provides functions to manipulate all generated mappings and to allow the user add other mappings. Mappings can also be compared, merged, and new mappings created automatically based on transitive relationships. The mappings can be exported in a simple RDF format or a CSV⁵ format. No details on the actual format have been published as yet however.

Analysis with respect to Requirements

COMA and COMA++ are implementation frameworks and do not include a process covering the entire ontology mapping lifecycle. COMA++ does support some activities of a management phase of the lifecycle. However, COMA++ does not support the core of the characterisation phase where the difficulty expected in mapping is evaluated. The original COMA system has shown however that a flexible matching architecture can be employed successfully to the schema-matching problem. Research has shown how the COMA system compares very favourably in terms of accuracy of match results, with respect to seven other schema-matching systems in an evaluation of authors' claims (Do et al. 2002), and with respect to Cupid (Madhavan et al. 2001) and Similarity Flooding (Melnik et al. 2002) in an independent evaluation (Yatskevich 2003). COMA++ promises an equally flexible matching architecture for ontologies but with additional features for mapping manipulation. In particular the clear separation between match and mapping generation offers the possibility for these steps to be done at separate times, resulting in more relevant mappings. However, the claim of the developers of the system, that it solves most problems of the EON 2004 ontology alignment contest with high accuracy, has yet to be independently verified. No evaluation has been published showing how the system compares in terms of reduction of effort and/or error rates of users. It is difficult to evaluate the substitutability features of the system as it has not been made publicly available for use. From the literature it is difficult to assess whether the Graphical User Interface incorporates any particular features beyond numeric similarity metrics that aid the user in evaluating whether suggested candidate matches or mappings should be altered or discarded.

⁵ A comma-delimited format

3.4.3 INRIA ontology alignment format

It is stated that the INRIA ontology alignment format can be very useful in various situations⁶ (Euzenat 2004a, Euzenat 2005a):

- For collecting mappings into libraries;
- For interchanging matching results between matching algorithms;
- For comparing results with each other or with possible “standard” results;
- For rendering results into different formats for exchange purposes.

The general format of an alignment description consists of a level and a set of correspondences.

The **level** is used to characterise the type of correspondence. Level 0 alignment states a correspondence between a pair of discrete entities. Level 1 alignment states that the correspondence is between pairs of lists of entities. Level 2 alignment states sets of expressions of a particular language with variables in these expressions (e.g. first order logic).

The **sets of correspondences** is of a general form: `reference1`, `reference2`, `relation`, `strength` and `id`. `reference1` and `reference2` are URL references to discrete entities, pairs of lists of entities or language specific expressions depending on the level of alignment. A discrete entity can be a class, property, individual or a complex expression (e.g. `fullname` is a concatenation of `firstname` and `surname`).

The `relation` part refers to the relation that is holding between the entities and can range from simple operators (“=”, “>” etc.) to semantic operators (subsumption, incompatibility etc.).

The `strength` denotes the confidence held in a correspondence and the `id` is an identifier for the correspondence.

In addition to the format, an API and a reference implementation has been provided. The API is expressed in Java and can be used for implementing the format and linking

⁶ The following is slightly altered to align with the terminology of this thesis

to matching algorithms and evaluation procedures. The reference implementation supplied includes a command line interface, example data, collection of simple matching algorithms (lexical based), thresholding methods for filtering out weak mappings, and sample output methods for rendering results into different formats (currently OWL, SWRL, XSLT, RDF, C-OWL).

Analysis with respect to Requirements

What is proposed by INRIA targets only the mapping phase in the ontology mapping lifecycle. The main advantages of the format is that it allows for the separation of matching results and mappings, and allows for mappings to be rendered into different mapping formats. The availability of the API as an open source project allows for adaptability and extensibility of the system. However, the associated API reference implementation is still immature and under development. For example, the reference implementation for the renderer of mappings into SWRL only copes with equivalence relations. In addition the format itself is still under development. For example, a disadvantage of the current format is that complex mappings are represented through a series of distinct URIs, making it difficult for a human to follow without tool support. For example, if there was a need to express a mapping such as “ontology one’s *full-name* entity is equivalent to the concatenation of ontology two’s *firstname* and *surname* entities”, this would be expressed as:

Cell#1

Entity1 = <http://someUrl/ontology1/#full-name>

Entity2=<http://thisDocument/#Cell2>

Relation = “equivalent”

Cell#2

Entity1 = <http://someUrl/ontology2/#firstname>

Entity2=<http://someUrl/ontology2/#surname>

Relation = “concat”

In addition, the format allows for the strength value to indicate a level of confidence in the matching. The strength of any particular match/mapping can be increased or decreased depending on what matcher in a particular sequence is applied. As the idea would be that these match/mapping results could be shared, the format could thus be

improved by including in the match result the calculation used or the matchers involved in calculating the strength value.

3.4.4 Protégé and PROMPT/Anchor

Protégé is an extensible, platform-independent environment for creating and editing ontologies and knowledge bases. Extensibility is achieved through a plug-in architecture. *There are five types of plug-ins:* (Protégé 2005)

- *Tab widget* - a user interface tab that appears in the main Protégé window alongside system tabs such as the Classes tab.
- *Slot widget* - appears on a form and is used to view and acquire a value for a slot at an instance.
- *Back-end* - specifies the mechanism that Protégé will use for storage (either as text or in a database).
- *Import or Export* - provides an extensible mechanism for importing and exporting Protégé knowledge-bases in a variety of formats, both file and database.
- *Project* - allows manipulation of a Protégé project and Protégé's project User Interface.

A wide range of plug-ins has been developed for the tool, including one for merging and alignment of ontologies called the Anchor-PROMPT plug-in. The PROMPT algorithm, shown in Figure 3-8, creates an initial list of matches based on class names and then loops through a series of steps, taking one of the PROMPT suggestions selected by the user, undertakes the operation involved, then automatically makes appropriate additional changes and finally updates the suggestions list accordingly. The initial suggestions are based on linguistic similarity matches. The operations available to the user involve: merging classes; merging slots; deep copying of classes; shallow copying (just the class but not its parents or the classes it refers to) of classes.

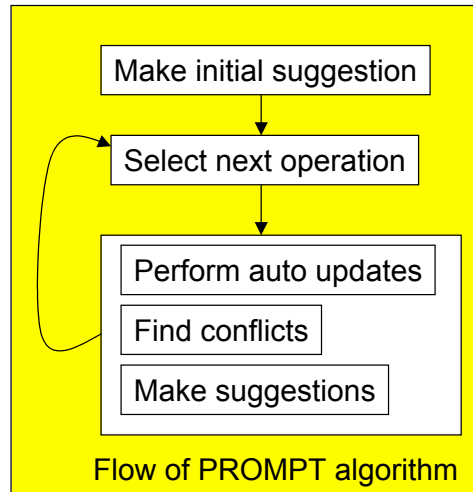


Figure 3-8: PROMPT algorithm overview

The Anchor algorithm is based on the observation that *if two pairs of terms from the source ontologies are similar and if there are paths connecting the terms, then the elements in those paths are often very similar as well* (Noy 2001). In the Anchor-PROMPT plug-in the Anchor algorithm is included as an option in those steps involved in making suggestions to the user, so that instead of just suggesting lexically similar terms, terms that are similar due to structure are also suggested.

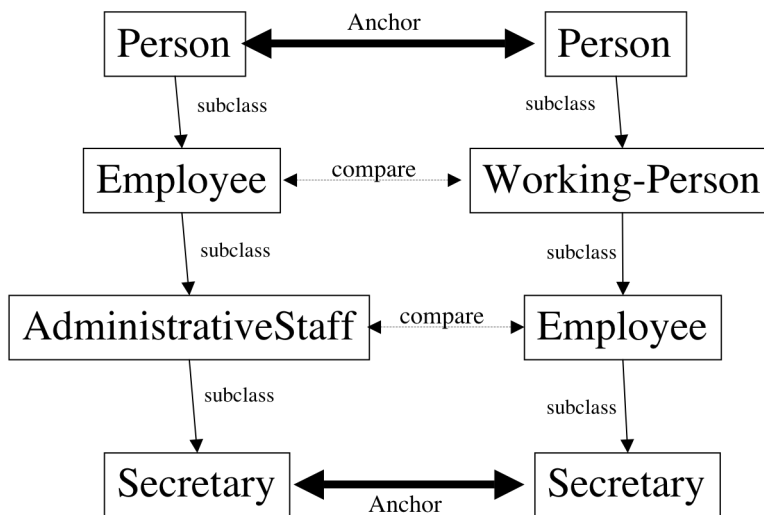


Figure 3-9: Example of anchor paths

First, the algorithm takes as inputs a set of “anchors” which are pairs of related terms in the two ontologies. In our scenario in Figure 3-9, this would be the `Person:Person` pair and the `Secretary:Secretary` pair. Second, all the paths between the two

anchors in each ontology are computed. This yields the paths `Person:Employee:AdministrativeStaff:Secretary` in the university ontology and `Person:Working-Person:Employee:Secretary` in the visiting lecturer ontology. Third, similarity scores are computed for the pairs of terms in the same positions in the paths, that is for `Employee:Working-Person`, `AdministrativeStaff:Employee`. The process is repeated for each pair of paths of the same lengths that have one pair of anchors as originating and another pair as terminating points. Finally, the similarity scores are aggregated from all traversals to calculate the final similarity score, such that terms that often appear in the same position on the paths going from one pair of anchors to another will get the highest score.

Analysis with respect to Requirements

The Anchor-PROMPT plug-in is intended to support the merging/integration of ontologies into a single ontology. Thus the mapping task is entwined in the integration task and it is not possible to document separately a set of mappings. If it was possible to separately document a set of mappings, then the plug-in could be seen as supporting only the mapping phase of the ontology mapping lifecycle. It can be argued that the wizard-like iterative approach of the tool can reduce effort on behalf of the user. This is because the user is presented with decisions to make rather than having to search out correspondences. However in our experimentation in chapter six, it was found that this approach can be disconcerting for a user as the rationale or traceability for why the system is offering certain suggestions has been found to be unclear to the user. The PROMPT system offers an extensible architecture through a plug-in API. Judging by the large set of contributed plug-ins found on the Protégé web site (Protégé 2005), this API is popular and usable. However it was decided against developing OISIN using the Protégé framework as it was considered that the framework is heavily biased towards supporting semi-automation, and one of our key aims was to adopt an implementation framework that would be supportive of manual, automated or semi-automated activities.

3.5 Key Requirements and Selected State of the Art systems

In section 3.3.2 a number of key requirements were introduced, and in section 3.4 the selected set of state of the art systems were analysed with respect to those requirements. This section provides a summary of the analysis through the introduction of a comparison framework based on the requirements.

For convenience, the following summarises the requirements detailed in section 3.3 and introduces the headings used for comparing those frameworks described in section 3.4:

1. The framework must support an ontology mapping lifecycle. The framework should support both the full mapping of ontologies and the partial mapping of ontologies. This requirement is examined under the “ontology mapping lifecycle” heading of the comparison framework.
2. The ontology mapping process of the framework must be specified in a manner that is independent of implementation. All activities should be defined so that they are potentially automatable. This requirement is examined under the “relationship between process and implementation” heading of the comparison framework.
3. Any implementation of the framework should support substitution of algorithms and tools. Where humans are involved in any activities, the implementation should reduce the amount of effort required and the error rates involved in mapping and enable the user to undertake the task in the context of the user’s organisation. This requirement is examined under the “extensibility” heading of the comparison framework.
4. The framework should be able to support the determination of mappings independent of any particular application and also support the determination of mappings dependent on a particular application’s perspective. In other words, mapping generation should be “relevant”. This requirement is examined under the “nature of mapping” heading of the comparison framework.

Allied to this requirement

- a. There needs to be a clear separation between candidate match generation and mapping generation. This requirement is examined

under the “match/mapping separation” heading of the comparison framework.

- b. The framework should support some mechanism to reduce the uncertainty involved in determining mappings from a set of candidate matches. This requirement is examined under the “reduction of uncertainty in mapping” heading of the comparison framework.
- c. The matching and mapping formats need to be widely interpretable. This requirement is examined under the “match/mapping format” heading of the comparison framework.

The comparison framework is presented in Table 3-1.

	MAFRA	COMA++	INRIA	Protégé plugins
Ontology mapping lifecycle	Does not include characterisation activities and only partially management phase	Does not include characterisation activities	Does not include characterisation activities or management phase	Does not include characterisation activities or management phase
Relationship between Process and implementation	Process is tied to implementation	Process is tied to implementation	No explicit process supported	Only ontology integration supported
Extensibility	Limited to fixed set of matcher types	API not publically available	Format enables extensibility	Substitution by source code alteration
Nature of Mapping	What constitutes a mapping determined by user	What constitutes a mapping determined by user	What constitutes a mapping determined by user	What constitutes a mapping determined by user
Match/Mapping separation	Exists and exposed	Exists but how it is achieved not exposed	Achieved through XML format	Separation does not exist
Match/Mapping format	Only map format XML	Only map format RDF, CSV	Map/match formats in XML	Not possible
Reduction of Uncertainty in mapping	Some inference based suggestions provided to user	Some uncertainty reduction based on tools to support reuse of mappings	Not supported	Some inference based suggestions provided to user

Table 3-1: Comparison of Selected SOA systems wrt key requirements

As agreement on what constitutes the ontology mapping lifecycle has not yet emerged, the comparison of the support of systems for the ontology mapping lifecycle is difficult. What can be stated is that MAFRA, COMA++, or INRIA do not as yet include support for the characterisation activities that we contend are important parts of the lifecycle. The Protégé plug-ins of PROMPT and Anchor-PROMPT purely focus on the mapping activity.

MAFRA and COMA++ have defined the ontology mapping process tied to their particular implementation. For example in MAFRA, the Normalisation activity is defined as “minimising the lexical heterogeneity of the ontology contents”. In COMA++ the process is described in terms of the order in which particular components are invoked. For example, the matching process is described in terms of iterations over the “component identification”, “matcher execution” and “similarity combination” components. Such implementation-tied process definitions would make it difficult for an organisation to determine which mapping activities could be supported in different ways. For example, the organisation may wish to have a manual matching process in place for certain domain-specific ontologies, whilst having automatic matching for general purpose ontologies.

Extensibility is supported to different extents by the various systems. MAFRA allows the substitution of different matching tools, but because the mapping format that is output is not extensible, the substituted matching tools must be of the same type. COMA++ claims the ability to add new matchers and choose from a library of matchers but because the API is not publicly available it has not been possible to examine this claim. Substitution of matching algorithms in Protégé PROMPT is only possible by altering the source code. A system that supports matchers that generate and consume the INRIA mapping format, should in theory be able to easily substitute matchers. This has yet to be proven.

The nature of the mappings generated by MAFRA, COMA++ and Protégé are determined and fixed by an expert user through the use of the tool. Although matching information in MAFRA and COMA++ is created separately from the mapping information, this matching information is not available from the tools. As will be seen in chapter four, this matching information could enable an application itself to determine a mapping in the context of its processing.

Finally, only limited support is provided in reducing the uncertainty involved in determining mappings from matching information. In the cases of MAFRA and Protégé, this support is in the form of attempting to infer some new mappings based on user specified mappings and the relationships between terms in the ontology. In

COMA++, an attempt is made to reduce uncertainty by presenting the user with previous mappings for the ontologies.

In summary, none of the systems reviewed, which are typical of systems in the key areas of the state of the art, address all the key challenges or requirements envisaged for a framework that supports practical ontology mapping.

3.6 Implementation Technologies

This section provides an overview of the technologies that were used in the implementation of the OISIN software tools that will be described in chapter five.

To achieve extensibility in the implementation and openness of the framework it was decided that each activity in the process should produce a concrete artefact that would be expressed in an open and structured data representation. The data representation chosen was XML as it is amenable to application processing and human interpretation. This latter point is important as for some deployments it may be appropriate to replace some of the software supported activities with manually based ones (e.g. characterising domain semantics). XML was also chosen due to its popularity and the wide availability of technology for processing. This results in a system where tools and algorithms can be easily substituted or extended.

The tools were implemented using a combination of Java and XML technologies. Java was chosen due to its portability, compatibility with XML and ability to incorporate useful third party libraries. The third party libraries used in this implementation, namely IPSI-XQ (IPSI 2005), Jena (Jena 2005), WordNet/JWNI (WordNet 2005, JWNL 2005) and SWOOGLE (SWOOGLE 2005) are described in Appendix A.

XML technologies were required to process and manipulate the information generated by the various tools. A number of different XML technologies could have been used to achieve the processing required, including SAX (SAX 2005), DOM (DOM 2005), XSLT (XSLT 2005), XPath (XPath 2005) and XQuery (XQuery 2005). SAX was chosen in the semi-automatic tools for reading and writing out XML files, as this is better suited to the underlying object infrastructure used in the graphical user interface.

XSLT could have provided some of the XML transformation functionality that was required but was very limited in its ability to query over several XML documents. Thus the majority of XML processing was achieved using XQuery. XPath was used as an integral part of XQuery. XQuery was chosen for its ability to query XML documents and its script-like capabilities in manipulating XML. This script-like capability meant that different parts of the tools could be rapidly altered when necessary and new tools added easily. Of course, it was known that the use of XQuery would have an impact upon performance due to the additional overhead of query processing. However it is expected that as the required functionality of individual tools become stable that the use of DOM would replace the use of XQuery.

This chapter has presented the state of the art in ontology mapping and described the challenges and requirements upon the framework that was derived. The next chapter describes the design of the process that was developed in response to these challenges and requirements.

4 DESIGN

This chapter describes the OISIN ontology mapping process that has been designed. Section 4.1 discusses the investigation that led to the identification of the four phases of the OISIN process. Section 4.2 briefly describes each of the activities of the phases, including motivation for the activity and possible implementation difficulties. A UML specification of the process is available in Appendix B.

4.1 *Genesis of the Phases*

In order to explore what might be needed in an ontology mapping process, it was decided to undertake an initial case study. The plan for this case study was to: (i) find two ontologies on the web that had a good overlap of domain representation; (ii) manually undertake a mapping between them; and (iii) use the mapping to create some software that would transform instance documents defined using terms from the ontologies.

The first task undertaken was the discovery of ontologies that would be candidates for mapping. It had been decided to concentrate on ontologies specified in OWL, as OWL has been standardised by the W3C and is being rapidly adopted as the way to publish ontologies on the web. A number of avenues were initially pursued to discover overlapping ontologies: search of the DAML website (DAML 2005), search of the Protégé library (Protégé 2005), and examination of the literature. However this did not yield ontologies that were sufficiently overlapping with respect to their domains. The search expanded by using a google query⁷ to identify candidate ontologies. This resulted in approximately one thousand candidates⁸, some of which were actually the same ontologies but which had been renamed. Each of the ontologies was manually examined in search of ontologies with some overlapping domain coverage.

⁷ The search query used was “<http://www.google.com/search?q=filetype:owl+owl>”

⁸ Queried in June 2003. The same query in June 2005 yielded ten thousand candidates

The ontology discovery activity proved to be a difficult task due to the following reasons:

- Some ontologies were uneven in depth (e.g. more taxonomy than ontology) and quality (e.g. incomplete or student homework attempts), but still required some time devoted to examination in order to discover this.
- Some ontologies had a file type of .owl but were in fact basic RDF representations. This meant that these were at best primitive ontology models.
- Some ontologies would not load into the Protégé tool that had been chosen (due to its wide availability and popularity) as the browser for the investigation. Failures mainly were as a result of incorrect use of OWL syntax.
- Identification of overlap or matches was difficult especially where one of the ontologies was more extensive or general than the other.

A number of conclusions were drawn from the experience gained in undertaking this task. First, for those use cases that require searching for overlapping ontologies from the internet, the search itself can be difficult. Even with the recent advent of semantic web specific search engines such as SWOOGLE (cf. Appendix A) it can be a non-trivial exercise as there are many ontologies duplicated across the net that are named differently. The second conclusion drawn was that irrespective of the use case, it would have been helpful to have some form of characterisation of the ontologies to indicate the complexity/difficulty that would be involved in attempting a mapping between them. This led us to introduce a “characterisation phase” in the OISIN ontology mapping process. Later review of the literature, as reported upon in chapter three, found that such a phase does not exist in other ontology mapping processes.

One candidate pair of ontologies covering the University/Research domain was chosen from the ontology candidates for mapping. The first ontology of the pair was published on the web by University of Manchester with fifty-six classes and sixty-six properties (Manchester 2003). The second ontology of the pair was authored by University of Aberdeen with ninety-six classes and ninety-two properties (Aberdeen 2003). This pair of ontologies is the basis for the “visiting lecturer” scenario examples used to illustrate points in chapter three.

Having identified a pair of ontologies to map, the mapping activity involved loading each ontology into a Protégé instance and examining each of the ontologies side-by-side to try to discover correspondences between elements in each ontology. Identifying similar classes in the “class view” of Protégé and similar properties in the “property view” of Protégé was straightforward. However, when using Protégé as a browser it was difficult to discover potential matches between classes and properties, as it required exhaustive examination of each of the ontologies. In addition, in confirming that a class was a candidate match, care needed to be taken to ensure that the is-a structures of the classes were compatible. For example, in our scenario (cf. Figure 3-1) the term `Researcher` is a potential match lexically in both, but `Researcher` is a subclass of `Employee` in the university ontology and `Researcher` is not a subclass of `Employee` in the visiting lecturer ontology, even though the `Employee` class exists.

This experience led us to the definition of the “Mapping Phase”, with the identified need to support activities that would highlight potential matches and highlight potential structural mismatches. The subsequent review of the literature (cf. chapter three) confirmed the need for the highlighting of candidate matches in order for mappings to be identified, and confirmed that checking the mappings for structural mismatches is not explicitly supported in most tools. From our experience, it was thought that explicit support is needed for these two tasks. The need for such support was subsequently confirmed in our evaluation experiment described in section 6.3.

The final task entailed using the mappings to automatically generate an XSLT script that could be used at runtime to automatically transform an instance of an XML document using element/attribute vocabulary drawn from one of the ontologies to an instance of an XML document using element/attribute vocabulary drawn from the other ontology. So for example, an instance in our scenario such as:

```
<ConferencePaper title="Semantic Web">
  The semantic web is a ....
</ConferencePaper>
```

using the university ontology as the vocabulary would get transformed to

```
<Paper has-title="Semantic Web">
  The semantic web is a ...
</Paper>
```

according to the visiting lecturer ontology’s vocabulary.

In our prototype built for the case study, the XSLT used to do the automatic transformation of the instance documents was created itself using an XSLT script. The XSLT script read the mappings expressed in an XML document and created the appropriate XSLT script from that (O’Sullivan 2003).

Experience from this task confirmed the utility of having the mappings expressed in a general XML format such that it could be interpreted in different ways or rendered into different forms at execution time. Thus the need for a separate “Execution Phase” of the framework was identified that could group execution time activities together.

Thus the initial case study was useful in helping to identify the characterisation, mapping and execution phases of the OISIN ontology mapping process. Subsequent investigations and review of literature also established the need for a management phase that was introduced into the process but not implemented as part of this thesis.

In summary, from the initial case study and subsequent review of literature, four phases of the OISIN ontology mapping process were identified: Characterisation, Mapping, Execution and Management. Each phase represents a distinct, coherent set of activities where each activity could potentially be undertaken by different roles and at different times.

4.2 The OISIN ontology mapping process

In Figure 4-1, a summary of the OISIN ontology mapping process is shown as a UML activity diagram. The breakdown of phases into activities and the definition of the activities themselves, were influenced by the following requirements outlined in section 3.3:

1. All activities should be defined so that they are potentially automatable;
2. Support a separation between candidate match generation and mapping determination;
3. Support both the full mapping of ontologies and the partial mapping of ontologies;

4. Support the generation of relevant mappings⁹;
5. Support the reduction of uncertainty in determining mappings from a set of candidate matches.

In order to address requirements one to three, a key design goal was to identify and document the activities in such a way that a continuum of deployment strategies would be supported. Requirements four and five are supported through the introduction of the concept of a “committed mapping”. A committed mapping represents a correspondence between elements in a set of ontologies that has been verified by a human. When mapping needs to take place to support a generic set of applications then the maximum number of committed mappings would need to be determined by an end user. On the other hand, when applications want to determine their own mappings, then a minimal number of these committed mappings would have to be identified beforehand in order for the algorithms used at execution time to have any chance of success (Campbell et al. 1995). Thus the breakdown of the mapping phase activities is derived from the need to identify committed mappings and to analyse the matching information based on these committed mappings.

For easier presentation and explanation, the UML activity diagram of figure 4.1 depicts the OISIN process in a summarised form. In particular, it is shown as if all the activities follow directly on from each other and as if the data entities flow directly between the activities. In reality, the activities are likely to be undertaken at different times and by different roles and so the detailed process includes UML signals and datastores. A more detailed presentation of the UML specification of the OISIN ontology mapping process is included in Appendix B.

⁹ This is a summarised version of the original requirement.

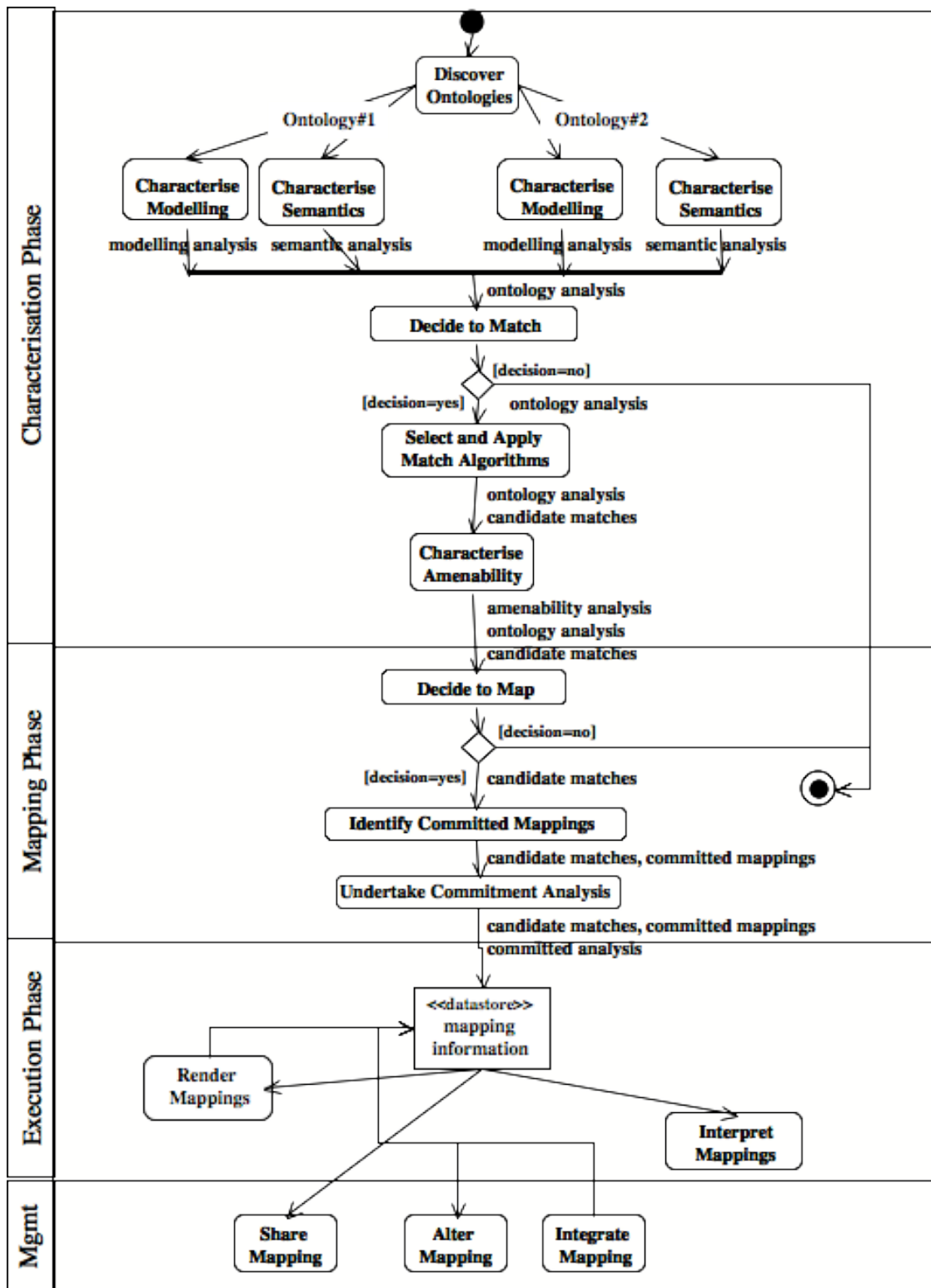


Figure 4-1: Summary UML Activity diagram for OISIN process

In Figure 4-2, a UML class diagram of the key entities shown in the activity diagram is presented. This class model does not show attributes or types, as it is intended that the class model would be extended with concrete entity specifications dependent on the deployment.

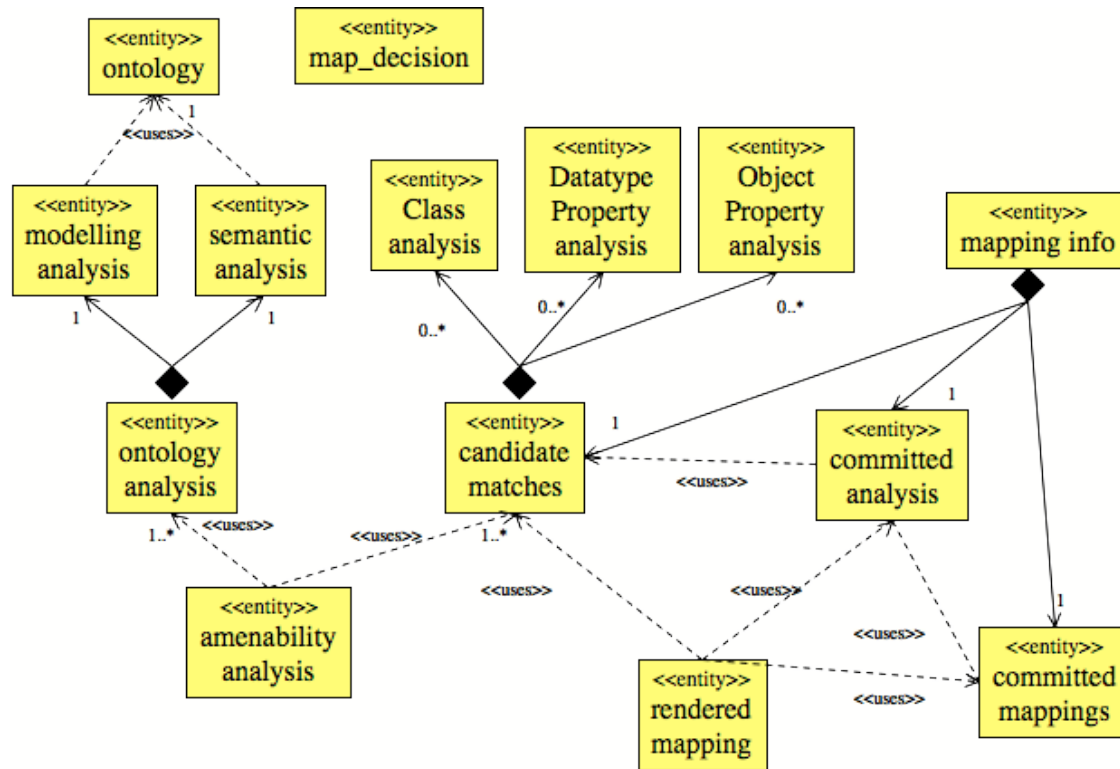


Figure 4-2: UML Class Diagram for extensible information model of OISIN

In the following subsections, the design of each of the phases and activities is described.

4.2.1 Characterisation Phase Activities

The Characterisation phase analyses the ontologies with respect to their amenability for mapping.

Motivation for this phase

This is an important phase for organisations in order to avoid wasted human effort or tool processing in attempting to map incompatible or poor quality ontologies. In addition, in cases where a mapping is required to be undertaken irrespective of the circumstances, then the difficulty of mapping should be anticipated. In current

ontology mapping systems (cf. section 3.2), ontology amenability characterisation is limited to whether the ontologies are in the right format for the system.

Activities breakdown

The breakdown for the characterisation phase activities follows on from our experience of the initial case study, plus the addition of the matching activity. First the ontologies should be individually assessed. This assessment is undertaken along two axes: the form and quality of the ontology (the *characterise modelling* activity) and the nature of the content of the ontology (the *characterise semantics* activity). Second the matching activity is undertaken to analyse the degree of overlap between the ontologies. The availability of the individual ontology analysis information together with the matching information provides a basis for characterising the difficulty that will be involved in mapping.

The main challenge in the design of the breakdown of activities for this phase lay in what activities should be chosen to support the characterisation of the individual ontologies. A design that analysed the ontologies in detail according to some of the quality metrics in the literature (Burton-Jones et al. 2003) would lead to the inclusion of activities that had no direct impact on the amenability of the ontologies for mapping. For example, Burton-Jones et al.'s "consistency metric" activity that examines an ontology to ensure that the terms used had consistent meaning in the ontology. Instead the focus of the activity design was on the evaluation of aspects that would have a direct impact on amenability for mapping. Two activities were designed, one focusing on the format and quality of the ontology and the other on the meaning of the ontology.

Discover Ontologies

This activity should discover the ontologies that will be subject to mapping.

Motivation for this activity

This activity will vary depending on the use case that the ontology mapping is supporting within an organisation. For a systems integration use case, the ontologies are likely to be known and presented to the integrator in this activity. For an open corpus integration use case, ontologies may need to be discovered through web search.

Difficulties involved in implementation

A difficulty for this activity is that ontologies have been published in many different knowledge sharing formats over the years (cf. chapter two). Thus an implementation of this activity will probably require some transformation of the ontologies into a canonical format. However, if one of these formats begins to emerge as the one most commonly used (e.g. OWL), the necessity for a canonical model in an implementation of this activity may disappear.

Characterise Modelling

This activity should analyse the quality of the ontologies and the modelling approach used.

Motivation for this activity

There is currently no accepted way to develop ontologies (Maedche and Staab 2001), and many ontologies *embody systematic errors or massive ontological unclarities* (sic) ... *predestined to yield an end-result that is of dubious merit* (Smith 2001). As was discovered in the initial case study, poor quality ontologies or ontologies with divergent modelling approaches can lead to wasted effort and frustration in attempting to map them. This problem occurs irrespective of whether the ontologies originate within the same organisation or not.

Difficulties involved in implementation

A number of difficulties need to be overcome in the implementation of this activity, especially when attempting to make the activity automatable. First, although there have been a number of efforts to define measures for characterising the quality of ontologies individually, these normally require an expert in the particular ontology domain to undertake the characterisation. For example, in the Burton-Jones et al. classification (Burton-Jones et al. 2003) the accuracy metric that is used requires a judgement as to whether the claims of an ontology are ‘true’. Thus, the challenge is to choose appropriate characteristics that can be evaluated by non-domain experts, or automatically, and/or to use some external sources of quality measurement such as the SWOOGLE website (cf. Appendix A) or some form of federation of trusted entities that will share opinion/evidence of quality.

Second, two ontologies may appear to be modelling an overlapping domain but ontology mismatch may occur due to model granularity or modelling style (Klein 2001). An example of model granularity mismatch is given by (Chalupsky 2000) where one ontology might model cars, trucks and all kinds of transport vehicles and another just models trucks at a fine level of detail including weight, purpose and so on. An example of a style of modelling mismatch is where a distinction between two classes can be modelled using a qualifying attribute (e.g. common class `student` with an attribute indicating `type` of undergraduate or postgraduate) or as separate classes (e.g. `postgraduate_student` and `undergraduate_student`). Characterising the approach that has been taken to modelling just based on the evidence of the ontology is difficult for a human expert and by extension an automated tool. Finally, one aspect of an ontology's modelling that can be analysed is the dimensions of the ontology. Characterising the dimensions of the ontology involves calculating the number of classes, properties and so on within the ontology, as well as calculating path lengths for different kinds of common relationships (such as `is-a` or `part_of` hierarchies). This dimension information is used in the *select and apply match algorithm* activity.

Characterise Semantics

This activity should analyse the content of the ontology in order to characterise the nature of the terms used in the modelling.

Motivation for this activity

The nature of the terms used for naming classes and properties in an ontology vary greatly. The majority of ontologies however compose simple terms together for names. For example the SWETO ontology has terms such as `Manmade_Structure` and `Political_Organisation` (SWETO 2003). Some ontologies are domain-specific but draw terms from general vocabulary. For example a clinical ontology has terms such as `LeftPulmonaryArtery` and `SoleOfFoot` (GALEN 2005). Other domain-specific ontologies do not draw from general vocabulary at all. For example the IUPUI protein ontology with terms such as `ADP-ribosylation` and `Hypusine` (IUPUI 2004). The intention of this activity is to analyse the content of the ontology from the perspective of characterising how amenable the ontology is to either human or tool-based matching. Understanding the nature of the terms can positively impact the later activity of the selection of matching algorithms for execution.

Difficulties involved in implementation

The most obvious choice for fulfilling one aspect of this activity is the WordNet tool introduced in section 3.6 and detailed in Appendix A, as it supports both user and application programmer interfaces. This can be used to determine whether a simple term, or a simple term that is part of a composite term, is drawn from general vocabulary. In this way it can be identified which terms are not domain-specific. The difficulty arises in endeavouring to identify where the remaining terms (if any) of an ontology that are not represented in WordNet come from. Some simple strategies such as attempting acronym expansion (Rahm and Bernstein 2001) may yield results, but in reality access to domain-specific thesauri online will be required. Thus the main difficulty in designing an implementation of this activity is the identification and integration of domain-specific thesauri.

Decide to Match Activity

This activity should decide whether matching should be attempted between ontologies.

Motivation for this activity

Like most processes in an organisation, policies need to govern the expenditure of resources, whether human or computing.

Difficulties involved in implementation

The difficulties involved in implementation of this activity relate to policy design and policy specification. Policy design for this activity involves the identification of the information that informs the “decision to match” and the criteria placed upon that information which will guide the decision-making. This involves selecting the parts of the characterisation information that will be used as the basis of the decision to match. For example, a policy might state that matching should not take place if either ontologies involve composite terms, as the matcher available may not be able to handle composite terms. A key difficulty with supporting policy design is in the refinement of the expressions of high level organisational policy goals into policies that are enforceable by Policy Based Management systems (Beigi et al. 2004).

Select and Apply Matching Algorithms Activity

This activity should support the selection and execution of matching algorithms upon the ontologies.

Motivation for this activity

As discussed in section 3.2.2, there are a wide range of match algorithms that can be applied depending on the type of approach desired, ranging from lexical match schemes right through to semantic model based schemes. In addition it has been shown that match algorithms perform differently depending on the characteristics of the schemas involved. For example, in a comparison of CUPID, COMA and Similarity Flooding SF it has been shown that CUPID performs worst as the schema size increases (Yatskevich 2003). In addition some algorithms take account of thesauri during lexical matching (e.g. CUPID) whilst others do not (e.g. COMA). Thus different choices, or combination of choices, of matchers may be desirable depending on the modelling and semantic characteristics of the ontologies.

Difficulties involved in implementation

Two difficulties arise when designing an implementation for this activity. First, this activity presupposes that matchers will produce matching results in a common format to aid flexible pipelining of matchers and aggregation of results. In systems like COMA and QOM, such formats are proprietary. No agreement has emerged as yet as to what a common format might look like, although the INRIA format (cf. section 3.4.3) may emerge as such a format given its widespread socialisation through ontology matcher contests during 2004. Such a format could enable an organisation to introduce and sequence different matchers, but will require that the organisation has a clear understanding of the meaning of the similarity measure scale of each matcher so that appropriate similarity measure combination strategies can be designed. The difficulty that remains with this approach is understanding the similarity measures when reusing match results gathered from elsewhere.

Second, there is a need to describe the various matchers in a manner that will allow selection based on the kind of semantic and modelling analyses that have preceded this activity. A first step is to distinguish whether the matcher is string, language, constraint, linguistic, reuse, graph, taxonomy, model or combination based (Shvaiko

and Euzenat 2004). A second step is to describe the suitability of the matcher for certain dimensions of ontologies, such as number of classes, number of properties, and average is-a hierarchy path lengths. The problem is that the information needed for this second step needs to be gleaned based on actual experience, as most evaluations to date of algorithms/systems have been based on author claims (Do et al. 2002, EON 2004, I3CON 2004).

Characterise Amenability Activity

This activity should take into account the characteristics of the ontologies and the nature of the candidate matches discovered in previous activities, and elicit key information items that indicate the expected difficulty in mapping.

Motivation for this activity

This activity provides a natural boundary between the characterisation and mapping phases, allowing for the different phases to be undertaken in different parts of an organisation or at different times. In addition a separate activity enables an organisation to choose characteristics of relevance for consideration, especially if third party tools that output a wide range of characteristics are used.

Difficulties involved in implementation

The main difficulty in design of an implementation of this activity is the choice of which information items arising from the characterisation phase are useful for the mapping phase. At the very least, the matching information generated is required. Otherwise, the information choice is dependent on the type of use cases the mapping process is intended to support in a particular deployment. In a purely automatic deployment, information drawn from modelling characteristics with respect to quality may be most important. In a deployment that requires human intervention in the identification of mappings, summary information drawn from the matching information may suggest some of the difficulties that might be faced in undertaking a mapping. For example, in manual mapping, effort is expended in deciding between matches for a particular entity. Thus one potential way to measure complexity is to summarise for each entity the number of matches that have been found by the matching activity. Taking an average of these would then provide an indicator of effort that will be required during the manual mapping process. That is, if there was a high

percentage of matching entities and, of these, nearly every entity had more than one potential match, then it can be reasonably assumed that a large amount of effort would be required in the determination of mappings.

4.2.2 Mapping Phase Activities

The Mapping Phase generates information that is necessary for the execution of mappings between a set of ontologies.

Activities breakdown

Based on our requirements, two use cases can be derived. In the first use case, the generation of mappings is to be determined by an expert user that would generally satisfy the needs of a set of applications. In the second use case, the generation of mappings is dependent on a particular application's perspective and context of usage. The challenge for the design of this phase was in the design of a set of activities that could be used in both use cases. This was achieved through the introduction of the concept of a "committed mapping". A committed mapping represents a correspondence between elements in a set of ontologies that has been verified by a human. The introduction of this concept means that both use cases are achievable with the one set of activities, but with a varying amount of effort required for the *identification of committed mappings* activity.

In the application generic use case, the aim of the *identification of committed mappings* activity would be to identify as many definite correspondences as possible.

For the application specific use case, the aim of the *identification of committed mappings* activity would be to identify the minimum number of definite correspondences. This is because these correspondences would not have been predetermined from the application's perspective and yet some correspondences need to be available in order for the application to reason over the matching information in its own determination of mappings.

Decide to Map Activity

This activity should execute organisational policies to determine if a mapping should proceed between a set of ontologies or if it has been already decided that a mapping should take place, to record the degree of difficulty anticipated.

Motivation for this activity

The motivation for the decide to map activity is identical to that for the decision to match activity. Organisations need to govern the expenditure of human or computing resources through policies. In some use cases there will be a choice as to whether effort should take place or not, and so the decide to map activity will concentrate on encoding the policies that govern that decision making. In some other use cases there will be no choice but to undertake the mapping, and so the decide to map activity will simply record information related to the mapping task that will be allocated. Examples include: recording an estimate of time needed based on the degree of difficulty from the amenability analysis; or recording to whom the mapping task will be assigned.

Difficulties involved in implementation

The difficulties for the *decide to map* activity is identical to that for the decision to match activity, that is, primarily the difficulties involved in policy design and encoding.

Identify Committed Mappings Activity

This activity should identify committed mappings between two ontologies.

Motivation for this activity

The problem with the determination of mappings by applications is that some certainties/points of reference need to exist in order to provide a context for the derivation of mappings from the matching information available. For example, in our scenario `AcademicStaff:Lecturer` (where “:” indicates a subclass) in the university ontology might potentially match lexically and structurally with `Academic:Lecturer-in-Academia` in the visiting lecturer ontology. However given the number of different uses of terms like `Academic` and `Lecturer`, it would be very difficult for a software agent to map these definitively unless some certainty correspondence is indicated somewhere else in the ontology. In this case, if a committed mapping existed between

the `Employee` classes in the ontologies or higher up in their is-a hierarchy between `Person` classes, then the application could have better confidence in making a determination that the following were mappings: `AcademicStaff<->Academic` and `Lecturer<->Lecturer-in-Academia`.

Difficulties involved in implementation

A number of difficulties arise in the implementation of this activity, namely:

- Identifying how a committed mapping can be detected;
- Determining when the activity is finished.

As discussed in section 3.2.2, support for derivation of mappings from candidate match information has primarily been in the form of graphical presentation of candidate matches or leading the user through possibilities via a rule-based system. Thus, the main difficulty with implementation of this activity is in the identification of the type of mapping derivation support that should be provided.

The second difficulty with implementation of this activity is a determination of when sufficient committed mappings have been identified. In terms of the application generic use case this depends whether a partial or full ontology mapping is being attempted and whether exhaustive mapping is required. In terms of the application specific use case, this is much more difficult to decide as by definition, it may not be known prior to this activity taking place which applications may use the resultant committed mappings, or in what context the mappings will be used. Thus, for the application specific use cases the activity also allows for the identification of the minimum number of committed mappings (so as to allow the maximum flexibility to the applications for mapping interpretation) but with as wide as coverage as possible (so as to allow the use of maximum matching information). This can be achieved through the provision of some form of guidance information.

Undertake Commitment Analysis Activity

This activity analyses the matching information in light of the committed mappings.

Motivation for this activity

Committed mappings can be considered as the strongest possible match between entities in the ontologies. As entities in an ontology are, by definition, highly

interrelated, the availability of information about strong matches has implications upon the other information about matches.

Difficulties involved in implementation

The main difficulties involved in implementation of this activity is deciding the impact of a committed mapping on the other match results and how that impact should be expressed. The nature of the impact depends on the kind of matchers applied in the process. A committed mapping may have little impact on lexical match results but may be used to strengthen or weaken other match results such as structural or semantic match results. For example, in our previous example a committed mapping between `Journal` and `Publication` could be used to strengthen any matches of subclasses.

One strength of having the committed mapping analysis activity separate from the capture activity is that it enables any kind of strategy or several different strategies for changing the match information based on the committed mappings. For example, one strategy might be to strengthen the match information for a particular class if it is related in some way to a committed mapping. Another strategy would be to ignore any class matches that are not related to a committed mapping via an is-a relationship.

4.2.3 Execution Phase Activities

In the execution phase, the mapping information (matching information, committed mappings and committed analysis) generated by the mapping phase is interpreted to create mappings that are relevant to the context of usage.

Activities breakdown

The challenge for this phase was the identification of activities that would support two possible execution strategies. One strategy is where the mapping would be created as a result of the application directly interpreting the mapping information generated by the mapping phase. Another strategy would be where the mapping information would be transformed into some standardised mapping format, which would then be used by the application. In order to enable these two strategies, two activities were designed.

In the *interpret mapping* activity, the mapping information may be interpreted directly in order to determine mappings. In the application generic use case, the amount of interpretation needed is minimal, as the committed mappings correspond to the mappings required.

Mapping information may also be rendered into a mapping format using the *render mapping* activity. These mappings may then be used during the mapping interpretation activity.

Render Mappings Activity

In this activity, mapping information is rendered (if required) into different mapping formats.

Motivation for this activity

Having a separate activity for mapping rendering allows for mappings to be presented in a multitude of formats. It may be required to exchange the mappings with other mapping systems that support specific formats (e.g. OWL or SWRL). Also the mapping may be used to tailor some instruction documents used by employees in a workflow. For example in an order fulfilment workflow for a general hardware merchant, a specific mapping could be translated into the instruction: “When you see a request for a VoltageChangingDevice, then this is equivalent to a request for one of our Transformers.”

Difficulties involved in implementation

The difficulty that needs to be addressed here is how to handle the need to transform complex mappings into languages that may not support directly all the necessary expressions (deBruijn et al. 2005b). For example, OWL provides the ability to express one-to-one equivalence mappings of properties through the `equivalentProperty` expression but not complex equivalence mappings (e.g. $\text{salary} = \text{pay} - (\text{pay} * \text{tax_rate})$).

Interpret Mappings Activity

In this activity the mapping information should be interpreted to enable semantic interoperability between applications or people.

Motivation for this activity

The ultimate aim of the mapping process is to enable semantic interoperability for a diversity of use cases.

Difficulties involved in implementation

To support the diversity of use cases, the mapping information needs to be available in human and software interpretable form. In addition, in line with our requirements, it should be a format that can be widely interpreted. A number of possibilities arise for implementation of this activity. One possibility is the adoption of a service-oriented architecture (SOA)¹⁰. In such an architecture, there would be a mapping interpretation service that would be available to humans via a graphical user interface and to applications via a web service. This service would take a request for transformation, interpret the mapping information and return a response. The advantage of this approach is that the mapping interpretation service could be based on reasoning capabilities that otherwise may be too expensive (in processing, cost or management terms) to embed in the client applications. One disadvantage is that the methods offered by the service may not suit the application in question. Another disadvantage is that service may require interactions which are too weighty in order to determine a simple mapping.

Thus, another possibility that may be attractive from an implementation perspective, is to provide the application itself with the means to interpret the mapping information. In order to do this, the technology required for interpretation needs to be mature and widely available.

¹⁰ A service-oriented architecture is a specific type of distributed system in which the agents are "services". A service is a software agent that performs some well-defined operation (i.e., "provides a service") and can be invoked outside of the context of a larger application. Furthermore, most definitions of SOA stress that "services" have a network-addressable interface and communicate via standard protocols and data formats. In essence, the key components of a Service Oriented Architecture are the messages that are exchanged, agents that act as service requesters and service providers, and shared transport mechanisms that allow the flow of messages. (W3C 2003a).

4.2.4 Management Phase Activities

In the management phase, the mapping information (matching information, committed mappings and committed analysis) generated by the mapping phase are managed and maintained until their retirement/withdrawal.

Share Mapping Activity

In this activity the mapping information is shared with others who require ontology mappings for semantic interoperability or who undertake an ontology mapping task.

Motivation for this activity

This activity is needed in order to distribute the generated ontology mappings so that they can be used widely within the organisation (and beyond) and/or enable the reuse of mappings during ontology mapping activities elsewhere.

Difficulties involved in implementation

The difficulties involved in implementation are typical of any distributed system, including the problems of replication, versioning and so on. One model would be for the mappings to be made available by a local web service, and interested parties either periodically poll for new mappings or are informed of their availability. Another model would be to proactively share the mappings with interested parties in a peer-to-peer configuration, or via a decoupled publish/subscribe arrangement. Each model comes with particular difficulties. The Wildflowr system (Conroy 2005), is an exploration of the difficulties involved with a peer-to-peer sharing of ontology. In a similar way, the difficulties involved with a content based routing approach to sharing of ontologically based information is presented in by Lynch (Lynch 2005).

Integrate Mapping Activity

In this activity mappings from outside the organisation are checked to see if they are conflicting or consistent with the existing mappings with a view to adding them to the system.

Motivation for this activity

This activity is a natural follow on from the need for organisations to reduce the effort involved in creation of mappings by potentially reusing mappings from other sources.

Difficulties involved in implementation

One of the difficulties involved in implementation is the fact that mappings may be shared in a variety of different formats, but this should be straightforward to overcome as long as the kinds of mapping information being exchanged is compatible. A more challenging difficulty is in the detection of mapping conflicts and development of a strategy to be employed for their resolution.

Alter Mapping Activity

In this activity existing mapping information is altered or withdrawn.

Motivation for this activity

This activity may be required due to the mappings being erroneous or use cases for the mappings changing or the need to retire a particular mapping.

Difficulties involved in implementation

Alteration of mappings within an organisation face the normal issues involved with the management of any critical information resource used by several applications. Does the requester of change have authorisation? Will the proposed alteration conflict with other existing mappings? How will the change be propagated inside and outside the organisation? How will the mappings being used be safely taken out of service so that the altered ones can be introduced? The implementation of this activity will naturally be influenced by how the Share Mapping and Integrate Mapping activities are implemented. The functionality for propagating change and conflict resolution being provided by each of the Share Mapping and Integrate Mapping activities respectively.

This chapter has presented the OISIN ontology mapping process, and described each activity in terms of an overview, motivation and expected difficulties involved in implementation. Chapter five proceeds to describe the implementation of the software tools designed to support the process.

5 TOOL IMPLEMENTATION

The OISIN ontology mapping process described in chapter four is designed so that it can be implemented in many different ways. This chapter describes the implementation of the supporting software tools that has been undertaken as part of this thesis research. During the implementation of the design, the following requirements identified in section 3.3 were addressed:

1. Should support substitution of algorithms and tools;
2. Matching and mapping formats need to be widely interpretable and not dependent on particular interpretation architectures;
3. Should reduce the effort required and the error rates involved in mapping for a user, and enable the user to undertake the task in the context of organisational requirements.

The scope of the implementation was to support the activities in the characterisation, mapping and execution phases of the ontology mapping process. Implementation of tool support for management activities has begun in research work allied to this thesis, and is also the subject of future work (cf. section 8.3).

Section 5.1 describes the OISIN applications developed and their support for different activities of the OISIN ontology mapping process. The implemented support for each activity is discussed in terms of strengths, weaknesses and limitations. Section 5.2 presents example screenshots from some of the semi-automatic tools implemented, in particular OisinAC-semi-v1, OisinMA and OisinAC-semi-v2.

5.1 *The OISIN applications*

In this implementation the process is supported by eight different types of application, as shown in the UML package diagram of Figure 5-1. As can be seen in the diagram, the applications do not call each other directly but depend on information that has been processed by other applications being available in the information model. This approach allows for the different activities of the process to be executed at different times and to be executed manually, semi-automatically, or automatically. This approach also facilitates the easy addition or substitution of applications.

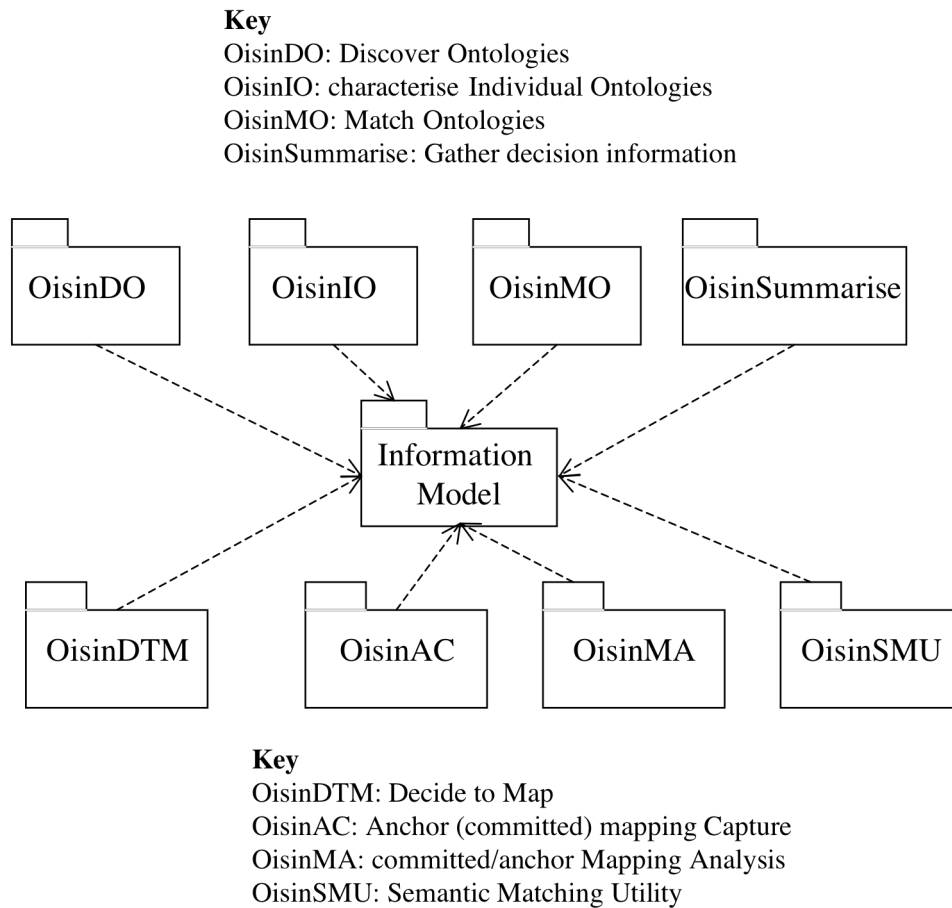


Figure 5-1: UML package diagram for OISIN applications

OisinDO discovers and transforms the ontologies into a canonical form. OisinIO characterises an individual ontology. OisinMO selects and applies different matchers to the ontologies. OisinSummarise gathers the information necessary in order to determine the difficulty in mapping. OisinDTM supports the decide to map task. OisinAC assists the identification of committed mappings and OisinMA analyses the match information in the light of the identified committed mappings. Finally, OisinSMU is a service to support applications interpreting the mapping information generated.

Figure 5-2 shows the relationship between the different applications and the OISIN mapping process. The UML package diagram notation¹¹ is used to show the scope of an individual application. Some of the currently implemented tools support more than one activity of the process.

¹¹ A large yellow rectangle which indicates the scope of the package, with a smaller yellow rectangle on top which indicates the package name

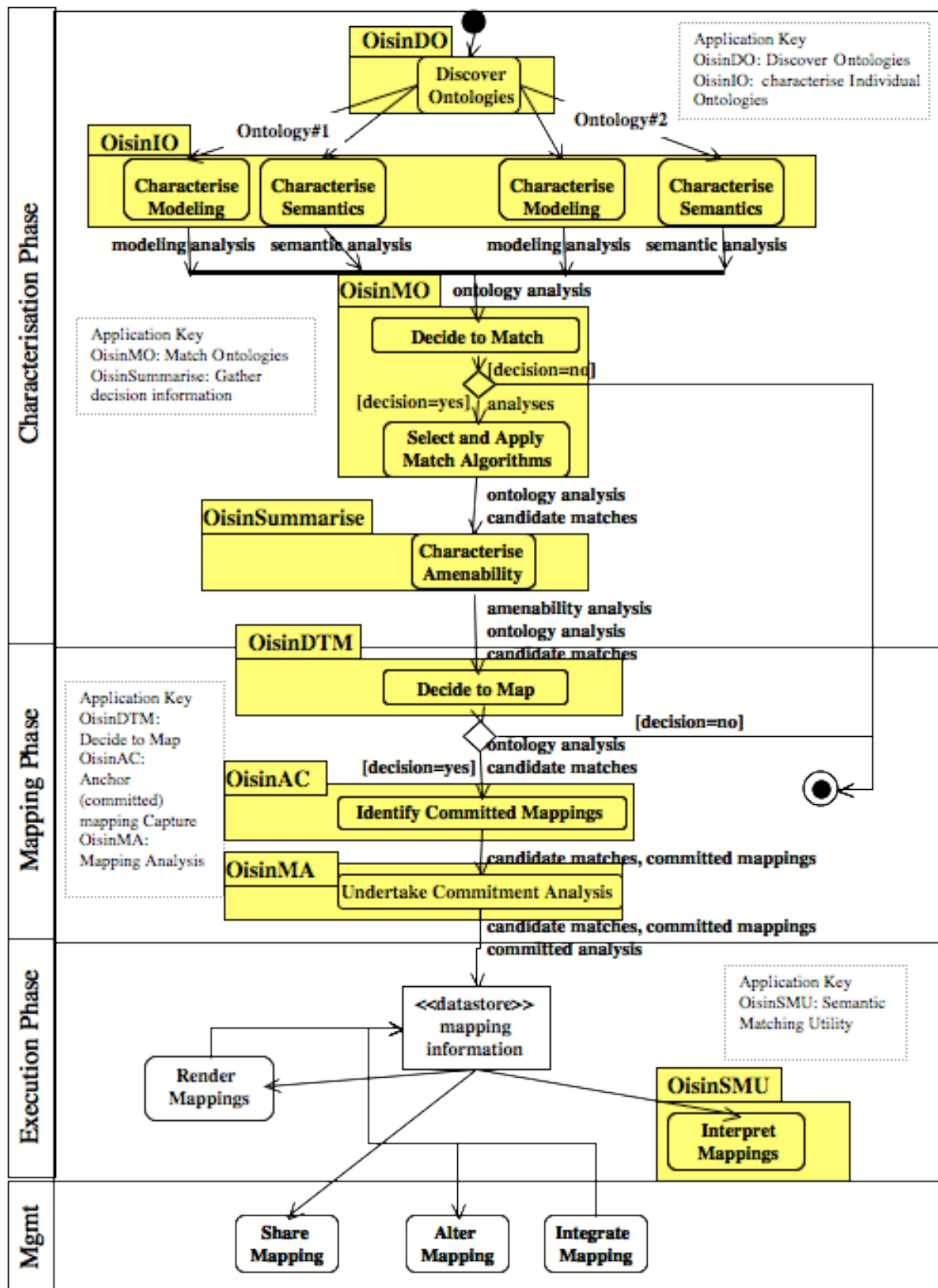


Figure 5-2: OISIN applications/process relationship

In order to examine if the designed process could support a continuum of deployment strategies, two organisational deployment scenarios were used. The first involves a situation where some of the activities would be semi-automated. The second deployment scenario involves a situation where an organisation requires the entire ontology mapping process to be automated.

The suffix “-auto” and “-semi” is used to distinguish where necessary between an automated version of a tool and a semi-automated version respectively (e.g. OisinDTM-semi and OisinDTM-auto). Where no suffix is denoted, the tool is an automated version (e.g. OisinIO).

The first scenario where some of the activities would be semi-automated, is likely to be a popular deployment scenario in the near term, given the known difficulties in the automatic derivation of mappings. In addition, the activity to determine the potential difficulty in undertaking a mapping task is not supported in the current state of the art and thus it was considered important to research semi-automatic support for this activity. This scenario involved the implementation of semi-automated tools to support the Decide to Map, Identify Committed Mappings and Undertake Commitment Analysis activities, with automated tools in support of the other activities (as shown in Figure 5-3). This deployment was used in the autumn of 2004 during the experimentation described in chapter six. Second versions of some of the tools were developed during the summer of 2005, namely: the OisinMO tool to support multiple matchers and the OisinAC-semi tool based on feedback from the experimentation.

In the second deployment, the semi-automatic activities were replaced by automated versions. The reader should note that these automated versions are unsophisticated as the focus of this deployment was merely to demonstrate that a fully automated implementation of the OISIN process is feasible.

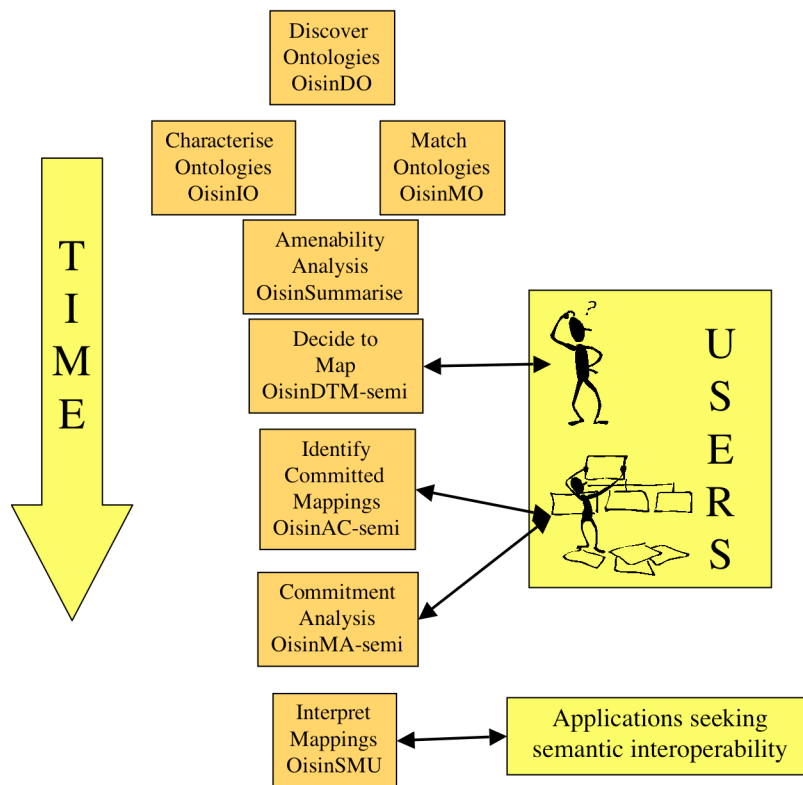


Figure 5-3: Overview of tools for first implementation deployment

Note also that in Figure 5-3, there is no designation of when things happen but rather just an indication that a certain sequence must be followed. Thus semi-automatic activities could just as easily take place at “runtime” (that is in reaction to an application request at runtime) as it would at “design time” (that is when the ontology mapping is planned in anticipation of runtime application requests). For the purposes of this section, it is assumed that the first deployment involves a situation where all but the mapping interpretation activity is undertaken at design time, and the second deployment involves a situation where all the activities must be undertaken at runtime.

To aid readability, the various applications and versions implemented will be described in the context of each activity of the process.

Section 5.1.1 discusses the Information Model used to support the implemented process. Section 5.1.2 provides an overview of the XQueries that were implemented. Sections 5.1.3 to 5.1.11 describes, for each activity, the implementation of the supporting application(s).

5.1.1 Information Model

The UML class model in the implementation extends the UML class model presented in chapter four, see Figure 4-2, and in Appendix B. It is not feasible to provide a single figure presenting the entire model that would be legible. Instead UML class models for each of the individual entities that have been extended are provided in the sections that describe the implementations that create them. Each entity is implemented as an XML document defined using a DTD in accordance with the UML class entity definition. These XML DTDs can be found in Appendix C.

5.1.2 Implementation technologies

As seen in Table 5-1, a number of XQuery functions were developed (within a namespace of “SU”¹²), grouped into XQuery files (.xq) and invoked from the named Java application. For example *SU:OWL_langpattern* is an XQuery that is stored in a file called *LangPattern.xq* and is invoked by the *OisinIO* application.

OisinIO	OisinMO	OisinMA
LangPattern.xq SU:OWL_langpattern SU:OWL_count_keywords Wordnet.xq SU:wordnet lex_analysis.xq SU:lex_analysis SU:detect_pattern SU:pattern OWL_DomainSemantics.xq SU:OWL_DomainSemantics SU:OWL_Reuse OWL_Dimensions.xq SU:OWL_Dimensions SU:top_concepts SU:count_OWL_concepts SU:count_OWL_atts SU:count_OWL_rels SU:OWL_shape SU:OWL_children SU:OWL_width SU:concept_dimensions SU:leaves	Match.xq SU:match_term SU:match_compositeterm SU:get_classes SU:match_onts Combine.xq SU:get_term_analysis SU:combine OisinSummarise AmmOntSummarise.xq SU:AmmOntSummarise	Anchor_analysis.xq SU:anchor_match SU:get_anchor SU:anchors SU:lowanchors SU:showpath SU:get_anchored_paths SU:count_length SU:top_concepts SU:check_anchor_paths SU:get_all_anchor_paths SU:anchor_analysis SU:analyse_class SU:analyse_properties SU:matches_analysis Integrate.xq SU:integrate OisinSMU OisinSMU.xq SU:OisinSMU

Table 5-1: XQuery functions developed

Execution of all the XQueries follows the common pattern shown in the UML sequence diagram of Figure 5-4. This common pattern queries and manipulates the XML files that comprise the information model.

¹² “SU” stands for Semantic Utility which was the original name for the OISIN tools.

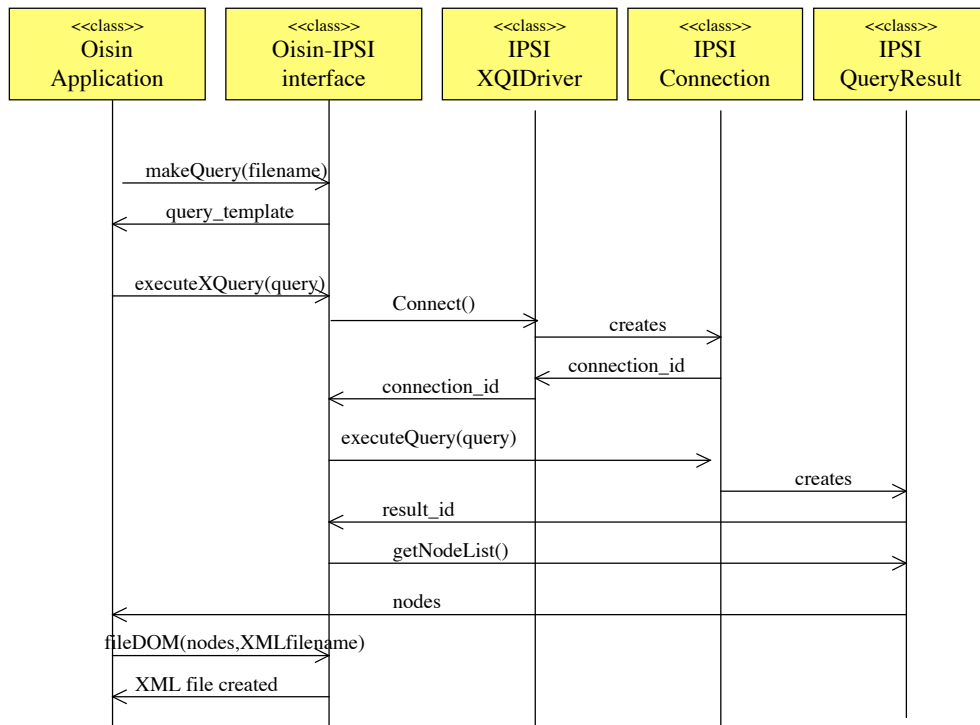


Figure 5-4: UML Sequence Diagram between Oisin applications/IPSI interface/Information Model

As can be seen from the diagram, the application first requests the Oisin-IPSI interface to read in the standard query for this task from a file. The application then tailors the parameters of the query depending on the name of the particular ontology and so on. When the application invokes the `executeXQuery` method of the Oisin-IPSI interface a connection is setup to the IPSI-XQ engine via the IPSI XQIDriver and the query passed to it. Typically the query is over one or more XML files in the Information Model. The result is passed back as a `QueryResult` object that is then transformed into a series of DOM objects by the `getNodeList()` method of the `QueryResult` object. The application finally requests the Oisin-IPSI interface to write this DOM representation of the result out to a requested XML file.

5.1.3 Discover Ontologies – OisinDO application

Ontology discovery is very dependent on the situation or task that the mapping implementation is supporting in an organisation. For ontology discovery from open corpus, then connection to the SWOOGLE web service (cf. Appendix A) could be useful to incorporate in this application. Local ontology discovery could involve

library or file system searches. However, for the purposes of this implementation, it was assumed that the ontologies are presented to the application for processing rather than having to be discovered.

Over the years, ontologies have been represented in a variety of different syntaxes, ranging from proprietary syntaxes (e.g. Protégé’s “pont” representation) to standards based syntaxes (RDF, RDFS, OWL). Even within the same language, different syntax can be used to express the same idea. A simple case in OWL for example would be where a property can be defined independently of a class and then associated with a class, or can be defined as part of a class definition.

In order to cater for this diversity of ontology representation, this activity converts the ontologies into a canonical format suitable for processing by OISIN tools. For this implementation, it was decided to cater for standards-based syntaxes only, as most proprietary ontology editing tools can now export their ontologies in one of these syntaxes.

The canonical format is constructed according to the UML class diagram shown in Figure 5-5. The canonical format represents the common concepts of interest for OISIN processing. This canonical format lists each class with its data properties, relationships, subclasses and superclasses. The subclasses and superclasses are listed for each class to enable is-a hierarchy path processing later in the mapping process.

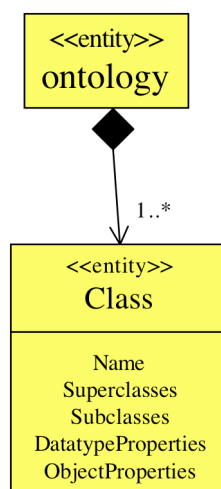


Figure 5-5: UML class diagram for ontology entity

To undertake the conversion into a canonical format, the Jena Java library (see appendix A) is used by the OisinDO application to load a representation of the ontology into a model in memory. This is achieved using the following method calls that first instantiates an in-memory object for the model and then reads the ontology file into that object:

```
OntModel m = ModelFactory.createOntologyModel();  
m.read("file:onts/" +ontFile);
```

This model allows access to the ontology statements via a collection of RDF data represented as a graph. The model is then queried by the OisinDO application, using various method calls available from the Jena library. The methods that are possible to invoke on the model depend on both the asserted statements in the underlying RDF graph, and the statements that can be inferred by the OWL reasoner being used. For example the methods `listSuperClasses` and `listSubClasses` enumerate the direct superclasses and subclasses of a particular class in the ontology. The OisinDO application uses the methods to access information in the ontology and to write it out as XML data in the OISIN specific canonical format.

Strengths/Weaknesses/Limitations

A weakness of the canonical format designed is that only an is-a inheritance taxonomic analysis is undertaken. This limits the possibility of using structural matchers based on other taxonomic relations (such as containment) in the current implementation.

On the other hand, a strength of the XML based canonical format is that the implementations of the subsequent mapping activities do not require the incorporation or use of an ontology reasoner. As the performance, ubiquity and maturity of ontology platforms improves then future implementations of the mapping process may dispense with a canonical format.

In addition, extending the canonical format to include other information from the ontology is easily achieved; as such extensions would not affect XML parsing or the XQueries being executed elsewhere in the process. This is because the XML processing is based on relative rather than absolute data formats.

The current implementation is limited to OWL ontologies only, although it could easily be extended to RDFS and DAML+OIL ontologies due to the capabilities of Jena. A risk however is the reliance on Jena, as Jena and its reasoner have been recently found to have poor performance on large OWL datasets (Guo et al. 2004). In addition, the implementation currently uses the default OWL reasoner. A consequence of this is that transitive or symmetric properties are not fully enumerated in the canonical model. This means that in some circumstances, some properties are not associated with a class when they should be, and the subsequent characterisation and matching activities will be working with some classes that have incomplete definitions.

5.1.4 Characterise Modelling – part of OisínIO application

The objective of the characterise modelling activity is to analyse the quality of an ontology and the modelling approach used. The challenge in the implementation of this activity was in choosing metrics that could equally well be automatically computed or determined by a non-domain expert. This section is subdivided into the metrics that were used to characterise the sophistication of the modelling and ontology quality. A UML class diagram illustrating the information entities involved is provided in Figure 5-6.

Sophistication of Modelling

In order to provide an indication of the sophistication of modelling, three aspects of an ontology were characterised. First, the application characterises how many language constructs are used in the ontology specified. This is achieved by querying the XML schema or DTD that specifies the ontology language to retrieve the terms used in the language. The ontology language terms used in the actual ontology under consideration are then analysed. This results in the production of a Language Analysis information object that is realised by an XML file named *LANGoooo*, where oooo indicates the ontology.

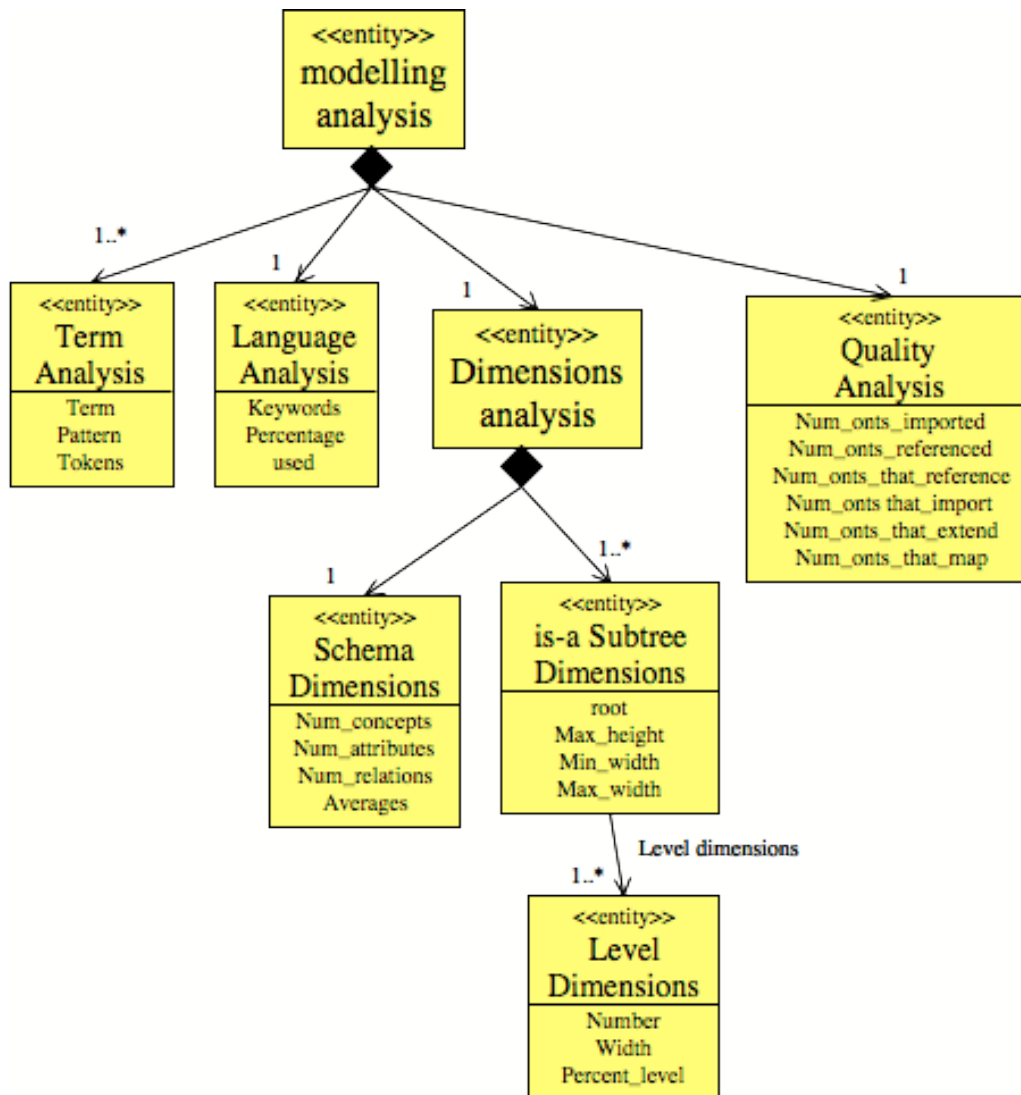


Figure 5-6: UML Class diagram for Modelling Analysis

Second, the nature of the terms (that is whether simple or composite) is analysed. A large number of composite terms (that is where a number of individual tokens are combined via hyphens, underscores or capitalisation) can cause difficulties in matching entities or identifying possible mappings, as the meaning of the combination can be quite different from the consideration of the meaning of the individual parts. For example, the meaning of `Student_Presentation` (possibly an assignment) would be quite different to `PresentationToStudents` (possibly a lecture) even though the two term combinations have two matching individual tokens. The application creates for each term used in the ontology a term analysis object that indicates the name of the term, and if the term is composite, the pattern detected and the individual tokens that constitute it. For example, `Student_Presentation` would have a pattern of “underscored” and have tokens `Student` and `Presentation`. Detection of composite

patterns required the implementation of several regular expressions that identified particular patterns for composite terms.

For example, the XQuery code in Figure 5-7 detects whether a hyphen/underscore combination (e.g. term1_term2-term3), underscore (e.g. term1_term2), hyphen (e.g. term1-term2) or middle capitalisation (Term1Term2) is used to separate terms in a composite term. This results in the production of Term Analysis information objects being realised in an XML file named *LEXoooo*, where oooo indicates the ontology.

```
declare function SU:detect_pattern($term)
{
  <pattern>
  {
    if (fn:matches($term,".+_.+") and fn:matches($term,".+-.+"))
      then
        "hyphenenated_underscored"
    else if (fn:matches($term,".+_.+"))
      then
        "underscored"
    else if (fn:matches($term,".+-.+"))
      then "hyphenated"
    else if (fn:matches($term,"(^[a-z]|[A-Z])[a-z]+([A-Z][a-z]+)+"))
      then "middleCapital"
    else
      "none"
  }
  </pattern>
};
```

Figure 5-7: Snippet of Detect Pattern XQuery

Third, the dimensions of an ontology are calculated. This information can provide an indication of the modelling approach used for the ontology. For example, an ontology with a very low average of properties per class may indicate a taxonomical approach to modelling, whereas one with a high number of attribute properties and very low number of relationship properties may indicate a poorly intra-related ontology. Most of the information is straightforward to gather, including:

- number of classes, attributes and relationships;
- minimum, maximum and average attributes per class;
- minimum, maximum and average relationships per class.

However, the dimensions of the is-a tree of an ontology required a number of recursive XQueries to be implemented. The is-a tree description is composed of descriptions of is-a subtrees dimensions. Each level in each subtree is described in terms of the

number of classes appearing at that level and the percentage of the ontology that represents. This information is used during the match algorithm identification and selection activity.

Figure 5-8 shows the three queries required to calculate the is-a tree dimensions: OWL_shape, OWL_children and OWL_width. OWL_shape (lines 1 to 41) first calls OWL_children (lines 42 to 54) to return all the classes at each level within a subtree. Then for each subtree the number of classes (width) at each level is calculated using OWL_width (lines 55 to 74). The query was designed in this way in order to reuse the OWL_children query that is also used in the match algorithm described later.

```

1. declare function SU:OWL_shape($root,$ont,$dim)
2. {
3. let $list := <top> {SU:OWL_children($root,$ont,0)} </top>,
4. $height := max($list/level)
5. return
6. (
7. let $width_list := <width> {SU:OWL_width($height,$list,1,$dim)}
   </width>
8. return
9. (
10.  "
11.  ",
12.  $width_list
13.  , "
14.  "
15.  ,
16.  for $most in $width_list/level
17.  where $most/percent_level = max($width_list//percent_level)
18.  return
19.  (
20.  <max_height>
21.  {
22.  $height
23.  }
24.  </max_height>,"
25.  "
26.  ,
27.  <majority_height> {$most/number/text()} </majority_height>
28.  ,
29.  <min_width>
30.  {min($width_list//width_level)}
31.  </min_width>,"
32.  "
33.  ,
34.  <max_width>
35.  {max($width_list//width_level)}
36.  </max_width>,"
37.  "
38.  )
39.  )
40.  )

```

```

41. };

42. declare function SU:OWL_children($findclass,$scanont,$level)
43. {
44.   for $subclass in
45.     doc($scanont)/rdf:RDF/su:Class[su:Name=$findclass]/su:SubClass/text(
46.     )
47.   return
48.   (
49.     let $level := $level + 1
50.     let $lev_all := (<level> {$level} </level> ,<concept>
51.       {$subclass} </concept>,"
52.       ",SU:OWL_children($subclass,$scanont, $level)) return ($lev_all)
53.   )
54. };

55. declare function SU:OWL_width($height,$list,$level,$dim)
56. {
57.   if ($level <= $height)
58.   then
59.     let $nodes := $list/level[text()=$level]
60.     return
61.     (let $lev2 := $level + 1
62.     return
63.     (
64.       <level>
65.       <number> {$level} </number>
66.       <width_level> {count($nodes)} </width_level>
67.       <percent_level> {count($nodes) div $dim//num_concepts}
68.       </percent_level>
69.       </level>,
70.       SU:OWL_width($height,$list,$lev2,$dim)
71.     )
72.   else
73.     ""
74.   };

```

Figure 5-8: Snippet of Ontology Shape XQuery

The schema dimension information object and the is-a subtree dimensions information objects are realised in an XML file named *DIMoooo*, where oooo indicates the ontology.

Quality measures

For the implementation it was decided to use the degree to which the ontology reuses other ontologies or is itself reused, as a measure of quality. This is achieved by

consulting SWOOGLE information available about the ontology. If the ontology is published on the web, then SWOOGLE will have some information on it. Naturally, the ontology under investigation may not be published on the web (and therefore not visible via SWOOGLE), in which case the quality may need to be measured using some other organisation specific criteria, such as the author credentials or quality of the provider of the ontology. In any case it would be important that the quality measurements could be automatically derived.

A snapshot of the mySQL relational database which underlies the SWOOGLE website¹³ was downloaded and is queried using Java Database Connectivity API (JDBC). In particular two relational tables are of interest, namely `ontology` and `ontologyrelation`. The `ontology` table keeps information about the ontologies that the web crawler discovers (e.g. file type, number of triples, number of classes, number of properties etc.) and the ranking for each ontology calculated by SWOOGLE. The `ontologyrelation` table stores information about the interrelationship of ontologies discovered during the crawling. Information is kept on each ontology as to whether the ontology:

- Imports other ontologies;
- Extends other ontologies (e.g. through `subClassof` statements);
- References other ontologies (e.g. through `range` statements or `sameAs` statements);
- Maps other ontologies (e.g. through `equivalentClass` statements).

This information can provide some indication of quality, in that ontologies that do not reference other ontologies or are not well referenced themselves, might be considered of poor quality in some situations.

In the current implementation, the application queries over the two tables using two SQL queries:

¹³ A Web Service interface to SWOOGLE has since become available and the database is no longer available for direct download

1. `select o.Url,ol.Url_source, ol.Relation,ol.Count from
Ontology o, ontologyrelation ol where o.Url=ol.Url_dest and
o.Url = ontology_url;`
2. `select o.Url, o.rank1,o.googlelinks from Ontology o where
o.Url = ontology_url;`

The first query returns the ontologies (*ol.Url_source*) that relate to the *ontology_url*, the nature (that is use, extends etc.) of that relationship through the *ol.Relation* attribute and the frequency of that relationship through the *ol.Count* attribute. The second query returns, for an ontology referred to by *ontology_url*, the SWOOGLE rank and the rank number according to GOOGLE for that ontology.

Strengths/Weaknesses/Limitations

A strength of the OISIN process as designed, is that further modelling and quality measures can be easily added or removed with the only impact being on the activity that summarises the measures for the determination of mapping difficulty. This provides for useful extensibility opportunities.

The current implementation is limited in the number of measures that are applied. The modelling measures are unsophisticated in that they concentrate on purely syntactic characteristics of the model (language constructs, nature of terms and dimensions). In a similar manner, the quality measures can also be considered syntactical. The rationale for choosing such syntactic approaches for inclusion was the desire to have an implementation that could be automated for the characterisation phase activities. Therefore, a greater range of modelling quality measures was sacrificed to ensure automation capabilities.

5.1.5 Characterise Semantics – part of OisinIO application

This part of the OisinIO application determines how many terms/tokens of the ontology are known/unknown in WordNet and returns for each term/token whether it is a verb, adjective or noun, and its synonyms. This analysis shows the degree to which the ontology uses non-domain-specific terminology, as well as providing useful information for matching used in the latter part of the characterisation phase.

For each simple term, composite term and token¹⁴ of a composite term, the application queries the WordNet online thesaurus to check whether the term is known or unknown. The application accesses WordNet through the JWNL application programmers interface (cf. Appendix A). In particular the main method used to achieve this is `Dictionary.getInstance().lookupIndexWord(pos,word)` where *pos* states which part of the dictionary to search (e.g. noun, adjective etc.) and *word* is the term to be searched for. If a term is known, synonyms for that term for a particular type (that is adjective, verb, noun etc.) is returned in a SynSet information object as shown in the UML class diagram in Figure 5-9.

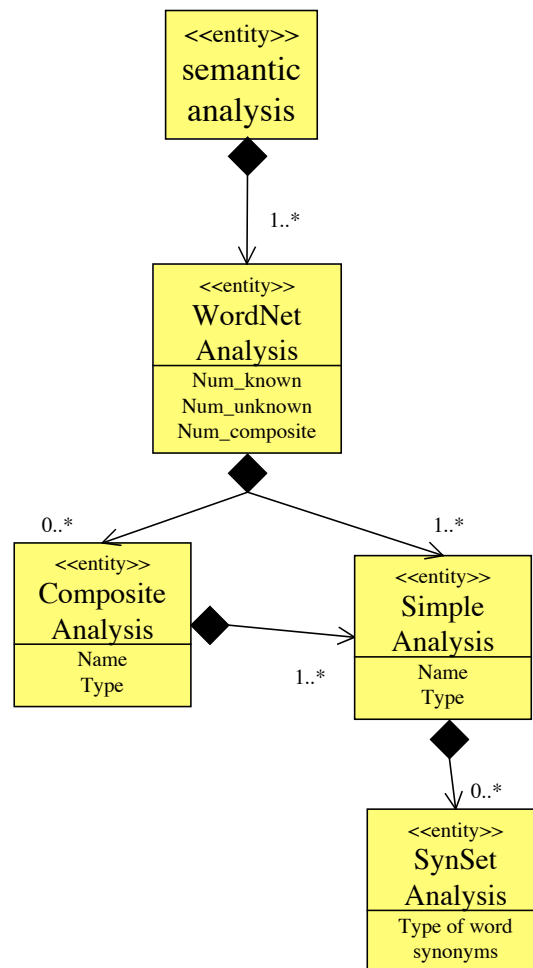


Figure 5-9: UML Class Diagram for Semantic Analysis

A synonym (synset entity) is composed of a statement of the type of the term in the “pos” element followed by the appropriate synonyms listed in “word” elements. To avoid the analysis of words that typically do not add any semantics to a term (such as

¹⁴ already known from the Term Analysis information objects processed by the characterise modelling activity

“of”, “the” and so on), a file called “extrawords” is consulted before WordNet is queried.

For example, the term *academic* would return two synsets. One synset represents the synonyms for *academic* when it is used as a noun, e.g. academician, faculty-member. The other synset represents the synonyms for *academic* when it is used as an adjective, e.g. donnish, pedantic. An example of a composite term *Affiliated-Person*, would return that this is not known as a composite term in WordNet but would return synsets for the tokens of the composite term of “affiliate” and “person”.

The simple, composite and synset analysis objects are returned in an XML file named *MERLINoooo*, where oooo indicates the ontology. The statistics about number of known terms, composite terms and unknown terms in an ontology is returned in an XML file named *WNEToooo*, where oooo indicates the ontology.

Strengths/Weaknesses/Limitations

The strength of this implementation is the use of a tokenisation approach in conjunction with the comprehensive dictionary/thesaurus WordNet (cf. Appendix A). This approach allows for the possible match set for an individual term/composite term to be widened considerably. A limitation of the current implementation is that it does not query any domain-specific online thesauri in order to attempt to classify unknown terms found. It would also be useful to incorporate a means to identify and expand abbreviations or acronyms.

5.1.6 Decide to Match, Select and Apply Match Algorithms – OisinMO application

In the OISIN process, a “decide to match” activity is included. The intention of the activity is to examine the characterisation information with a view to deciding whether or not to undertake matching. In the current implementation this activity is not implemented. Currently, the OisinMO application just assumes that a decision to match has been taken.

In theory, any matching algorithm can be applied as long as it produces the necessary information according to the UML class diagram shown in Figure 5-10. Unfortunately the INRIA format for match information emerged too late for use within the initial implementation of OISIN but is being considered for the next version (cf. section 8.3).

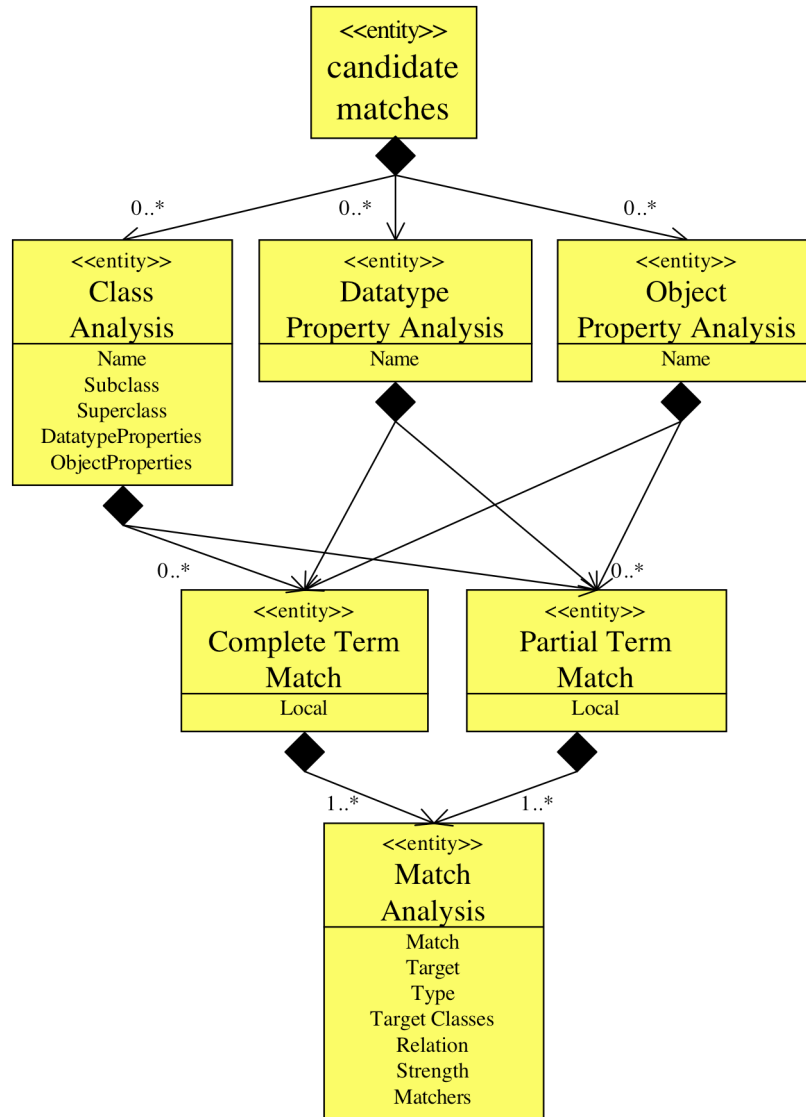


Figure 5-10: UML Class Diagram for Candidate Matches

Two matchers have been implemented so far (OisinMO-lex and OisinMO-props) and a basic result combination application OisinMO-combine. Applying the Shvaiko and Euzenat classification (Shvaiko and Euzenat 2004) that has been outlined in section 3.2.2, OisinMO-lex is a language/linguistic resource based matcher and OisinMO-props is a structure and constraint based matcher. Another matcher called OisinMO-nwmgmt is under development to support matching between network management based ontologies. OisinMO-nwmgmt does not fit into the Shvaiko classification and so

we have termed it a “domain-specific” matcher. OisinMO-nwmgmt matches two ontologies derived from network management information models but using information from the original models in addition. OisinMO-nwmgmt is discussed further in section 8.3 on future work. OisinMO-lex was used for the experimentation described in chapter six, whereas initial implementations of OisinMO-props and OisinMO-combine have only recently been developed.

The matchers produce information objects according to the UML class diagram shown in Figure 5-10. The Candidate Matches analysis consists of information for each Class, DatatypeProperty and ObjectProperty in the ontology. Information is copied from the canonical format about the entity, even if a candidate match is not found for that entity. Thus for a Class, this will include information about subclasses, superclasses and so on.

A match analysis entity is constructed for each match comprising, match, target, relation, target_classes, strength and matchers attributes. The match attribute indicates which of six term match combinations is involved. Term-Term indicates that both the matching source and target ontology terms are simple terms. CTerm-CTerm indicates that both the terms involved are either composite terms or part of composite terms. In the same way Term-CTerm and CTerm-Term combinations are also possible. The final combination involves a term matching to a synonym, yielding Syn-Term and Term-Syn.

The target attribute names the matched term in the other ontology. The type attribute can be one of seven combinations. Class-Class signifying that the term in the local ontology represents the name of a class and matches with a term in the other ontology that also represents a class. In a similar way Class-DatatypeProperty, Class-ObjectProperty, DatatypeProperty-DatatypeProperty, ObjectProperty-ObjectProperty, DatatypeProperty-Class and ObjectProperty-Class indicate the types of the terms in the match. If the term in the other ontology represents a property, then the analysis object will also list the classes in which that property also appears in the other ontology in the target_classes attribute.

The `relation` attribute indicates the type of the match (by default “=”). The `strength` attribute indicates the strength/similarity measure for the match. The `matchers` attribute indicates the type of matchers that contribute to the strength value, depending on whether it is a lexical or semantic match and so on. The `relation`, `matchers` and `strength` attributes have only been recently implemented during the Summer of 2005 and are used in OisinAC-semi-v2. For OisinAC-semi-v1, a lexical matcher and a one to one equality matching relation was assumed.

Finally, the match analysis objects are classified as being one of two term match possibilities. A complete term match is one where the complete term matches to a term or token in the other ontology. A partial term match is one where only a token of the term matches a term or token in the other ontology.

5.1.6.1 OisinMO-lex

The first matcher implemented for OISIN is based on a lexical matching approach. This involves taking a term and attempting to find a match amongst the terms, synonyms or composite terms in the other ontology. The matcher reuses the WordNet analysis information already generated by the OisinIO application.

Figure 5-11 shows the XQuery used to match a simple term against simple terms, synonyms and composite terms in the other ontology. In a similar way an XQuery is used to find a match for a composite term and tokens of a composite term.

```

1. declare function SU:match_term ($term1,$ont2,$canont)
2. {
3.     for $term2 in doc($ont2)/wordnet_analysis/term
4.     return
5.     if (lower-case(normalize-
        space($term1/name/text()))=lower-case(normalize-
        space($term2/name/text())))
6.     then
7.     <analysis><target> {$term2/name/text()} </target>
8.     {let $temp:= concat($term1/type/text(),"-")
9.     return
10.     if (contains($term2/type/text(),"Property"))
11.     then
12.     (
13.     SU:get_classes($term2/name/text(),$canont),
14.     <type>{concat($temp,$term2/type/text())}</type>
15.     )
16.     else <type>{concat($temp,$term2/type/text())}</type>

```

```

16. }
17.     <match>Term-Term</match></analysis>
18. else
19.     let $synlist1 := $term1/synset/sense/word
20.     where (some $syn1 in $synlist1 satisfies lower-
        case(normalize-space($syn1/text()))=lower-case(normalize-
        space($term2/name/text()))
21.     return
22.     <analysis><target> {$term2/name/text()} </target>{let
        $temp:= concat($term1/type/text(),"-")
23.     return
24.     if (contains($term2/type/text(),"Property"))
25.     then
26.     (
27.     SU:get_classes($term2/name/text(),$canont),
28.     <type>{concat($temp,$term2/type/text())}</type>
29. )
30. else
31.     <type>{concat($temp,$term2/type/text())}</type>
32. } <match>Syn-Term</match></analysis>,
33. for $termX in
    doc($ont2)/wordnet_analysis/compositeterm
34. for $term2 in $termX/term
35. where replace(lower-case(normalize-
        space($term1/name/text())),"_|-","") = replace(lower-
        case(normalize-space($term2/name/text())),"_|-","")
36. return
37.     <analysis><target> {$termX/name/text()} </target>
38.     {let $temp:= concat($termX/type/text(),"-")
39. return
40.     if (contains($termX/type/text(),"Property"))
41. then
42. (
43.     SU:get_classes($term2/name/text(),$canont),
44.     <type>{concat($temp,$termX/type/text())}</type>
45. )
46. else
47. <type>{concat($temp,$termX/type/text())}</type>}<match>Ter
    m-CTerm</match></analysis>
48. };

```

Figure 5-11: Snippet of Term Match XQuery

Lines 7 to 17 is the analysis production where a simple term matches with a simple term in the other ontology. First the name of the *target* is written out, and if the type of that target is a property then the classes in which that property appears is returned as *target classes* by invoking the *get_classes* function. Lines 18 to 32 produce an analysis where the matching term is a synonym. Lines 33 to 48 produce an analysis where the matching term is a token of a composite term in the target ontology.

The more recent version of OisinMO-lex includes in the output for each match, the type of the matcher (“lexical”), the strength of the match, and “=” in a relation element.

The candidate matches are returned in an XML file named *CMB $o1$ - $o2$* , where *o1* is the name of the ontology from whose perspective the match was undertaken with *o2* being the target ontology. A CMB match file is returned for each ontology.

Figure 5-12 is an example taken from the *CMBa-b.xml* file showing an analysis of the “Journal” class term.

```

1. <su:Class>
2. <su:Name>Journal</su:Name>
3.
4. <complete_match>
5. <local>Journal</local>
6.
7.   <analysis>
8.     <target>Journal</target>
9.     <type>Class-Class</type>
10.    <match>Term-Term</match>
11.  </analysis>
12.
13.  <analysis>
14.    <target>journal</target>
15.    <target_classes>
16.      <class>JournalArticle</class>
17.    </target_classes>
18.    <type>Class-ObjectProperty</type>
19.    <match>Term-Term</match>
20.  </analysis>

21.    <analysis>
22.      <target>JournalArticle</target>
23.      <type>Class-Class</type>
24.      <match>Term-CTerm</match>
25.    </analysis>
26.
27. </complete_match>
28.
29. <su:SubClass>Article</su:SubClass>
30.
31. <su:SuperClass>Publication</su:SuperClass>
32.
33. <su:DatatypeProperty>has-page-numbers</su:DatatypeProperty>
34. <su:DatatypeProperty>has-volume</su:DatatypeProperty>
35. <su:DatatypeProperty>has-ISSN-number</su:DatatypeProperty>
36. <su:DatatypeProperty>has-issue-number</su:DatatypeProperty>
37. <su:DatatypeProperty>has-title</su:DatatypeProperty>
38. <su:DatatypeProperty>has-publisher</su:DatatypeProperty>
39. <su:DatatypeProperty>has-ISBN-number</su:DatatypeProperty>
40. <su:ObjectProperty>has-editor</su:ObjectProperty>
41. <su:ObjectProperty>has-author</su:ObjectProperty>
42. <su:ObjectProperty>has-subject-area</su:ObjectProperty>
43. <su:ObjectProperty>has-publication-date</su:ObjectProperty>
44. </su:Class>

```

Figure 5-12: Example of XML for a class match

This shows in lines 4 to 27 that the complete term `Journal` matches three targets. The first target `Journal` (lines 7 to 11) is a class and the terms match exactly. The second target `journal` (lines 14 to 20) is a object property which appears in the class `JournalArticle`. The third target is `JournalArticle` (lines 22 to 24) that matches because `Journal` is a token of the composite term. The rest of the analysis shows the direct subclass, superclass and properties of the `Journal` class (lines 28 to 43).

The property term matches are listed separately after all the class information, as this saves unnecessary repetition given that properties appear in several classes. Figure 5-13 provides an example taken from `CMBA-b.xml`.

```

1.  <su:ObjectProperty>
2.  <name>author-of</name>
3.    <partial_match>
4.      <local>author-of</local>
5.      <analysis>
6.        <target>author</target>
7.        <target_classes>
8.          <class>WorkshopPaper</class>
9.          <class>TechnicalReport</class>
10.         <class>SpecialIssuePublication</class>
11.         <class>JournalArticle</class>
12.         <class>Book</class>
13.         <class>Journal</class>
14.         <class>OnlinePublication</class>
15.         <class>Publication</class>
16.         <class>ConferencePaper</class>
17.         <class>ArticleInBook</class>
18.         <class>Article</class>
19.       </target_classes>
20.       <type>ObjectProperty-ObjectProperty</type>
21.       <match>CTerm-Term</match>
22.     </analysis>
23.   </partial_match>
24. </su:ObjectProperty>

```

Figure 5-13: Example of XML for a property match

Figure 5-13 shows that the `author-of` object property is a composite term that matches against a term (line 21) that represents an object property (line 20) that is called `author` (line 6) in the target ontology. The target term appears in a number of classes in the target ontology (lines 8 to 18).

5.1.6.2 OisinMO-props

The OisinMO-props matcher examines class to class matches that have been identified by a previous matcher. It determines the strength of an individual class match based on examination of the properties of the classes involved. The matcher examines the

properties of each the matched classes, identifies whether any of the candidate matches for an individual property appears in the other class and establishes if the type of the properties are the same. The match strength is increased in line with the proportion of the property/type matches identified to the number of properties in the class.

5.1.6.3 OisinMO-combine

The OisinMO-combine application is a basic implementation of a combination of match results application. This first implementation of OisinMO-combine takes as input the result of several matchers. It then checks where there is overlap in matches identified by the different matchers, and if so it adds the corresponding strengths together. Obviously, more sophisticated strength combination strategies can be deployed where strengths of individual matches can be increased or decreased dependent on the organisational policy.

Strengths/Weaknesses/Limitations

A strength of the current matching tools is the ability to analyse not just based on the term itself but also based on tokens of composite terms. A weakness, on the other hand, is that this provides an explosion of possible matches and consequently can lead to a significant amount of matching information being generated. Take for example, two of the ontologies used in the experiments. One ontology to be considered consists of 56 classes and 66 properties and is 32KB in canonical format. The other ontology, which consists of 96 classes and 92 properties, is 88KB in canonical format. The matching files for the two ontologies are 64KB and 104KB. This pattern of increased file size of between twenty and fifty percent is typical.

In addition, the implementation of this activity could be improved through the addition of more matchers (e.g. semantic model based) and more sophisticated strength combination strategies. The design and implementation of the OISIN framework allows for the addition of more matchers and more sophisticated strength combination components to be easily achieved. However, the difficulty with implementation will be the design of how the various strengths will be computed and the weighting of the matches from various kinds of matchers. Currently there are no “best practice” heuristics that can be called upon in this regard, and it is likely that deployment experience will need to be harnessed in order to tune the implementation. This is akin

to the problem of distributed database schema design where only a certain amount of the strategy can be determined before deployment and the rest through experience.

Finally, the matchers currently output information in the CMB extensible format. Although extensible, this format would still be considered a proprietary format. It would be useful to investigate if this information can be rendered into the INRIA format, especially if that format becomes a *de facto* standard. Work in this direction is underway as part of future work that has been initiated (cf. section 8.3).

5.1.7 Characterise Amenability – OisinSummarise application

The current implementation of this activity summarises the characterisation information generated. The schema for summarising the information is drawn from the data that is needed to characterise the degree of mapping difficulty. This schema is discussed in detail in the next section 5.1.8.

The summary is achieved by executing an XQuery over the XML files that comprise the characterisation information and then rendering this information appropriate to the *decide to map* application. For the semi-automatic version of OisinDTM, this information is rendered into a comma-delimited list that can be imported in to Microsoft Excel. For the automatic version of OisinDTM, the information is queried directly.

Strengths/Weaknesses/Limitations

A strength of this approach is that it decouples the implementation of the characterisation phase activities from the mapping phase activities, such that different summary information can be produced as more characterisation information becomes available or as the criteria of the organisation changes. Being XQuery based is an advantage for this application as the kind of summary information to be generated can be quickly and easily changed by an ontology engineer. However, XQuery is not easily generated/altered by software. This is a limitation as ideally it would be beneficial to open up the specification of what information to summarise to other more non-technical roles (such as a manager) in an organisation. This limitation could be overcome by switching to an XQueryX (XQueryX 2003) format for the query (which

is an XML representation for XQuery). However query engines that interpret XQueryX are only beginning to emerge. To support a semi-automatic approach to this summarisation would then require a simple tool to allow a user specify the summary information that is required and have this tool parameterise an XQueryX query. To fully support an automatic approach, a tool could be designed to analyse the decision to map policies of the organisation automatically and derive what summary information is required to be queried as XQueryX. The complexity of this automatic derivation would then depend on the complexity of the policies themselves and how they are expressed.

5.1.8 Decide to Map – OisinDTM Application(s)

This section describes two implementations of OisinDTM: OisinDTM-semi and OisinDTM-auto.

5.1.8.1 OisinDTM-semi

OisinDTM-semi implements a semi-automatic approach, where the characterisation information is gathered and presented automatically, while the user manually makes the decision to map. It was decided to support the use case where a decision had to be made rather than the one that simply records the allocation of the mapping task (cf. section 4.2.2). To support this activity, it was decided to use a technology that aids decision making and that is familiar to most professional workers, that is Microsoft Excel. This was considered to be a useful approach for the following reasons:

- Characteristic information could be presented numerically and graphically;
- Information could be easily imported;
- Organisational policies and calculations related to the characteristic information could be embedded in the spreadsheets;

Thus OisinDTM-semi is implemented as an Excel spreadsheet that imports the comma delimited list version of the characteristic summary information produced by the OisinSummarise application.

The characterisation information is presented in a four page Excel spreadsheet using numeric and graphical means. The first page shown in Figure 5-14 (and see Appendix

D-2 for another example) presents syntactic and semantic information related to each ontology. The syntax analysis section provides information about whether the ontology compiles and the percentage of OWL keywords used in the ontology (which provides an indication of the depth of the modelling used). A basic OWL ontology would be around 20 percent. In addition the percentage of the terms that are composite is presented. This information is useful as it provides an indication as to how complex the matching and mapping potentially will be. The WordNet analysis section provides an indication of how many of the terms or parts of composite terms are known. The ontology references section presents information gathered from SWOOGLE about the number of other ontologies the ontology references or imports and how the ontology is ranked and reused. This latter information provides another indication of quality.

Ontologies

x.owl y.owl

Page 1

Syntax Analysis

“Term” is a Class or Property Name

e.g. full-lecturer →

e.g. full_lecturer →

e.g. fullLecturer →

compiles?	yes	yes
% OWL keywords used	29	20
Number of Terms	105	150
% Terms which are Composite	52.38	68.67
Number of Hyphenated Composite Terms	55	0
Number of Underscored Composite Terms	0	3
Number of MiddleCapital Composite Terms	0	100

WORDNET Analysis

Terms known directly →

Term Combinations where some constituent terms known →

Unknown Terms →

% of ontology whose terms are unknown →

Terms known directly	30	67
Term Combinations where some constituent terms known	75	73
Unknown Terms	0	10
% of ontology whose terms are unknown	0	7

e.g. blxyLecturer →

e.g. amghik →

Ontology References Analysis

Number of Ontologies that it imports	0	1
Number of Ontologies that it references	5	3

Analysis using SWOOGLE

“GOOGLE” for ontologies which are published on web

Rank	0.2	0.3	(Top Ten normally in range 0.2 to 0.5)
Ontologies which import	1	2	
Ontologies which reference	1	2	
Ontologies which extend	1	1	
Ontologies which map	0	0	

Figure 5-14: Example page 1 of OisindTM excel spreadsheet

The second page of information (Figure 5-15, and Appendix D-2 for another example) presents information about the dimensions of the ontologies. The basic information on number of classes, attributes and relationships is useful for understanding the nature of the individual ontologies (e.g. taxonomic, highly interrelated etc.) as well as providing an indicator of the complexity of matching due to the cardinalities involved. The numeric and graphic information showing the shape of the ontologies is a useful indicator of the scale of traversal of the ontologies that may be needed in searching for mappings.

Dimensions Analysis

	x.owl	y.owl
Number of Classes	55	100
Number of Unique Attributes	20	30
Number of Unique Relationships	40	20

Shape of Ontologies

Level	x.owl Width	y.owl Width
0	5	10
1	10	30
2	20	45
3	20	10
4	0	5
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0

Number of Classes at each level

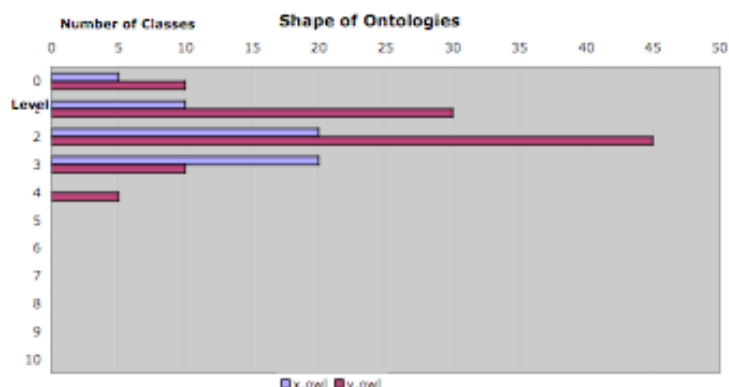


Figure 5-15: Example page 2 of OisindTM excel spreadsheet

The analysis generated by the overall amenability activity is presented in pages three and four of the spreadsheet. The candidate matches of terms/composite terms are presented on page three (Figure 5-16, and Appendix D-2 for another example) in table and graphical format according to type, that is classes, datatype properties and object properties. This provides an indication of the number of terms that have a candidate match in the other ontology. The candidate matches to potential mappings numeric table and chart shows the number of potential mappings arising. For example the class term `Lecturer` may potentially map to a class term named `Academic-Lecturer` or a property of the `course` class named `course_lecturer`, showing that one candidate match potentially has two different mappings. Thus, this information can provide an indication of the number of choices the user will need to consider during the mapping stage and so can be an indicator of the amount of effort that may be required.

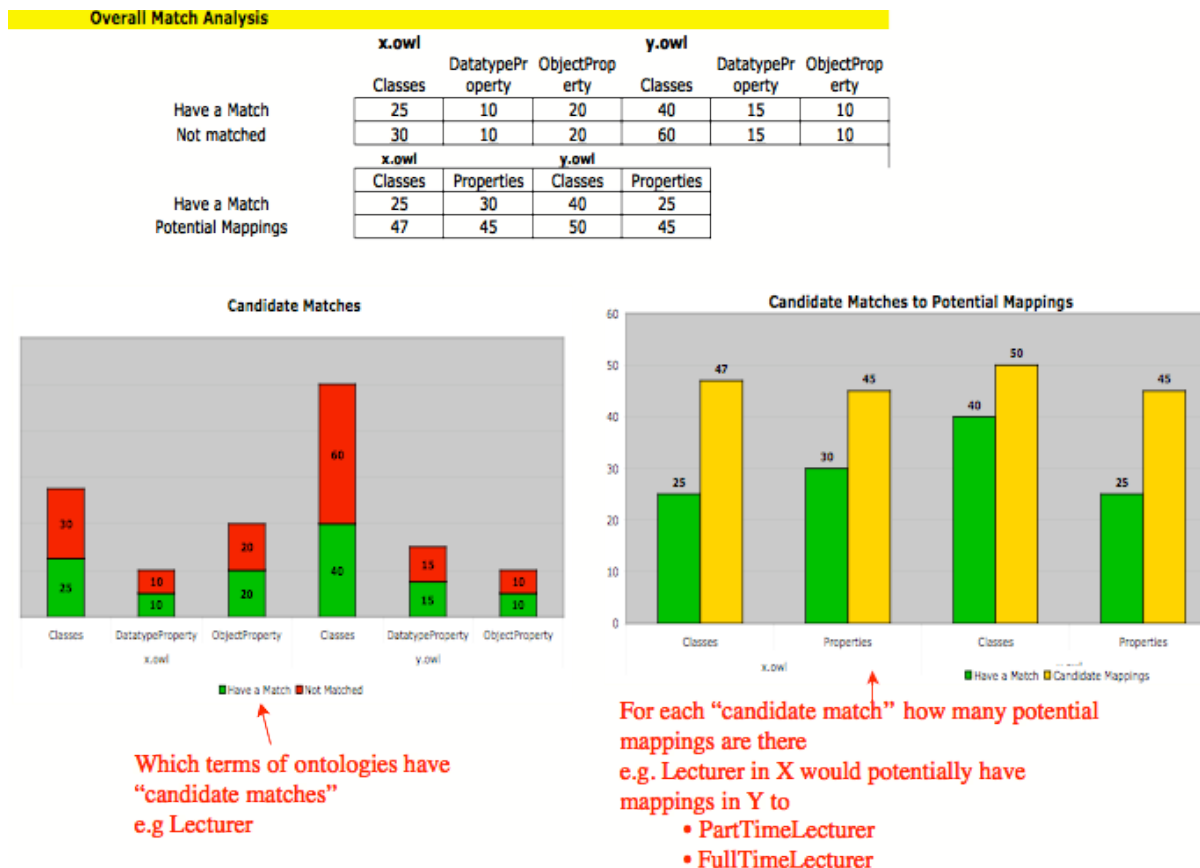


Figure 5-16: Example page 3 of OisindTM excel spreadsheet

A further indication of the degree of complexity that may be involved in the mapping discovery phase is provided on page four of the spreadsheet (Figure 5-17, and Appendix D-2 for another example). Here more detail about the potential mappings are shown with one table showing the candidate matches of terms and the other the candidate matches of parts of composite terms. The table breakdown indicates the types of the terms involved in a potential mapping. The chart shown summarises the potential mappings with respect to whether there are type mismatches involved or not. A high degree of type mismatching would be an indicator that mapping could prove difficult as the ontologies would have a different perspective on how the domains were modelled. For example one ontology may model mainly in terms of properties and the other in terms of classes.

x.owl	y.owl
SMU	SMU
CUPID	CUPID

Potential Mappings: Match of Term to Term or Term in Composite Term

Class to Class	27	26
DatatypeProperty to DatatypeProperty	1	16
ObjectProperty to ObjectProperty	7	34
Property to Class	0	11
Class to Property	8	0
DatatypeProperty to ObjectProperty	0	0
ObjectProperty to DatatypeProperty	0	0

Same Type Mapping	70	89
Different Type Mapping	32	17

Potential Mappings: Match of Term in Composite Term to Term or Composite Term

Class to Class	9	13
DatatypeProperty to DatatypeProperty	15	0
ObjectProperty to ObjectProperty	11	0
Property to Class	13	6
Class to Property	3	0
DatatypeProperty to ObjectProperty	1	0
ObjectProperty to DatatypeProperty	7	0

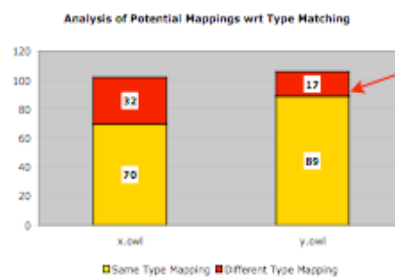


Figure 5-17: Example page 4 of OisindTM excel spreadsheet

It is envisaged that the actual decision to map would be based on a combination of guided examination of the characterisation information and examination of the ontologies. Guidance for interpretation of the excel spreadsheet could be easily incorporated into the spreadsheet itself or provided as a separate document. This guidance can be based on the policies and best practice experience of the organisation where the deployment is being situated. An example of such guidance and the issues in its design are described in section 6.2, where the experiment to evaluate the *decide to map* activity is presented.

5.1.8.2 OisindTM-auto

The automated version of the OisindTM application is implemented via an XQuery that queries the characteristic summary information and makes a decision based on the guidance used in the experiment described in section 6.2. The XQuery is shown in Figure 5-18.

1. declare function
SU:DecideMapAuto(\$dim1,\$wnet1,\$lang1,\$lex1,\$dim2,\$wnet2,\$lang2,\$lex2,\$cmb)
2. {
3. let \$anal := (<analysis>

```

4.      <ont1>
5.      {
6.      SU:IndivOnt($dim1,$wnet1,$lang1,$lex1)
7.      }
8.      </ont1>
9.      <ont2>
10.     {
11.     SU:IndivOnt($dim2,$wnet2,$lang2,$lex2)
12.     }
13.     </ont2>
14.     <ratios> {SU:ratio($cmb)} </ratios>
15.     </analysis>)
16.     return
17.     (
18.         let $res := <result>
19.         {
20.             if (($anal//ont1//unknown/text() > 0.11) or
($anal//ont1//owl < 0.2) or ($anal//ont1//composite < 0.5)) then
(<ont1_failure/>,$anal//ont1)
21.             else <ok/>,
22.             if (($anal//ont2//unknown/text() > 0.11) or
($anal//ont2//owl < 0.2) or ($anal//ont2//composite < 0.5)) then
(<ont2_failure/>,$anal//ont2)
23.             else <ok/>,
24.             if ($anal//matches/text() !=0)
25.             then
26.             (
27.                 if ($anal//matches/text() div $anal//potential/text() <
0.5)
28.                 then
29.                 (<mapping_ratio_failure/>,$anal/ratios)
30.                 else
31.                 <ok/>,
32.                 if ($anal//mismatch_type/text() div $anal//types/text() >
0.3)
33.                 then
34.                 <type_ratio_failure/>
35.                 else
36.                 <ok/>
37.                 )
38.                 else
39.                 <no_matches/>
40.                 }
41.             </result>

42.         return
43.         if (exists($res//no_matches) or count($res//ok) < 4)
44.         then
45.         <final> {$res} </final>
46.         else
47.         <final> {true()} </final>
48.         )
49.     };

50.     declare function SU:IndivOnt($dim1,$wnet1,$lang1,$lex1)
51.     {
52.         let $percent_unknown := (doc($wnet1)//unknown_terms/text()
div
53.         (doc($dim1)//num_concepts/text() +
doc($dim1)//num_unique_attributes/text() +
doc($dim1)//num_unique_relationships/text())),
54.         $percent_owl := doc($lang1)//percent_keywords_used/text(),
55.         $percent_hyphen := count(distinct-values(for $pat in

```

```

doc($lex1)/lex_analysis/term_analysis[pattern/text()="hyphenated
"] return $pat/term/text())) div doc($wnet1)//composite_terms,
56.   $percent_under := count(distinct-values(for $pat in
doc($lex1)/lex_analysis/term_analysis[pattern/text()="underscore
d"] return $pat/term/text())) div doc($wnet1)//composite_terms,
57.   $percent_middle := count(distinct-values(for $pat in
doc($lex1)/lex_analysis/term_analysis[pattern/text()="middleCapi
tal"] return $pat/term/text())) div
doc($wnet1)//composite_terms,
58.   $comp_list := ($percent_hyphen,$percent_under,
$percent_middle),
59.   $max_composite := max($comp_list)
60.   return
61.   (
62.   <char>
63.   <unknown> {$percent_unknown} </unknown>
64.   <owl> {$percent_owl} </owl>
65.   <composite> {$max_composite} </composite>
66.   </char>

67.   )
68.   );

69.   declare function SU:ratio($combont)
70.   {
71.   <ratios>
72.   <numtypes> {count(doc($combont)/*/type)} </numtypes>
73.   <mismatch_type> {count(for $type in doc($combont)/*/type
where $type/text()="Class-ObjectProperty" or
$type/text()="Class-DatatypeProperty" or
$type/text()="DatatypeProperty-ObjectProperty" or
$type/text()="ObjectProperty-DatatypeProperty" return
$type/text())} </mismatch_type>
74.   <matches> {count(distinct-values(for $class in
doc($combont)*/su:Class
75.   where exists($class/complete_match) or
exists($class/partial_match)
76.   return
77.   $class/su:Name/text()))} </matches>,
78.   <potential> {count(for $class in doc($combont)*/su:Class
79.   return
80.   $class/*/analysis)} </potential>
81.   </ratios>
82.   };

```

Figure 5-18: Snippet of OisinDTM-auto XQuery

First (lines 3 to 15) the summary information for each ontology is gathered through the use of the SU:IndivOnt (lines 50 to 68) and SU:ratio (lines 69 to 82) functions. This information is then examined in lines 17 to 49 to determine if a mapping should take place or not. Different XML elements are returned (e.g. mapping_ratio_failure) indicating success or failure. Based on this information OisinDTM-auto either proceeds to the next application or returns a failure for that ontology set.

Strengths/Weaknesses/Limitations

One strength of OisinDTM-semi is the use of a tool that is familiar to most computer users, Microsoft Excel, in the presentation of the ontology characterisation. This allows a wide range of users to avail of the information and make a determination as to how difficult the mapping will be. It also allows for the user to specify other formulas to be applied to the summary data, perhaps encoding local expertise, or for the characterisation charts to be presented differently, perhaps personalised to the preferences of different users.

A weakness of the current OisinDTM-semi implementation is that the spreadsheet design was undertaken without potential user involvement. A limitation of the implementation is that it relies on the user using an Excel spreadsheet in parallel with an ontology browser (such as Protégé). It would be interesting to see if simple ontology browsing functionality could be implemented as an Excel pivot table, because in this activity all that is required is some form of ontology visualisation. If this was possible, then the same environment could be used both for browsing and characterisation information analysis.

The strength of the approach is also seen in the successful implementation of OisinDTM-auto. This has shown that heuristic guidance provided to a human user of OisinDTM-semi can be successfully encoded when the heuristic is precise enough. The difficulty will be in the determination of heuristic guidance that can be encoded.

A weakness of OisinDTM-auto is that it relies on the heuristic guidance being encoded as a static query posed upon the characterisation information. This approach is limited due to its static nature. Similar to comments made with respect to the OisinSummarise application in the last section, it would be interesting to explore the use of XQueryX instead of XQuery for the query expression so as to enable query manipulation by software applications. This would allow the decision criteria for OisinDTM-auto to be automatically derived from evolving organisational policies.

5.1.9 Identify Committed Mappings – OisinAC application(s)

As discussed in section 4.2.2, most runtime algorithms require some certainties to be specified to act as a frame of reference for processing. In the OISIN framework these certainties have been termed “committed mappings”. For this implementation, the idea of an “anchor” has been borrowed from Noy and Musen (Noy and Musen 2001), as a concrete way of expressing a committed mapping. Furthermore the notion of an “anchor path” is borrowed. The idea of an anchor path is that if two anchors are specified in a hierarchy of Ontology A it is likely that the classes which appear in the intervening path (based on some relation such as is-a or containment) may correspond with those on the path of the corresponding anchors in Ontology B. The reader is referred to section 3.4.4 for a description of how the anchor concept is used in Protégé. In this implementation an anchor path is limited to the is-a hierarchy relationship .

This section describes three implementations of OisinAC: OisinAC-semi-v1; OisinAC-semi-v2 and OisinAC-auto.

5.1.9.1 OisinAC-semi-v1

The OisinAC-semi-v1 tool is a graphical user interface (GUI) for identifying anchor mappings from the set of candidate matches previously identified and from the ontologies themselves. The GUI was implemented using the Java Swing library, and the parsing of the input XML documents was performed using a SAX parser. Figure 5-19 provides a summarised view of the key classes implemented. The terms “accessor methods” and “add methods” are used as shorthand to indicate that the data in each of the attributes of the classes can be retrieved/updated. For conciseness, detail is not supplied here on the classes that implement the behaviour of the `SAXHandler` or `GUIHandler`. The methods implemented in the `GUIHandler` allow the display, searching and sorting of the main classes of `ClassPackage`, `Attribute`, `CompleteMatches`, `PartialMatches`. The methods implemented for the `SAXHandler` include event handlers that are called based on the XML element that is read from the CMB input file. These event handlers create vectors of objects to represent the ontology class and ontology property information and vectors that list the classes whose term names are a complete or partial match. A walk through of the user interface is provided in section 5.2.1.

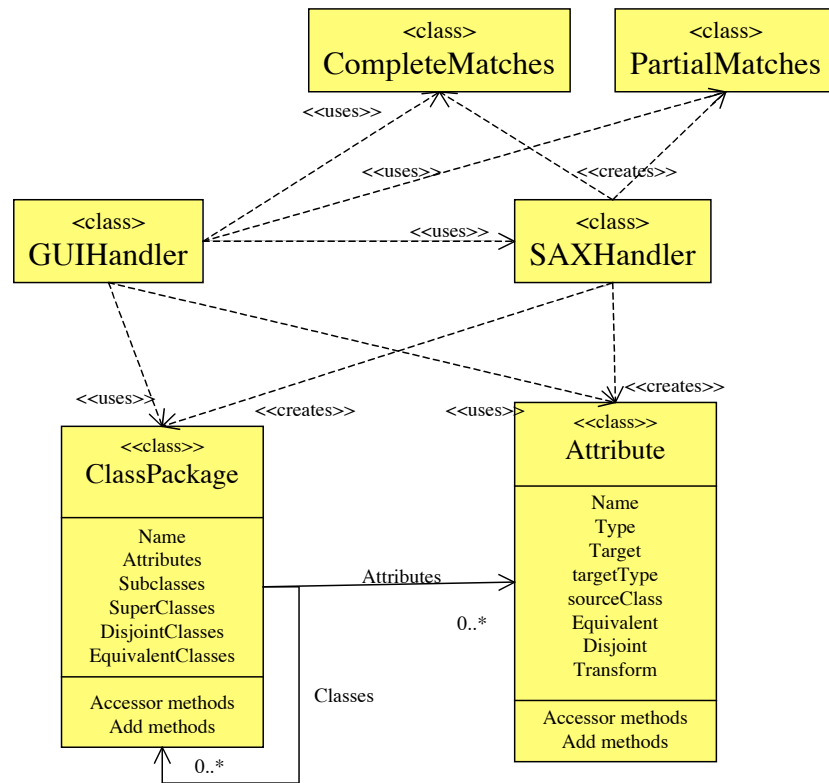


Figure 5-19: UML Class Diagram for OisinAC-semi-v1 application

The class `ClassPackage` represents an ontology class and the class `Attribute` represents a property of that ontology class. All but the `DisjointClasses` and `EquivalentClasses` attributes of `ClassPackage` are set by the `SAXHandler` based on the candidate match information described in section 5.1.6. Information for the attributes `DisjointClasses` and `EquivalentClasses` is supplied by the user when a mapping is created using the `GUIHandler`. Similarly the CMB input document is parsed to fill in the name, type, target, targetType and sourceClass attributes for an `Attribute` object. Again the `Equivalent`, `Disjoint` and `Transform` attributes are created through user interaction during the creation of property mappings. The `CompleteMatches` and `PartialMatches` objects are created by the `SAXHandler` and keep a list of the classes that have complete or partial matches respectively.

The committed mappings are returned in an XML file named `ANCoI-o2`, where *o1* is the name of the ontology from whose perspective the match was undertaken with *o2* being the target ontology. The mapping is expressed using the `equivalentClass`, `equivalentProperty` and `disjointClass` syntax of OWL. An ANC mapping file is returned for each ontology.

OisinAC-semi-v1 was developed during the summer of 2004 and used in the experiments described in chapter six in the autumn of 2004.

5.1.9.2 OisinAC-semi-v2

During the summer of 2005, OisinAC-semi-v1 was improved based on the feedback received during the experimentation undertaken in the autumn of 2004.

In particular, the following significant features were added:

- Ability to invoke the OisinMA application directly from the tool;
- Ability to save or retrieve mapping sessions was added;
- Ability to retrieve and display an existing mapping file in order to support reuse of mappings;
- Ability to display matches and strengths of matches from different matchers;
- Reminder to user upon exit of potential class matches that were not as yet used in mappings;
- Inherited properties are shown in a different colour in the property pane;

An example screenshot of the new improved interface is provided in Section 5.2.2.

5.1.9.3 OisinAC-auto

The OisinAC-auto tool is a basic implementation of an automated approach to the identification of committed mappings between one ontology and another ontology. This implementation simply selects a previously computed file of relevant committed mappings and invokes the OisinMA application to undertake the committed mappings analysis. This invocation is necessary as the candidate matches involved may be different (due to different matching algorithms being used) than those that were created when the committed mappings were computed. Recall that the OisinMA application analyses and manipulates candidate matches with respect to identified committed mappings. As discussed in section 4.2.2, a more sophisticated implementation of the OisinAC-auto application could be developed that would apply selected mapping patterns to the candidate matches in order to detect committed mappings.

Strengths/Weaknesses/Limitations

A particular strength of the OISIN approach is the ability to use it in the “minimum” and “maximum” anchor capture modes. In addition, the OISIN approach allows for the generation of a comprehensive set of information that can be used by applications themselves to decide on mappings. As can be seen in the walk through in section 5.2, a number of features have also been implemented to reduce the amount of effort involved in finding mappings (e.g. the display match buttons) and documenting mappings (e.g. the feature that allows the user make a property match between two anchors, a mapping for all the classes in which that property appears).

Another strength of the semi-automatic implementations is that the user is visually guided to potential matches but not forced to follow a system driven agenda. In line with this, the system encourages the reduction of errors (e.g. pointing out structural mismatches and reminding of unmapped candidate matches) rather than preventing errors from occurring. This contrasts strongly with the implementation of the Anchor-PROMPT plug-in for Protégé.

The current semi-automatic versions are limited however in not providing more heuristic support for identifying potential mappings. Perhaps some systematic support for mapping patterns (in a manner similar to the SEKT project) would be useful.

5.1.10 Undertake Commitment Analysis – OisinMA application(s)

The intention of the commitment analysis activity is to analyse the candidate matches in the light of the committed mappings chosen. In the case of the anchor approach taken in the OISIN implementation, this involves the generation of “anchor paths” and analysis of the candidate matches with respect to these paths. The introduction of the concept of anchor path is to take advantage of the intuition expressed by Noy and Musen, *that if two pairs of terms from the ontologies are similar and there are paths connecting the terms, then the elements in those paths are often similar as well* (Noy and Musen 2001). In the OISIN implementation, an anchor path represents the classes that appear in the is-a hierarchy between two anchors in an ontology. This contrasts with Noy and Musen, where the anchor path could be any path that can be traversed

over an ontology based on a specific relation, but where the relation could be any chosen (e.g. containment, is-a, user specified and so on).

The information about the anchors, anchor paths and candidate matches (which, in OISIN, together comprise what is termed **mapping information**) can be used by an application at runtime to determine, in the light of the context of processing, what constitutes a sufficient mapping. Obviously, an application may have most confidence in classes that appear in anchor mappings, have less confidence in classes that match and appear in anchor paths, and less confidence still about classes that have been matched but do not appear as anchors or in anchor paths. The determination of what is or is not considered a mapping can thus be undertaken in the context of the applications involved in using the mappings and what they are trying to achieve.

First, information about the ontologies is examined in order to retrieve the anchor paths. This is achieved using the XQuery snippet shown in Figure 5-20. In this snippet, all the anchors for an ontology are retrieved (line 5), then for each anchor (line 6) the corresponding anchors lower in the is-a hierarchy (using the `lowanchors` XQuery) is returned (lines 7 to 13). Of course there may be none or a number of lower anchors involved. Finally, given an anchor and one of the lower anchors, the `get_anchored_paths` XQuery is called to show the classes on the path (lines 14 to 26).

```
1. declare function SU:get_all_anchor_paths($canont,$eont)
2. {
3.   <anchor_paths>
4.     {
5.       let $anchors := (SU:anchors($canont,$eont))
6.       for $concept in $anchors
7.         let $lwancs := (SU:lowanchors($concept,$canont,$eont))
8.         let $lowancs := <ancs> {for $concept in $lwancs
9.           where exists($concept)
10.            return
11.              (
12.                $concept
13.              )} </ancs>
14.       let $paths :=
15.         (SU:get_anchored_paths($concept,$canont,$lowancs))
16.       return
17.         (
18.           if (exists($paths/*))
19.           then
20.             (

```

```

21.         )
22.     else
23.         (
24.             ""
25.         )
26.     )
27. }
28. </anchor_paths>
29. };

```

Figure 5-20: Snippet of XQuery to retrieve all anchor paths

Second, OisinMA checks whether there are any structural mismatches arising from the anchor mappings. In our definition of an anchor path, the assumption is that corresponding anchors are on the same is-a tree. A structural mismatch occurs where the corresponding start and end anchors of an anchor path appear in a different is-a tree. For example in Figure 5-21 if the mapping between `Employee` terms (dotted line) and the mapping between `AcademicStaff` (dashed line) terms is made, then if a mapping between `Researcher` is made (solid line) there is a structural mismatch. In one ontology a `Researcher` is modelled as a subclass of `AcademicStaff` and in the other a `Researcher` is a superclass of `Academic-Staff`. Thus a structural mismatch would indicate that the anchor paths in question could not be considered similar and thus would be contrary to the definition of an anchor path. Practically it would mean that an application interpreting the mapping information would mistakenly assume that because a matching class was part of an anchor path that it could have more confidence in assuming the corresponding match would make a good mapping.

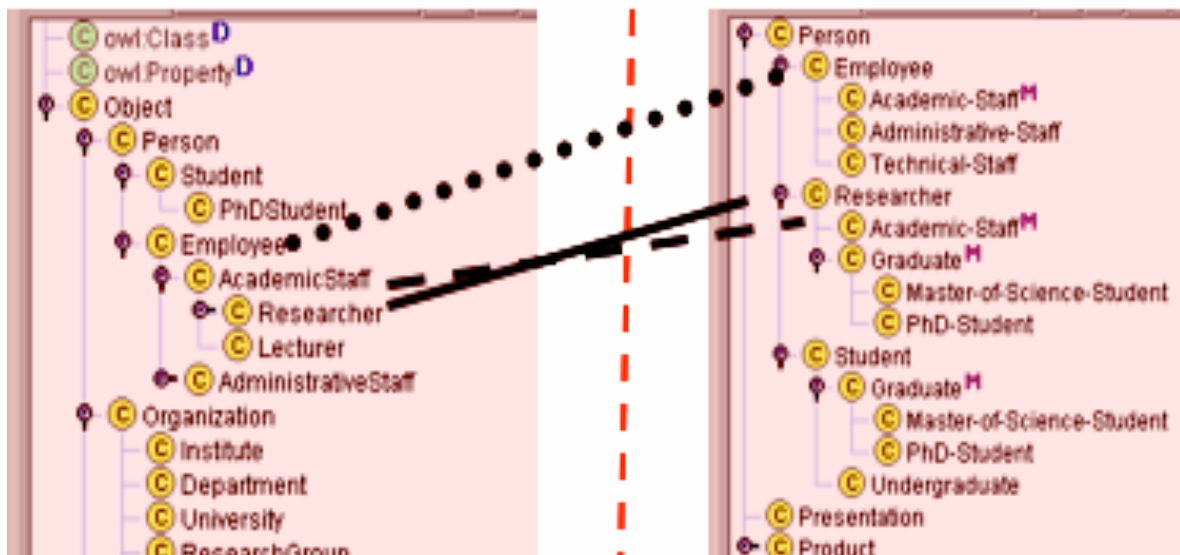


Figure 5-21: Example of ontology structural mismatch

Figure 5-22 shows a snippet of the code used to implement the anchor analysis. Before this snippet of code gets executed, the paths (`$a1_paths` and `$a2_paths`) have already been generated using a XQuery called `get_all_anchor_paths` which enumerates the classes that appear on the is-a tree in between pairs of anchors within an ontology. First (line 1) all the paths of each ontology are checked by `check_anchor_paths`. This takes the head and tail of each path, returns the corresponding anchor, and checks whether a path comprised of the corresponding head and tail anchors appears in the other path list. From this it can be determined (lines 3 to 6) which anchors are not on a path. It also indicates which paths do not have corresponding paths that are connected via the is-a hierarchy (line 11). It then proceeds to produce an analysis for the valid paths (lines 13 to 37).

```

1. let $checkpaths := <paths_analysis>
   {SU:check_anchor_paths($canont1,$canont2,$a1_paths,$a2_paths,$eont1)} </paths_analysis>
2. let $all := distinct-values($checkpaths/valid[exists(matches)]/*>nd/text())
3. let $not_on := for $anc in $anchors return
4. (
5.   if (some $a1 in $all satisfies $a1=$anc) then "" else
   <not_on_path> {$anc} </not_on_path>
6. )
7. return
8. (
9.
10.   <anchors_not_on_paths> {$not_on} </anchors_not_on_paths>,
11.   <structural_mismatches>
   {$checkpaths/valid[not(exists(matches))]}/*>
   </structural_mismatches>,
12.   <number_of_valid_paths>
   {count($checkpaths/valid[exists(matches)])}
   </number_of_valid_paths>,
13.   <valid_paths>
14.     {for $v in $checkpaths/valid[exists(matches)]
15.      return
16.      <valid_path>
17.      {
18.        $v/anchor_path,
19.        $v/matches
20.      }
21.      <length> {$v/path/length/text()} </length>
22.      {for $valid in $v/path/nd
23.       return
24.       (
25.         if ($valid/text()!="SU:ANCHOR")
26.         then
27.         <nd>
28.         <name> {$valid/text()} </name>
29.         {SU:analyse_class($valid/text(),$canont1,$eont1)}
30.         </nd>
31.         else
32.         ""

```

```

33.      )
34.      }
35.      </valid_path>
36.      }
37.      </valid_paths>
38.      )

```

Figure 5-22: Snippet of XQuery to implement anchor analysis

Next a summary of the mapping analysis is created. The information comprising the mapping analysis is shown as part of the UML diagram in Figure 5-23. This analysis, which is output as an XML file of the form *MAoI-o2*, presents information from the perspective of ontology *o1*. In summary, it includes information about:

- Dimension of schema (number of concepts etc.);
- Analysis of Candidate Matches (number of complete/partial matches of names of classes, properties);
- Analysis of Anchor classes specified by the user (number properties, number property matches, number of equivalent properties specified etc.);
- Listing of all Anchor Paths (listing the head and tail of anchor paths deduced from anchors specified);
- Listing of the anchors which do not participate in a path;
- Listing of anchor paths which are considered structural mismatches and are not valid paths;
- Analysis of Valid Anchor Paths (what the head/end anchors match to in the other ontology and listing each class along the path and whether they match to anything or not).

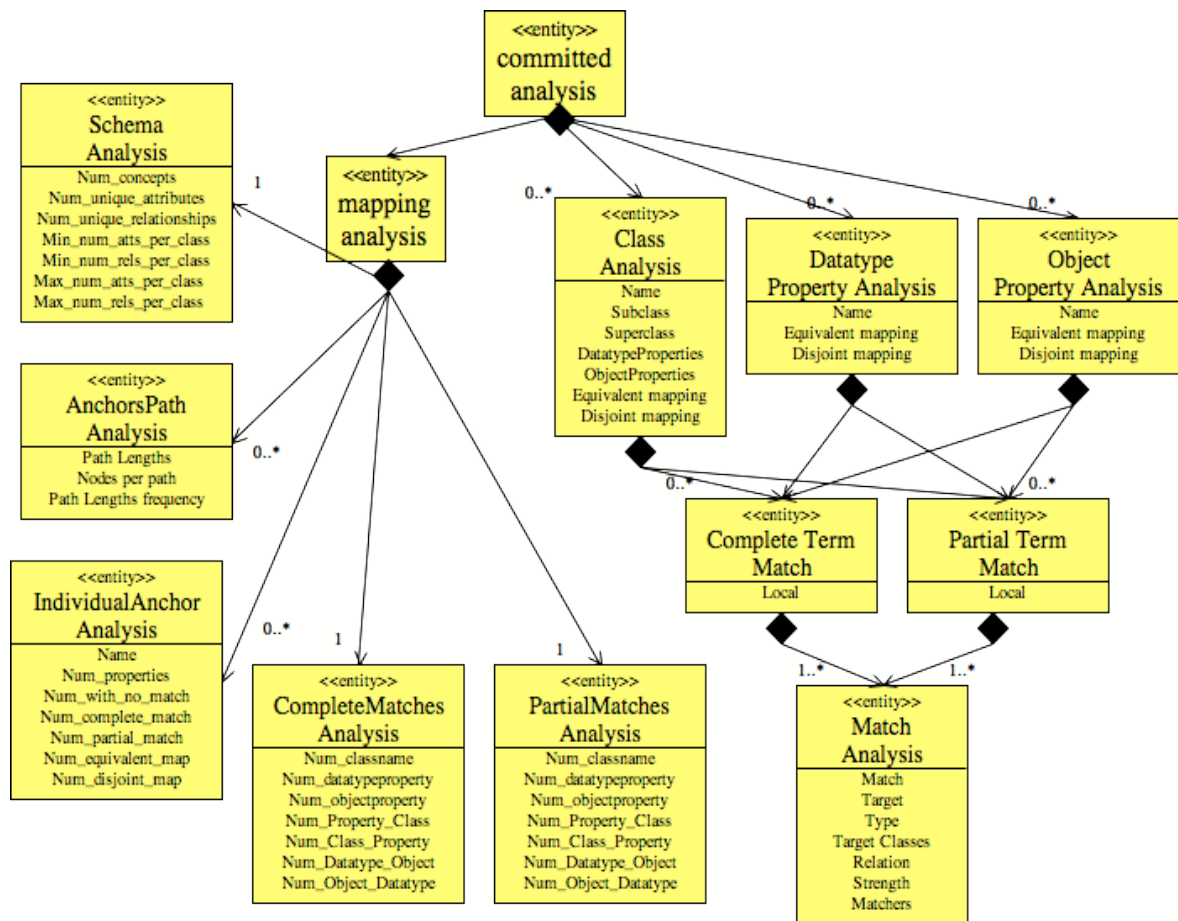


Figure 5-23: UML Class Diagram for Committed Analysis

Originally, OisinMA was implemented as a separate stand-alone application that was used in conjunction with the OisinAC-semi-v1 application. More recently the application has been incorporated as a button on the OisinAC-semi-v2 user interface. In both cases, once the mapping analysis is complete, a window with two scrollable panes appears. The automated version of OisinMA undertakes the analysis and outputs the results to file, and does not need to display results in a window. Figure 5-24 shows an example of the analysis window displayed. Each pane shows the mapping analysis from the perspective of one of the ontologies.

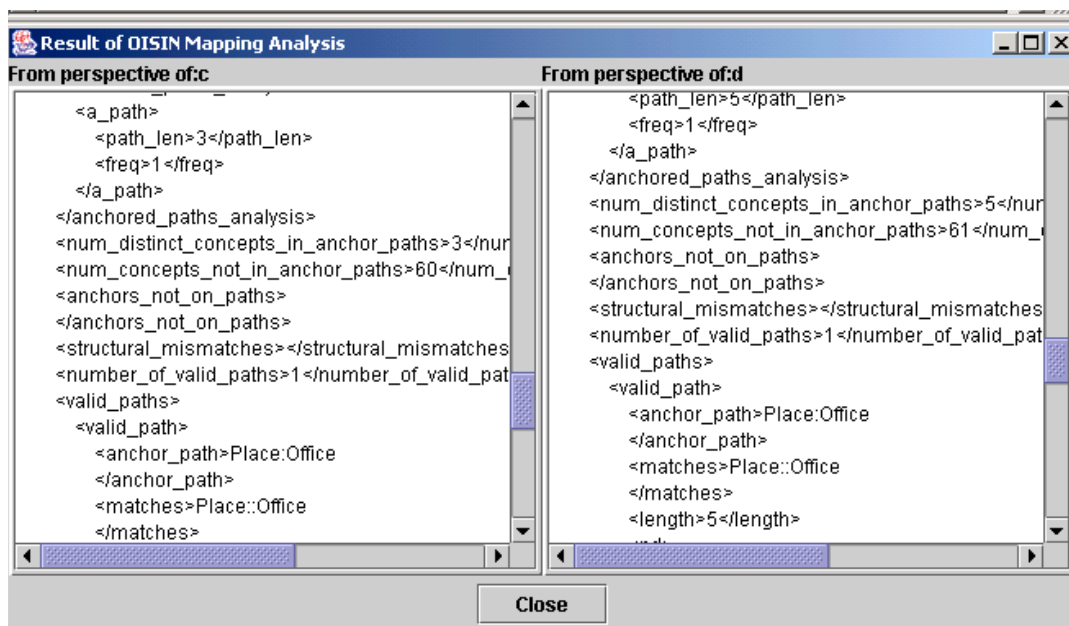


Figure 5-24: Example OisnMA window presenting Mapping Analysis

Finally, as can be seen from Figure 5-23, OisnMA alters the candidate match information of the CMB files to include the equivalent and disjoint mappings identified as committed mappings. The updated candidate match analysis is returned as an XML file named *INToI-o2*, where *oI* indicates the related source ontology. The availability of INT files provides an application with the opportunity either to use the anchor information within the ANC files or to have access to more complete match and mapping information through the INT files.

Strengths/Weaknesses/Limitations

One strength of the current implementation is the wealth of information that is generated and made available to any person or application that wishes to interpret mappings. As shown in Figure 5-23, the committed analysis information not only includes the match analysis, anchor mapping and anchor path information but also summary analyses of the ontology, path and matches information. In essence, the application has access to all the information that was made available in the *decide to map* and *anchor capture* activities, as well as the information arising from the anchor capture and anchor path analysis activities.

On the other hand, this strength is a potential weakness in that more information naturally leads to large analysis file sizes. File size is not a storage issue but rather

potentially impacts on the amount of time that will be necessary for an application to query or interpret the information. Take for example, two of the ontologies used in the experiments. The “a” ontology (56 classes and 66 properties) is 32KB in canonical format and the “b” ontology (96 classes and 92 properties) is 88KB in canonical format. The mapping information file INTa-b file is 92KB and the INTb-a file is 152KB. This type of increase in file size is not atypical. This weakness potentially could be reduced through the optimisation of tag names or the application of classic file organisation techniques such as indexing. However, this would be at the cost of the introduction of mediated access to the information rather than allowing the applications operate over the information directly. Another potential weakness of the current anchor path implementation is that it is limited to paths generated from the is-a inheritance hierarchy. However it could be argued that this is not a major limitation as, from experience, most ontologies use the is-a relationship as a fundamental modelling relation.

5.1.11 Interpret Mappings – OisinSMU service

OisinSMU is currently implemented as an XQuery (see Figure 5-25) that takes a term as well as the anchors file (`$anchors` parameter), mapping analysis file (`$mapping_analysis` parameter) and candidate match information file (`$details` parameter) as input and returns relevant information about the term. If it is a class term that has an anchor mapping (lines 8-9), the analysis of its properties (e.g. number of mapped properties) from the mapping analysis file is returned. If it is a class term that is not an anchor but appears on an anchor path then the name of the anchor path on which it appears is returned and in addition the match analysis for that term is returned if it exists (lines 10-21). If the term is a class term but does not appear on an anchor path, then the match analysis for that term is returned if it exists (lines 22-33). In all cases where a class is involved, full details of the class (subclasses, superclasses, properties and their mappings or matches) is also returned using the `su:get_full_analysis_class` XQuery. If the term is a property term, then the equivalent mapping is returned (lines 42-44) if it exists, or the match analysis for that property is returned if it exists (45 to 56).

```

1. declare function
   SU:SMU($termlist,$anchors,$mapping_analysis,$detail)
2. {
3.   for $term in $termlist
4.   return
5.   (
6.     let $anchor :=
       doc($mapping_analysis)/*/*/anchor[normalize-
       space(name/text())=normalize-space($term)]
7.     return
8.       (if (exists($anchor/name))
9.         then ($anchor,
10.            SU:get_full_analysis_class($term,$anchors,$mapping_analysi
11.            s,$detail))
12.         else
13.           let $onpath_list := (for $onpath in
14.             doc($mapping_analysis)/*/*/valid_path,
15.             $onpathnode in $onpath/nd
16.             where normalize-
17.             space($onpathnode/name/text())=normalize-space($term)
18.             return
19.               <class_in_path>
20.               <name>{$term} </name>
21.               <in_path>{$onpath/matches/text()}</in_path>
22.               {$onpathnode/match}
23.               </class_in_path>)
24.           return
25.           (
26.             if (exists($onpath_list/*))
27.             then ($onpath_list[1], <full_details>
28.                SU:get_full_analysis_class($term,$anchors,$mapping_analysi
29.                s,$detail) </full_details>)
30.             else
31.             (
32.               let $class :=
33.                 (doc($detail)/*/*su:Class[normalize-
34.                 space(su:Name/text())=normalize-space($term)])
35.               return
36.               if (exists($class/*))
37.               then
38.                 (
39.                   if (exists($class/complete_match) or
40.                     (exists($class/partial_match)))
41.                   then
42.                     <match_class> {$class}
43.                     </match_class>
44.                   else (<nomatch_class> {$class}
45.                        </nomatch_class>),
46.                   <full_details>
47.                   SU:get_full_analysis_class($term,$anchors,$mapping_analysi
48.                   s,$detail) </full_details>
49.                 )
50.               else
51.               let $properties := (<props>
52.                 {for $property in
53.                   (doc($detail)/*/*su:Class/su:DatatypeProperty,doc($detail)/
54.                   /*/*su:Class/su:ObjectProperty),$propertyname in
55.                   $property/su:Name/text()
56.                   where normalize-space($propertyname)
57.                   = normalize-space($term)
58.                   return $property } </props>)
59.               return
60.               (

```

```

42.         if
43.         (exists($properties/*[owl:equivalentProperty]))
44.         then
45.             <equivalent_property>
46.             {$properties/*[owl:equivalentProperty]}
47.             </equivalent_property>
48.         else
49.         (
50.             let $ps := (for $p in
51.             (doc($detail)/*su:DatatypeProperties/su:DatatypeProperty,
52.             doc($detail)/*su:ObjectProperties/su:ObjectProperty), $pname
53.             in $p/name/text()
54.             where normalize-space($pname) =
55.             normalize-space($term)
56.             return $p)
57.             return
58.             (
59.                 if (exists($ps//complete_match) or
60.                 ($ps//partial_match))
61.                 then
62.                     <match_property> {$ps} </match_property>
63.                 else
64.                     <nomatch_property> {$term,$ps}
65.                     </nomatch_property>
66.                 )
67.             )
68.         )
69.     )
70. )
71. )
72. }

```

Figure 5-25: Snippet of OisinSMU application XQuery

Strengths/Weaknesses/Limitations

A strength of the OisinSMU implementation is the amount of information that is returned. This provides the application with the opportunity to map at a deep or shallow level as all the information related to committed mappings and candidate matches is made available. A limitation of the current implementation is that the interface is based on an individual term basis and thus several interactions may be required with the service as an application creates its mappings. For example, if a term is found to be a class on an anchor path, then the other terms on that anchor path would have to be requested separately if an analysis of those terms is needed. This problem could be overcome by including in the API, a parameter to indicate whether associated information is required.

The next section presents a walk through for the Anchor Capture tools that have been developed.

5.2 Graphical User Interface walk through for Anchor Capture

This section provides a brief overview of Graphical User Interfaces implemented for the semi-automated versions of the application that supports the committed/anchor mapping capture activity. Section 5.2.1 presents a walk through of the OisinAC-semi-v1 application. Section 5.2.2 describes the OisinAC-semi-v2 interface.

5.2.1 OisinAC-semi-v1 walk through and workflow

The OisinAC-semi-v1 mapping capture tool (see Figure 5-26) allows the user to load into the classes pane, the candidate match (CMB) files generated in the characterisation phase.

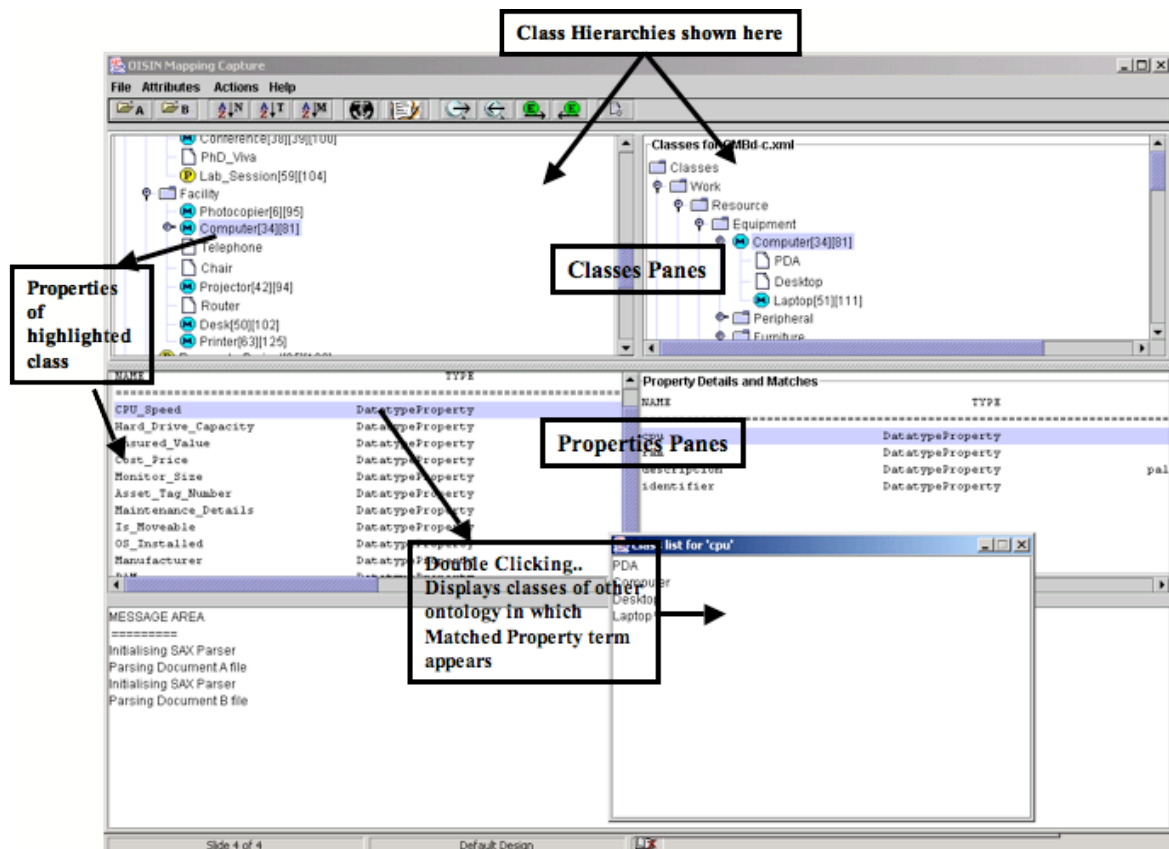


Figure 5-26: Main window of OisinAC-semi application

The classes pane shows the is-a hierarchies of each ontology and whether a class has a candidate match. An **m** icon indicates that the complete term is a candidate match with some term in the other ontology. A **p** icon indicates that part of the term is a candidate match with a term in the other ontology. The numbers in brackets, shown after class names that have candidate matches, are included for the purposes of debugging.

Clicking on a class will show the properties of that class in the properties pane and whether those properties have any candidate matches in the other ontology. Double-clicking on a property that has a property candidate match will show the classes in which that candidate match appears in the other ontology. This latter feature can be useful in trying to determine the usage of that property.

The toolbar for OisinAC-semi is shown in Figure 5-27. The user can select a class of interest and by pressing the appropriate `display candidate match` buttons the user can have the candidate matches for that class highlighted in the other ontology's class pane. In addition, by using the `sort properties` buttons the properties in the panes can be reordered according to property name, property type, or candidate matches.

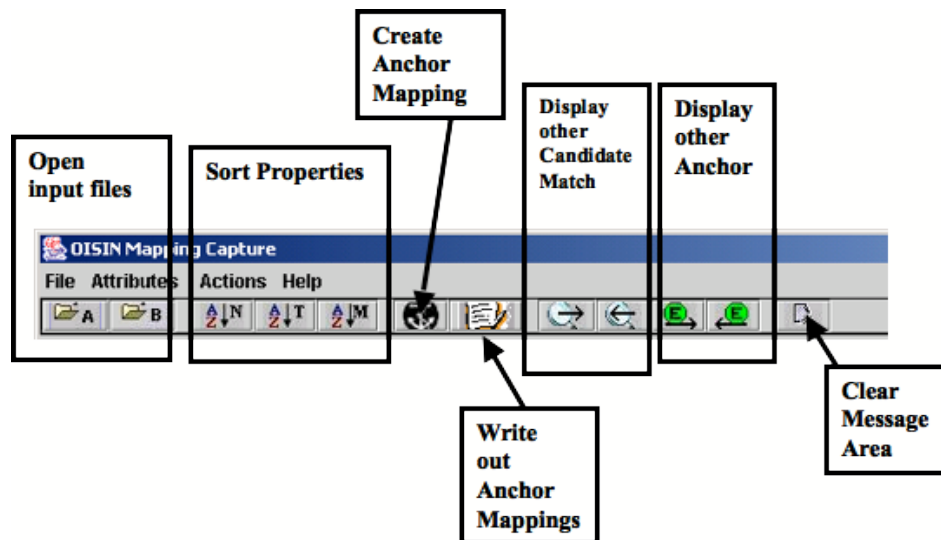


Figure 5-27: Overview of toolbar of OisinAC-semi

In order to create an anchor mapping between two classes, the user selects one class from each class pane and presses the `create anchor mapping` button. Note that the selected classes do not have to be ones discovered by OISIN as candidate matches, as the user has full control as to which classes or properties are mapped. Once the `create anchor mapping` button is pressed, the `Create Mapping` pane shown in Figure 5-28 appears.

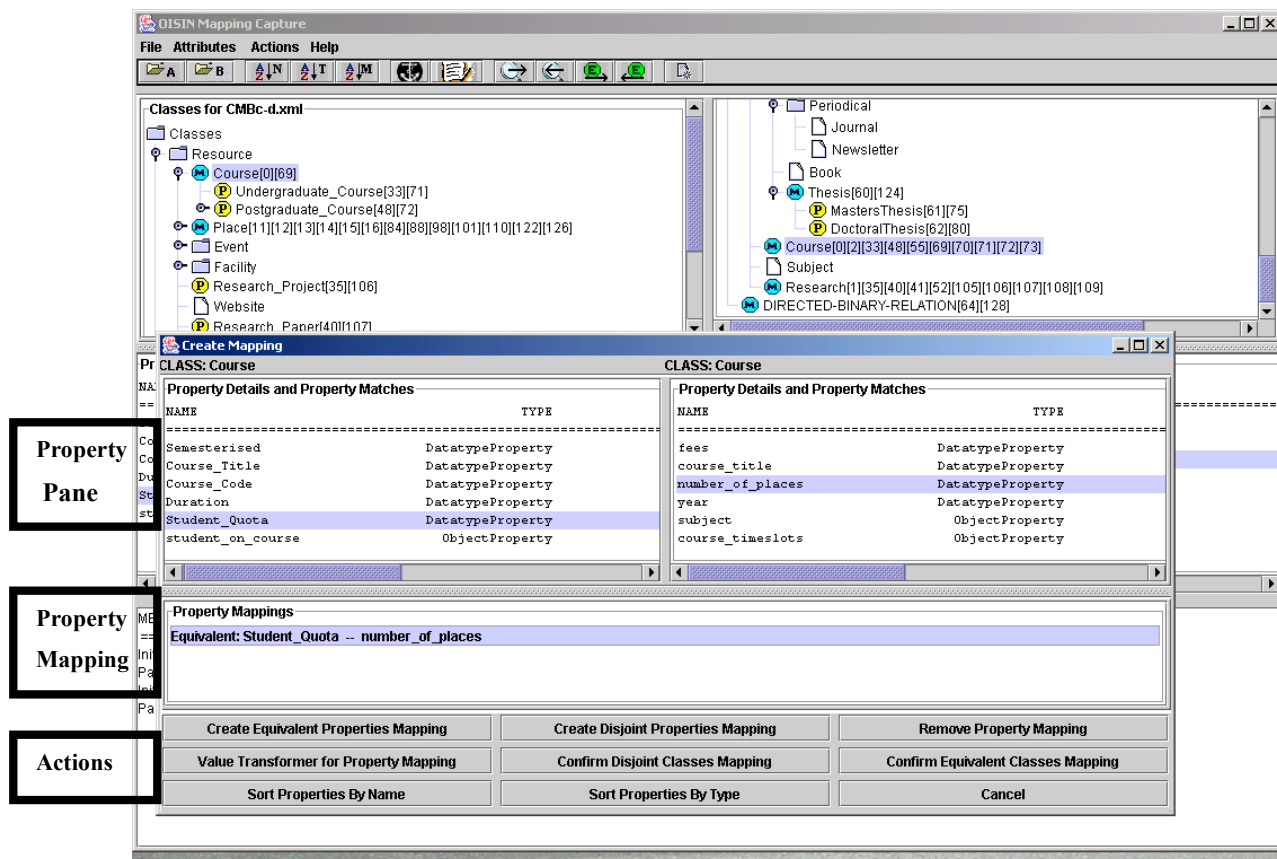


Figure 5-28: Overview of Anchor Capture Panel

The Create Equivalent Properties Mapping and Create Disjoint Properties Mapping buttons will create a mapping between the properties that are selected (student_quota and number_of_places in the example screenshot). The user is asked if this mapping should be created wherever these properties are found in a class. Obviously the user needs to be clear that this mapping holds true for all situations in order to answer this. Once created, the mapping appears in the property mapping pane. Property mappings can be removed by selecting the mapping in the property mapping pane and pressing the Remove Property Mapping button. The value Transformer for Property Mapping button allows the user to associate a code file that can be used at runtime to transform values from one property range to another. The Confirm Equivalent Classes Mapping and Confirm Disjoint Classes Mapping buttons are used to create a class to class mapping. Note that this can be done without having any property mappings specified.

Once a class is mapped as equivalent an E symbol will appear beside it. In a similar way a D denotes a disjoint mapping. It is possible for a class to be involved in several

equivalent or disjoint anchor mappings at the same time, simply by creating mappings as indicated earlier. To find the equivalent mappings for a class, the user single-clicks the class so that it is highlighted and presses one of the `Display Anchor` buttons shown in Figure 5-27, depending on the pane on which the class resides. The equivalent mappings for that class will be highlighted in the other pane. To display the equivalent and disjoint mappings for a class at the same time, the user double clicks the class, and both equivalent and disjoint mappings are displayed in a new pane. For example, the details of mappings for `Thesis` is shown in Figure 5-29. To delete a mapping, the user selects the relevant mapping and presses the appropriate `Delete` button. To alter a mapping, the user proceeds as if creating a mapping between the two classes and alters the mapping (e.g. delete property mappings etc.) as needed.

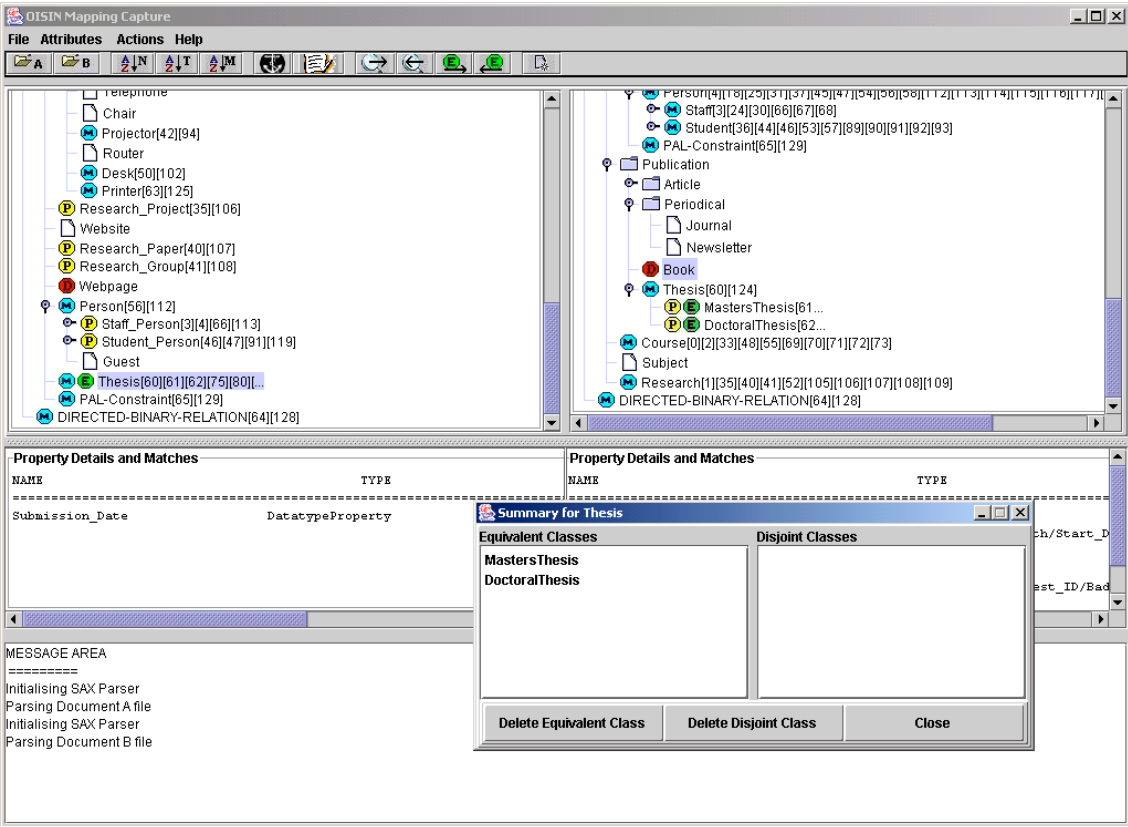


Figure 5-29: Example of Mapping Details window

A workflow was designed to support a user of a semi-automatic application in undertaking the anchor mapping capture. This workflow has been designed such that it can capture a minimum or maximum number of anchors.

The minimum number of anchors would need to be captured in those cases where the determination of mappings is being left to an application. For example, take the case of

a visiting lecturer whose profile is defined in ontological terms defined by his home university. A wide range of applications will need to discover for themselves the appropriate mappings between the university ontologies. For a room booking application the mapping could be based on a weak match as the application is not expending expensive organisational resources. Thus the fact that `Associate Lecturer` only weakly matches to `Assistant Lecturer` but the superclasses of both `Lecturer` and `FullLecturer` have been identified as an anchor mapping, could be sufficient to allow the visitor book a room. On the other hand, for some application requiring the expenditure of budget (e.g. buying equipment) then the mapping would need to be based on an anchor.

The maximum number of anchors could be captured in a situation where all the mappings are to be determined by an expert user. This would be the case if the envisaged scenario for usage of mappings is well known and the mappings themselves will be the same across a wide range of applications.

The workflow designed to support the minimum and maximum identification of anchors is shown in Figure 5-30. The workflow does not require every anchor mapping between the two ontologies to be enumerated and the extent of the mapping discovery can be determined depending on the envisaged deployment of the mappings. One ontology is chosen (the source) and each is-a subtree is examined one at a time starting from its root if a full mapping approach is being taken. Otherwise if a partial mapping approach is being taken, a subset of is-a subtrees are examined.

Starting at the root, each class in the subtree is considered and a potential equivalent class in the other ontology (the target) is searched for. The user is not limited to the marked candidate matches. If a potential anchor pairing is identified, the anchors of the pair are checked to see if they can be confirmed as equivalent. This confirmation can be undertaken by examination of the properties, or other criteria can be used. If confirmed as an anchor pairing, then the workflow suggests looking at the classes of the is-a subtree of the anchor of the ontology, working from the leaves upwards in an attempt to identify the lowest possible anchors of each subtree where the anchor is a root.

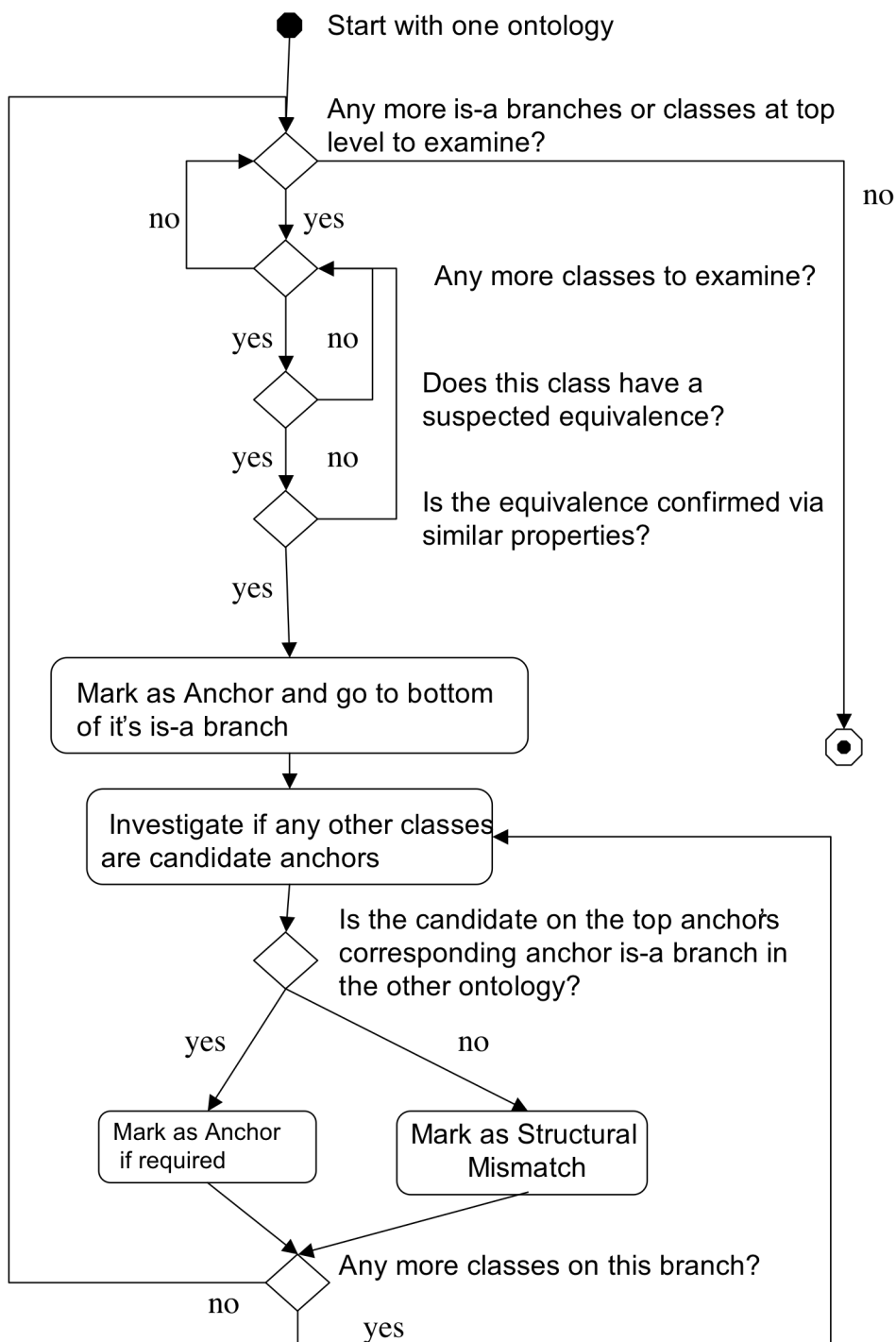


Figure 5-30: Anchor Capture Workflow

A structural mismatch could be detected at this stage if the corresponding anchor of the lowest possible anchor is on a different subtree in the target ontology. Although OisinMA provides a means to identify such mismatches later automatically, the workflow is designed to help the user to spot and avoid the mismatches in the first place. Once the lowest anchor is determined, either more anchors are sought in this

subtree (if taking a maximum mapping approach) or the focus shifts to another subtree (if a minimum mapping approach is being taken).

5.2.2 OisinAC-semi-v2

A number of improvements (cf. section 5.1.9.2) were made to the OisinAC application based on feedback from the experimentation.

Some of these improvements were made to improve usability of version one. As can be seen in Figure 5-31, the inherited properties are now distinguished by font colour (with blue indicating an inherited property), the Mapping Analysis application can now be invoked from the Mapping Capture window to check for structural mismatches and the ability to save/retrieve in-progress mapping sessions has been added. In addition, before OisinAC closes down, it now reminds the user of candidate class matches that have not been mapped.

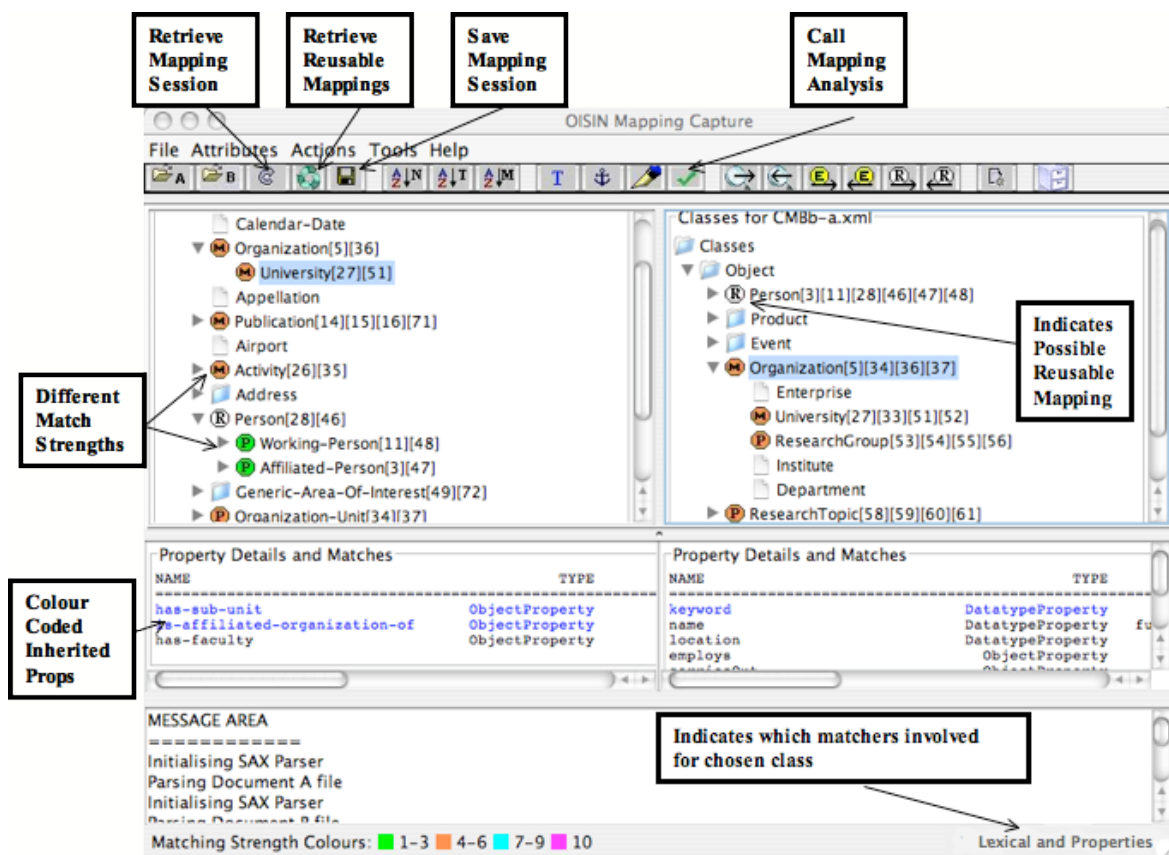


Figure 5-31: OisinAC-semi-v2 screenshot

Other improvements that provide additional functionality are also highlighted in Figure 5-31. The first addition is the retrieve reusable mappings functionality that allows the user to select pre-existing anchor mapping files and have the mappings for classes displayed on the screen. This is indicated using the `R` icon. This is purely for indicative purposes however, and the user is still required to create mappings in the normal manner. The second addition is the colour coding of matches based on the combined strength element of the match entity. This was not used in version one as only one matcher (OisinMO-lex) was used. In addition, if the user highlights a candidate match pair, an indication will be provided in the bottom right hand corner of the window as to the matcher strengths, which were combined to achieve the combined strength score.

In this way, the OISIN implementation has evolved over the lifetime of the thesis and further enhancements are planned for the future.

This chapter has presented the software tools that have been developed to support the OISIN ontology mapping process. Chapter six presents the experiments that were undertaken to evaluate the developed process and supporting software tools and which informed the evolution of the implementation.

6 EVALUATION

This chapter presents how the ontology mapping process and tools that were developed and implemented for the thesis (described in chapters four and five respectively) have been evaluated. Section 6.1 presents an introduction to the experiments undertaken. Sections 6.2 and 6.3 describe details of each of the experiments undertaken and the findings that resulted.

6.1 Introduction

Two experiments were designed. The first experiment evaluated the Characterisation Phase by means of focusing on the *Decide to Map* activity. The second experiment evaluated the *Mapping Phase* activities.

Decide to Map experiment

It was decided to evaluate the process and tools of the characterisation phase by evaluating the usefulness of the amenability analysis information created during the phase, as this would be more insightful than an evaluation that just concentrated on the execution of the calculations. Recall that the amenability analysis information comprises information generated by all tools in the phase. The evaluation focused on the *Decide to Map* activity where the information from the characterisation phase is envisaged being used as a means of indicating the potential difficulty in mapping between a set of ontologies. Another reason for this focus was that the *Decide to Map* activity is one that is seldom, if ever, explicitly supported by state of the art frameworks. It was our belief that the common practice of making the decision to map based on just examining the ontologies was too subjective and error prone to be an industrial strength activity useful within an organisation. Thus two hypotheses were derived for testing.

The first hypothesis was that making the decision to map by just examining the ontologies directly would lead to inconsistent recommendations on the same sets of ontology pairs. This is also very subjective, and dependent on the expertise of the user involved. Testing this hypothesis would confirm or deny our contention.

The second hypothesis was that the information produced as a result of the characterisation phase would be useful in the generation of recommendations to map that would be as good, if not better, than those recommendations made as a result of browsing the ontologies directly. The significance of proving this hypothesis would be that: the information produced by the characterisation phase would be shown to be useful and that potentially the decision to map could be automated without loss of accuracy, with the potential benefits of time saving, labour saving and scalable deployment.

The design of the experiment involved:

- the discovery of five pairs of ontologies whose semantics overlap to different extents;
- the use of OISIN characterisation phase tools to pre-process the five pairs of ontologies;
- the establishment of a gold-standard set of decisions to map for the five pairs of ontologies;
- the discovery of what decisions to map would be made on the five pairs of ontologies by ten participants who just examined the ontologies directly; Protégé was used as a browser which was most familiar to the experiment participants.
- the establishment of what decisions would be made on the five pairs of ontologies by a different set of participants who used the OISIN tools;
- the comparison of the non-OISIN supported decisions, the OISIN supported decisions, the gold-standard decisions; This comparison of results was possible as each person from each group had been paired based on their ability before the experiments began. Given the small population sample size and the unknown standard deviation, the decisions of each student pair were analysed using the 'student' t-distribution. The data was also statistically analysed using a general linear model and an analysis of variance (ANOVA) was also undertaken.
- the gathering of feedback via a questionnaire on the difficulties found by the participants in making the decision to map recommendations and the confidence they had in those recommendations.

Details of this experiment and the resultant findings can be found in section 6.2.

Committed Mappings experiment

As established in our initial investigation work described in section 4.1, the “identification of mappings” task and “identification of structural mismatch” task were identified as tasks requiring support in the mapping phase, especially when undertaking a minimum number of committed mappings between ontologies. Two evaluation questions naturally arose:

- Would others find that the tasks identified need to be supported?
- To what degree do the tools as currently implemented assist the undertaking of the tasks?

Considering our two evaluation questions, two hypotheses were tested. The first hypothesis was that the mapping and structural mismatch identification tasks do require tool support. The second hypothesis was that the OISIN tools are highly usable, assist the undertaking of these tasks in the mapping process and lead to the successful generation of mappings.

The design of the experiment involved:

- the discovery of two pairs of significantly overlapping ontologies;
- the application of OISIN characterisation phase tools to pre-process the two pairs of ontologies for use by the four groups (consisting of three participants each) taking part in the experiment;
- the establishment of a gold-standard set of mappings for the ontology pairs;
- the discovery of what the mapping results for the ontology pairs would be without the support of any dedicated mapping tools; Protégé and the PROMPT plug-in were offered to the experiment groups for partial support;
- the establishment of what the mapping results for the ontology pairs would be as a result of the use of OISIN tools; The same groups undertook the mapping tasks but upon the pair of ontologies which they had not previously mapped.
- the comparison of the OISIN supported mappings, the non-OISIN supported mappings and the gold-standard mappings. In addition, the performance of each group with and without OISIN support needed to be examined.
- the gathering of feedback on the usability of the tools via a questionnaire.

Details of this experiment and the resultant findings can be found in section 6.3.

6.2 Decide to Map Activity Experiment

One potential benefit of the *Decide to Map* activity is the avoidance of mapping ontologies that do not meet organisation specified criteria, e.g. minimum degree of overlap, semantic quality and so on. Another benefit is in the indication of the degree of difficulty that may be involved in a mapping task, even in cases where there is no choice but to map. Section 6.2.1 presents the objective of the experiment and provides an overview. The results of each part of the experiment are then presented in Sections 6.2.2 and 6.2.3, with the statistical analysis of results and discussion of findings described in section 6.2.4.

6.2.1 Overview

The first hypothesis to be tested in this experiment was that making the decision to map by just examining the ontologies directly would lead to inconsistent recommendations on the same sets of ontology pairs. The second hypothesis was that by just using the pre-processed information about ontologies generated by OISIN, that recommendations to map would be determined, that would be at least as good as those made as a result of examining the ontologies directly.

For this experiment nine ontologies were chosen and split into five pairs. Each ontology was assigned a letter, and each pairing was named using the combination of these letters. As shown in Table 6-1, the pairings were chosen such that two pairs were non-overlapping; one pair was partially overlapping but had different domains; and two pairs were overlapping.

Ont Pairs	Ont	Ref	# Classes	# Properties	Domain
AB	A	(Aberdeen 2003)	56	66	University
CT	B	(Manchester 2003)	96	92	University
FH	C	(TCD1 2004)	63	143	University
PE	D	(TCD2 2004)	66	92	University
CD	E	(TCD1 2005)	45	79	Social Life
	F	(W3C 2003b)	137	15	Wine
	H	(Stanford 2003)	34	10	Travel
	P	(IUPUI 2004)	75	0	Protein
	T	(SWETO 2003)	114	56	General

Table 6-1: Overview of Experiment Ontologies

Having examined closely each pair of ontologies by hand, a “proceed to mapping” decision was made by this author for each pair of ontologies in the range: No, Probably Not, Possibly, Definitely. This scale mapped to an anticipated degree of mapping difficulty. Of course in any particular organisation the scale may differ or relationship of the degree of difficulty to the scale would be different. Two ontology characteristics in particular were chosen as the main criteria to be used in the decision making for this experiment. The first criterion was applied where overlap in the ontology domains occurred. If overlap occurred the criterion to be checked was whether there was good candidate match prospects between the classes in the overlapping part. The second criterion related to the quality of the ontologies. The resulting decisions formed the gold-standard for the experiment and these are shown in Table 6-2. The gold-standard was verified by an independent researcher with similar ontology mapping experience. This gold-standard was used to analyse the decisions resulting from each part of the experiment.

Ontology Pair	Decision	Reason
AB	Definitely	Large overlap and quality of each ontology is good
CT	Possibly	Small overlap but doubt about quality of C
FH	No	Not overlapping, although in theory could be
PE	No	Not overlapping, and ontology P is very domain specific
CD	Possibly	Large overlap, but doubt about quality of both C and D

Table 6-2: Gold-standard mapping decisions

Twenty students participated in the experiment, ten postgraduate-by-research students from the Knowledge and Data Engineering Group and ten final year Computer Science students from the 2004/2005 BA/BAICT course. To enable later statistical analysis of the decisions, the students were paired off in terms of similar ability and then one from each pair randomly allocated to the first or second part of the experiment, ensuring that each experiment had five postgraduate and five final year students each. There was a number of non-native English speakers included but these were not treated differently.

The experiment was designed in two parts. In the first part, a group of students were asked to examine the ontologies using the Protégé tool and to decide whether or not to map the pairs of ontologies. This tested our first hypothesis that just examining the ontologies would lead to inconsistencies in recommendations, as well as providing a baseline set of information to test our second hypothesis. The second part of the

experiment required the second group of students to decide whether to map the same pairs of ontologies, but without sight of the ontologies themselves and using only information generated by the OISIN characterisation phase. This information was presented to the participants via a Microsoft Excel spreadsheet, as would be presented via OisinDTM-semi described in section 5.1.8. This tested our second hypothesis that OISIN supported decisions without sight of the actual ontologies would be at least as good as decisions based on examining the ontologies directly.

The decisions from the two parts of the experiment were then statistically compared with respect to the gold-standard set of decisions generated.

6.2.2 Decisions based on examining Ontology directly

The first group of students were requested to make their decisions by examining the ontologies side by side using Protégé as a browser. The decision sheet (Appendix D-1) asked the participants to record:

- experience of working with ontologies;
- experience of mapping between ontologies;
- their decisions for each ontology pair and any associated comments;
- level of difficulty found in making the decisions;
- confidence in the decisions made.

All students were told at the start of the experiment that the threshold for what would be a match would be quite low, and that the matching should really be concentrated at the class level. The two main criteria used in the creation of the gold-standard decisions (that is good match prospects of any overlap and quality of ontologies) were explained. For the overlap criteria, this involved looking for overlap by simply browsing each ontology side by side. For the quality criterion, it was explained that this would be difficult just by browsing alone but that one possible indicator would be the way in which composite terms presented themselves in an ontology. Inconsistent patterns for composite terms (e.g. a mixture of hyphenated and underscored patterns) would be an indicator of a poorly engineered ontology. Table 6-3 represents the decision sheets of students one to five and Table 6-4 shows the information from the decision sheets for students six to ten.

Summary of Decision Sheets where
ontologies examined using Protégé

	Previous experience of working with ontologies	Previous experience of mapping between ontologies	Comment
Student 1	a lot	very little	Time taken: 23 minutes
	Should map AB?	Definitely	Most classes have equivalences in other ontology
	Should map CT?	Definitely	Some overlap structure from T might map to Building/Space in C
	Should map FH?	Probably Not	F is much more fine grained about food while H is more general
	Should map PE?	No	nothing in common
	Should map CD?	Definitely	Lots of overlap
	Difficulty coming to decisions?	Fairly Easy	Scanning the concepts allows you to quickly ascertain if the ontologies are about the same things
	Confidence in decisions	Confident	Most pairs were easy to identify whether they were mappable once the concept hierarchy had been opened
Student 2	just theory	none	Time taken: 15 minutes
	Should map AB?	Possibly	No 1 to 1 match but some mapping possible
	Should map CT?	Probably Not	very different domains IMHO
	Should map FH?	No	zero overlap maybe they belong in same ontology?
	Should map PE?	No	
	Should map CD?	Definitely	Not 1 to 1 but clear overlap
	Difficulty coming to decisions?	Fairly Easy	Not sure sometimes as to what defines a solid definitely but its usually fairly apparent if a mapping is possible
	Confidence in decisions	Very Confident	I think I understand notions of equivalency, "has-a" and "is-a" sufficiently to be able to backup my decisions
Student 3	just theory	none	Time taken: 25 minutes
	Should map AB?	Possibly	
	Should map CT?	Probably Not	T is much more higher level, C delves down
	Should map FH?	No	completely different concepts
	Should map PE?	No	completely different concepts
	Should map CD?	Definitely	Some similar elements
	Difficulty coming to decisions?	Fairly Difficult	It was relatively easy for similar ontologies and likewise for completely different ones but harder for ones with some elements in common
	Confidence in decisions	Sort of Confident	I would be afraid that I missed some similarities between some ontologies
Student 4	some	none	Time taken: 35 minutes
	Should map AB?	Definitely	Good for partial match
	Should map CT?	No	
	Should map FH?	No	
	Should map PE?	No	
	Should map CD?	Definitely	Good matching
	Difficulty coming to decisions?	Fairly Difficult	Easier for full match than partial match where needed to
	Confidence in decisions	Confident	
Student 5	none	none	Time taken: 24 minutes
	Should map AB?	Definitely	Similar ontologies
	Should map CT?	Definitely	Mapping similarities
	Should map FH?	No	No similarities
	Should map PE?	No	No similarities
	Should map CD?	Definitely	ontologies have mapping similarities
	Difficulty coming to decisions?	Easy	
	Confidence in decisions	Very Confident	

Table 6-3: Summary of Decision Sheets for Protégé supported Students 1 to 5

**Summary of Decision Sheets where
ontologies examined using Protégé**

	Previous experience of working with ontologies	Previous experience of mapping between ontologies	Comment
Student 6	some	very little	Time taken: 19 minutes
	Should map AB?	Definitely	same general domain of academia
	Should map CT?	Probably Not	only academic department mildly overlaps with research
	Should map FH?	No	totally different areas of interest
	Should map PE?	No	Again no overlap between what domain each ontology is
	Should map CD?	Possibly	partial overlap
	Difficulty coming to decisions?	Fairly Easy	
Student 7	a lot	very little	Time taken: 23 minutes
	Should map AB?	Definitely	there are a lot of concepts common to both
	Should map CT?	Possibly	C it too specific, T is too general
	Should map FH?	No	different domains. No obvious overlap except maybe for
	Should map PE?	No	Completely different domains
	Should map CD?	Probably Not	same domain different concepts little overlap
	Difficulty coming to decisions?	Fairly Easy	Easy to identify when domains are different. Easy to see what concepts are included in each ontology. Main classes are usually a good indicator.
Student 8	some	very little	Time taken: 27 minutes
	Should map AB?	Possibly	Some similarity but each concentrates on different areas
	Should map CT?	No	Little similarity between C and T, except in names of upper
	Should map FH?	Probably Not	no obvious mappings apparent
	Should map PE?	Probably Not	no obvious mappings apparent
	Should map CD?	Possibly	in same domain and plenty of similarities but similar classes
	Difficulty coming to decisions?	Fairly Easy	Just looked to see if ontologies in same domain and to see if any classes matched up and looked at their structure.
Student 9	some	none	Time taken: 30 minutes
	Should map AB?	Possibly	A seems more logical and some overlaps
	Should map CT?	Probably Not	No overlaps in the first classes
	Should map FH?	No	F focuses on food whilst H focuses on travel
	Should map PE?	No	Makes no sense to try to map these
	Should map CD?	Possibly	Overlaps in several classes
	Difficulty coming to decisions?	Fairly Easy	The first ontology comparison was more difficult because I had no experience. So I would think with more and more ontologies seen and compared my decisions are more confident
Student 10	some	very little	Time taken: 26 minutes
	Should map AB?	Possibly	
	Should map CT?	Definitely	
	Should map FH?	No	
	Should map PE?	No	
	Should map CD?	Definitely	
	Difficulty coming to decisions?	Fairly Easy	As decision not made not on details
	Confidence in decisions	Confident	If I understood the instructions correctly

Table 6-4: Summary of Decision Sheets for Protégé supported Students 6 to 10

Although all students reported very little or no experience of mapping ontologies, only three of the students reported very little or no experience of working with ontologies. On average the task took twenty-three minutes to complete, with the first being finished after fifteen minutes and the last student concluding after thirty-five minutes. The fifty decisions resulting from this task were then analysed to see how far these decisions were from the gold-standard. Figure 6-1 shows a distribution of the decisions with twenty-eight decisions aligning with the gold-standard, twenty decisions being either side of the gold-standard and two decisions being two points away from the gold-standard.

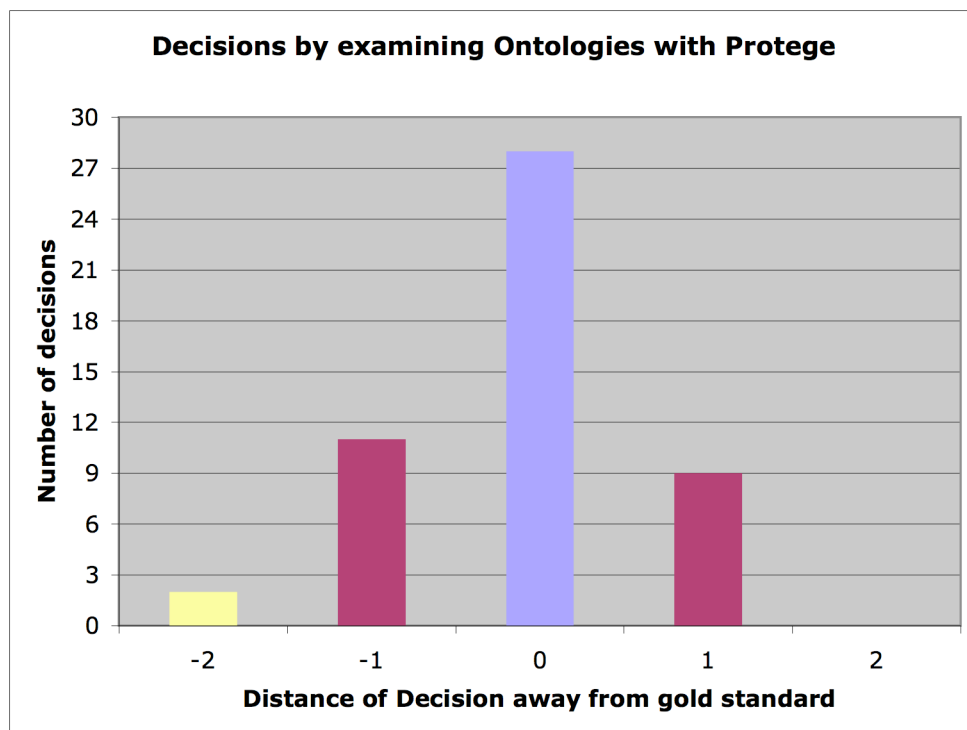


Figure 6-1: Chart of Results for Protégé supported decisions

The students were also requested to comment on the confidence they had in their decisions and the difficulty of the task. Eight of the ten students were both “confident” or “very confident” in their decisions and found the task “fairly easy”.

Discussion and Findings

The analysis shows that inconsistent recommendations did result when the same pairs of ontologies were examined by different people, supporting our hypothesis that just relying on examining the ontologies directly as a basis for decision making would lead

to inconsistencies. The inconsistent recommendations may arise from the diverse approaches to browsing taken. Evidence of diversity of browsing approaches can be found in some of the comments made by the participants. For example student six stated “I thought about the domain of interest each ontology was representing and not considering the class hierarchies or properties”. In contrast student eight commented “Just looked to see if ontologies in same domain and to see if any classes matched up and looked at their structure.” Inconsistencies could also have arisen from students having difficulty applying the decision-making criteria suggested. However it is difficult to find direct evidence for this point in the student comments.

The question which naturally arises is whether the inconsistencies arise more because of the browsing nature of the examination of the ontologies or because of difficulty in the application of the decision making criteria. This question is explored further in the next part of the experiment.

6.2.3 Decisions based on OISIN generated information

The second group of students was requested to make their decision to map recommendations by examining the information that was pre-processed for each of the ontologies by OISIN and without being able to examine the ontologies directly. This information was presented in the form of a four page Microsoft Excel spreadsheet. The rationale for each piece of information has already been outlined in Section 5.1.8.1. For convenience, a summary of the information that appears on the four pages is provided in Table 6-5. An example of the Microsoft Excel spreadsheet information generated for the AB mapping is also included as Appendix D-2.

Page 1	Basic lexical information about the ontologies, including number of terms, number of composite terms, and the make up of those composite terms (hyphenated, underscored or middleCapital);
	The number of terms or parts of composite terms which matched against the WORDNET database; The reuse of the ontologies of other ontologies and how well reused the ontology is (through SWOOGLE information);
Page 2	The dimensions of the ontologies (number of classes, properties, relationships) and the shape of the ontologies (that is number of classes at each level of the is-a hierarchy);
Page 3	The number of candidate matches based on lexical matching and an indication of the number of potential mappings arising;
Page 4	The number of potential mappings that has matching types (that is class to class, property to property).

Table 6-5: Overview of OISIN generated Excel spreadsheet

Before the experiment the main information elements were explained in detail. The two main criteria used in the creation of the gold-standard decisions (that is good match prospects of any overlap and quality of ontologies) were explained. In the first part of the experiment the students were requested to make their decisions purely based on the information without any further guidance. The decision sheet used was exactly the same as that used by the group who made their decisions based on just examining the ontologies using Protégé. Table 6-6 represents the decision sheets of students one to five and Table 6-7 shows the information from the decision sheets for students six to ten. The tables show for each student the prior knowledge of ontologies and ontology mapping, the decisions, the comments, the difficulties/confidence in making the decisions, and the time taken to complete the task.

Summary of Decision Sheets where ontologies examined using OISIN generated information only

	Previous experience of working with ontologies	Previous experience of mapping between ontologies	Comment
Student 1	a lot	very little	Time taken: 21 minutes
	Should map AB?	Possibly	B has more terms than A yet A has a larger % composite terms therefore may be similar at another level Yet matches are not strong and strong type matching
	Should map CT?	No	Large volume of terms known. Low matches
	Should map FH?	No	Large difference between no of terms, matches extremely low
	Should map PE?	No	Large difference between no of terms, known terms and matches. P very low on unique attributes versus E
	Should map CD?	Definitely	Similar number of terms Imports similar no of ontologies Similar overlap in dimensions. Good match ration and type match results
	Difficulty coming to decisions?	Fairly Difficult	I thought data was more assisted guide. Kept thinking what I like/dislike could be way off reality. I could have done it easier with added info about each ontology but maybe that's the point
	Confidence in decisions	Sort of Confident	Tough to know if my method was sound
Student 2	just theory	none	Time taken: 15 minutes
	Should map AB?	Possibly	Both have good wordnet analysis but fairly poor potential matches
	Should map CT?	Probably Not	Similar shape and style. Very good wordnet analysis. Very poor candidate matching
	Should map FH?	No	Terrible matching stats, but good wordnet analysis
	Should map PE?	Probably Not	Terrible matching stats. Possibly bad spelling in P
	Should map CD?	Definitely	Similar layout, good matching stats
	Difficulty coming to decisions?	Fairly Difficult	Obviously more information is required
	Confidence in decisions	Not Confident	Could be wrong depending on domain
Student 3	a lot	very little	Time taken: 16 minutes
	Should map AB?	Definitely	Syntactically they are quite similar with the syntactic differences identified it should be easy to map
	Should map CT?	Probably Not	Although type mapping is quite high, based on % of unknown terms in C it seems to be quite domain specific
	Should map FH?	No	Seems like no real similarities
	Should map PE?	No	Over 30% unknown terms in P. Both are not well referenced. Potentially not well written given percentage of OWL keywords
	Should map CD?	Possibly	Relatively high percentage of similar type mappings
	Difficulty coming to decisions?	Fairly Easy	
	Confidence in decisions	Confident	
Student 4	just theory	none	Time taken: 23 minutes
	Should map AB?	Possibly	Very high same type mapping. High percentage of potential matches although different shapes
	Should map CT?	Probably Not	Same shape and composite terms but low percentage matches and mappings
	Should map FH?	No	Similar shape and structure but basically no direct or potential matches
	Should map PE?	No	Very different shape, high % unknown in P. Very low % matching and mapping
	Should map CD?	Definitely	Similar shape. D has one extra level. Extremely high potential mappings
	Difficulty coming to decisions?	Difficult	Hard to decide with no subject information and some conflicting analysis results
	Confidence in decisions	Sort of Confident	Not very confident but a bit more so having gone through five pairs
Student 5	none	none	Time taken: 22 minutes
	Should map AB?	Probably Not	Difference in depths
	Should map CT?	Possibly	C could be a subset of T
	Should map FH?	No	Don't show good matches
	Should map PE?	No	Don't show good matches
	Should map CD?	Definitely	Good symmetry, Good matches
	Difficulty coming to decisions?	Fairly Difficult	I don't have a very good understanding of OWL and terminology. Graphs useful though.
	Confidence in decisions	Not Confident	I based my decisions on some very limited experience I am not sure I have a broad enough understanding

Table 6-6: Summary of Decision Sheets for spreadsheet supported Students 1 to 5

Summary of Decision Sheets where ontologies examined using OISIN generated information only

	Previous experience of working with ontologies	Previous experience of mapping between ontologies	Comment
Student 6	some	some	Time taken: 13 minutes
	Should map AB?	Probably Not	Differing no of levels in ontologies and not good mappings but good type mappings
	Should map CT?	Probably Not	same no of levels and similar widths but poor mappings
	Should map FH?	Probably Not	Tree structure looks similar but virtually no matches
	Should map PE?	No	Different tree structure and poor matches
	Should map CD?	Possibly	Similar tree structure, good class mappings but poor type matches
	Difficulty coming to decisions?	Fairly Easy	I concentrated my decisions on the general shape of the tree structure and the number of matches made
	Confidence in decisions	Confident	Should the above criteria be valid then I am confident just hope I did not discard vital information
Student 7	some	very little	Time taken: 13 minutes
	Should map AB?	Definitely	Felt B was a more complex version of A
	Should map CT?	Probably Not	What portion that was similar seemed to be very similar
	Should map FH?	No	Overall shape seemed similar but not enough similarities elsewhere
	Should map PE?	No	Felt that similarities in match analysis could be attributed to coincidence
	Should map CD?	Definitely	Seemed very similar
	Difficulty coming to decisions?	Fairly Easy	As progressed through tasks it became easier. First one or two were hard.
	Confidence in decisions	Confident	I am reasonably sure that I have been accurate in my analysis. It seemed reasonably obvious to me which ontologies to map.
Student 8	some	very little	Time taken: 20 minutes
	Should map AB?	Probably Not	Too many differences in syntax and wordnet. Matching not too bad
	Should map CT?	Possibly	The shape and wordnet are good and references not known and matches not good
	Should map FH?	No	Too different and references only to F
	Should map PE?	Probably Not	no obvious mappings apparent
	Should map CD?	Possibly	Shape and matching not good
	Difficulty coming to decisions?	Fairly Difficult	
	Confidence in decisions	Sort of Confident	
Student 9	some	none	Time taken: 14 minutes
	Should map AB?	Possibly	Differences but a half part match and have potential mapping. Types are similar
	Should map CT?	Definitely	Even if not a lot of matches, all are candidate mappings
	Should map FH?	No	Not enough candidate matches or mappings
	Should map PE?	Probably Not	A lot of unknown words, different shapes, only few matches
	Should map CD?	Possibly	Quite same structure, size similar and level of matches is medium and a lot potential mappings
	Difficulty coming to decisions?	<i>not indicated</i>	
	Confidence in decisions	<i>not indicated</i>	
Student 10	just theory	none	Time taken: 18 minutes
	Should map AB?	Possibly	B is bigger and more levels but lots of known or combined terms. Quite good type mappings
	Should map CT?	Probably Not	Even if the dimensions are close, matches are too few
	Should map FH?	No	Nearly no candidate matches
	Should map PE?	Probably Not	Lots of unknown terms in P. Very few matches
	Should map CD?	Definitely	Lots of known terms. Nearly same size and structure. Candidates for matching and good proportion of same type mappings
	Difficulty coming to decisions?	Fairly Difficult	Do not know if the choices are the good ones to make decisions
	Confidence in decisions	Confident	My approach seems logical

Table 6-7: Summary of Decision Sheets for spreadsheet supported students 6 to 10

The average time taken to complete this task was eighteen minutes with the first student completing in thirteen minutes and last student completing in twenty-five minutes. The decisions when compared against the gold-standard are provided in Figure 6-2. This shows twenty-two decisions aligned with the gold-standard, twenty-five one decision point away and three decisions more than one decision point away.

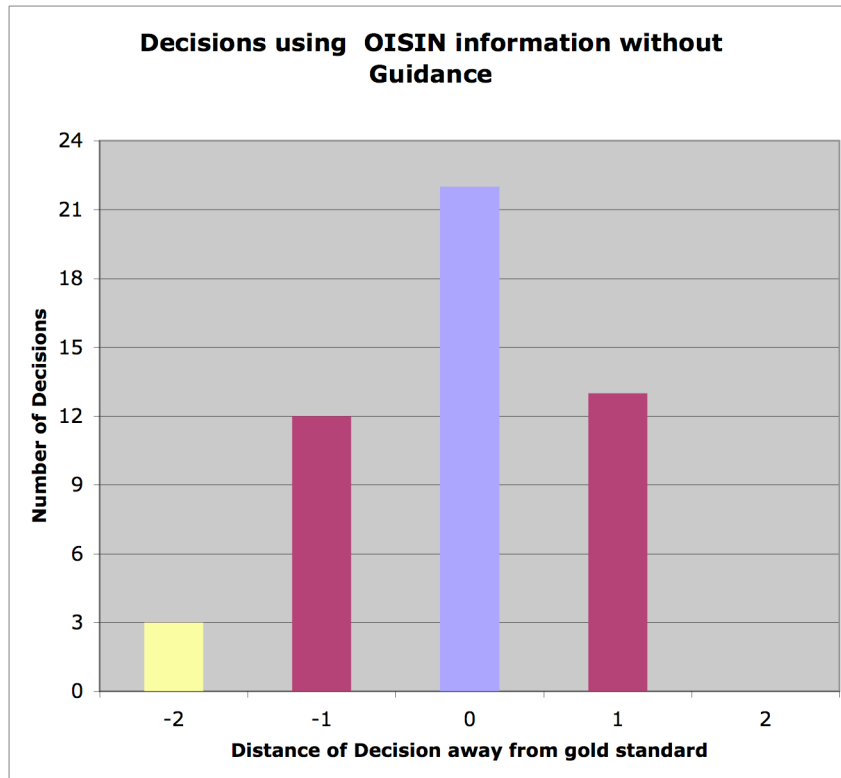


Figure 6-2: Chart of Results for spreadsheet supported decisions

The students were also requested to comment on the confidence they had in their decisions and the difficulty of the task. Seven of the ten students were only “sort of confident” in their decisions and found the task “fairly difficult”.

For the second part of the experiment, the same students were requested to make the decision to map again but this time using a decision sheet that had guidance incorporated on how to interpret the information. Again they were not allowed examine the ontologies directly. Due to the rigidity of the guidance supplied, it is our belief that the experiences of the participant in the first part of the experiment had no influence in their undertaking of the second part.

The guidance was designed to provide concrete advice on how the two overall decision criteria of overlap and quality could be evaluated. Due to the lack of research into the development of such concrete advice, the guidance designed for this experiment is based on the author's experience and essentially are heuristics. Of course, this is a potential limitation as it was not possible to independently verify the guidance or thresholds. However from an experimental point of view, this is not critical as an evaluation of the affect of guidance is being examined rather than the affect of this particular guidance.

The guidance sheet used in the experiment is included in Appendix D-3. The design of the guidance itself was a non-trivial matter, given it required the design of heuristics that could be easily applied. This meant identifying a subset of the measurements generated by the characterisation tools that would be helpful, for designing useful heuristics. In the end it was decided to select two measurements related to the modelling characteristics, three from the semantics characteristics, and two related to the candidate match characteristics.

In summary, the guidance suggested that if less than 10 class matches were found then immediately the decision should be "no", otherwise decisions were suggested dependent on the number of negative responses to a series of questions. Zero or one negative response should indicate a "definitely" decision; two or three negative responses should indicate a "probably" decision and so on.

The questions are listed below and include a brief commentary explaining the rationale for each question:

- Does each ontology compile and have greater than twenty percent of OWL keywords used? This is one partial indicator of quality. A simple ontology uses about twenty percent of all the OWL keywords.
- Does each ontology have greater than ninety percent of composite terms predominantly of one type (that is, hyphenated, underscored or middleCapital)? This is one partial indicator of quality. Well engineered ontologies would have a consistency in the pattern used for naming classes or properties.

- Does each ontology have a low percentage (less than ten percent) of terms unknown in WORDNET? This is an indicator of how domain-specific the ontology is.
- Does each ontology either import or reference other ontologies? This is one partial indicator of quality. Poor ontologies do not attempt to reuse at all.
- Is there SWOOGLE information for each ontology, and if so does it indicate a well reused or referenced ontology? If the ontology is published on the web then SWOOGLE information will exist and reusability/referencing provides one measure of quality.
- Is the majority of the ratios of Candidate Matches to Potential Mappings less than one is to two? This is an indicator for overlap and complexity in undertaking the mapping, with a ratio greater than one is two indicating more complexity.
- Is the percentage of Potential Mappings that do not match with respect to type mapping less than thirty-three percent? This is an indicator of differences in modelling styles.

Table 6-8 shows the decisions made and time taken for each student.

	Time taken in minutes	Should map AB?	Should map CT?	Should map FH?	Should map PE?	Should map CD?
Gold-Standard		Definitely	Possibly	No	No	Possibly
Student 1	19	Definitely	Possibly	No	No	Possibly
Student 2	15	Definitely	Possibly	No	No	Possibly
Student 3	24	Definitely	Possibly Not	No	No	Possibly
Student 4	14	Definitely	Possibly	No	No	Possibly
Student 5	20	Definitely	Possibly	No	No	Possibly Not
Student 6	29	Definitely	Possibly	Possibly Not	No	Possibly
Student 7	16	Definitely	Possibly	No	No	Possibly
Student 8	11	Definitely	Possibly	No	No	Possibly
Student 9	17	Definitely	Possibly	No	No	Possibly
Student 10	22	Possibly	Possibly Not	Possibly Not	No	Possibly Not

Table 6-8: Decisions made with guidance sheet

The decisions shown with a bold typeface are decisions that deviate from the gold-standard. The decision sheets for these students were examined to see why these deviations occurred. Student 3 decided to ignore the guideline for this decision

“because without SWOOGLE analysis an element of mistrust would influence my decision”. Student 5 misinterpreted the match to mapping ratio question (Q9). Student 6 had identified the decision should be a NO according to question Q1, and reinforced this with a comment “the number of matches is so poor I would not try to match” but still marked the decision box as “possibly not”. This incorrect decision was later confirmed with the student as being a transcription error. Student 10 appeared to have difficulty in interpretation of guidance questions Q2, Q5, Q9, Q10 as for most pairs of ontologies these were answered incorrectly. The student is not a native English speaker and in subsequent discussions this was seen as the decisive factor.

The average time taken to complete this part of the experiment was nineteen minutes. As shown in Table 6-8 above, the first student completed in eleven minutes and last student completed in twenty-nine minutes. Figure 6-3 shows the resulting decision distribution.

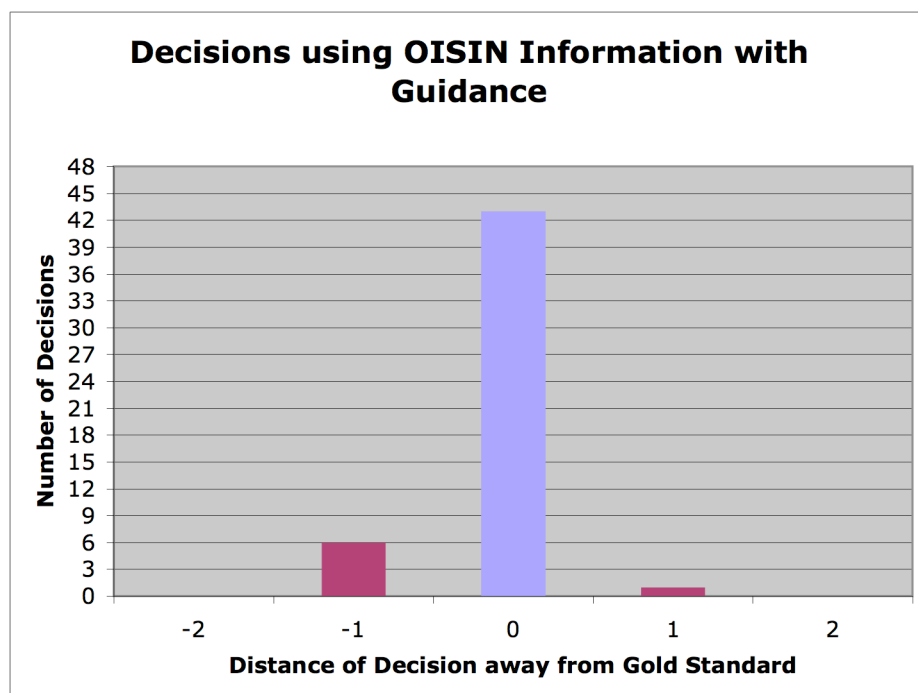


Figure 6-3: Chart of Results for guidance supported decisions

Figure 6-3 shows forty-three decisions aligned with the gold-standard, seven decisions one away from the gold-standard and none two decisions away. Again, the students commented on the confidence they had in their decisions and the difficulty of the task. These comments are presented in Table 6-9.

	After using the guidance, how difficult was it to come to an overall decision for a pair of ontologies?		After using the guidance, how confident are you in the decisions that you made	
Student 1	Fairly Easy	As the methodology just had to be followed	Confident	So long as the methodology is correct
Student 2	Fairly Easy	Very step by step - good	Not confident	Still no domain information
Student 3	Fairly Easy		Confident	
Student 4	Fairly Easy		Confident	
Student 5	Easy	Guidelines were explicit	Very Confident	Explicit guidelines made it difficult to be wrong
Student 6	Fairly Easy	The guidelines are very rigid	Confident	Since the guidance was so rigid it is difficult to disagree
Student 7	Easy		Very Confident	
Student 8	Fairly Easy	The guidance showed clearly what was important	Confident	
Student 9	Easy		Very Confident	
Student 10	Fairly Easy	When you know what to look at and number of yes and nos that cause a decision it becomes easier	Confident	

Table 6-9: Opinions on Confidence in decisions and Difficulty of decision making

Nine of the ten students responded that they were “very confident” or “confident” in their decisions and all found the task “easy” or “fairly easy”. In addition the task questionnaire (see Appendix D-4) sought feedback on the usefulness of the guidance and on what parts of the spreadsheet or guidance could be improved. Table 6-10 presents the responses related to the usefulness of the guidance.

	Were the guidance questions provided helpful in coming to a decision?	Which (if any) numeric or chart information was particularly helpful in reaching your recommendation?
Student 1	Yes as long as the methodology is correct	Similar number of terms and matching
Student 2	Yes sample/acceptable values are useful. Not sure that each should have equal weighting	Page 2 graph, easy to see similar layout. Page 3 graph, easy to see matching.
Student 3	Yes although I used similar metrics for the first part of the experiment	The graphs on page 3 and 4. Also the "term" info on page 4.
Student 4	Yes provides clarity on what attributes to examine	Candidate versus potential mappings gave a good indication of similarities
Student 5	Yes they were very specific, pointed out useful areas on the spreadsheet	Graphs on pages 1, 3, 4 and tables on page 1
Student 6	Yes more attention was paid to the statistics on page 1. The tree structure was less vital	The number of matches - I wouldn't consider mapping unless there were a high proportion of matches
Student 7	Yes it gave an idea of what percentages to expect	Shape of ontologies and candidate matches
Student 8	Yes they made what I was looking at more clear and made what was acceptable level of matching clear	Matching classes, syntax analysis and swoogle were most helpful as indicated amount of work needed to do mapping
Student 9	Yes to find key factors of decision like number of candidate matches for classes	All very helpful
Student 10	Yes I knew better what to look for. Some of them are logical	Percentage of unknown terms

Table 6-10: Opinions on guidance

As can be seen, the guidance was universally found to be useful, as encapsulated in the quote of one student: “the guidance questions made what I was looking at more clear, and made what was acceptable level of matching clear”. There was a diversity in the numeric or chart information that was considered particularly helpful, with nearly all graphs or numeric information getting a positive comment from someone. The one exception is the numeric information related to the dimensions/shape of ontologies on page 2 of the guidance. This is reinforced in the responses to the questions related to what was unclear about the guidance and what numeric information/charts were not particularly useful. These responses are presented in Table 6-11.

	Which guidance questions (if any) were unclear?	Which (if any) numeric or chart information was <u>not</u> particularly helpful in reaching your recommendation?
Student 1	Q2 (compilation and %OWL keywords), more mathematically confusing	
Student 2	Q9 (ratios of matches to mappings) and Q10 (percentage type matching) takes a bit of work to find correct table/graph to do calculation	Page 3 and Page 4 graphs. Both based on only positive matches
Student 3	All were perfectly clear	Shape of the ontology. This can vary dramatically based on domain subject and granularity
Student 4	Q3 (composite terms), as to whether it mattered if the composite types differed	None
Student 5	None	Tables on Page 2, Graphs were sufficient
Student 6	The number of levels in the tree was unimportant (Q8)	The composite terms and swoogle analysis did not generally impact my decisions
Student 7	None	Swoogle analysis seemed unimportant compared to other information
Student 8	Q4 (percentage unknown terms) seems less helpful than rest	Shape of ontologies
Student 9		Shape of ontologies
Student 10	Don't understand aim of Q9 (ratio of matches to mappings)	Q9 (ratio of matches to mappings)

Table 6-11: Opinions on unhelpful guidance or information

The table shows that six of the ten guidance questions were found to be unclear in some aspect by different students. Two students responded that the ratio of matches to potential mappings (Q9) was unclear. With respect to feedback on numeric information/chart information, there was common agreement that the information related to the shape of the ontologies was not considered particularly useful.

The students were also requested to make suggestions as to what improvements could be made to the information or guidance sheets. The improvements suggested are summarised in Table 6-12.

Do you have any suggestions for numeric information, charts or guidance that might be added to help in this task?

Content related	Domain similarity value e.g. Tyres and clutches 0.95; Boxing and computers 0.15
	Domain topic analysis
Layout related	Put % symbol in text of tables after figures
	Improve layout page 3
	State page 4 is based on data on page 3
	More descriptions

Table 6-12: Suggested improvements to guidance and information

The layout related suggestions are minor and the two content related suggestions are similar, in that they both suggest that some form of topic analysis would be helpful. This suggestion could be easily addressed through the application of Latent Semantic Indexing techniques from the Information Retrieval field of study.

6.2.4 Analysis of Experiment Results

Figure 6-4 provides an initial overall comparison of the decisions from the two parts of the experiment outlined in section 6.2.2 and 6.2.3.

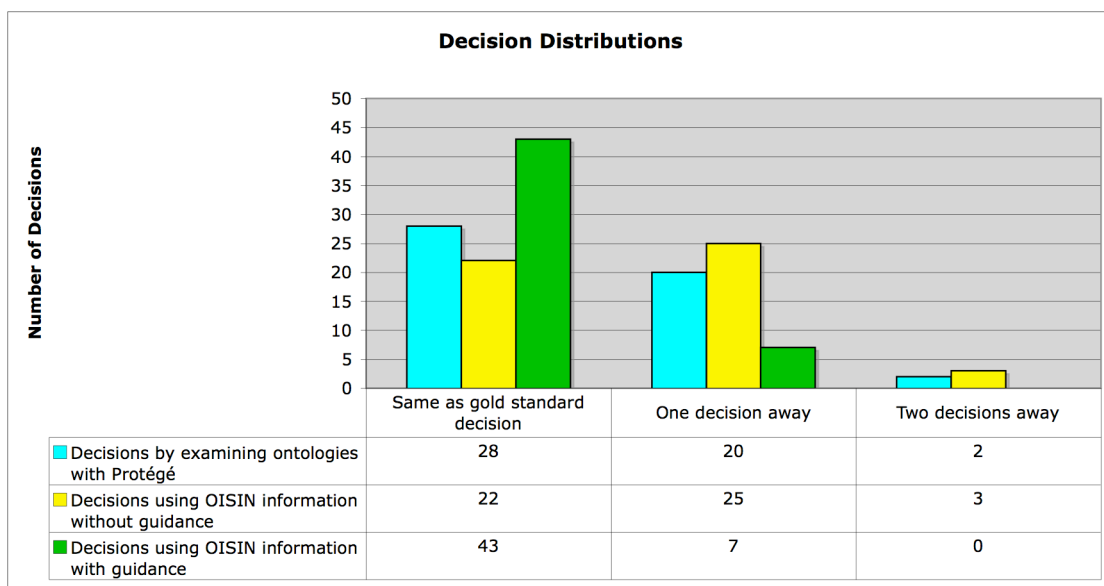


Figure 6-4: Overall comparison of decisions

A variety of statistical analyses were required to examine the data resulting from the experiment. MINITAB software (MINITAB 2005) was used to perform the analysis described in this section. The output from MINITAB is included as Appendix D-5. A statistician independently verified the following results. It was first noted that the decision distributions approximated normal distributions. Due to the small sample size,

the approximated normal distributions of the decisions and the fact that the standard deviations are estimated, a number of paired t-tests (Reilly 1997) were performed over the three sets of results in order to test the hypothesis that using the spreadsheet information would be as good as examining the ontologies directly. The main results of the paired t-tests were:

- There was no evidence of a difference between examining the ontologies directly and just examining the spreadsheet without guidance.
- There was significant evidence ($p < 0.05$) of a difference between examining the ontologies directly and examining the spreadsheet with guidance. The estimated magnitude of this difference is between 0.17 and 0.51 mean score, with ninety-five percent confidence. This means for example that when trying to make decisions like those in the experiment, people examining the ontologies directly will have a mean score of at least 0.17 worse than those people using the spreadsheet with guidance.
- There was significant evidence ($p < 0.05$) of a difference between looking at the spreadsheet without guidance and looking at the spreadsheet with guidance. The estimated magnitude of this difference is between 0.31 and 0.65 mean score, with ninety-five percent confidence. This means for example that when trying to make decisions like those in the experiment, people examining the ontologies using the spreadsheet without guidance will have a mean score of at least 0.31 worse than those people using the spreadsheet with guidance.

A General Linear Model (STATSOFT 2005) was then used to perform a fuller analysis, to consider the effect of the three factors - Person, Question, Group – on the decisions, where person represented the participant, question the ontology pair and group representing each group in the experiment. The results of these analyses were consistent with the earlier conclusions: no difference could be established between making the decision based on examining the ontologies directly and examining the spreadsheet without guidance, but differences between these information sources and the spreadsheet with guidance were re-asserted. The fuller analysis reaffirmed that there are differences between the questions (mapping decisions were not equal in "difficulty") and no differences were proven between persons (persons may be equal in "ability"). This General Linear Model was constructed on the assumptions that both the

people and the questions were only samples from larger populations of people and ontology pairs.

Finally a One-Way ANOVA analysis (Reilly 1997) was undertaken to analyse variance and attempt to identify its sources. With One-Way ANOVA two sources of variation can be suggested (i) difference in treatment and (ii) chance. This analysis did not yield additional qualitative insights, but the output (Figure 6-5) is useful because the mean scores for ontology-only decisions (SCORE 1), spreadsheet without guidance decisions (SCORE 2A) and spreadsheet with guidance (SCORE 2B) are shown.

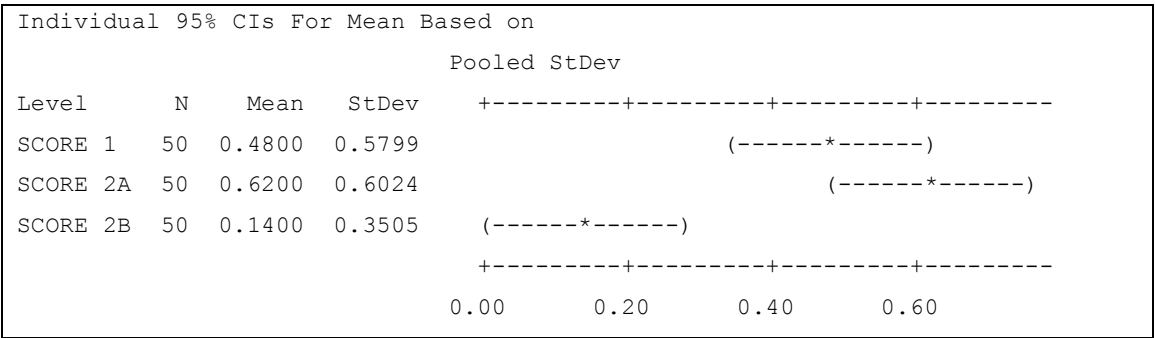


Figure 6-5: ANOVA analysis

Discussions and Findings

Cursory comparison of the decision distributions (cf. Figure 6-4) for decisions made using the ontology only information and OISIN information “without guidance” may suggest that the latter yields in general to less accurate decisions. However undertaking paired t-tests which examines the individual decisions of the students, paired based on ability indicates, with ninety-five percent confidence, that there is no evidence of a difference. This finding would indicate that examining information about an ontology is at least as good as looking at the ontology itself. This finding is particularly significant in scenarios where the ontologies that need to be decided upon will be large and where examining the ontologies directly would be impractical or too time consuming. In these situations, there can be confidence that looking at the generated spreadsheet information is as good as examining the ontology directly.

It was also found that examining the OISIN information with guidance as opposed to examining the spreadsheet without guidance or examining the ontologies directly, will

yield decisions significantly closer to the gold-standard. This finding demonstrates that when the decision criteria are more directly supported by the design tools (in this case incorporating the guidance into the spreadsheet) that there are significantly less inconsistencies in recommendations. This is important as it implies that policies based on quantifiable criteria can be encoded in either semi-automatic or automatic tools. However, from the experiment it is not possible to judge the affect that the individual criteria and thresholds that were incorporated into the guidance, had on the improvement. The author's current view is that the affect will be directly in line with the quality of the guidance. Good guidance yields better results.

Given that a similar number of inconsistencies were found when examining the ontologies directly and when examining the summary information produced by OISIN without guidance, it can be argued that inconsistencies in recommendations are likely to be as a result of not applying the decision criteria correctly. It can thus be argued that similar reductions in inconsistencies could be achieved if a specific ontology browsing methodology or tool was designed to directly support the decision criteria and the decision to map activity. This was not explored in this research but is worthy of further study.

The implemented guidance was found to be very useful and feedback indicates minor layout corrections, the addition of information related to topic analysis and removal of information about the "Shape of Ontologies". From the experience gained in the experiment, it is clear that when humans are involved in interpretation of the guidance, the guidance needs to be simple (to avoid misinterpretation) and clear (to empower non-native English speakers).

With respect to how easy the students found each task and the confidence that they had in their decisions, it was found that there was not much difference between those students who could examine directly the ontologies using Protégé and those who examined the OISIN information with guidance. However those students who used the OISIN information without guidance found the task difficult and consequently were not confident in their decisions. Thus even though a similar number of inconsistencies arose, there was a significant difference in the confidence that students had in browsing the ontologies as opposed to examining summary information about the ontologies. A

number of points can be argued based on this. One is that examining the ontologies directly provides a false sense of confidence in undertaking the task whereas only examining the summary information results in a user experience that is less confident than it should be. In future work therefore it would be interesting to measure the confidence of OISIN framework users who had both access to the summary information backed up by the ability to examine the ontologies directly.

Comparing the amount of time that was necessary to reach a decision shows that the average time taken was twenty-three minutes for examining the ontologies directly and eighteen minutes for examination of the OISIN information (with or without guidance). On the surface this is not much of a difference in time, but it would be expected that looking at the ontologies directly would be even longer as the size of the ontologies became bigger, whereas the time taken to examine the spreadsheet information would remain fairly constant.

In summary, it was found that students who undertook the task using the OISIN information with guidance found the task easy, were confident, and took decisions close to the gold-standard in a timely way. Students who undertook the task by just examining the ontologies directly found the task fairly easy, were confident but took longer to make decisions that were a little further away from the gold-standard. Finally, students who undertook the task by examining the OISIN information without guidance found the task difficult, made timely decisions equally as good as those who had sight of the ontologies but were not confident at all about the decisions that were made.

6.3 Committed Mapping Activities Experiment

As discussed in Section 4.1, identifying mappings and finding structural mismatches are difficult tasks even when one is attempting to map two ontologies fully, let alone when one is only attempting to map them partially. Noy and Musen describe what is necessary to set up an experiment to evaluate whether a tool is “good enough” for the user’s task (Noy and Musen 2002). This involves finding source ontologies covering the same domain, then the manual creation of a gold-standard mapping to serve as a benchmark, and finally the comparison of the user created mappings versus the gold-

standard mappings. This was the approach followed in this experiment. Section 6.3.1 presents the objective of the experiment and provides an overview. The results of each part of the experiment are then presented in Sections 6.3.2 to 6.3.6, with the overall analysis of results from the experiment and discussion of findings described in section 6.3.7.

6.3.1 Overview

The first hypothesis of this experiment to be tested was that the identification of committed mappings and identification of structural mismatch tasks require tool support. In order to test this hypothesis, participants in the experiment were requested to undertake a mapping between a pair of ontologies with no dedicated mapping tool support. The Protégé tool was made available to the participants to aid in the examination of the ontologies. Protégé was chosen as being representative of a typical ontology browsing tool. In addition, it was suggested that the PROMPT plug-in for ontology integration could be used if the participants found it helpful in undertaking the task. The mapping results produced by the participants, as well as questionnaire feedback, were analysed in order to evaluate whether these activities do indeed require support as postulated in the hypothesis. The results from this part of the experiment are described in detail in section 6.3.2.

The second hypothesis was that the OISIN tools are highly usable, assist the committed mapping related activities and lead to the successful generation of mappings. In order to test this hypothesis the participants in the experiment were required to undertake another mapping task but this time on the ontology pair that they had not mapped previously and using the semi-automatic versions of the Anchor Capture and Mapping Analysis tools, that is OisinAC-semi and OisinMA. The mapping results produced by the participants, as well as questionnaire feedback, were analysed in order to evaluate the usability of the OISIN tools, the supportiveness of the committed mapping activities and the success in generating mappings as put forward in the hypothesis. The results from this part of the experiment are described in detail in section 6.3.3.

For the experiment two pairs of overlapping ontologies representing a University/Research domain were used. It was decided to pick pairs of ontologies in the University/Research domain as this would be familiar to the students participating

in the experiment. The first pair has already been introduced in our scenario and was used in our case study, that is the University of Manchester ontology and the University of Aberdeen ontology. These ontologies will be referred to hereafter as Ontology A and Ontology B respectively. The second pair (hereafter referred to Ontology C and Ontology D) had been produced for a course project by first year students of the M.Sc. in Ubiquitous Computing course in Trinity College Dublin in 2003 (TCD1 2004, TCD2 2004). The “C” ontology involved sixty three classes and one hundred and forty-three properties, and the “D” ontology involved sixty-three classes and ninety two properties. The mapping between ontology A and ontology B will be referred to as the “AB mapping” and the mapping between ontology C and ontology D will be referred to as the “CD mapping”.

A gold-standard mapping for the AB pair and CD pair was created by this author, before the experiment began and are shown in Table 6-13. The gold-standard mappings were independently verified by another researcher with similar ontology mapping experience. In summary there are:

- sixteen class mappings with one structural mismatch mapping in the AB mapping;
- twenty six class mappings with zero structural mismatches in the CD mapping.

Only one structural mismatch was included in the experiment, and that was in the AB mapping. This mismatch occurs naturally (that is it was not artificially introduced) in the ontologies published by the universities of Manchester and Aberdeen, and is a good example of how this can occur. As can be seen in Figure 3-1 (as these ontologies are used in the scenario), the University ontology’s `Researcher` is modelled as a subclass of `AcademicStaff` and that is a subclass of `Employee`, whereas in the visiting lecturer’s ontology, a `Researcher` is a subclass directly of `Working-Person`, but also has `Academic` as a subclass of `Employee`. If a user creates a mapping between `Employee` and `Employee`, `Academic` and `AcademicStaff` then there is not a problem. However if the user adds a mapping between `Researcher` classes then there is a structural mismatch.

AB mapping

Ontology A	Ontology B
Book	Book
Paper	ConferencePaper
Journal	Journal
Lecturer-In-Academia	Lecturer
Organization	Organization
Person	Person
PhD-Student	PhDStudent
Project	Project
Publication	Publication
Researcher-In-Academia	Researcher
Research-Area	ResearchTopic
Secretary	Secretary
Student	Student
Tech-Report	TechnicalReport
University	University
Paper	WorkshopPaper

CD mapping

Ontology C	Ontology D
Administration_Person	AdministrativeStaff
Building	Building
Computer	Computer
Conference_Room	ConferenceRoom
Course	Course
Desk	Desk
Equipment	Facility
Postgrad_Student_Person	GraduateStudent
Corridor	Hallway
Laboratory	Lab
Laptop	Laptop
Lecturer_Staff_Person	Lecturer
Lecture_Theatre	LectureTheatre
Library	Library
Office	Office
Person	Person
Photocopier	PhotoCopier
Place	Place
Printer	Printer
Projector	Projector
Toilets	Restroom
Chair	Seat
Staff_Person	Staff
Student_Person	Student
Thesis	Thesis
Undergrad_Student_Person	UndergraduateStudent

Table 6-13: Gold-standard mappings for AB and CD ontology pairs

The twelve first year students of the 2004/2005 M.Sc. in Ubiquitous Computing class participated in this experiment. Students on this course come with Computer Science undergraduate degrees from a wide variety of universities. The students were formed into four groups of three. The students in the M.Sc. class have technical undergraduate degrees from a wide range of universities. Thus the students were asked to work together on two pre-experiment activities designed to provide a common basis for the groups with respect to working with ontologies and using the Protégé tool. The first activity was to build an ontology using Protégé for a Social domain. The second activity involved specifying XQuery based applications based on one of the A, B, C, D ontologies. The results of this second task were used as input into the trial described in chapter seven. The groups then participated in the experiment, first undertaking a mapping with support of the Protégé/PROMPT tool (detailed in section 6.3.2), and then undertaking a mapping using the OisinAC-semi and OisinMA tools (detailed in section 6.3.3).

6.3.2 Mappings with support from Protégé/PROMPT tool

For the first task, each group was allocated one of the pairs of ontologies under test and required to produce an XML document (according to the DTD shown in Figure 6-6) showing what they believed were committed mappings expressed as anchors as part of the longest possible anchor paths. Recall the anchor path as defined in Section 5.1.10, represents the classes that appear in the is-a hierarchy between two anchors. The `start` and `end` elements (lines 7 and 8) indicate the start and end of the anchor path. In addition any mappings found in the corresponding is-a path between the two anchors were to be represented as an `anchor` element (line 6). Asking for the longest possible anchor path was designed to elicit the maximum number of mappings without requiring all anchor paths to be enumerated. For any anchor mappings, the students were also requested to document the mappings of their respective properties. If a mapping would cause a structural mismatch it was to be indicated as a `mismatch` element (line 3). If an anchor was standalone (that is not part of a path), it was expressed as a path with the same start and end elements.

```
1. <!ELEMENT name (#PCDATA)>
2. <!ELEMENT target (#PCDATA)>
3. <!ELEMENT mismatch (name,target)>
4. <!ELEMENT property (name,target)>
5. <!ELEMENT properties (property)>
6. <!ELEMENT anchor (name,target,properties?)>
7. <!ELEMENT start (anchor)>
8. <!ELEMENT end (anchor)>
9. <!ELEMENT path (start,(anchor|mismatch)*,end)>
10.<!ELEMENT mappings (path)*>
```

Figure 6-6: Experiment mapping DTD

The students were requested to use Protégé as a means of exploring the ontologies, and were encouraged to use the PROMPT tab of the Protégé tool to support the mapping phase. A one hour lab session was held at the beginning of the task in order to familiarise the students with the mapping phase and the PROMPT tab of Protégé. It was explained to the students that the threshold for what would be a match would be quite low, and that the matching should really be concentrated at the class level. The groups were encouraged to follow the mapping workflow described in section 5.2.1, Figure 5-30.

In order to evaluate the accuracy of the resultant mappings, the mappings identified by the groups were compared against the previously generated gold-standard mappings

and analysed in order to assess how many mapping errors had been made during each task. The errors discovered fall into three categories:

- Mappings that cause structural mismatches;
- Incorrect mappings;
- Gold mappings that were missed.

The AB mappings created by the two groups are presented in Table 6-14 and the CD mappings generated by the other two groups are presented in Table 6-15. The tables show the mappings of each group and categories of errors discovered.

Gold standard AB mapping			
Ontology A	Ontology B	Group 1	Group 2
Book	Book	included	included
Paper	ConferencePaper	included	missed
Journal	Journal	included	included
Lecturer-In-Academia	Lecturer	included	included
Organization	Organization	included	included
Person	Person	included	included
PhD-Student	PhDStudent	included	included
Project	Project	included	missed
Publication	Publication	included	included
Researcher-In-Academia	Researcher	included	included
Research-Area	ResearchTopic	included	missed
Secretary	Secretary	included	included
Student	Student	missed	included
Tech-Report	TechnicalReport	included	included
University	University	included	included
Paper	WorkshopPaper	missed	missed

Group 1 Additional	Category
Generic-Area-Of-Interest == Object	incorrect
Organization-Unit == Object	incorrect
Activity == Object	incorrect
University-Faculty == Department	incorrect
Academic == AcademicStaff	causes structural mismatch
Employee == Employee	causes structural mismatch

Group 2 Additional	Category
System-Administrator == TechnicalStaff	incorrect
Academic == AcademicStaff	causes structural mismatch
Employee == Employee	causes structural mismatch

Table 6-14: AB ontology mappings created manually by experiment groups

Gold standard CD mapping			
Ontology C	Ontology D	Group 3	Group 4
Administration_Person	AdministrativeStaff	missed	missed
Building	Building	included	included
Computer	Computer	included	included
Conference_Room	ConferenceRoom	included	included
Course	Course	missed	missed
Desk	Desk	included	included
Equipment	Facility	included	included
Postgrad_Student_Person	GraduateStudent	included	included
Corridor	Hallway	included	included
Laboratory	Lab	included	included
Laptop	Laptop	included	included
Lecturer_Staff_Person	Lecturer	included	included
Lecture_Theatre	LectureTheatre	included	included
Library	Library	missed	included
Office	Office	included	included
Person	Person	included	included
Photocopier	PhotoCopier	included	included
Place	Place	included	included
Printer	Printer	included	included
Projector	Projector	included	included
Toilets	Restroom	included	included
Chair	Seat	included	included
Staff_Person	Staff	included	included
Student_Person	Student	included	included
Thesis	Thesis	missed	included
Undergrad_Student_Person	UndergraduateStudent	included	included

Group 3 Additional

Space == AtomicPlace
 Technician == SystemsStaff
 External_Lecturer_Staff_Person == VisitingStaff
 Common_Room == Restroom

Category

incorrect
 incorrect
 incorrect
 incorrect

Group 4 Additional

Space == AtomicPlaceInBuilding
 Tutorial_Room == LectureRoom
 Common_Room == StaffRoom
 Administration_Person == SystemsStaff
 External_Lecturer_Staff_Person == VisitingStaff

Category

incorrect
 incorrect
 incorrect
 incorrect
 incorrect

Table 6-15: CD ontology mappings created manually by experiment groups

6.3.3 Difficulties Questionnaire

At the end of the first mapping task each group was requested to complete a short questionnaire (see Appendix E-1). The first question in the questionnaire sought to identify information about the level of difficulty found undertaking different parts of the mapping task. The answers are presented in Table 6-16.

Part of Mapping Task	Group 1	Group 2	Group 3	Group 4
Identifying the mappings according to the suggested mapping workflow	Straightforward	Straightforward	Very Difficult	Difficult
Identifying the concepts that might be a candidate anchor pair	Straightforward	Difficult	Very Difficult	Difficult
Identifying the longest possible valid anchor path	Straightforward	Difficult	Straightforward	Difficult
Identifying whether anchor pairs lie on a similar is-a path	Straightforward	Straightforward	Difficult	Difficult
Identifying mismatch anchor pairs	Difficult	Straightforward	Very Difficult	Difficult
Identifying properties of a candidate anchor pair considered equivalent pairs	Straightforward	Straightforward	Straightforward	Very Easy
Documenting the anchors and anchor pairs	Very Difficult	Very Easy	Very Easy	Very Easy

Table 6-16: Summary of mapping task difficulties experienced by experiment groups

Two of the four groups found the identification of mappings according to the suggested workflow straightforward. Three out of the four groups found that it was difficult or very difficult to identify candidate matches/anchor pairs. Two of the groups reported that it was difficult or very difficult to identify matches/anchor pairs that lie on a similar is-a hierarchy path. Three of the four groups found the identification of matches/anchor pairs that caused mismatches difficult or very difficult. Ironically the group that stated that this part of the task was straightforward missed the structural mismatch in the AB mapping allocated to them. The other group which missed the structural mismatch in the AB mapping recognised this part of the task was difficult. All groups agreed that it was very easy or straightforward to identify property matches. One group found the task of documenting the mappings according to the DTD very difficult, but in a comment elaborated by explaining that this was due to their lack of experience with XML in general.

The second question in the questionnaire sought to gather feedback on the usage of the PROMPT tab of Protégé during the task. All groups reported how they had attempted to use the tool to guide their mapping task but in each case they decided not to use it extensively. Various reasons were given (shown in Table 6-17) but they can be summarised as follows. Students did not find PROMPT to be of support in undertaking the task and were unhappy with the lack of transparency with respect to the mapping decisions suggested by PROMPT. These issues are highlighted in the following particular quotes: “It seems to be based only on word matching and not semantic or

thesaurus based, so we preferred to proceed manually” and “We felt PROMPT wouldn’t be able to give us adequate reasons why anchor pairs matched”.

Group 1	We used Protégé to display the Ontologies, their classes and their properties side by side but we did not use PROMPT at all. It was easier to visually see the trees on each display rather than read the suggested mapping. We did use PROMPT as some form of validation of our selections, that is to make sure we didn't leave out anything or get things completely wrong.
Group 2	We didn't use PROMPT for our project as we felt that the workflow suggested in the notes gave us a simple manageable way to discover the anchor pairs between the ontologies. We felt that PROMPT would provide an extra layer of complexity to the discovery process and also that the workflow gave us more control during this process. Another issue why we didn't use PROMPT was that by doing the mapping manually we would have sufficient reasons why we felt certain concepts were equivalent. We felt PROMPT wouldn't be able to give us adequate reasons why anchor pairs matched and also that by doing it manually we would gain a better grasp of the whole subject of semantic mapping
Group 3	PROMPT mapping was not accurate. It seems to be based only on word matching and not semantic or thesaurus based, so we preferred to proceed manually.
Group 4	After initially using PROMPT in this assignment we quickly decided against it as we couldn't find any obvious benefit in it. We had to go through the ontologies manually anyway and the PROMPT results only mapped the obvious pairs that anyone would have seen straight off. As we didn't trust it to arrive at similar results to our own or add anything new, we didn't proceed any further with it. The fact that it wasn't particularly intuitive to use didn't help its case either.

Table 6-17: Group comments about usefulness of Protégé/Anchor-PROMPT

6.3.4 Mappings using the OISIN tools

In the second mapping task, each group was allocated the pair of ontologies that they had not mapped during the first task. The motivation for this approach was to ensure that the group would come to the mapping task anew and with no previous bias in mind¹. The pairs of ontologies provided to the groups had already been pre-processed using the OISIN characterisation phase tools. Again a one hour lab session was held at the beginning of this task in order to familiarise the students with the OISIN mapping tools. It was explained to the students that the threshold for what would be a match would be quite low, and that the matching should really be concentrated at the class level. The AB mappings created by the two groups are presented in Table 6-18 and the CD mappings generated by the other two groups are presented in Table 6-19. The tables show the mappings of each group and categorises any errors discovered.

¹ Note: It had been explicitly stated to the groups the importance of not discussing their mapping work with other groups during and between the two tasks. This request was respected.

Gold standard AB mapping			
Ontology A	Ontology B	Group 3	Group 4
Book	Book	included	included
Paper	ConferencePaper	included	missed
Journal	Journal	included	included
Lecturer-In-Academia	Lecturer	included	included
Organization	Organization	included	included
Person	Person	included	included
PhD-Student	PhDStudent	included	included
Project	Project	included	included
Publication	Publication	included	included
Researcher-In-Academia	Researcher	not included	not included
Research-Area	ResearchTopic	missed	missed
Secretary	Secretary	included	included
Student	Student	included	included
Tech-Report	TechnicalReport	missed	included
University	University	included	included
Paper	WorkshopPaper	included	missed

Group 3 Additional

AcademicStaff== Educational-Employee

Employee== Employee

Category

okay as avoided mismatch
by not including Researcher
anchor

okay as avoided mismatch
by not including Researcher
anchor

Group 4 Additional

AcademicStaff== Educational-Employee

Category

okay as avoided mismatch
by not including Researcher
anchor

Table 6-18: AB ontology mappings created by experiment groups using OISIN

Gold standard CD mapping			
Ontology C	Ontology D	Group 1	Group 2
Administration_Person	AdministrativeStaff	included	included
Building	Building	included	included
Computer	Computer	included	included
Conference_Room	ConferenceRoom	included	included
Course	Course	included	missed
Desk	Desk	included	included
Equipment	Facility	missed	included
Postgrad_Student_Person	GraduateStudent	included	included
Corridor	Hallway	included	included
Laboratory	Lab	included	included
Laptop	Laptop	included	included
Lecturer_Staff_Person	Lecturer	included	included
Lecture_Theatre	LectureTheatre	included	included
Library	Library	included	included
Office	Office	included	included
Person	Person	included	included
Photocopier	PhotoCopier	included	included
Place	Place	included	included
Printer	Printer	included	included
Projector	Projector	included	included
Toilets	Restroom	included	included
Chair	Seat	missed	included
Staff_Person	Staff	included	included
Student_Person	Student	included	included
Thesis	Thesis	included	missed
Undergrad_Student_Person	UndergraduateStudent	included	included

Group 1 Additional

Research== Research_Paper

Category

incorrect

Group 2 Additional

Lecture_Theatre == LectureRoom

Technician == SystemsStaff

Space == AtomicPlaceInBuilding

Category

incorrect

incorrect

incorrect

Table 6-19: CD ontology mappings created by experiment groups using OISIN

6.3.5 Mappings Analysis

As discussed in section 4.1, our initial investigation of mapping showed that structural mismatches are particularly difficult to identify and the consequences for the OISIN approach of not discovering them can lead to incorrect usage of mappings at runtime. During the introduction of the task to the groups, the problem of structural mismatches was highlighted and examples explored. During the first task, just using Protégé/PROMPT, both groups that were allocated the AB mapping included mappings that would result in this structural mismatch. During the second task, using the OISIN mapping tools, both groups avoided any structural mismatches. In the second task, both groups avoided mapping the Researcher classes and were consequently able to provide additional mappings.

Incorrect mapping errors occur when there is a misinterpretation of the classes being mapped. For example (as seen in Figure 6.15) in undertaking the CD mapping, group four included a mapping between `visitingStaff` and `External_Lecturer_Staff`. The former refers to a visiting professor and the latter refers to some form of externally contracted lecturer. As can be seen in Figure 6-7, the number of incorrect mappings generated using the OISIN tools was significantly reduced.

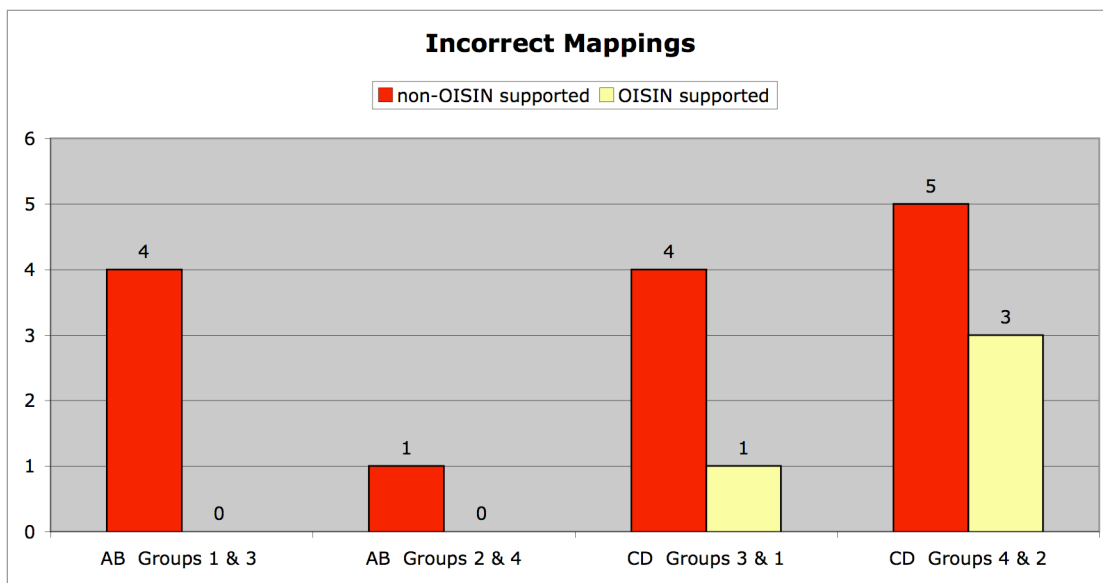


Figure 6-7: Analysis of Incorrect Mappings created

In addition, the mappings were examined to see how many gold mappings were missed in each task. As can be seen from Figure 6-8, the results here were mixed with very little difference as a result of being non-OISIN supported or OISIN supported. Upon

closer examination the mappings missed can be broken down into two types: arguable mappings and surprising omissions, with surprising omissions representing the majority of the missing gold mappings. Arguable mappings relate to gold mappings that were considered “gold” but may not be depending on one’s perspective. Seven of the eighteen missed mappings fall into this category. Five arise from where *WorkshopPaper* and *ConferencePaper* in ontology B was mapped to *Paper* in ontology A in the gold mapping. Only one group included these two mappings, one group included only one of the mappings and two groups failed to map these terms at all. Two missed mappings arise from the gold-standard mapping *Research-Area* to *ResearchTopic*, which is arguable and the properties of the classes involved do not help to make a determination one way or the other.

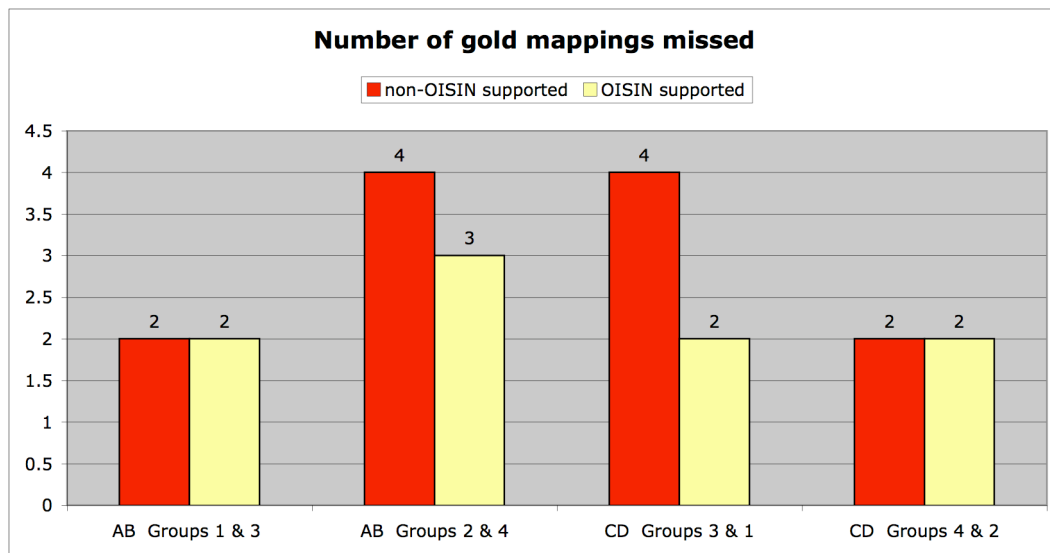


Figure 6-8: Analysis of Gold Mappings missed

Surprising omissions are mappings that clearly should be mapped but were not. Eleven of the eighteen missed mappings fall into this category, i.e. *Thesis* to *Thesis* (twice missed); *Student* to *Student*; *Project* to *Project*; *Course* to *Course* (twice missed); *Library* to *Library*; *Tech-Report* to *TechnicalReport* (twice missed); *Equipment* to *Facility*; *Chair* to *Seat*.

Finally, the performance of each individual group was analysed in terms of the total number of missed gold mapping or incorrect mapping errors. Figure 6-9 shows a drop in the number of errors for three of the four groups of at least fifty percent. However it is difficult to assess whether this decline can be attributed to the usage of the OISIN

mapping tools or due to the mapping experience built up during the first task. From the data it is difficult to determine why there was no improvement in group two's performance.

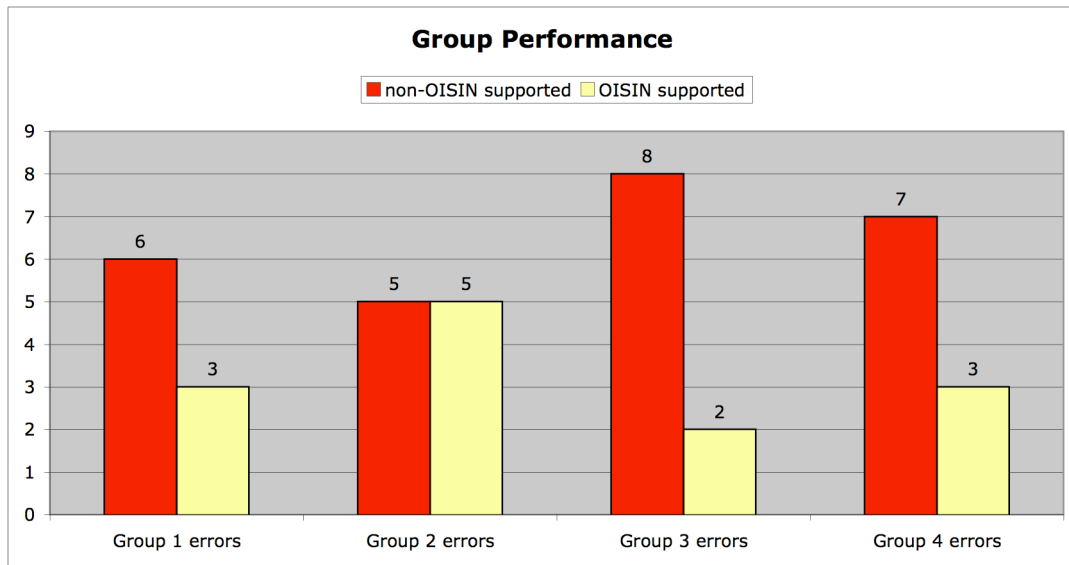


Figure 6-9: Analysis of performance of groups in undertaking ontology mapping

6.3.6 OISIN tools usability analysis

The questionnaire issued at the end of the second task (see Appendix E-2), sought to gather feedback on the usability of the OISIN tools and whether the difficulties encountered during parts of the first task were reduced during the second task. Ten out of the twelve participants provided a response to this questionnaire. Three of these students (student three, four and five) provided a common and agreed response. Respondents were asked to rate a number of statements. The student responses and student comments for the following statements are presented in tables 6-20, 6-21, 6-22:

- The OISIN tools allowed ontology mappings to be identified and documented easily;
- Some of the difficulties experienced in undertaking the mapping task documented during the last assignment were eased due to the use of the OISIN tools;
- It was easier to identify mappings with OISIN tools than with the Protégé/PROMPT tool.

The OISIN tools allowed ontology mappings to be identified and documented easily

Student	Rating	Comment
Student 1	Agree	I consider OISIN a useful tool because: the GUI helps to visually map classes in the two ontologies by clicking on the nodes; automatically generates the documentations; the suggested mappings are generally accurate; it's possible to automatically check for structure mismatches
Student 2	Strongly Agree	This was clearly the case especially in documenting of mappings though it was also a much more efficient process to identify mappings because the software's setup meant it cut down on manual work. Where it came into its own was that OISIN allowed you to automatically create XML documents creating all the relevant paths, anchors and supplementary information. Furthermore it created a b-a as well as a-b mapping concurrently which apart from saving huge amounts of time eliminated the potential for human error in the documenting of mappings.
Student 3,4,5	Strongly Agree	The visual presentation of the two ontologies side by side allows structure to be assessed quickly and accurately.
Student 6	Agree	You do not have to analyse the whole ontology manually, the program points out anchor candidates - that really speeds up main process
Student 7	Agree	Performing the ontology mapping this time around was much easier with OISIN. At first this mainly was because it presents the ontologies very well on screen, with colour coded partial and full mappings highlighted.
Student 8	Agree	The tool seems handy. It allows to identify the mappings quite quickly. The potential matches are very easy to compare. What is more, the tool generates two mappings simultaneously.
Student 9	Strongly Agree	The tool allowed some of the more obvious mappings to be quickly identified and documented. Certainly the automatic generation of the XML mapping document is a benefit, removing a tedious and time consuming task from the concern of the user.
Student 10	Agree	Yes I agree with this statement. The whole process of identifying and documenting ontology mappings was a lot easier and quicker. This time around we got all the mappings in less than an hour whereas in the last assignment it took a lot longer. The program main advantage was it gave us help and vital feedback on our mappings which allowed us to modify them as we saw fit. In the last assignment when we were in the process of identifying mappings we sometimes felt we didn't know if we were doing it right or not, whereas with OISIN you always knew where you stood, i.e. whether the mapping contained structural mismatches. The documenting part of the assignment was also a whole lot easier as the XML mapping documents were generated for you automatically.

Table 6-20: Opinions on usefulness of OISIN tools for ontology mapping

Some of the difficulties experienced in undertaking the mapping task documented during the last assignment were eased due to the use of the OISIN tools

Student	Rating	Comment
Student 1	Agree	I agree because the tool helped us to generate the "inverse" mapping (from b to a) and to discover some structural mismatches that we quickly fixed by hand. Also it was very easy to "see" class correspondences using the two panes interface
Student 2	Strongly Agree	The whole process of mapping was considerably eased by using the OISIN tools. It was far clearer what were potential matches and cut down much manual checking of the ontologies. The matching properties was also much more straightforward and a nice feature was that properties inherited by superclasses were automatically matched in any matching subclasses. Efficiency was increased greatly and most importantly you were far less likely to miss out on any potential mappings using the OISIN tools
Student 3,4,5	Strongly Agree	Especially the typing! And it eliminated the effort required to ensure the output XML file was valid according to the supplied DTD
Student 6	Agree	Program automatically generates XML file what makes the work easier. Program creates two way mappings. Mapping analysis tool is able to find mismatches automatically.
Student 7	Agree	Potential mappings can be viewed at the touch of a button, one at a time. This is good and saves a huge amount of time compared to the pen and paper approach. It makes it much easier to work down one of the ontologies top to bottom, identifying and confirming mappings on the way. OISIN allowed and encouraged a far more structured approach to performing the mapping and automated all the monotonous XML etc. OISIN removed the work element of performing the mapping and allowed us to concentrate on the decisions.
Student 8	Agree	The last time we spent a lot of time comparing the ontologies on paper. We had to print the whole ontology structure and then compare it. With OISIN it was much easier. The process was partly automated, so it basically was checking whether classes OISIN found similar were actually a match. In PROTEGE we had to open two applications to compare the ontologies. IN OISIN we had the two opened all the time.
Student 9	Agree	It certainly speeded up the process and the ability of OISIN to determine some property mappings aided that process. The lack of clutter in the screen and the ease with which classes could be compared certainly helped ease the process of deciding on mappings.
Student 10	Agree	In the previous assignment when we were trying to map the two ontologies we had to open up two versions of PROTÉGÉ., we then had to take a screenshot of each tree hierarchy and paste into Word. Printing them enabled us to try and match up the classes according to structures. While this allowed us to find structural mismatches it did not let us see if classes were actually equivalent as generally we had to compare properties to do this. This was very awkward as we had to go to and fro between each of the ontologies and PROTEGE to compare the properties. Meanwhile on the other hand OISIN let us view both ontologies at the same time and also let us view and match their properties where appropriate. Overall it was more time efficient, as allowed automatic XML generation.

Table 6-21: Opinions on whether OISIN alleviated difficulties

It was easier to identify mappings with OISIN tools than with the Protégé/PROMPT tool

Student	Rating	Comment
Student 1	Strongly Agree	In the last assignment we decided not to use the Prompt tool because the suggested class mappings were too few. I think this is due to the linguistic rules that are used to related class names. Also the GUI of OISIN is a lot more user friendly and easy to learn.
Student 2	Strongly Agree	The simple fact that you could open up the two ontologies at the same time and view them side by side, made it a lot easier to identify potential mappings. This coupled with the ability to use the OISIN tools to confirm whether its suggested equivalences were accurate meant the whole process was a lot more intuitive and straightforward than the cumbersome approach necessitated by using the Protege/PROMPT tool. You now no longer had to do screen grabs, join them together, print them out and use pens to make mappings between the two.
Student 3,4,5	Strongly Agree	<i>no comment provided</i>
Student 6	Strongly Agree	The layout of the program makes work easier - left and right panes display needed information, highlight candidate anchors. Just one click enables us to create match between anchors and their properties what takes a few minutes by doing manually.
Student 7	Agree	This is the main reason we didn't use PROMPT that much in the last assignment. PROMPT didn't seem to allow the more intuitive approach to performing the mapping that OISIN did. Potential mappings could be considered one at a time, this made the process a lot more transparent.
Student 8	Strongly Agree	We didn't use the PROMPT tool because the output seemed irrelevant to the task. With OISIN the task seemed quite easy to accomplish.
Student 9	Agree	Yes, certainly the identification by OISIN of possibly equivalent classes sped up the process allowing the user to concentrate on determining equivalence rather than spending time searching through the entire ontology for possible matches
Student 10	Strongly Agree	While we didn't use the Protégé/PROMPT tool in the last assignment I still strongly agree with the statement. The reason we didn't use PROMPT was because of the lack of feedback generated by it. We felt that when you compared or merged two ontologies and given suggestions of what to merge we didn't really know which we were right and which were wrong. We couldn't really compare properties easily unless we left PROMPT and compared the properties using PROTEGE directly. It didn't feel like you had control as the interface seemed a bit confusing as to what was going on. On the other hand, OISIN you could compare ontologies easily with two separate windows and when comparing particular classes you could view the properties of both classes and easily match them or disregard them accordingly. You were give much better visual feedback in OISIN than in PROMPT which gave you the user better control and understanding of what you were doing.

Table 6-22: Opinions on OISIN versus Protégé/Anchor-PROMPT

A five point Likert scale was used (with 1 = strongly disagree, 2 = disagree, 3 = neutral, 4 = agree, 5 = strongly agree) in order to analyse the responses to the three questions. Figure 6-10 shows the average ratings for each of the statements. In summary it can be seen that: it was strongly agreed that it was easier to identify mappings with the OISIN tools than with Protégé/PROMPT; there was considerable agreement that the OISIN tools allowed ontology mappings to be identified and documented easily; and it was agreed that some of the difficulties in mapping experienced during the first mapping task were eased due to the OISIN tools.

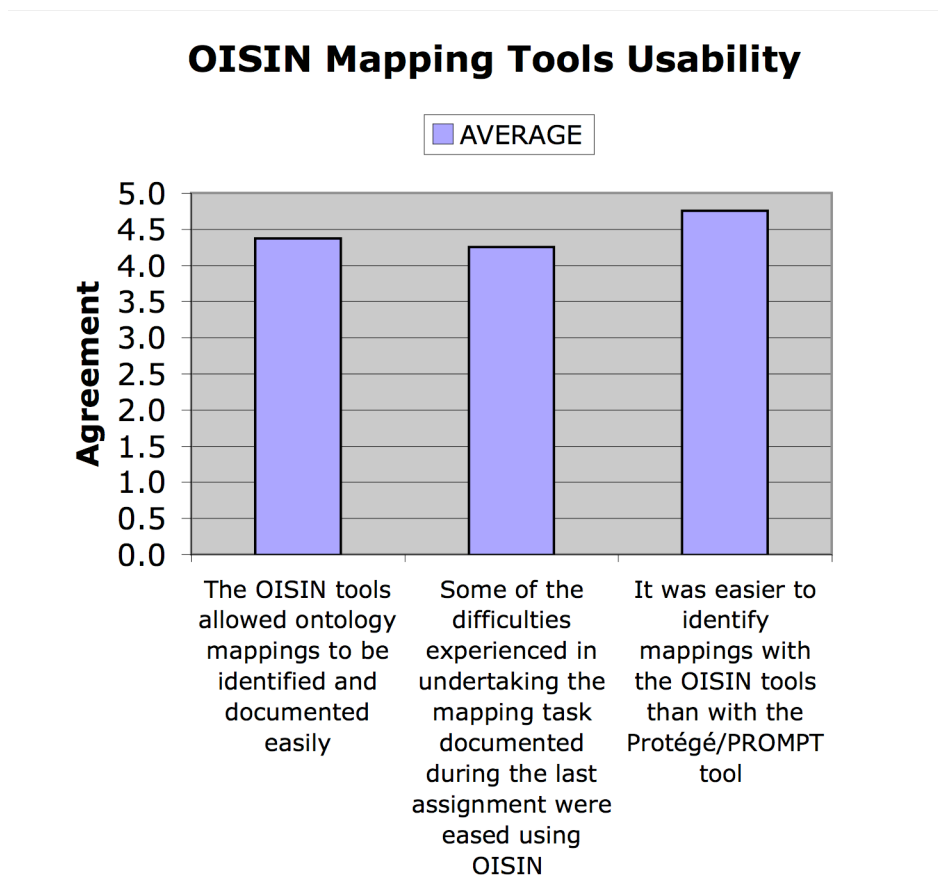


Figure 6-10: Opinions on OISIN Mapping Tools usability

The comments supplied by the students, confirm these positive indications. For example comments in support of the claim that OISIN allows ontology mappings to be identified and documented easily, include: “Potential mappings could be considered one at a time, this made the process a lot more transparent” and “The lack of clutter on the screen and the ease with which classes could be compared certainly helped ease the process of deciding on mappings”. With respect to easing the difficulties experienced

during the first task, the following extracted quotes are representative: “I agree because the tool helped us to discover some structural mismatches that we’ve quickly fixed by hand” and “OISIN removed the work element of performing the mapping and allowed us to just concentrate on the decisions”. Finally the following comments arose when discussing whether it was easier to identify mappings with the OISIN tools than with Protégé/PROMPT tool: “PROMPT didn’t seem to allow the more intuitive approach to performing mapping that OISIN did” and “You were given much better visual feedback in OISIN than in PROMPT, which gave you, the user, better control and understanding of what you were doing”.

Questions five and six of the questionnaire sought feedback on what parts of the mapping task were OISIN tools particularly useful or not useful. The responses are summarised in Table 6-23.

Parts of the mapping task where OISIN tools considered to be useful or not useful?

	Identifying the mappings according to the suggested mapping workflow	Identifying concepts that might be a candidate anchor pair	Identifying whether anchor paths lie on similar is-a paths	Identifying mismatch anchor pairs	Identifying equivalent properties of candidate anchor pairs	Documenting anchors and anchor paths
Student 1	useful	useful	not useful	not useful	useful	useful
Student 2	useful	useful	useful	useful	useful	useful
Student 3,4,5	useful	useful	useful	useful	useful	useful
Student 6	useful	useful	not useful	not useful	useful	useful
Student 7	useful	useful	not useful	not useful	useful	useful
Student 8	useful	useful	useful	useful	useful	useful
Student 9	useful	useful	useful	useful	useful	useful
Student 10	useful	useful	useful	useful	useful	useful

Table 6-23: Opinions for what parts of mapping task OISIN tools useful

In general it can be seen that OISIN tools was found to be useful for the majority of the tasks. From the table, it is clear that three students found OISIN not particularly useful in assisting to identify mismatch anchor pairs or anchor pairs lying on similar is-a paths. One of the associated comments reveal that the integration of the tools was the main issue: “It did not seem to help with identifying anchor paths, not visually anyhow. It did throw up structural mismatches in the XML output, but none of these were obvious in OISIN itself. The identification was left up to the user to work out.”

The final question in the questionnaire asked the students to suggest improvements for the OISIN tools to better support the task of mapping. The most common improvement suggested was the integration of the Mapping Analysis tool with the Anchor Capture tool in such a manner that structural mismatches would be graphically displayed. Other suggestions included:

- Distinguishing inherited properties through colour/font;
- Automatic/Sensible sizing for windows;
- Shortcut for select and find button combinations;
- Addition of a “Word Find” option.

The suggested improvements above have already been incorporated as part of Oisin-AC-semi-v2 (cf. section 5.1.9.2).

Finally, the students were asked to estimate the number of hours taken in completing the two tasks. Table 6-24 presents these results. The students were asked not to include any time spent in learning to use the tools involved but rather asked to focus on just the time taken for the mapping task.

Number of Hours expended	Group 1	Group 2	Group 3	Group 4
non-OISIN supported task	4	6	3	9
OISIN supported task	1	1.5	1.5	1.5

Table 6-24: Hours spent on each mapping task

For the first task the time taken was minimum three hours and maximum nine hours. In comparison, the second task (using OISIN) was undertaken in minimum one hour and maximum one and a half hours. This shows a significant time saving when groups used the OISIN tools.

6.3.7 Discussion and Findings

The first hypothesis of this experiment to be tested was that the identification of mapping and structural mismatch activities requires tool support. Analysis of the mappings produced as a result of the first part of the experiment would indicate that this support would be beneficial, as the structural mismatch involved was not identified by either group; on average there were two incorrect and three missed gold mappings

out of sixteen mappings for the AB pair and there were four incorrect and three missed gold mappings out of twenty-six mappings for the CD pair. This finding is supported in the analysis of the responses to the first questionnaire that found that the parts of the mapping task that were most difficult were the identification of candidate mappings and identification of structural mismatches.

The second hypothesis was that the OISIN tools are highly usable, assist the activities in the committed mapping process and lead to the successful generation of mappings. In general, the findings indicate that the OISIN mapping tools allow users to identify more accurate mappings than using a handcrafting approach supported by Protégé/PROMPT. It is particularly useful that it was found that the OISIN tools enabled the user to avoid the creation of structural mismatches. The fact that the two groups involved with the AB mapping in the first task failed to avoid such mappings despite particular emphasis being placed on watching out for such situations, clearly shows that the OISIN Mapping Analysis tool is useful. Although not definitive, the trend of decline in incorrect mappings is also indicative of the ability of the OISIN mapping tools to support more accurate mappings. In addition, the improvement in mapping performance by three of the four groups supports the claim of increased accuracy using the OISIN mapping tools. However, the lack of improvement in the number of gold mappings that were missed by groups in the two tasks indicates that the OISIN mapping tools could be enhanced to try to better support the user in this respect. A simple improvement that has since been implemented is to remind users at the end of a mapping session of candidate class matches that have not been mapped. This might help to reduce the number of “surprising omissions” mappings but this has yet to be confirmed.

In addition, the responses from the second questionnaire confirm that the OISIN tools were found useful in support of the majority of the mapping tasks and in alleviating the difficulties found during the first mapping task. For a small number of students however the OISIN tools were not considered useful in the determination of structural mismatches, but upon further investigation this was due to the fact that the OISIN Mapping Analysis tool is currently not fully or visually integrated with the OISIN Anchor Capture tool. This improvement has been started as part of the future work that has begun and described in section 8.3.

Protégé/PROMPT's wide availability and popularity as an ontology engineering environment would tempt some organisations to use the tool for mapping tasks. The analysis though confirms that Protégé/PROMPT was not helpful when undertaking the mapping task. As discussed previously, this is not surprising (and not a criticism) as the tool primarily is intended to be a support for ontology merging rather than mapping.. However, its attempted usage by the groups in the first task reaffirms that a dedicated mapping tool is required and appreciated.

It was also found that undertaking a mapping using a mapping tool can be up to four times faster than using pen and paper. The importance of leaving the user in control and the importance of graphical feedback also emerged strongly from the responses.

In summary, the OISIN mapping tools were found to provide practical ontology mapping support by alleviating some of the difficulties found in the mapping task and by speeding up the process. In addition, the OISIN tools were found to be very usable and allowed ontology mappings to be identified and generated easily, although a graphical means to represent structural mismatch would be appreciated.

This chapter has presented the experiments that were undertaken to evaluate the OISIN ontology mapping process and supporting tools. Chapter Seven continues by describing the trial that was undertaken to demonstrate the usage of the generated mapping information artefacts at runtime to support semantic interoperability between applications. In addition it compares OISIN with related work.

7 TRIAL AND RELATED WORK

This chapter describes the trial created to demonstrate the use by the OisinSMU service of the information artefacts generated by the OISIN mapping tools. Section 7.1 introduces the trial and the detail of the trial is described in section 7.2. Finally section 7.3 compares the OISIN framework with the related systems in the state of the art.

7.1 Introduction

The aim of the trial was to show that the generated mapping information from the mapping phase could be used at runtime by the OisinSMU service (described in section 5.1.11) to translate terms for applications which had committed to different ontologies. In particular, the aim of the trial was to investigate the kind of success that the OisinSMU service would have in returning mapping information to an application. The trial involved the development by four groups of students, of sixteen different query sets, representing sixteen different applications that would use terms drawn from four different ontologies. The twelve students involved were not aware of the mappings that were available between these ontologies so that their choice of terms would not be influenced. The terms from the student queries were submitted to the OisinSMU service and the mapping information that was returned was analysed.

7.2 Interpret Mapping Activity Trial

Sixteen independently authored applications were used to undertake this trial. This section provides an overview of the trial, details the applications and the trial procedure, and finally presents results.

7.2.1 Overview

The trial was undertaken as part of the preparation for the *Committed Mapping* activities experiment. The motivation for taking this approach was:

- To provide the participants experience of how ontologies could be used by applications, before trying to map such ontologies;
- To have the participants author the applications before being exposed to the subject of ontology mapping or the ontology mapping experiment.

The four groups of M.Sc. Ubiquitous Computing students were allocated one of the A, B, C, D ontologies relating to the University domain and requested to:

1. Create an XML instance document based on the classes/properties of the allocated ontology such that at least fifteen different classes (taken from across the ontology) should be used during the specification of the instances (with three properties used in each);
2. Imagine four different ubiquitous computing applications that might use the information in the XML document and for each application to define five useful, interesting and distinct XQuery/XPath queries over the XML instance document.

Details of the resulting applications are provided in section 7.2.2. The ontological terms used by these applications were then drawn from the XPath/XQueries and formed the input to the OisinSMU service, as explained in section 7.2.3. The results of the trial are presented in section 7.2.4.

7.2.2 Applications

The terms used by all the applications are summarised in Table 7-1. An example XML instance document that was created using terms from the “C” ontology is presented in Appendix F-1.

Group 1 Ontology A	Group 2 Ontology B	Group 3 Ontology C	Group 4 Ontology D
Academic	Article	Abstract	AcademicStaff
Affiliated-Person	ArticleInBook	Academic_Space	adjacent
Article	author	Asset_Tag_Number	Article
author-of	BName	Associated_Pages	assigned_to
Book	Book	Associated_Website	author
Chapter	carriedOutBy	Attaced_Research_Project	Book
Educational-Employee	containsArticle	Attached_Course	Building
Employee	content	attached_lab	building_number
faculty-of	CountyOfOrigin	Attached_Research_Group	Campus
family-name	DevelopmentProject	Author	CompoundPlace
full-name	editor	Base_Url	Computer
Generic-Area-Of-Interest	employees	Capacity	contain
given-name	firstPage	Computer	Course
has-affiliation	groupHead	Conference	course_teaching
has-affiliation-to-unit	GroupItsPartOf	Contact_Person	course_title
has-appellation	hasPartsInstitute	Course	cpu
has-author	hasStudent	Course_Code	date
has-chapter	headEmployee	Course_Coordinator	department
has-editor	Institute	course_for_lecture	description
has-faculty	JName	course_for_tutorial	Desktop
has-gender	Organisations	Course_Lecture	Equipment
has-ISBN-number	Journal	Course_Title	GraduateStudent
has-issue-number	JournalArticle	Course_Tutorial	hardback
has-page-numbers	keyword	Current_Contact_Hours	have_equipment
has-publication-date	lastPage	Current_Location	ID
has-publisher	Lecturer	Duration	ID_number
has-room-number	memberOfGroup	Employee_Number	identifier
has-subject-area	memberResearcher	Event	is_part_of
has-sub-unit	Name	Event_Duration	Journal
has-supervisor	organisation	Exam	journal
has-title	Person	exam_for_course	Laptop
is-affiliated-organization-of	PhDStudied	exam_invigilated	latitude
is-member-of-project	price	Exam_Invigilator	located
Journal	Project	examined_course	longitude
Magazine	projectInfo	Facility	number_of_places
Organization	publisher	Full_Url	PDA
Organization-Unit	reportNo	GPS_Location	Peripheral
Person	Researcher	ID	Person
PhD-student	ResearchGroup	Index_Page	personal_phone_no
PhD-Thesis	ResearchProject	Input_Devices	Place
Professor-In-Academia	researchStudent	Is_Colour	presentation
Publication	secretaryOfGroup	Is_working	Printer
Research-Area	Secretary	Lab_Demonstrator	Publication
Secretary	staffId	Lab_Session	published_work
Student	Student	Laboratory	publisher
studies-at	studentOfCollege	Lecture	qualification
studies-in-unit	TechnicalReport	Lecture_Lecturer	qualification_held
Thesis	TechnicalStaff	Lecture_Theatre	ram
University	University	Lecturer_Staff_Person	salary
University-Faculty	year	Lenght	Scanner
Working-Person		Location	sex
works-for		member	size
works-in-unit		Monitor_Size	spatially_subsumes
		Name	speed_per_minute
		ORI	Staff
		OS_Installed	street_address
		Page_Title	Student
		paper_published_by	supervised_by
		Person	takes_course
		Place	takes_subject
		Postgrad_Student_Person	taught_by
		Postgraduate_Course	Thesis
		Postgraduate_Course_Taken	title
		Printer	Work
		Printer_Type	year
		Projector	year_of_entry
		Research_Group	
		Research_Paper	
		Research_Project	
		resource	
		Room_Code	
		Second_Name	
		Semesterised	
		Space	
		Staff_Person	
		Start_Time	
		Student_ID	
		Student_on_course	
		Student_Person	
		Student_Quota	
		Taught_Postgraduate_Course	
		Theatre_Name	
		Thesis	
		Title	
		Topic	
		Tutorial	
		Tutorial_Room	
		Tutorial_Tutor	
		Webpage	
		Website	

Table 7-1: Terms extracted from Student XML instance documents

A wide ranging set of applications were produced upon these XML instance documents. Each application consists of five XQueries and some sample queries from group three are presented in Appendix F-2. The applications are summarised in Table 7-2.

	Group 1	Group 2	Group 3	Group 4
Ontology	A	C	B	D
Application 1	Publication Information	Library Queries	Web Page Search	Publication Information
Application 2	College Prospectus	Project Information Finder	Student Calendar	University Course Viewer
Application 3	Student Registration	PDA Person Locator	Computer Services Search	Staff/Student Manager
Application 4	Human Resources	Organisation Queries	Electronic Nameplate	Electrical Equipment Monitor

Table 7-2: Summary of Student Applications

Eleven of the sixteen applications defined by the students are traditional ones, targeting similar areas of the university domain, i.e. research projects, publications, staff/students, courses. The other five (shaded in the table) are interesting deviations from the traditional and were defined with a ubiquitous computing based university in mind:

- *Computer Services Search* was designed to allow students/staff find facilities based on current location, and focuses on equipment;
- *PDA Person Locator* is designed for visitors trying to find people quickly, and focuses on location information;
- *Student Calendar* was designed to gather events and lecture information of interest, and focuses on timetables and events;
- *Electrical Equipment Monitor* was designed to allow the university manage electricity consumption, and focuses on equipment information;
- *Electronic Nameplate* was designed to hang outside a room and to change automatically based on activity ongoing inside, and focuses on timetable and location information.

The terms drawn from the queries therefore represent a coherent set of independently selected terms from the four ontologies used in combination. These terms, grouped by application usage, are summarised in Table 7-3.

Group 1 Ontology A	Group 2 Ontology B	Group 3 Ontology C	Group 4 Ontology D
College Prospectus	Organisation	Student Calendar	Staff/Student Manager
Organization	Organisations	Person	university
University	ResearchGroup	Student_Person	person
rdfs:label	Name	Postgrad_Student_Person	staff
has-faculty	Book	Second_Name	department
Human Resources	Article	Student_ID	personal_phone_no
is-affiliated-organisation-of	Journal	Course	salary
Working-Person	Institute	Postgraduate_Course	ID_number
Employee	GroupItsPartOf	Taught_Postgraduate_Co	Qualification
has-room-number	employs	Student_on_course	year_of_entry
given-name	Person Locator	Course_Title	takes_course
family-name	Person	Course_Code	have_equipment
works-in-unit	staffId	Event	Course Viewer
Publication Information	Student	Lecture	work
Publication	Name	course_for_lecture	course
Book	phone	Exam	course_title
has-ISBN-number	Researcher	Place	taught_by
Magazine	memberOfGroup	Space	year
Article	groupHead	Academic_Space	number_of_places
has-title	Name	Lecture_Theatre	Publication Information
has-author	Lecturer	Location	work
Student Registration	Researcher	Topic	publication
Person	Secretary	Postgraduate_Course_Take	title
Affiliated-Person	TechnicalStaff	Computer Services	author
Student	Student	Place	ID
has-affiliation-to-unit	Name	Space	book
full-name	staffID	Academic_Space	article
author-of	Project Information	Room_Code	publisher
studies-in-unit	Project	Facility	presentation
PhD-student	Name	Printer	thesis
has-affiliation	carriedOutBy	Current_Location	Journal
	Person	Is_Colour	journal
	Institute	Computer	date
	ResearchGroup	OS_Installed	Equipment Monitor
	Library Queries	Lecture_Theatre	equipment
	Journal	Projector	identifier
	Name	Laboratory	computer
	editor	Webpage Search	laptop
	JournalArticle	Full_Url	assigned_to
	Name	Webpage	description
	Jname	Person	peripheral
	author	Lecturer_Staff_Person	scanner
	lastPage	Staff_Person	size
	firstPage	Employee_Number	located
	Book	Second_Name	ram
	author	Thesis	
	publisher	Title	
	price	paper_published_by	
	Article	Research_Paper	
		Webpage	
		Author	
		Page_Title	
		Research_Group	
		Name	
		Associated_Website	
		Associated_Pages	
		Website	
		Student_ID	
		Person	
		Postgrad_Student_Person	
		Second_Name	
		Student_ID	
		Electronic Nameplate	
		Event	
		Start_Time	
		Topic	
		Academic_Space	
		Room_Code	
		Capacity	
		GPS_Location	
		Start_Time	
		Location	
		Topic	
		Place	
		Space	
		Academic_Space	
		Event	
		Room_Code	
		Location	
		Start_Time	
		Event	
		Exam	
		Topic	
		examined_course	
		Person	
		Student_Person	
		Postgrad_Student_Person	
		Postgraduate_Course_Taken	
		Second_Name	

Table 7-3: Query terms grouped by application

7.2.3 Trial Setup

The sets of terms described in table 7-3 were then filtered to create for each group a distinct list of terms. At the same time, terms that were used by the applications but did not appear in the original ontology were removed (e.g. the term *JName* from Group two). These distinct lists of terms were then used as input to the OisinSMU service. The ubiSMU XQuery (see Figure 7.1) analyses the files that contain the sets of terms by invoking the OisinSMU for each term (lines 3 to 10). It returns:

- The number of anchor mapping classes contained in the set; (line 14)
- The number of classes that were found in an anchor path; (line 16)
- The number of classes that had a match but were not considered an anchor or did not appear in an anchor path; (line 15)
- The number of classes that did not match at all; (line 17)
- The number of properties that were mapped; (line 18)
- The number of properties that matched but were not mapped; (line 19)
- The number of properties that did not match or map. (line 20)

```
1. declare function ubiSMU($termlist,$AAfile,$INTfile)
2. {
3.     let $result:= <result> {for $t in
4.         doc($termlist)/ont/t/text()
5.         return
6.         let $detail := SU:SMU($t,$AAfile,$INTfile)
7.         return
8.         (<term> <name> {$t} </name>
9.          <type> {node-name($detail)} </type>
10.         </term>)}
11. return
12.     ($result,
13.      <totals> {count($result/term)} </totals>,
14.      <equivalences_anc> {count($result/term[type="anchor"])}
15.      </equivalences_anc>,
16.      <matches_class> {count($result/term[type="match_class"])}
17.      </matches_class>,
18.      <in_path> {count($result/term[type="class_in_path"])}
19.      </in_path>,
20.      <unknown_class>
21.      {count($result/term[type="nomatch_class"])} </unknown_class>,
22.      <equivalences_prop>
23.      {count($result/term[type="equivalent_property"])}
24.      </equivalences_prop>,
25.      <matches_prop> {count($result/term[type="match_property"])}
26.      </matches_prop>,
27.      <unknown_prop> {count($result/term[type="nomatch_property"
28.      or type="nomatch"])} </unknown_prop>
29.      )
30. }
```

Figure 7-1: XQuery used to invoke OisinSMU

Executing the ubiSMU XQuery over the sets of terms generated by the applications and the AB and CD gold-standard mappings, yielded the information which provides the basis for the SMU success rate evaluation described in the next section.

7.2.4 Results

Application requests can be categorised into four types, reflecting the information that is returned by the OisinSMU service in response to a term matching request:

- Requests upon terms which match to an identified anchor mapping;
- Requests upon terms which match to terms that appear on some anchor path;
- Requests upon terms which match to terms that were identified as candidate matches but not as an anchor;
- Requests upon terms that do not match to any of the above.

The application can decide to use the information returned by OisinSMU or not.

Table 7-4 shows an analysis of the terms for each of the ontologies, the XML instance documents and the trial applications and whether they appear as anchor mappings, within anchor paths, as candidate matches only and those which do not appear at all in the mapping information. For example from the top table it can be seen that fifty-five percent of the classes in ontology “A” appear in the mapping information in some way or other. Seventy-seven percent of these classes were chosen by chance by group one to appear in the XML instance document as shown in the middle table. Finally the bottom table shows that ninety-one percent of the XML instance document terms used by the group to develop their application queries had anchor mapping or candidate matching information returned from the SMU.

ONTOLOGY versus Mapping Analysis	A	B	C	D
Total Number of Classes	56	96	63	66
Anchor Mapping Classes	15	16	26	26
In Anchor Path Classes	9	10	20	9
No mapping but Candidate Match Classes	7	4	3	7
No match or mapping	25	66	14	24
Percentage Classes that Map/Match	55%	31%	78%	64%
Total Number of Properties	66	92	143	92
Anchor Mapping Properties	17	17	19	17
No mapping but Candidate Match Properties	20	18	41	23
No match or mapping	29	57	83	52
Percentage Properties that Map/Match	56%	38%	42%	43%

INSTANCE DOCUMENT versus Mapping Analysis	Group 1	Group 2	Group 3	Group 4
Total Number of Classes	22	17	30	24
Anchor Mapping Classes	9	10	14	12
In Anchor Path Classes	3	3	10	1
No mapping but Candidate Match Classes	5	1	2	3
No match or mapping	5	3	4	8
Percentage Classes that Map/Match	77%	82%	87%	67%
Total Number of Properties	32	33	61	42
Anchor Mapping Properties	13	4	6	13
No mapping but Candidate Match Properties	4	2	14	6
No match or mapping	15	27	41	23
Percentage Properties that Map/Match	53%	18%	33%	45%

APPLICATIONS versus Mapping Analysis	Group 1	Group 2	Group 3	Group 4
Total Number of Classes	11	12	25	14
Anchor Mapping Classes	6	8	14	7
In Anchor Path Classes	3	1	2	1
No mapping but Candidate Match Classes	1	0	5	1
No match or mapping	1	3	4	5
Percentage Classes that Map/Match	91%	75%	84%	64%
Total Number of Properties	16	10	26	26
Anchor Mapping Properties	8	4	4	8
No mapping but Candidate Match Properties	1	1	6	5
No match or mapping	7	5	16	13
Percentage Properties that Map/Match	56%	50%	38%	50%

Table 7-4: Analysis of Terms with respect to Gold-standard Mapping Information

The top part of Table 7-4 also provides an indication that a good percentage of the terms from all the ontologies appear in the mapping/match information, ranging from thirty-one percent to seventy-eight percent of classes and thirty-eight to fifty-six percent of properties.

Figure 7-2 presents the information about the success rate for applications for class terms graphically.

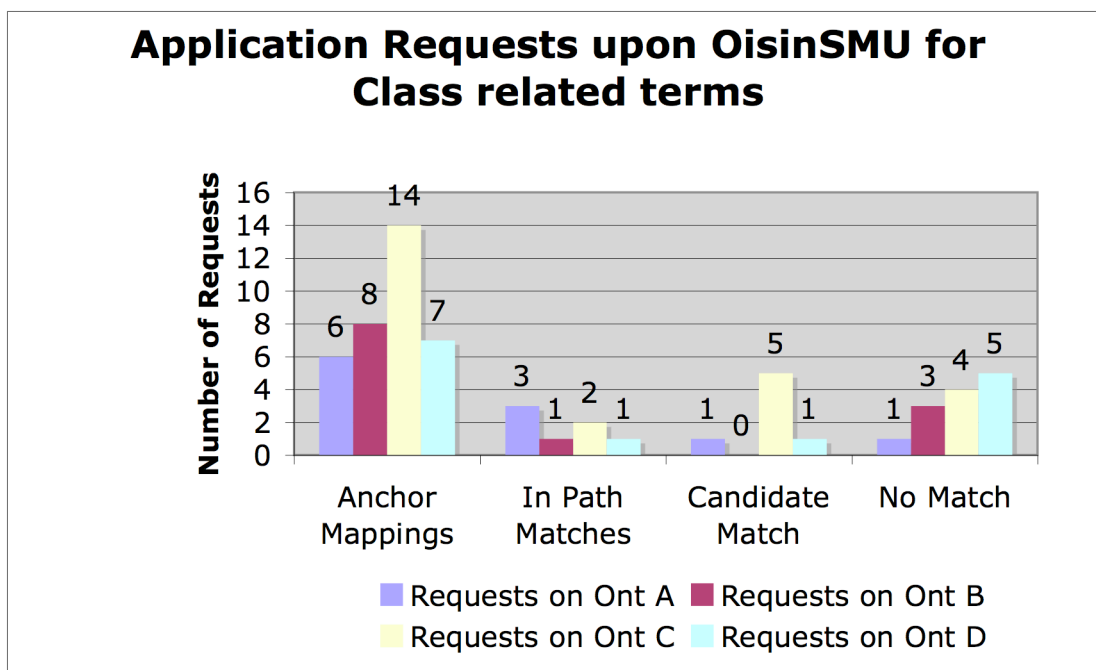


Figure 7-2: Breakdown of Mapping Information kept for Class related terms

Figure 7-2 shows (on a per ontology basis) that the majority of classes have either anchor mappings or they appear in anchor paths, which would provide the application with a good degree of confidence when deciding whether to use the information as the basis of a mapping or not. Candidate matches could of course be used too, but with lower confidence.

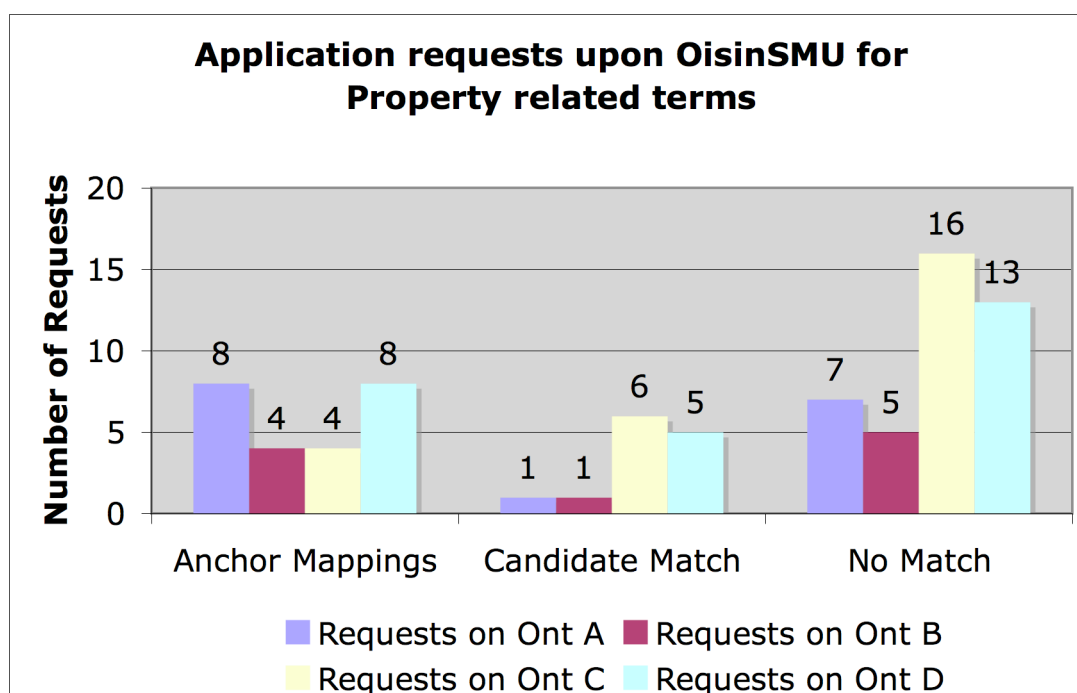


Figure 7-3: Breakdown of Mapping Information kept for Property related terms

As can be seen from Figure 7-3, the ratio (on a per ontology basis) between property terms that have some form of mapping/match and those that do not have any is in the majority of cases one-to-one. The high proportion of unmatched/unmapped properties is not surprising as the modelling of individual properties for two classes can vary widely dependent on the perspective of the original ontology designer.

A request from an application can be considered successful if the OisinSMU returns information about an anchor mapping, information about an anchor path or a term that has candidate matches. The success rate (in terms of percentages) of the OisinSMU in response to the application requests for class and property terms together has been summarised in Figure 7-4.

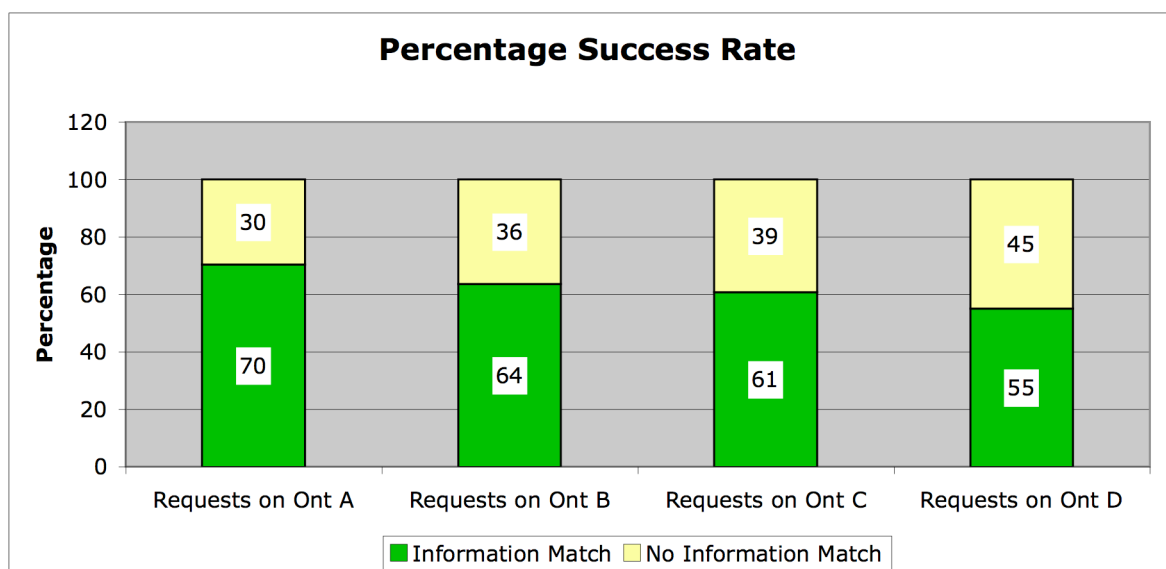


Figure 7-4: Overall success rate

Discussions and Findings

It should be recalled that the AB and CD gold mappings used in this trial were generated as a result of a task that attempted to identify the maximum number of anchor mappings in the ontology pairs. This explains the large number of anchor matches, and it can be inferred that the small number of “in-path” matches is due to the fact that these are not anchors. In fact, a minimum anchor identification approach to mapping AB and CD would not be hugely different, with there being an overall reduction of three anchors in the CD mappings and one anchor in the AB mappings.

This is because the particular ontologies involved are wide and shallow ontologies, leading to short anchor paths.

The trial demonstrates however that OisinSMU performed reasonably well against a series of sixteen applications authored independently of the ontology mappings. The ratios of mapping/match information returned versus no information returned for class terms for individual ontologies were as follows: 10:1 for ontology A; 21:4 for ontology C; 9:3 for ontology B and 9:5 for ontology D. For properties however the ratios were approximately 1:1 success versus failure.

Of course there is nothing to guarantee that OisinSMU would perform so well given a totally different set of applications as the success rate is highly dependent on: the breadth of the ontology being referred to; the part of that ontology being used by the application (in our case represented by XML instance document and XQueries posed upon it) and the nature of the anchor mapping between the ontologies.

7.3 Comparison of OISIN to related work

In the analysis parts of section 3.4, selected state of the art systems were analysed with respect to the key requirements introduced in Section 3.3. In Table 7-5, the comparison table of the summary section 3.5, is repeated but this time with the addition of a column reflecting how OISIN meets the key requirements set out.

	MAFRA	COMA++	INRIA	Protégé plugins	OISIN
Ontology mapping lifecycle	Does not include characterisation activities and only partially management phase	Does not include characterisation activities	Does not include characterisation activities or management phase	Does not include characterisation activities or management phase	includes all phases
Relationship between Process and implementation	Process is tied to implementation	Process is tied to implementation	No explicit process supported	only ontology integration supported	Process specified independent of implementation
Extensibility	Limited to fixed set of matcher types	API not publically available	format enables extensibility	substitution by source code alteration	XML based interfaces allow for substitution of all tools/algorithms
Nature of Mapping	What constitutes a mapping determined by user	What constitutes a mapping determined by user	What constitutes a mapping determined by user	What constitutes a mapping determined by user	What constitutes a mapping determined by user or by application as appropriate
Match/Mapping separation	exists and exposed	exists but how it is achieved not exposed	achieved through XML format	separation does not exist	exists and exposed
Match/Mapping format	only map format XML	only map format RDF, CSV	map/match formats in XML	not possible	map/match formats in XML
Reduction of Uncertainty in mapping	Some inference based suggestions provided to user	Some uncertainty reduction based on tools to support reuse of mappings	not supported	Some inference based suggestions provided to user	Uncertainty reduced through committed mappings and reuse

Table 7-5: Comparison of OISIN with selected state of art systems

MAFRA has defined a good ontology mapping lifecycle, but the process is not specified in detail but rather is closely aligned with its implementation. Thus it would be difficult for an organisation to take the MAFRA process and attempt to implement it or part of it in a different way. OISIN has specified a technology-neutral process. In addition, MAFRA does not have any activities that support a user in the tasks that may need to be undertaken before a mapping phase, for example determination of mapping difficulty. OISIN has introduced the characterisation phase and decision to map activity. The management activities of MAFRA relate to versioning of the mappings or collaboration in the creation of the mappings and do not address sharing or integration of mappings with others. OISIN includes sharing and integration of mappings in its

management phase. The MAFRA implementation is extensible but limited to a fixed set of matcher types. OISIN allows the introduction of any kind of matcher. In MAFRA, the determination of mappings from matching is system led with some user interaction but results in a fixed set of mappings that are not widely interpretable. OISIN allows for the user or an application to determine what is a mapping, and these mappings are interpretable by any applications with access to basic XML technologies.

COMA++ has a good architecture for generating and managing mappings but has no clear implementation independent process. It does not support the user in the determination of what would be involved in undertaking a mapping, and could benefit from the introduction of a characterisation phase. Although the matching and mapping activities are separate in the architecture, there is no published API so it is difficult to determine if the matching information could be published separately such that an application could determine a mapping itself as in OISIN. In addition, although the architecture allows for different matchers to be substituted into the system, this feature is difficult to assess due to the lack of availability of the API. In contrast, the implemented OISIN information model has been exposed as XML, allowing for other matchers to be easily introduced. COMA++ includes some useful detailed mapping manipulation features (e.g. MatchCompose, Diff, Compare) that might influence the functionality of an implementation of tools in support of a mapping management phase of OISIN.

The INRIA API provides a means to manipulate matches and mappings expressed in the INRIA format. It does not have a ontology mapping lifecycle or process associated with it. Rather than being a system itself, it allows the creation of ontology matching/mapping systems. However the current implementation of the API is very basic and not fully functional (e.g. in its rendering of mappings into SWRL). The format however looks promising and its use is being evaluated as part of an extension to OISIN that has begun called EOIN (Extended OisIN) which is discussed in section 8.3. This extension will initially allow for mappings to be saved in the INRIA format.

The Protégé Anchor-PROMPT plug-in is intended to support the merging/integration of ontologies into a single ontology, and thus is not separable for inclusion in an ontology mapping process. The mapping task is entwined in the integration process

such that match and mapping information is not exposed. OISIN provides a technology-neutral specification and tool implementation for an ontology mapping process. Although some inference based suggestions are made by the Anchor-PROMPT plug-in to the user with respect to determination of mappings, these have been found to be unhelpful during a ontology mapping process (cf. section 6.3.1). In contrast OISIN allows for the user to be fully in control of the mapping process.

An additional discussion of how well OISIN meets the framework requirements are presented in section 8.1.

8 CONCLUSIONS

This chapter presents how well the objectives of the thesis were achieved (section 8.1), summarises the contribution made (section 8.2), describes some ongoing work and work which may be undertaken in the future (section 8.3) and concludes with some final remarks (section 8.4).

8.1 Objectives

This thesis identifies the activities, information artefacts, heuristics and guidance required to support a practical ontology mapping process, and examines how they can be coordinated and sequenced in the ontology mapping process. In order to investigate this research question, three objectives were derived:

1. Research and identify models and processes that support ontology mapping;
2. Develop an integrated set of process and software tools to support practical ontology mapping;
3. Evaluate the use and benefit of the process and the software tools developed to generate ontology mappings, and demonstrate the usage of the generated mapping information artefacts at runtime to support semantic interoperability between applications.

This thesis proposes the OISIN framework comprising an ontology mapping process supported by software tools. The process consists of activities (cf. chapter four and appendix B), information artefacts (cf. chapter four and appendix B), heuristics (cf. section 6.2.3) and guidance (cf. section 5.2.1). The literature reviews undertaken for objective one, shows that the proposed OISIN framework addresses a gap in the state of the art. The OISIN process and OISIN software tools have been developed in response to objective two. The process and software tools meet the framework requirements set out in section 3.3. In summary, OISIN supports a full ontology mapping lifecycle; is extensible and adaptable and supports the creation of relevant mappings, ranging from supporting a user's determination of mappings to an application's determination of mappings. Currently there is no other framework that meets all of these requirements. Furthermore, to our knowledge, no other proposal allows for both user determination of mappings and application determination of mappings within the same framework. Finally, objective three has been achieved

through two experiments, a trial and through usage in two major research projects. Experience from these activities demonstrates that both the process and software tools have been found to be useful and beneficial.

The rest of this section discusses how well these objectives have been achieved in more detail.

State of the Art Research Objective

The state of the art review presented in chapter three details the research that was undertaken in the achievement of the first objective. Both the initial and recent literature reviews reveal that progress has been made in the development of new algorithms and tools for matching of ontologies but very little progress has been made on the development of process and software tools to support practical ontology mapping process over the full ontology mapping lifecycle. Thus the proposed OISIN framework addresses a gap that exists in the state of the art.

Process and Tools Development Objective

The objective of developing an integrated set of process and software tools to support practical ontology mapping has been substantially achieved. The process has been specified in a technology-neutral form using UML (cf. chapter four and appendix B). The implementation comprises of: software tools developed to support the process activities (cf. chapter five); information artefacts (cf. extensible UML class model in appendix B and XML DTDs used in the implementation of the tools in appendix C); heuristics and guidance (cf. appendix D-3, Figure 5-30). The degree of success in reaching this objective can be seen through the examination of how the developed OISIN framework meets the *key practical framework requirements* outlined in section 3.3. For each requirement that has been identified, a brief review of how that requirement has been met in OISIN is presented in the following discussion.

The framework must support an ontology mapping lifecycle.

As yet, there is no commonly agreed ontology mapping lifecycle. The OISIN process proposes the characterisation phase, the mapping phase, the execution phase and the management phase as the four phases that comprise a full ontology mapping lifecycle. Some of these phases appear individually in parts of the state of the art, but not yet at

the same time in the one framework. The characterisation phase characterises the ontologies with respect to their amenability for mapping. The mapping phase enables the identification of mappings between ontologies. The execution phase supports the use of the mappings to enable runtime semantic interoperability between applications. The management phase provides for the alteration, sharing and reuse of mappings. This thesis has successfully specified and implemented the activities, and their sequencing, for the characterisation, mapping and execution phases. Examination of the management phase activities has mainly been through work aligned with this thesis, and will be further examined through proposed future work (cf. section 8.3).

The ontology mapping process of the framework must be specified in a manner that is technology-neutral.

The process has been specified in a technology-neutral form using UML (cf. chapter four and appendix B). It is argued that the benefit of having the process specified using UML is that the process can be tailored and situated in many different parts of an organisation in support of different tasks (e.g. ontology integration, peer-to-peer querying etc.). Experience of using UML in the specification of processes elsewhere (e.g. in the Unified Process and in TeleManagement Forum) informed this decision, and it was not considered something that had to be proven in the thesis work.

All activities should be defined so that they are potentially automatable.

Care was taken in the specification of the process to define activities that could equally be supported through manual, semi-automated or automated means. It has been shown that some of the more complex activity definitions meet this requirement, through the implementation of semi-automated and automated versions of the *Decide to Map*, *Anchor Capture*, and *Mapping Analysis* activities. Meeting this requirement allows organisations to tailor the implementation of the process to suit their particular organisational needs with respect to deployment.

Support the determination of mappings independent of any one particular application and also support the determination of mappings by an application itself dependent on that particular application's perspective. In other words, mapping generation should be "relevant".

OISIN can support the traditional semi-automated mode of undertaking maximum mappings determined by an expert, independently of any particular application. In addition, OISIN can support the creation of mappings by applications themselves relevant to the context of its processing. This has been achieved through the use of committed mappings and the exposure of candidate match and committed mapping information for direct use (or through a software service) by applications at runtime. In the implementation, anchors were used as the means to express committed mappings. The trial has shown that applications can use the information generated by OISIN at runtime to aid the achievement of semantic interoperability. In addition, the trial was useful in evaluating the success rate of the mapping information in response to queried terms arising from sets of independently authored applications. However the applications designed were limited in scope and ambition (due to the constraints of the experiment), and it would be very informative to undertake further trials with more fully featured applications that require mapping information.

Should support substitution of algorithms and tools.

In the initial implementation of OISIN, the information models used by tools in the implementation are defined using XML DTDs (cf. Appendix C), ensuring that the inputs/outputs of all tools are both machine and human processable. This allows for the easy substitution of algorithms and tools, as the tools merely need to produce/consume XML documents according to the specified DTDs in order to participate in the system. This feature has been demonstrated through the recent introduction of the OisinMO-props, OisinMO-combine and OisinMO-nwmgmt matchers. Independent researchers developed these matchers according to the DTD. The matchers were introduced into the OISIN framework without any changes to the OisinAC applications being necessary. Naturally, the overall OISIN implementation would have benefited from a more widespread availability of independently developed matching software that could have been tailored for use within OISIN. Unfortunately all that is typically published in the state of the art are descriptions of the matching algorithms, rather than actual software implementations.

Support both the full mapping of ontologies and the partial mapping of ontologies.

Although the mapping experiment showed how OISIN could be used to support the full mapping of two pairs of ontologies, there is nothing to prevent the user only undertaking a partial mapping. Wider use of the framework in the future is likely to demonstrate this.

Clear separation between candidate match generation and mapping determination.

In OISIN, candidate match generation and mapping determination have been clearly specified and demonstrated as separate activities. Match generation is undertaken in the characterisation phase. In the case where all of the mappings are determined by the user, the mapping determination is undertaken during the mapping phase. In the case where the mappings are determined by the application at runtime, a combination of the mapping phase (to identify committed mappings) and execution phase (where committed mappings, analysis and match information is interpreted) is used. The primary benefit of such a separation is that it facilitates the application of different strategies for match identification and mapping determination.

Reduce the uncertainty involved in determining mappings from a set of candidate matches.

The availability at runtime of the committed mappings and committed mappings analysis, is intended to reduce the uncertainty involved in the generation of mappings by an application itself. It is argued in this thesis that such reductions will be achieved. At the moment this is difficult to prove, as it would require comparison with some other approaches, not typically available in the state of the art, that allow an application to determine mappings at runtime.

With respect to user determination of mappings, the structural mismatch feature of the OisinMA application and the display of reusable mappings in OisinAC, have been introduced to support the reduction of uncertainty. It can be argued that the reduction of mapping errors in the committed mapping experiment could be as a result of such features. However, it is acknowledged that more active support for mapping determination (e.g. mapping patterns) could be more directly beneficial for a user in the future.

Matching and mapping formats need to be widely interpretable and not dependent on particular interpretation architectures.

OISIN produces the mapping information (committed mappings, match information, committed mapping analysis) as a series of XML documents that can be interpreted by any application that has an XML DOM/SAX parser or access to an XPath/XQuery/XSLT engine. Such XML parsers and engines are ubiquitous relative to ontology reasoners and proprietary engines (e.g. MAFRA). This means that mappings could be easily determined by applications themselves using a variety of widely available free technologies through a number of simple APIs.

Where semi-automatic activities are involved, the framework should reduce the effort required and the error rates involved in mapping and enable the user to undertake the task in the context of the user's organisational requirements.

The *Committed Mapping* experiment has shown that the OISIN implementation reduced the effort required in ontology mapping and reduced mapping error rates. In addition, the *Decide to Map* experiment illustrated how organisational requirements might be encoded in the Microsoft Excel spreadsheet used to guide the decision-making. These experiments indicate that the OISIN framework aids the reduction of effort and the reduction of error rates, and supports organisational requirements.

In summary, the author argues that OISIN has met each of the key requirements.

Evaluation Objective

The first part of the final objective of evaluating the use and benefit of the developed process and software tools, was achieved through the execution of two experiments described in chapter six. The *Decide to Map* experiment showed that the information generated by the activities in the characterisation phase was useful in identifying the mapping difficulty that may be involved in a mapping task. In particular, the finding that it was as beneficial (and in some cases more beneficial) to examine the characterisation information about ontologies than to examine the ontologies directly in the determination of mapping difficulty, has potential significance for the design of an industrial strength characterisation phase for an ontology mapping process. The significance lies in the fact that the usability of such characterisation information is not

adversely affected by the size of the ontologies involved. Furthermore, the characterisation information is amenable as a target for the execution of encoded organisational policies with respect to decision-making. The *Committed Mapping* experiment showed that the semi-automated tools, artefacts and guidance of the Mapping Phase were useful in the identification of mappings and structural mismatches. In addition, the tools were found to be beneficial in the reduction of effort required and in the reduction of mapping errors made.

The second part of the final objective involved demonstrating that the usage at runtime of the mapping information artefacts generated by OISIN, would support semantic interoperability between applications. This was achieved through the trial described in chapter seven. This trial showed that sixteen independently authored applications (written specifically for the experiment) successfully used the mapping information generated by OISIN.

Although this trial was encouraging, further evaluation would be beneficial. Such wider evaluation is planned as part of the ongoing usage of OISIN tools in two projects whose application domains require ontology mapping solutions. The first project is Science Foundation Ireland's Centre for Telecommunications Value Chain Research² (CTVR). In this project, OISIN tools are being used to support management knowledge delivery between autonomic nodes. The second project is the Higher Education Authority's M-Zones³ project. This project focuses on the management of ubiquitous computing environments. In this project, OISIN tools are being used to support the intelligent ontology-based querying of context information and the semantic interoperability of user policies. As will be seen in the next section the initial usage of OISIN within these projects has been described in publications.

² www.ctvr.ie

³ www.m-zones.org

8.2 Contribution

The first contribution of the thesis is the development of a technology-neutral specification using UML for the ontology mapping process. To our knowledge, such a specification has not been published before. The process supports a full ontology mapping lifecycle and features a characterisation phase that is currently lacking in the state of the art. This is important, as in order to be practical, some set of activities that help characterise the potential difficulty in mapping between ontologies is required if deployment of ontology mapping in industrial settings is to be successful.

The design of the process supports a continuum of deployment strategies from all activities being undertaken at design time right through to all activities being undertaken at runtime. In addition, organisations are enabled to specialise activities and define which activities will be supported through manual, semi-automated or automated means depending on the intended deployment scenario.

Initial descriptions of the process has been published in:

O'Sullivan D., Lewis D., "Semantically Driven Service Interoperability for Pervasive Computing", 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access, San Diego, USA, 19 September 2003, ACM Press, pages17-24.

O'Sullivan D., Lewis D., Wade V., "Enabling Adaptive Semantic Interoperability for Pervasive Computing", 2nd International Workshop on Managing Ubiquitous Communications and Services (MUCS), Dublin, Ireland, December 13-14, 2004.

It is planned to submit a more detailed description of the process based on this thesis to the European Semantic Web Conference (EWSW) 2005 and selected journals.

In addition, this process can be specialised to suit many different use cases. The OISIN process and tools have also been used in two different projects. In the M-Zones project, OISIN has been applied to the problem of user policy mobility between managed

systems. This work has been published at one of the major network management conferences called Integrated Management:

van der Meer S., O'Sullivan D., Lewis D., Agoulmine N., "Ontology Based Policy Mobility for Pervasive Computing", 9th IFIP/IEEE International Symposium on Integrated Management, Nice, France, May 2005.

OISIN has also been used in M-Zones in research examining the problem of context information management, and initially published as:

Power R., O'Sullivan D., Lewis D., Conlan O., Wade V., "A Context Information Service using Ontology-Based Queries", First International Workshop on Advanced Context Modelling, Reasoning And Management, UbiComp 2004, Nottingham, England, September 7 - 10, 2004.

Furthermore, OISIN has also been applied in the management of autonomic communications in the CTVR project and this work has been published in:

Lewis D., O'Sullivan D., Power R., Keeney J., "Semantic Interoperability for an Autonomic Knowledge Delivery Service", 2nd IFIP TC6 International Workshop on Autonomic Communications (WAC 2005), Athens, Greece, October 3-5, 2005.

Finally, the same process can be used whether the determination of what is a mapping is to be undertaken by an expert user, or by the application itself. This allows for the flexible deployment of the tools in many different situations and means that the same tools can be used for either use case. This increases the general utility of the tools and process to an organisation and reduces the amount of user training involved.

A further contribution is the finding arising from our evaluation experiments that the inspection of information about ontology characteristics is as useful, and in some cases more useful, in determining ontology mapping difficulty, than the current informal practice of examining the ontologies directly. This is important, as it has demonstrated that a repeatable process can be employed and encoded in tools, in support of the

determination of difficulty of mapping, and that this determination processing is not impacted by the size of the ontologies.

8.3 Future Work

Arising out of the research conducted for this thesis, a number of areas are being pursued. First, a number of improvements to the developed tools are being pursued (described in section 8.3.1) in the development of Extended OisIN (EOIN). Second, tools and approaches to support the OISIN management phase are being researched (section 8.3.2). Finally, the deployment of the tools in different application areas is raising issues that need to be addressed as discussed in section 8.3.3.

8.3.1 Extended OisIN (EOIN)

Recently implementation has begun on a new version of the Anchor Capture tool, called EOIN-AC. The main aim of the new tool is to enable the capturing of complex mappings that involve more than one class or property from each ontology, and a variety of mapping operators (equivalence, subsumption, “>” and so on). This involves a change to the graphical user interface and the export of the mapping information in the INRIA format.

As part of this work it is planned to source different types of matchers (e.g. semantic model matchers) as they become available, and/or implement such matchers based on algorithm descriptions. The intention would then be to derive, through experimentation, strategies for the combination of such matchers. The publication of such strategies would fill a gap currently present in the state of the art.

In addition, the introduction of support for mapping derivation from candidate match information will be explored. Currently this is achieved through the advisory display to the user of mappings that others have chosen for a set of ontologies. For example, the idea of leveraging mapping patterns will be investigated. Also, the possibility of generating recommendations of mappings based on peer-analysis of mappings (in a manner like recommender systems), is a potential avenue for exploration.

8.3.2 Management Phase Activities

Recent M.Sc. projects that the author has supervised have initiated exploration into the issues and technologies that may be involved in the sharing and integration of mappings between peer mapping systems. Conroy has explored the issues involved in setting up a peer-to-peer network that would share such mappings (Conroy 2005), whereas Lynch has explored how a Content Based Network approach (Carzaniga et al. 2001) could be altered to allow for the efficient distribution of ontologically based information such as mappings (Lynch 2005). In addition, recent work has explored the availability of engines suitable for undertaking inferences over a set of mappings when integrating a mapping into an existing set. In particular, the Semantic Web Rule Language (SWRL) was explored as the possible interchange format for the mappings and consequently engines capable of working with SWRL have been examined. This strand of research is being continued as part of the CTVR project but is as yet unpublished.

8.3.3 New application areas

In the CTVR project, OISIN is being applied to map between ontologies that have been derived from a Management Information Base (MIB) based information model and a Common Information Model (CIM) based information model. As discussed in Keeney et al. (Keeney et al. 2005), it was found that the lexical matcher implemented was inadequate in support of the task, as the naming conventions adopted by the MIB and CIM standards meant that each name of class had a high degree of repeated text reflecting a particular sub-area of interest being modelled (e.g. PrinterStatus, PrinterColour, PrinterPages and so on). Upon examination it was concluded that even if OISIN had implemented some of the other matchers classified by Shvaiko and Euzenat, that the candidate matches produced would still be inadequate. In this case, it was decided to implement a domain-specific matcher that used supplementary information available in the models (but not in the ontologies derived from the models) to supplement the lexical matching. The addition of this matcher was straightforward due to the extensible nature of the OISIN framework.

This work has shown that the application of OISIN in different application domain areas has the potential to raise some interesting issues and challenges, and so the

intention is to continue to pursue the application of the OISIN framework to different domains.

8.4 Final Remarks

With the rise of applications that need to operate in *ad hoc* or autonomic or ubiquitous computing environments, there is a need to allow for mapping determination to be undertaken at runtime by the applications themselves. However, fully automated ontology mapping is as yet impractical. Although what is proposed in the OISIN framework falls short of allowing the achievement of all mapping tasks to be undertaken by an application at runtime (in that “committed mappings” need to be available a-priori), it is the author’s view that what is proposed provides a concrete contribution in that direction.

Finally, the problem of Semantic Interoperability is not new, but the emergence of ontologies as an approach to model semantics has provided an opportunity for the semantic correspondences between applications to be specified in a more precise manner as mappings. However in order for the use of such mappings to gain widespread acceptance, a practical ontology mapping lifecycle needs to be put in place. The author believes that the specification of the OISIN process in UML will encourage debate as to what constitutes an ontology mapping lifecycle, hopefully leading to a common understanding. This common understanding should then result in the emergence of a vibrant market for software solutions to support ontology mapping.

BIBLIOGRAPHY

- Aberdeen 2003 "University Ontology", University of Aberdeen, Available at: http://www.csd.abdn.ac.uk/~cmckenzi/playpen/rdf/akt_ontology_LITE.owl, Last visited: September 2005. Version used available at: www.cs.tcd.ie/~osulldps/thesis/a.owl.
- Aumueller et al. 2005 Aumüller D., Do H., Massmann S., Rahm E., "Schema and Ontology Matching with COMA++", Proc. SIGMOD 2005 (Software Demonstration), Baltimore, June 2005.
- Bechofer et al. 2000 Bechhofer S., Broekstra J., Decker S., Erdmann M., Fensel D., Goble C., van Harmelen F., Horrocks I., Klein M., McGuinness D., Motta E., Patel-Schneider P., Staab S., Studer R., "An informal description of Standard OIL and Instance OIL", available on-line as <http://www.ontoknowledge.org/oil/downl/oil-whitepaper.pdf>.
- Beigi et al. 2004 Beigi M., Calo S., Verma D., "Policy Transformation Techniques in Policy-based Systems Management", Proceedings of IEEE Policy 2004, Yorktown, NY, June 2004.
- Benjamin et al. 2001 Benjamin V., Contreras J., Corcho O., Gomez-Perez A., "Six challenges for the semantic web", Poster at International Semantic Web Conference 2002, Sardinia, Italy, June 2002.
- Berners-Lee et al. 2001 Berners-Lee T., Hendler J., Lassila O., "The Semantic Web", Scientific American, May 2001.
- Bowers and Delacambre 2000 Bowers S. and Delacambre L., "Representing and transforming model-based information", Workshop on the Semantic Web at the Fourth European Conference on Digital Libraries, Portugal, 2000.

- Burton-Jones et al. 2003 Burton-Jones A., Storey V., Sugumaran V., Ahluwalia P., "Assessing the Effectiveness of the DAML Ontologies for the Semantic Web", NLDB 2003 Natural Language Processing and Information Systems, 8th International Conference on Applications of Natural Language to Information Systems, pages 56-69, June 2003, Burg (Spreewald), Germany. LNI 29 GI 2003, ISBN 3-88579-358-X.
- Campbell et al. 1995 Campbell A., Chalupsky H., Shapiro S., "Ontological mediation: An analysis", Unpublished manuscript, 1995.
- Carzaniga et al. 2001 Carzaniga A., Rosenblum D., Wold A., "Design and Evaluation of a Wide-Area Event Notification Service", ACM Transactions on Computer Systems, Volume 19, Issue 3, August 2001.
- Castano et al. 2001 Castano S., De Antonellis V., De Capitani di Vimercati S., "Global viewing of heterogeneous data sources", IEEE Transactions on Knowledge and Data Engineering, Volume 13, Issue 2, pages 277-297, 2001.
- Chebotko et al. 2004 Chebotko A., Lu S., Fotouhi F., "Challenges For Information Systems Towards The Semantic Web", AIS SIGSEMIS Semantic Web and Information Systems Newsletter, Volume 1, Issue 1, April 2004.
- Cohen et al. 1998 Cohen P. et al., "The DARPA High Performance Knowledge Bases Project", Artificial Intelligence, Volume 19, Issue 4, pages 25-49, 1998.
- Conroy 2005 Conroy C., "Wildflowr: P2P Sharing of Ontology Mappings", M.Sc. Dissertation, Trinity College Dublin, May 2005.
- Corcho and Gomez-Perez 2000 Corcho O. and Gomez-Perez A., "Evaluating knowledge representation and reasoning capabilities of ontology specification languages", ECAI 2000 Workshop on Applications of Ontologies and Problem-Solving Methods, Berlin, 2000.

- Cross 2003 Cross V., "Uncertainty in the Automation of Ontology Matching", Proceedings of 4th International Conference on Uncertainty Modelling and Analysis, ISUMA '03, IEEE Computer Society, 2003.
- Cui et al. 2002 Cui Z., Jones D., O'Brien P., "Semantic B2B integration: issues in ontology-based approaches", ACM SIGMOD Record, vol 31 no 1, pages 43-48, ACM Press.
- Daconta et al. 2003 Daconta M., Obrst L., Smith K., "The Semantic Web: A Guide to the Future of XML", Web Services and Knowledge Management, Wiley, ISBN 0-471-43257-1-53500.
- Damianou 2002 Damianou N., "A Policy Framework for Management of Distributed Systems", PhD. Thesis chapter 2, Available at: <http://www.doc.ic.ac.uk/~ncd/files/thesis-ncd.pdf>, Last visited: August 2005.
- DAML 2005 DAML Ontology Library, Available at: <http://www.daml.org/ontologies/>, Last visited: October 2005.
- deBruijn et al. 2005a deBruijn J., Martin-Recuerda F., Ehrig M., Polleres A., Predoiu L., "Ontology Mediation Management", IST SEKT project deliverable, 4.4.1, February 2005.
- deBruijn et al. 2005b deBruijn J., Foxvog D., Zimmerman K., "Ontology Mediation Patterns Library", IST SEKT project deliverable, 4.3.1, February 2005.
- Do and Rahm 2002 Do H. and Rahm E., "COMA - A System for Flexible Combination of Match Algorithms", VLDB 2002.
- Do et al. 2002 Do H, Melnik S., Rahm E., "Comparison of Schema Matching Evaluations", Proceedings of Workshop on Web and Databases, Erfurt, Germany, October 2002.
- Doan et al. 2002 Doan A., Madhavan J., Domingos P., Halevy A., "Learning to map between ontologies on the semantic web", 11th International World Wide Web Conference (WWW 2002), Hawaii, USA, 2002.

- DOM 2005 "Document Object Model (DOM)", W3C, Available at: <http://www.w3.org/DOM/>, Last visited: September 2005.
- Dou et al. 2002 Dou D., McDermott D., Qi P., "Ontology translation by ontology merging and automated reasoning", Proceedings of EKAW2002 Workshop on Ontologies for Multi-Agent Systems, pages 3-18, 2002.
- Dumais et al. 1996 Dumais S., Landauer T., Littman M., "Automatic cross-linguistic information retrieval using Latent Semantic Indexing", In SIGIR'96 - Workshop on Cross-Linguistic Information Retrieval, pages 16-23, August 1996.
- Ehrig and Staab 2004 Ehrig M. and Staab S., "QOM - Quick ontology mapping", International Semantic Web Conference (ISWC 2004), LCNS 3298, pages 683-697, Springer, Berlin, Germany, 2004.
- Ehrig and Sure 2004 Ehrig M. and Sure Y., Ontology mapping - an integrated approach, Proceedings of ESWS, pages 76-91, 2004.
- Ehrig et al. 2003 Ehrig M., Haase P., van Harmelen F., Siebes S., Staab S., Stuckenschmidt H., Studer R., Tempich C., "The SWAP data and metadata model for semantics-based peer-to-peer systems", In Proceedings of MATES-2003. First German Conference on Multiagent Technologies. Erfurt, Germany, September 22-25. LNAI, Springer, 2003.
- EON 2004 EON 2004 contest, Proceedings of the 3rd Evaluation of Ontology-based tools (EON), Available at: <http://km.aifb.uni-karlsruhe.de/ws/eon2004/>, Last visited: August 2005.
- Euzenat 2004a Euzenat J., "An API for ontology alignment", International Semantic Web Conference (ISWC 2004), LCNS 3298, pages 698-712, Springer, Berlin, Germany, 2004.
- Euzenat 2004b Euzenat J., "Evaluating ontology alignment methods", Semantic Interoperability and Integration, Dagstuhl Seminar Proceedings, Germany, September, 2004.

- Euzenat 2005a Euzenat J., “An API for ontology alignment (version 1.3)”, Available at <http://co4.inrialpes.fr/align/align.pdf>, June 2005. Last visited: August 2005.
- Euzenat et al. 2004 Euzenat J. (editor), “State of the art on ontology alignment”, Deliverable, IST Knowledge Web NoE, August 2004.
- Fensel 2003 Fensel D., “Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce”, 2nd edition, Springer-Verlag, Berlin, 2003.
- Firat et al. 2004 Firat A., Madnick S., Grosz B., “Contextual Alignment of Ontologies for Semantic Interoperability”, MIT Sloan Working Paper No. 4515-04; CISL Working Paper No. 2004-17, (October 2004).
- GALEN 2005 "Clinical Ontology", Available at: <http://www.cs.man.ac.uk/~horrocks/OWL/Ontologies/galen.owl>, Last visited: September 2005.
- Giunchiglia and Shvaiko 2003 Giunchiglia F. and Shvaiko P., “Semantic matching”, Proc. of IJCAI 2003 Workshop on ontologies and distributed systems, pages 193-146, Acapulco, Mexico, 2003.
- Giunchiglia et al. 2004 Giunchiglia F., Shvaiko P., Yatskevich M., “S-Match: an algorithm and an implementation of semantic matching”, Proceedings of ESWS, pages 61-75, 2004.
- Gray et al. 1997 Gray P., Preece A., Fiddian N., Gray W., Bench Capon T., Shave M, Azarmi N., Wiegand M., Ashwell M., Beer M., Cui Z., Diaz B., Embury S., Hui K., Jones A., Jones D., Kemp G., Lawson E., Lunn K., Marti P., Shao J., Visser P., “KRAFT: Knowledge Fusion from Distributed Databases and Knowledge Bases”, Database and Expert Systems Applications, DEXA '97, Toulouse, France, 1997.

- Grosso et al. 1998 Grosso W., Gennari J., Fergerson R., Musen M., "When knowledge models collide (how it happens and what to do)", 11th Workshop on Knowledge Acquisition, Modelling and Management (KAW '98), Banff, Canada, 1998.
- Gruber 1993 Gruber T, "A Transistional Approach to Portable Ontology Specifications", Knowledge Acquistion, 5, pages 199-220, 1993.
- Guo et al. 2004 Guo Y., Heflin J., Pan Z., "An Evaluation of Knowledge Base Systems for Large OWL Datasets", Technical Report, CSE department, Leigh University, 2004.
- Hameed et al. 2004 Hameed A, Preece A., Sleeman D., "Ontology Reconcilliation, Handbook of ontologies", Stabb S. and Suder R. (eds), International Handbooks on Information Systems, pages 231-250, Springer Verlag, Berlin, Germany, 2004.
- Horrocks et al. 2000 Horrocks I., Fensel D., Broekstra J., Decker S., Erdmann M., Goble C., van Harmelen F., Klein M., Staab S., Studer R., Motta E., " OIL: The ontology inference layer", Technical Report IR-479, Vrije Universiteit Amsterdam, 2000.
- I3con 2004 I3con contest, NIST Performance Metrics for Intelligent Systems (PerMIS) Workshop, August 2004, <http://www.atl.external.lmco.com/projects/ontology/i3con.html>, Last visited: August 2005.
- IPSI 2005 The IPSI-XQ XQuery engine, Available at: http://www.ipsi.fraunhofer.de/oasys/projects/ipsi-xq/index_e.html, Last visited: August 2005.
- IUPUI 2004 "Protein Ontology", Available at: <http://mensa.sl.iupui.edu/ontology/NormalProteinWKeywords.owl>, Last visited: September 2005. Version used available at: www.cs.tcd.ie/~osulldps/thesis/p.owl.

- Jacobson et al. 1999 Jacobson I., Booch G., Rumbaugh J., "The Unified Software Development Process", Addison-Wesley, February 1999, ISBN:0201571692.
- Jasper and Uschold 1999 Jasper R. and Uschold M., "A Framework for Understanding and Classifying Ontology Applications", 12th Workshop on Knowledge Acquisition Modeling Management, KAW '99, 1999.
- Jena 2005 "Jena - A Semantic Web Framework for Java", Available at <http://jena.sourceforge.net/>, Last visited: August 2005.
- JWNL 2005 Java API for Wordnet, Available at: <http://jwordnet.sourceforge.net/>, Last visited: August 2005.
- Kalfoglou and Schorlemmer 2003 Kalfoglou Y. and Schorlemmer M., "Ontology mapping: the state of the art", The Knowledge Engineering Review, Volume 18, Issue 1, pages 1-31, 2003.
- Kalyanpur et al. 2004 Kalyanpur A., Sirin E., Parsia B., Hendler J., "Hypermedia inspired ontology engineering environment: Swoop", Proceedings of 3rd International Semantic Web Conference (ISWC-2004), Japan (Poster).
- Keeney et al. 2005 Keeney J., Lewis D., O'Sullivan D., RoelensA., Wade V., Boran A., Richardson R., "Runtime Semantic Interoperability for Gathering Ontology-based Network Context" , Submitted to NOMS 2006.
- Kitakami et al. 1996 Kitakami H., Mori Y., Arikawa M., "An intelligent system for integrating autonomous nomenclature databases in semantic heterogeneity", Database and Expert Systems Applications, Dexa '96, LCNS no. 1134, pages 187-196, Switzerland, 1996.
- Klein 2001 Klein M., "Combining and relating ontologies: an analysis of problems and solutions", IJCAI-2001 Workshop on Ontologies and Information Sharing, pages 53-62, Seattle, WA, 2001.

- Kong et al. 2004 Kong C., Wang C., Lau F., "Ontology Mapping in Pervasive Computing Environment", International Conference on Embedded and Ubiquitous Computing (EUC-04), pages 1014-1023, Aizu, Japan, 26-28 August 2004.
- Lacher and Groh 2001 Lacher M. and Groh G., "Facilitating the exchange of explicit knowledge through ontology mappings", 14th International FLAIRS Conference, Key West, FL, USA, 2001.
- Lewis et al. 2005 Lewis D., O'Sullivan D., Power R., Keeney J., "Semantic Interoperability for an Autonomic Knowledge Delivery Service", 2nd IFIP TC6 International Workshop on Autonomic Communication (WAC 2005), Athens, Greece, 2005.
- Lynch 2005 Lynch D., "A Proactive Approach to Semantically Oriented Service Discovery", M.Sc. Dissertation, Trinity College Dublin, September 2005.
- Madhavan et al. 2001 Madhavan J., Bernstein P., Rahm E., "Generic Schema Matching with Cupid", Proceedings of International Conference on Very Large Databases (VLDB), 2001.
- Maedche and Staab 2001 Maedche A. and Staab S., "Learning Ontologies for the SemanticWeb", Special Issue on Semantic Web. IEEE Intelligent Systems, Volume 16, Issue 2, March/April 2001.
- Maedche et al. 2002 Maedche A., Motik B., Silva N., Volz R., "MAFRA - A Mapping FRamework for Distributed Ontologies in the Semantic Web", Workshop on Knowledge Transformation for the Semantic Web (KTSW 2002), ECAI 2002, pages 60-68, Lyon, France, 2002.
- Mafra 2004 "MAFRA Toolkit, Open Source toolkit", Available at: <http://mafra-toolkit.sourceforge.net/>, Last visited: August 2005.

- Manchester 2003 "Knowledge Acquisition Ontology", University of Manchester, Available at:
<http://protege.stanford.edu/plugins/owl/owl-library/ka.owl>,
 Last visited: September 2005. Version used available at:
www.cs.tcd.ie/~osulldps/thesis/b.owl.
- Melnik et al. 2002 Melnik S., Garcia-Molina H., Rahm E., "Similarity Flooding: A Versatile Graph Matching Algorithm", ICDE 2002.
- Miller et al. 2000 Miller R., Haas L., Hernandez M., "Schema mapping as query discovery", Proceedings of International Conference on Very Large Databases (VLDB), pages 77-88, Morgan Kaufmann, 2000.
- MINITAB 2005 "Minitab software for statistical analysis", Available at:
<http://www.minitab.com/>, Last visited: October 2005.
- Mitra et al. 2004 Mitra P., Noy N., Jaiswal A., "OMEN: A Probabilistic Ontology Mapping Tool", Proceedings of Workshop on Meaning coordination and negotiation, ISWC 2004, Hiroshima, Japan, 2004.
- Neches et al. 1991 Neches R. et al., "Enabling Technology for Knowledge Sharing", AI Magazine Volume 12, Issue 3, pages 36-56, 1991.
- Noy 2004 Noy N., "Semantic Integration: A Survey of Ontology-Based Approaches", Special Issue on Semantic Integration, SIGMOD Record, Volume 33, Issue 4, pages 65-70, December 2004.
- Noy and Musen 2001 Noy N. and Musen M., "Anchor-PROMPT: Using Non-Local Context for Semantic Matching", Proceedings of IJCAI workshop on Ontologies and Information Sharing, pages 63-70, 2001.
- Noy and Musen 2002 Noy N. and Musen M., "Evaluating ontology-mapping tools: requirements and experience", Proceedings of 1st Workshop on Evaluation of Ontology Tools (EON 2002), EKAW '02, 2002.

- Noy and Musen 2003 Noy N. and Musen M., "The PROMPT suite: Interactive tools for ontology merging and mapping", *International Journal of Human-Computer Studies*, Volume 59, Issue 6, pages 983-1024, 2003.
- Obsrt and Liu 2003 Obsrt L. and Liu H., "Knowledge Representation, Ontological Engineering and Topic Maps", Chapter 7, *XML Topic Maps*, Jack Park Ed., Addison-Wesley, ISBN 0-201-74960-2, 2003.
- ODP 1993 "Open Distributed Processing – ODP Trading Function", ISO/IEC JTC1/SC21/WG7 N880, November 1993.
- Ogden and Richards 1946 Ogden C. and Richards L., "The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism", 8th ed., Harcourt Brace, New York, 1946.
- O'Sullivan 2003 O'Sullivan D., Lewis D., "Semantically Driven Service Interoperability for Pervasive Computing", 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access, San Diego, USA, 19 September 2003, ACM Press, pages17-24, 2003.
- O'Sullivan 2004 O'Sullivan D., Lewis D., Wade V., "Enabling Adaptive Semantic Interoperability for Pervasive Computing", 2nd International Workshop on Managing Ubiquitous Communications and Services (MUCS), Dublin, Ireland, December 13-14, 2004.
- Oszu and Valduriez 1999 Oszu M. and Valduriez P., "Principles of Distributed Database Systems: (2nd Edition),Prentice Hall, ISBN 0-13-659707-6, 1999.
- OWL 2004a "OWL Web Ontology Language, Use Cases and Requirements", W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-webont-req-20040210/>, Last visited: August 2005.

- OWL 2004b “OWL Web Ontology Language Overview”, W3C Recommendation, 10 February 2004, Available at: <http://www.w3.org/TR/2004/REC-owl-features-20040210/>, Last visited: August 2005.
- OWL-S 2004 “OWL Web Ontology Language for Services (OWL-S)”, Submitted to W3C, November 2004, Available at: <http://www.w3.org/Submission/2004/07/>, Last visited: September 2005.
- Protégé 2005 Protégé Plug-in Library, Available at: <http://protege.stanford.edu/download/plugins.html>, Last visited: August 2005.
- Rahm and Bernstein 2001 Rahm E. and Bernstein P., “A survey of approaches to automatic schema matching”, VLDB Journal, Volume 10, Issue 4, pages 334-350, 2001.
- RDF 2004 "RDF/XML Syntax Specification (Revised)", W3C, February 2004, Available at: <http://www.w3.org/TR/rdf-syntax-grammar/>, Last visited: September 2005.
- RDFS 2004 "RDF Vocabulary Description Language 1.0: RDF Schema", W3C Recommendation, February 2004, Available at: <http://www.w3.org/TR/rdf-schema/>, Last visited: October 2005.
- Reilly 1997 Reilly J., "Understanding Statistics and its application in Business, Science and Engineering", Folens, ISBN:0861219805.
- SAX 2005 "Simple API for XML", Available at: <http://www.saxproject.org/>, Last visited: September 2005.
- Sheth 1999 Sheth A., "Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics", Interoperating Geographic Information Systems, Kluwer Academic Press, 1999.

- Shvaiko and Euzenat 2004 Shvaiko P. and Euzenat J., "A Survey of Schema-based Matching Approaches", Submitted to Journal of Data Semantics, Long version in DIT Technical Report DIT-04-87, 2004.
- Silva and Rocha 2003 Silva N. and Rocha J., "Ontology Mapping for Interoperability in Semantic Web", Proceedings of the IADIS International Conference WWW/Internet 2003, ICWI 2003, Algarve, Portugal, November 5-8, 2003.
- Smith 2001 Smith B., "Ontology and Information Systems", Lecture Text, Available at: http://fis.vse.cz/dek4/www_fak4.nsf/V-WebLookup/ScientiaSmith, Last visited: August 2005.
- Stanford 2003 "Travel Ontology", Stanford University, Available at: <http://protege.stanford.edu/plugins/owl/owl-library/travel.owl>, Last visited: September 2005. Version used available at: www.cs.tcd.ie/~osulldps/thesis/h.owl.
- STATSOFT 2005 "History and explanation of General Linear Models", Available at: <http://www.statsoft.com/textbook/stglm.html>, Last visited: October 2005.
- SWETO 2003 "Semantic Web Technology Evaluation Ontology", Available at: http://lsdis.cs.uga.edu/projects/semdis/sweto/testbed_v2_3.owl Last visited: September 2005. Version used available at: www.cs.tcd.ie/~osulldps/thesis/t.owl.
- SWOOGLE 2005 "Swoogle: search and metadata for the Semantic Web", Available at: <http://swoogle.umbc.edu/>, Last visited: August 2005.
- SWRL 2004 "SWRL: A Semantic Web Rule Language", W3C submission, May 2004, Available at: <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>, Last visited: October 2005.

TCD1 2004	"University Ontology", M.Sc. student project 2004, Trinity College Dublin, Version used available at: www.cs.tcd.ie/~osulldps/thesis/c.owl .
TCD1 2005	"Social Life Ontology", M.Sc. student project 2005, Trinity College, Version used available at: www.cs.tcd.ie/~osulldps/thesis/e.owl .
TCD2 2004	"University Ontology", M.Sc. student project 2004, Trinity College Dublin, Version used available at: www.cs.tcd.ie/~osulldps/thesis/d.owl .
TMF 2005a	"The eTOM model", TeleManagement Forum, http://www.tmforum.org/browse.asp?catID=1647 , Last Visited August 2005.
TMF 2005b	"eTOM success stories", TeleManagement Forum, http://www.tmforum.org/browse.asp?catID=1660 , Last Visited August 2005.
Tonti et al. 2003	Tonti G., Bradshaw J. M., Jeffers R., Montanari R., Suri N., Uszok, A., "Semantic Web languages for policy representation and reasoning: A comparison of KAoS, Rei, and Ponder", In D. Fensel, K. Sycara, & J. Mylopoulos (Ed.), The Semantic Web—ISWC 2003, Proceedings of the Second International Semantic Web Conference, Sanibel Island, USA, 2003, LNCS 2870, pages 419-437, Berlin: Springer.
Truran et al. 2005	Truran M., Goulding J., Ashman H., "Co-Active Intelligence for Image Retrieval", Proceedings of the 13th annual ACM international conference on Multimedia, Singapore, pages 547-550, ACM Press, 2005.
UDDI 2003	"Universal Description, Discovery and Integration (UDDI)", Available at: http://www.uddi.org/specification.html , Last visited: September 2005.

- UML 2004 "UML 2.0 Superstructure specification", Open Management Group, Available at:
<http://www.omg.org/cgi-bin/doc?ptc/2004-10-02>, Last visited: September 2005.
- Uschold and Gruninger 2004 Uschold M. and Gruninger M., "Ontologies and Semantics for Seamless Connectivity", Special Issue on Semantic Integration, SIGMOD Record, Volume 33, Issue 4, pages 58-64, December 2004.
- van der Meer 2005 van der Meer S., O'Sullivan D., Lewis D., Agoulmine N., "Ontology Based Policy Mobility for Pervasive Computing", 9th IFIP/IEEE International Symposium on Integrated Management, Nice, France, May 2005.
- van Harmelen et al. 2001 van Harmelen F., Patel-Schneider P., Horrocks I., "DAML+OIL (March 2001) version (revision 4.1)", <http://www.daml.org/2001/03/daml+oil-index.html>.
- Visser et al. 1997 Visser P., Jones D., Bench-Capon T., Shave M., "An analysis of ontological mismatches: Heterogeneity versus interoperability", AAAI 1997 Spring Symposium on Ontological Engineering, Stanford, USA, 1997.
- W3C 2003a "Web Services Architecture", W3C, Available at:
<http://www.w3.org/TR/2003/WD-ws-arch-20030808/>,
Last visited: September 2005.
- W3C 2003b "Food Ontology", W3C, Available at:
<http://www.w3.org/TR/2003/PR-owl-guide-20031209/food>,
Last visited: September 2005. Version used available at:
www.cs.tcd.ie/~osulldps/thesis/f.owl.
- Wache et al. 2001 Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H., Hübner S., "Ontology-based Integration of Information - A Survey of Existing Approaches," In: Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, 2001, Volume pages 108-117.

Wiederhold 1998	Wiederhold G., "An algebra for ontology composition", Workshop on Formal Methods, pages 56-61, US Naval Postgraduate School, Monterey, CA, 1994.
WordNet 2005	"The WordNet database", Available at: http://wordnet.princeton.edu/ , Last visited: August 2005.
WSDL 2001	"Web Services Description Language (WSDL) 1.1", W3C, March 2001, Available at: http://www.w3.org/TR/wsdl , Last visited: September 2005.
WSMO 2005	"Web Service Modeling Ontology (WSMO)", Submitted to W3C, June 2005, Available at: http://www.w3.org/Submission/WSMO/ , Last visited: September 2005.
XML 2000	"Extensible Markup Language (XML) 1.0 (Second Edition)", W3C, October 2000, Available at: http://www.w3.org/TR/2000/REC-xml-20001006 , Last visited: September 2005.
XMLSchema 2004	"XML Schema Part 0: Primer Second Edition", W3C, October 2004, Available at: http://www.w3.org/TR/xmlschema-0/ , Last visited: September 2005.
XPath 2005	"XPath", W3C, Available at: http://www.w3.org/TR/xpath , Last visited: September 2005.
XQuery 2005	"XQuery", W3C, Available at: http://www.w3.org/XML/Query/ , Last visited: September 2005.
XQueryX 2003	"XML Syntax for XQuery 1.0 (XQueryX)", W3C, December 2003, Available at: http://www.w3.org/TR/2003/WD-xqueryx-20031219/ , Last visited: September 2005.
XSLT 2005	"The Extensible Stylesheet Family", W3C, Available at: http://www.w3.org/Style/XSL/ , Last visited: September 2005.

- XUpdate 2000 "XUpdate -XML Update Language", XML:DB, September 2000, Available at:
<http://xmldb-org.sourceforge.net/xupdate/xupdate-wd.html>,
Last visited: September 2005.
- Yatskevich 2003 Yatskevich M., "Preliminary Evaluation of Schema Matching Systems", Technical Report #DIT-02-028, University of Trento, May 2003.

GLOSSARY

Mapping Process	Process used to discover the correspondences between information elements in a set of schemas.
Matching Process	Process used to discover candidate correspondences between information elements in a set of schemas based on the application of some matching criteria.
Schema	Set of information elements connected by some structure.
Semantic Interoperability	Occurs where consistent semantic interpretation of the same information is achieved across systems.
Semantic Interpretation	Mapping between some structured subset of data and a model of some set of objects in a domain with respect to the intended meaning of those objects and relationships between those objects.
Ontology	A formal, explicit specification of a shared conceptualization.
Conceptualization	Abstract model of how people think about things in the world from a particular perspective.
Explicit Specification	Assigning names and definitions to concepts or relations.
Committed Mapping	Represents a correspondence between elements in a set of ontologies that has been verified by a human.
Candidate Match	A match resulting from a matching process that is potentially a candidate mapping correspondence.

APPENDICES

The appendices present support information for the thesis.

- Appendix A provides an overview of the technology used in the thesis.
- Appendix B presents a UML specification for the OISIN process.
- Appendix C includes the XML DTDs used in the implemented information model.
- Appendix D provides supporting information for the *Decide To Map* experiment.
- Appendix E includes supporting information for the *Committed Mapping* experiment.
- Appendix F presents supporting information for the OisinSMU trial.

Appendix A. Technologies

This appendix presents the technologies used in the implementation of OISIN. The XQuery Engine IPSI-XQ is presented in section one. The Jena ontology environment used in transforming an ontology into canonical format is presented in section two. A description of WordNet and JWNL used for lexical analysis is used in section three. Finally, the online semantic web search engine SWOOGLE is presented in section four.

1. IPSI-XQ

The following description is adapted from the IPSI-XQ documentation (IPSI 2005). The IPSI-XQ engine has been developed by the Fraunhofer Institute, in Germany. It is implemented in Java and available at no cost for non-commercial usage. It comes with several user interfaces: a graphical, a command line and web interface. It includes a Java Application Programmers Interface for integration with other applications.

The IPSI-XQ engine uses three kinds of input: query expression, optional input/output types, and one or more input documents. As shown in Figure A-1, processing a query involves two major phases: Query Analysis and Query Evaluation.

During Query Analysis, first the XQuery Expression and the (optional) Input and Output Type Expressions are parsed to produce an `xQuery OperatorTree` and `xQuery Type Tree`. Then the `xQuery Normalizer` expands the operator tree into an `xQuery Core Operator Tree`. Guided by "type-inference-rules", the `Type Inference` component walks down the core operator tree until it encounters nodes with a declared input type to produce the `Result Type Tree` of the query. When an output type is explicitly declared, the `Type Checking` component checks whether the inferred result type is subsumed by the declared output type, by comparing the tree-automata for the two types. If not, a `Static Error` is raised. Otherwise, the `xQuery Compiler` prepares the core operator tree for evaluation by generating a query-plan consisting of Datamodel method calls and basic Functions & Operators.

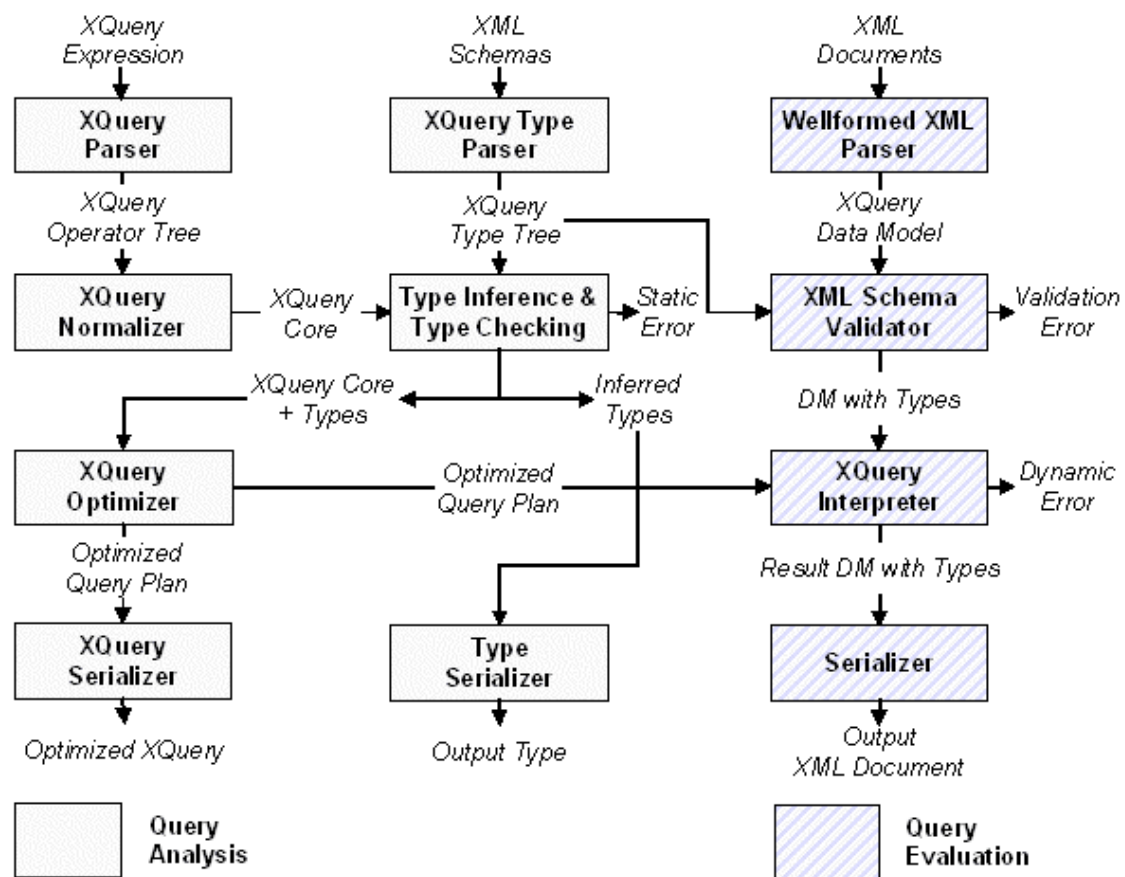


Figure A-1: Overview of IPSI-XQ Execution

As a first step during Query Evaluation the input XML documents are processed using Xerces to instantiate a Datamodel tree. Then the xQuery Type Validator validates the Datamodel tree against the declared input types, and adorns the Datamodel nodes with Types. If validation fails, a validation Error is raised. Otherwise, the xQuery Processor evaluates the query-plan to produce the Result tree. Explicitly declared and built-in runtime type checks can lead to a Dynamic Error.

Finally, the Serializer produces the XML Output Document, and the Type Tree Serializer produces the xQuery Result Type.

IPSI-XQ version 1.3.2 is used as the XQuery execution engine in the OISIN tools. The general pattern of usage in our implementation is that an XQuery query is read from a file into a string. This string is then passed to the XQuery engine via the java API for execution. The results of the query are returned in the form of a Document Object Model (DOM). The resulting DOM is navigated and written out to an XML file.

2. Jena

The following description is drawn from the Jena documentation (Jena 2005). Jena is an open source Java framework developed by Hewlett Packard for building Semantic Web applications. It provides a programming environment for working with RDF, RDFS and OWL files, and includes a rule-based inference engine. The ontology API is language-neutral with each language having a profile that lists the permitted constructs and the URI's of the classes and properties. Thus in the DAML profile, the URI for object property is `daml:ObjectProperty`, in the OWL profile is `owl:ObjectProperty` and in the RDFS profile it is `null` since RDFS does not define object properties. The profile is bound to an ontology model, which is an extended version of Jena's `Model` class. The general `Model` allows access to the statements in a collection of RDF data. `OntModel` extends this by adding support for the kinds of objects expected to be in the ontology: classes (in a class hierarchy), properties (in a property hierarchy) and individuals.

The properties defined in the ontology language, map to accessor methods. When a method is called, the information is retrieved from the underlying RDF statements. All of the classes in the ontology API that represent ontology values have `OntResource` as a common super-class. The properties of `OntResources` can be interrogated using a common pattern of accessor methods as shown in Table A-1. Thus, ontology classes and ontology properties are represented as subclasses of `OntResource`, specifically `OntClass` and `OntProperty`, and so can be examined using the same pattern of accessor methods.

Method	Effect
<code>add<property></code>	Add an additional value for the given property.
<code>set<property></code>	Remove any existing values for the property, then add the given value.
<code>list<property></code>	Return an iterator ranging over the values of the property.
<code>get<property></code>	Return the value for the given property, if the resource has one. If not, return null. If it has more than one value, an arbitrary selection is made.
<code>has<property></code>	Return true if there is at least one value for the given property. Depending on the name of the property, this sometimes is a <code><property></code> .
<code>remove<property></code>	Removes a given value from the values of the property on this resource. Has no effect if the resource does not have that value.

Table A-1: Jena accessor methods

The statements that the ontology Java objects see depend on both the asserted statements in the underlying RDF graph, and the statements that can be inferred by the reasoner being used (if any). As seen in Figure A-2, the asserted statements are held in the base graph. The reasoner, or inference engine, can use the contents of the base graph and the semantic rules of the language, to show a more complete set of statements - i.e. including those that are entailed by the base assertions. The same Graph interface is used at both levels, such that models can be built with no reasoner, or with one of a variety of different reasoners, without changing the ontology model.

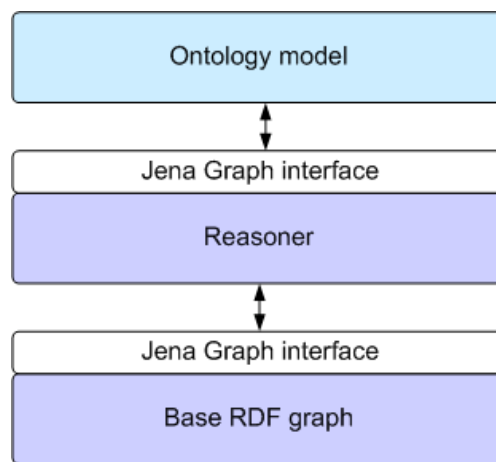


Figure A-2: Jena model decomposition

Jena version 2.1 is used in the Oisindo tool to load in the ontology and to write it out according to an internal canonical model, as discussed in chapter four. Jena was chosen due to its ability to cope with several ontology formats and its ability to allow use of different ontology reasoners.

3. Wordnet/JWNL

The following description of WordNet is adapted from the WordNet documentation (WordNet 2005). WordNet is an online lexical reference system, developed by Princeton University Cognitive Science Laboratory. English words are stored in different files called lexicographer files that organise nouns, verbs, adverbs and adjectives into groups of synonyms, each representing one underlying lexical concept.

For example the noun “mapping” has two senses (accepted meaning of the word):

1. mapping, map, correspondence -- (a function such that for every element of one set there is a unique element of another set)
2. mapping, chromosome mapping -- ((genetics) the process of locating genes on a chromosome)

The verb “map” has six senses

1. map -- (make a map of; show or establish the features of details of; "map the surface of Venus")
2. map -- (explore or survey for the purpose of making a map; "We haven't even begun to map the many galaxies that we know exist")
3. map -- (locate within a specific region of a chromosome in relation to known DNA or gene sequences; "map the genes")
4. map, map out -- (plan, delineate, or arrange in detail; "map one's future")
5. map -- (depict as if on a map; "sorrow was mapped on the mother's face")
6. map, represent -- (to establish a mapping (of mathematical elements or sets))

Different relations link the synonym sets and these relationships are implemented via pointers between synonym sets in the files. Many different semantic relationships are implemented. Two such examples are: hypernymy/hyponymy: describes the more general/more specific relationship between words; and homonyms: where words are pronounced or spelled the same way but have different meanings.

For example the hypernyms for each sense of the word “mapping” is:

Sense 1

mapping, map, correspondence -- (a function such that for every element of one set there is a unique element of another set)

=> function, mathematical function -- (a mathematical relation such that each element of one set is associated with at least one element of another set)

=> mathematical relation -- (a relation between mathematical expressions (such as equality or inequality))

=> relation -- (an abstraction belonging to or characteristic of two entities or parts together)

=> abstraction -- (a general concept formed by extracting common features from specific examples)

Sense 2

mapping, chromosome mapping -- ((genetics) the process of locating genes on a chromosome)

=> procedure, process -- (a particular course of action intended to achieve a result; "the procedure of obtaining a driver's license"; "it was a process of trial and error")

=> activity -- (any specific activity; "they avoided all recreational activity")

=> act, human action, human activity -- (something that people do or cause to happen)

The WordNet database is available for Windows and Unix environments and can be accessed online or downloaded as a package.

WordNet 2.0 is used in the OisinIO application to lexically analyse the terms in an ontology (see Section 5.1.5). WordNet was chosen due to its extensive database of approximately one hundred and fifty thousand unique terms and its ability to quickly return an analysis of any individual term. Access to WordNet is achieved through the Java WordNet Library (JWNL) API. JWNL is an open source project that is designed to work with any dictionary that uses the WordNet architecture (JWNL 2005).

4. SWOOGLE

SWOOGLE is under development by the UMBC university in Maryland, USA (SWOOGLE 2005). SWOOGLE is intended to be the "google for the semantic web" enabling software agents to locate and use semantic web documents (SWD). An SWD is an online document written in a semantic web language such as RDF, RDFS or OWL. A wide range of queries can be posed over the data gathered, related to the ontologies themselves (e.g. which ontologies have the concept "professor" defined?), the instances defined (e.g. get instances of class professor in TCD), as well as characterising the semantic web itself (e.g. what is most popular ontology?).

Figure A-3 illustrates the SWD discovery phase of the SWOOGLE architecture:

- A Google Crawler uses Google to find URLs of websites that potentially host ontology files.
- A Focused Crawler crawls through HTML files recursively within the given website (SWD Reader).
- A SWD Crawler crawls through the discovered SWDs and discovers more URLs according to term semantics (IR analyzer).

The metadata creation phase generates metadata for the ontology discovered, such as the ratio of ontology data to instance data in the ontology and the interrelationships of this ontology with other ontologies. This metadata is analysed in the data analysis phase and the ontologies are ranked.

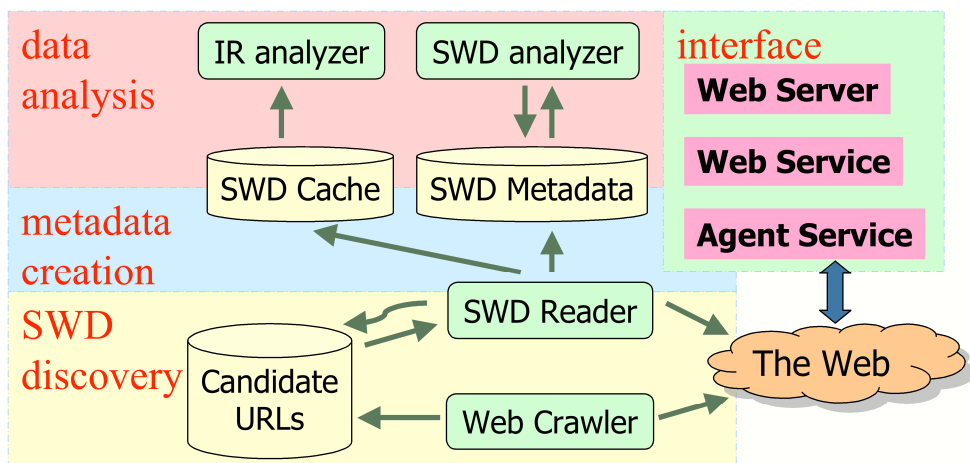


Figure A-3: SWD Discovery phase of SWOOGLE architecture

A version of the SWOOGLE database was downloaded for use in the research. In the OisínIO application, this database is queried (as described in Section 5.1.4) in order to provide a partial indication of the quality of an ontology.

Appendix B. UML specification of OISIN ontology mapping process

In this appendix, the OISIN ontology mapping process is presented as a UML specification. In section one, a Use Case diagram for the process is presented. Section two describes activity diagrams and activity descriptions for the process. Finally, section three presents an extensible UML class diagram that describes the core entities involved in the process.

1. Use Case Diagram

In this section the use cases for the OISIN ontology mapping process are discussed. Each use case is briefly described, and the activities that are expected to fulfil them are referenced.

The actors/roles for the process are presented in Table B-1.

Actors/Roles	
Software Agent	This actor represents an external software application or agent that requires ontology mapping tasks to be undertaken.
Researcher	This role is played if the ontologies for the mapping task need to be discovered.
Ontology Engineer	This role is undertaken by someone with experience of working with ontologies.
Manager	This role is played by someone with responsibility for mapping management and maintenance.
Intepreter	This actor is required if the mapping needs to be manually interpreted.

Table B-1: Actors/Roles for OISIN

The use case diagram for the process is presented in Figure B-1 and each use case is described now.

The goal of the **Find Ontologies** use case is to search for ontologies that fulfil certain requirements specified by the organisation. This use case is optional as a set of

ontologies could simply be presented to the system for mapping rather than having to be discovered. If required however, the scope of this search could be local to the organisation, or external to the organisation in ontology repositories or on the internet. The use case is initiated by the actor, based on the needs of the organisation. The use case is fulfilled through the *Discover Ontologies* activity of the characterisation phase of OISIN.

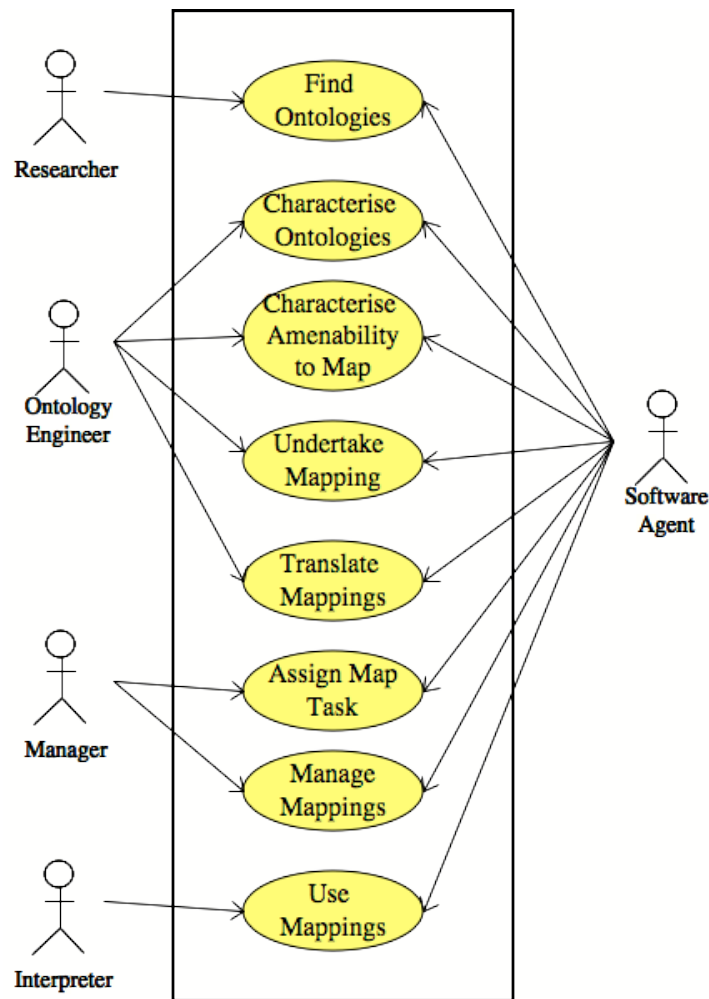


Figure B-1: Use Case Diagram

The goal of the **Characterise Ontologies** use case is to analyse a set of ontologies with respect to certain characteristics. The characteristics examined will vary from organisation to organisation but the aim of the characterisation is to discover the information necessary to support the *Characterise Amenability to Map* use case. The use case is initiated by the actor, based on a request coming from the organisation. The use case is fulfilled through the *Characterise Semantics* and *Characterise Modelling* activities of the characterisation phase of OISIN.

The goal of the **Characterise Amenability to Map** use case is to determine the difficulty that may be involved in undertaking a mapping between a set of ontologies. The information on which the analysis is based is dependent on the organisation concerned and so may or may not require the *Characterise Ontologies* use case to have been previously undertaken. However, it would be expected that some form of ontology analysis information would be available. The use case is initiated by the actor, based on a request coming from the organisation. The use case is fulfilled through the *Decide to Match, Select and Apply Match Algorithms* and *Characterise Amenability* activities of the characterisation phase of OISIN.

The goal of the **Assign Map Task** use case is to allocate a task of mapping between a set of ontologies to some actor(s). In some organisations initiation of the mapping task may depend upon the successful outcome of applying organisational criteria to the amenability analysis for the ontologies set involved. Thus a possible outcome is a failure response stating the reason for not undertaking the mapping. However, in some organisations this step may not be necessary. In addition, the assign map task is an optional use case depending on deployment within an organisation, as the undertake mapping task may simply be executed by the same actor that undertook the other uses cases for a particular set of ontologies. Where it exists, the use case is initiated by the actor, based on a request coming from the organisation. The use case is fulfilled through the *Decide to Map* activity of the mapping phase of OISIN.

The goal of the **Undertake Mapping** use case is to determine committed mapping information for a set of ontologies. The use case is initiated by the actor, based on a request coming from the organisation. The use case is fulfilled through the *Identify Committed Mappings* and *Undertake Commitment Analysis* activities of the mapping phase of OISIN.

The goal of the **Translate Mappings** use case is to transform the mapping information for a set of ontologies from one format to another. The use case is initiated by the actor, based on a request coming from the organisation. The use case is fulfilled through the *Render Mappings* activity of the execution phase of OISIN.

The goal of the **Manage Mappings** use case is to manage and maintain the generated mapping information over their lifetime. The use case is initiated by the actor, based on a request coming from the organisation. The use case is fulfilled through the *Integrate mappings*, *Alter mappings* and *Share mapping* activities of the management phase of OISIN.

The goal of the **Use Mappings** use case is to employ the mappings to achieve some transformation between information expressed using terms from different ontologies, that is to achieve semantic interoperability. This may involve manual or runtime transformation of documents depending on the organisational deployment. The use case is initiated by the actor. The use case is fulfilled through the *Interpret activity* of the execution phase of OISIN.

2. Activity Diagrams

In this section the OISIN ontology mapping process is described through a series of UML Activity Diagrams representing the different phases of the process. These activity diagrams provide an indication of the sequencing of the activities and the entities that are involved. However, **it is intended that these activity diagrams are indicative rather than prescriptive**, as different organisations may exhibit different needs. For example, how and when signals are created is highly dependent on the organisational situation in which the mapping process is deployed. For example, in an organisation that is running some form of portal website, the *discover ontologies* signal is likely to be created periodically and automatically. Whereas, in a situation involving a merger of organisations, the creation of the *discover ontologies* signal is likely to be once off.

Each activity in an activity diagram is described in terms of a brief description, characteristics of execution, input, output, preconditions, actors and behaviour governance. The behaviour governance field provides an indication of the entity that can be used to adapt the behaviour of the activity. The entities and signals are described in section three. The parameter “set_id” is used as a means of identifying a particular set of ontologies that are the subject of mapping.

2.1 Characterisation Phase

In Figure B-2 the activity diagram for the characterisation phase is presented.

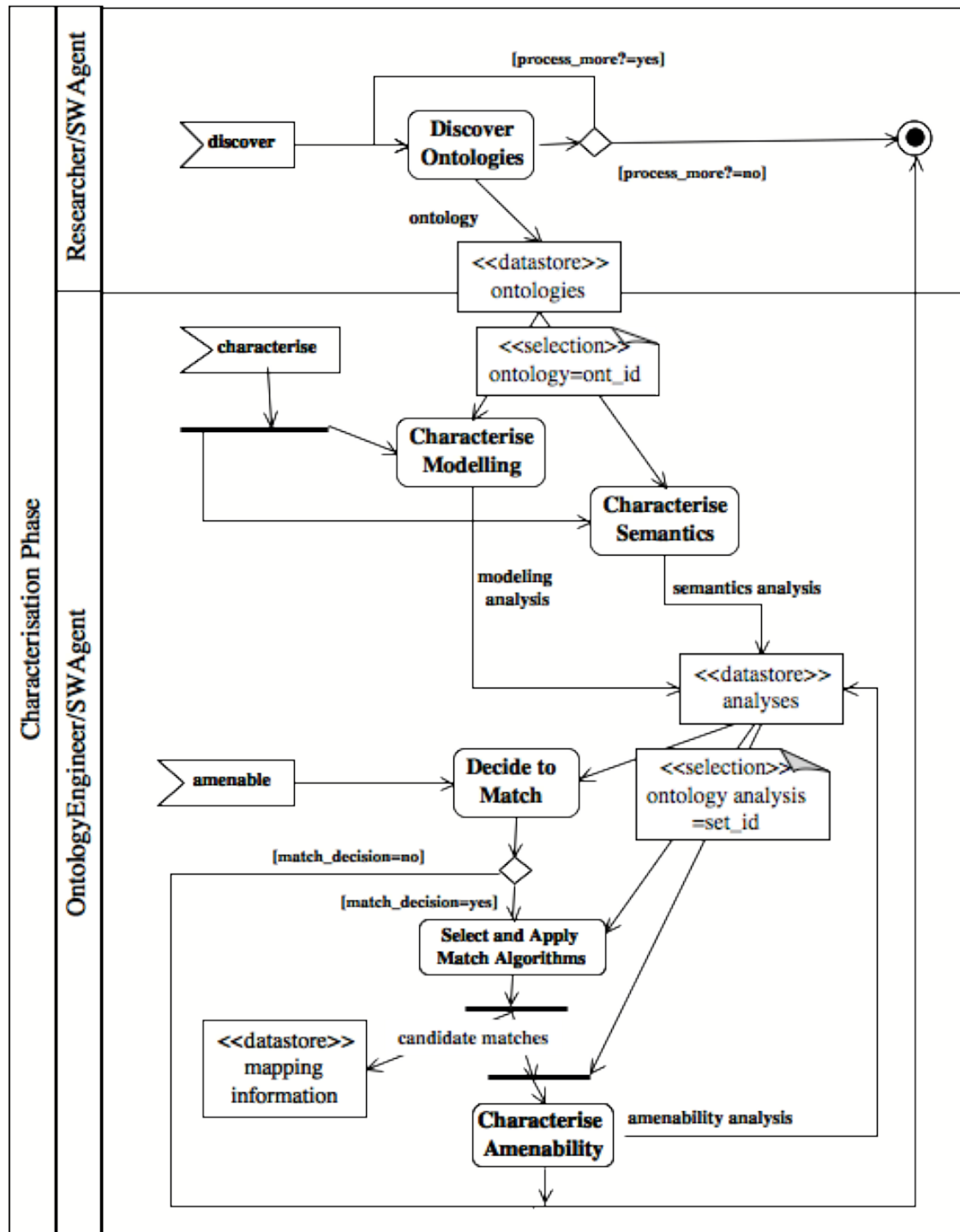


Figure B-2: Activity Diagram for Characterisation Phase

The activities of the activity diagram are overviewed in the following tables.

Discover Ontologies	
Description	This activity discovers ontologies that will be subject to mapping
Characteristics	Optional execution depending on situation
Input	goal_description
Output	a number of ontology entities or failure signal
Preconditions	signal to discover exists
Actors	Researcher or Software Agent
Behaviour Governance	acceptance_criteria

Characterise Modelling	
Description	This activity analyses the quality of an ontology and the modelling approach used
Characteristics	Executed once for an ontology
Input	ontology
Output	modelling analysis
Preconditions	signal to characterise exists
Actors	Ontology Engineer or Software Agent
Behaviour Governance	modelling_directions

Characterise Semantics	
Description	This activity analyses the content of the ontology in order to characterise the nature of the terms used in the modelling
Characteristics	Executed once for an ontology
Input	ontology
Output	semantics analysis
Preconditions	signal to characterise exists
Actors	Ontology Engineer or Software Agent
Behaviour Governance	semantics_directions

Decide to Match	
Description	This activity decides whether matching should be attempted between ontologies
Characteristics	Executed once for a set of ontologies
Input	ontology analyses
Output	match_decision_event
Preconditions	ontology analyses exist
Actors	Ontology Engineer or Software Agent
Behaviour Governance	match_decision_criteria

Activity descriptions continued on next page

Select and Apply Match Algorithms	
Description	This activity selects and executes the matching algorithms upon the ontologies
Characteristics	Possibly executed several times because criteria for algorithm selection may be altered to yield better results
Inputs	ontologies, ontology analyses
Outputs	candidate matches
Preconditions	match_decision_event is yes
Actors	Ontology Engineer or Software Agent
Behaviour Governance	algo_selection_criteria

Characterise Amenability	
Description	This activity selects information from the analyses generated in previous activities in order to characterise amenability of the ontologies for mapping
Characteristics	Possibly executed several times if amenability directions relating to what information to gather changes
Inputs	ontology analyses, candidate matches
Outputs	amenability analysis
Preconditions	input exists
Actors	Ontology Engineer or Software Agent
Behaviour Governance	amenability_directions

Table B-2: Activity Descriptions for Characterisation Phase

2.2 Mapping Phase

In Figure B-3 the activity diagram for the mapping phase is presented.

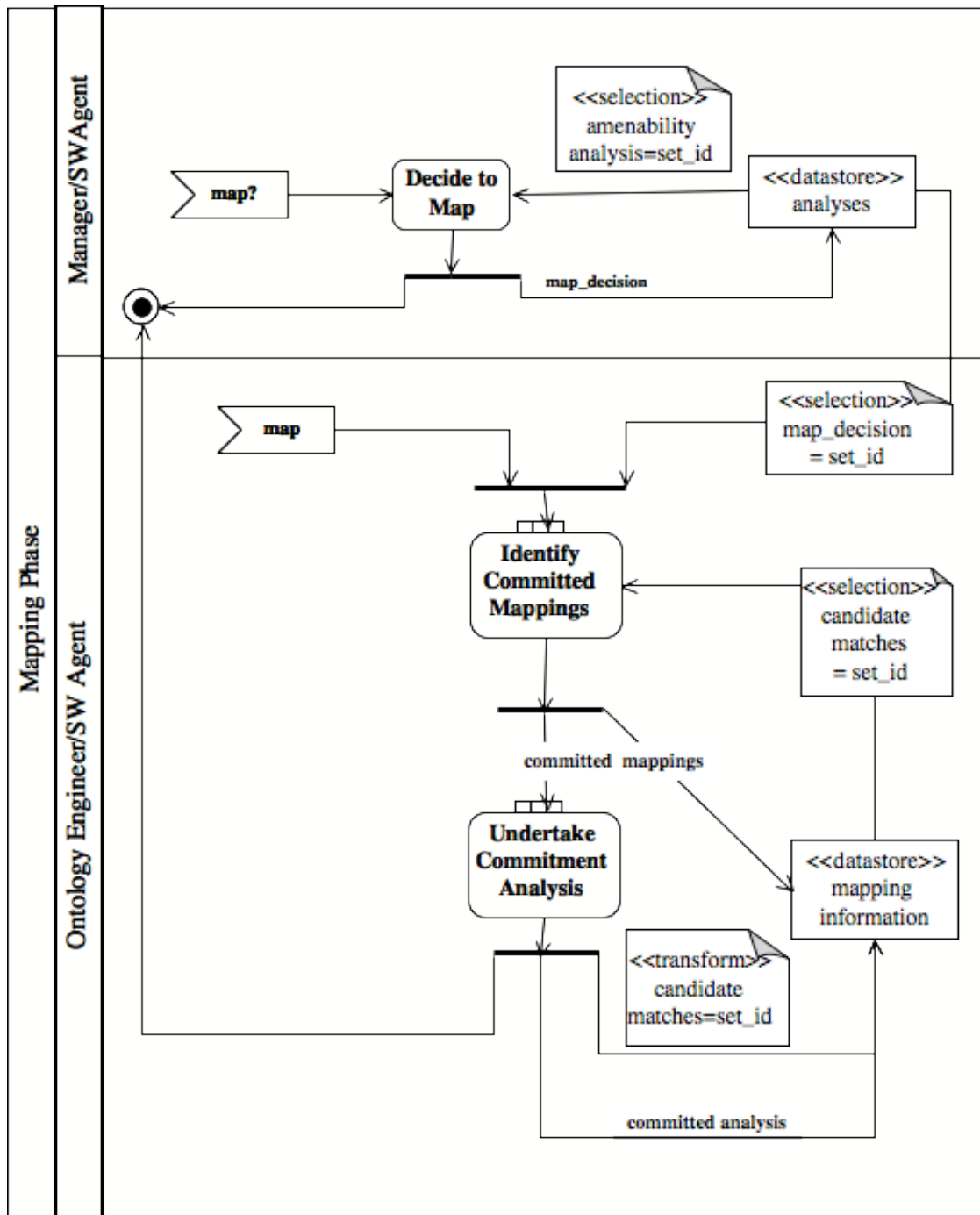


Figure B-3: Activity Diagram for Mapping Phase

The activities of the activity diagram are overviewed in the following tables.

Decide to Map	
Description	This activity determines the difficulty in mapping between the set of ontologies
Characteristics	Executed once for a set of ontologies in a queue
Input	amenability_analysis
Output	map_decision_event
Preconditions	map? signal exists
Actors	Manager or Software Agent
Behaviour Governance	map_decision_criteria

Identify Committed Mappings	
Description	This activity identifies committed mappings between a set of ontologies
Characteristics	Could be executed several times over a given set of ontologies. Sets of ontologies in queue
Input	ontologies, candidate matches
Output	committed mappings
Preconditions	map_decision is yes and map signal exists
Actors	Ontology Engineer or Software Agent
Behaviour Governance	mapping_detection_guidance

Undertake Commitment Analysis	
Description	This activity analyses the matching information in light of the committed mappings
Characteristics	Executed once over a set of committed mappings. Each set from a queue
Input	candidate matches, committed mappings
Output	committed analysis, transformed candidate matches
Preconditions	none
Actors	Ontology Engineer or Software Agent
Behaviour Governance	match_alteration_strategy

Table B-3: Activity Descriptions for Mapping Phase

2.3 Execution Phase

In Figure B-4 the activity diagram for the execution phase is presented.

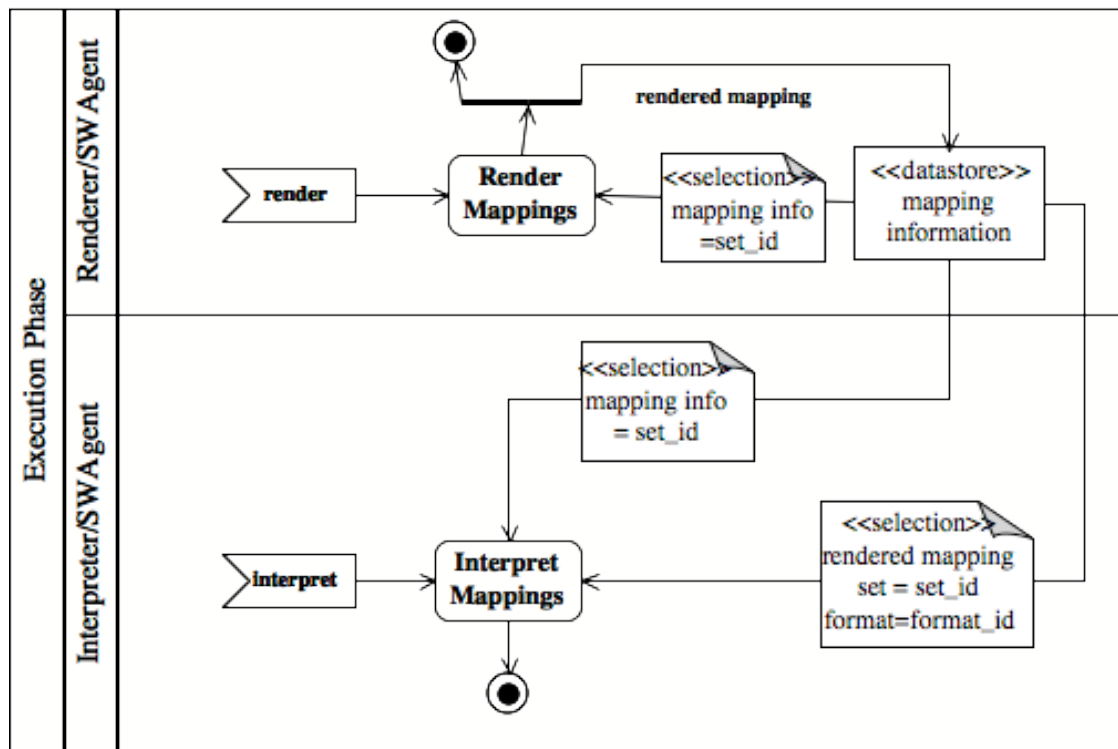


Figure B-4: Activity Diagram for Execution Phase

The activities of the activity diagram are overviewed in the following tables.

Render Mappings	
Description	This activity renders mapping information into different mapping formats
Characteristics	Executed once over a set of committed mappings
Input	mapping information, required_format
Output	format_specific_mapping
Preconditions	signal to render exists
Actors	Ontology Engineer or Software Agent
Behaviour Governance	formatting_directions

Interpret Mappings	
Description	This activity interprets the mapping information to enable semantic interoperability between applications or people
Characteristics	Can be executed several times over any given set of mappings
Input	mapping information or format_specific_mapping
Output	format_specific_mapping
Preconditions	signal to interpret exists
Actors	Interpreter or Software Agent
Behaviour Governance	none

Table B-4: Activity Descriptions for Execution Phase

2.4 Management Phase

In Figure B-5 the activity diagram for the management phase is presented.

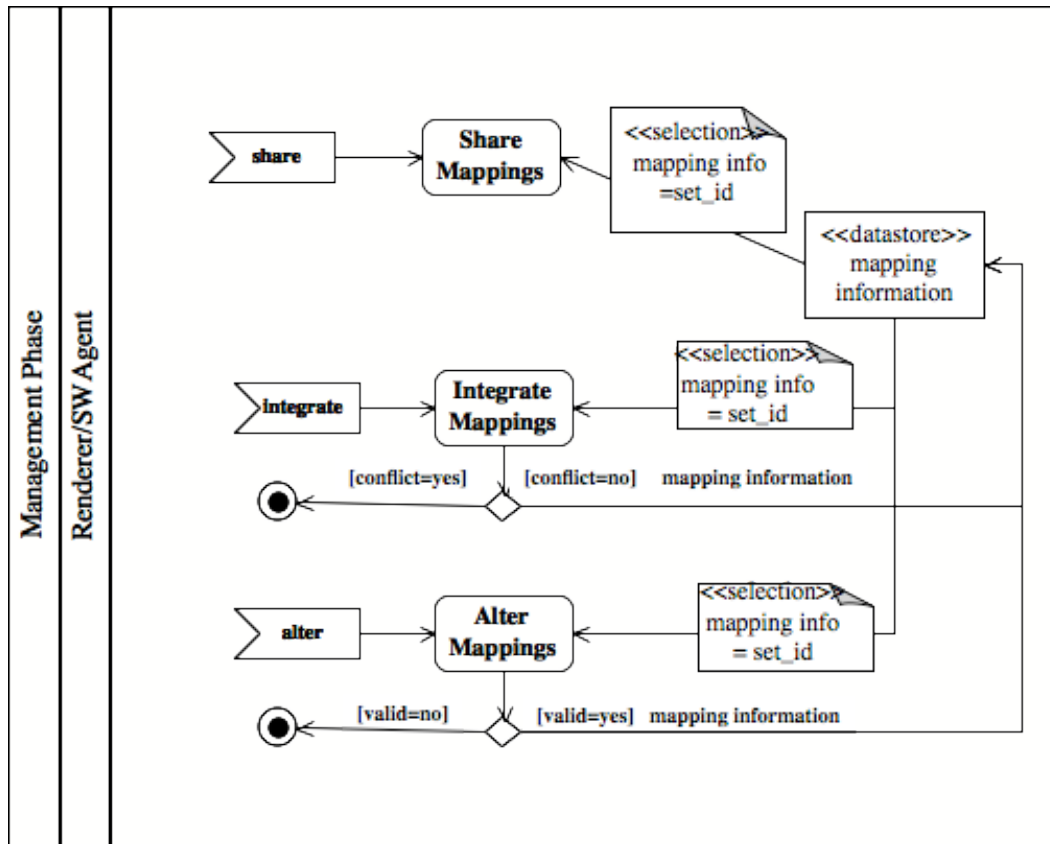


Figure B-5: Activity Diagram for Management Phase

Integrate Mappings	
Description	This activity attempts to integrate an externally generated mapping into the mapping set of the system
Characteristics	Executed once for each mapping to be integrated
Input	external mapping information or format specific mapping
Output	accepted or rejected
Preconditions	signal to integrate exists
Actors	Manager or Software Agent
Behaviour	conflict_directions
Governance	

Alter Mappings	
Description	This activity allows the alteration (including withdrawal) of a mapping, logs the change and disseminates a change event to other systems which have copies of the mapping
Characteristics	Executed as often as changes required
Input	list of systems that have copies of mapping, mappings
Output	change_event
Preconditions	signal to alter exists
Actors	Manager or Software Agent
Behaviour	alteration_authorisation_criteria
Governance	

Share Mappings	
Description	This activity shares a mapping or groups of mappings with other systems either proactively or in reaction to a request
Characteristics	Executed once for each mapping to be shared if proactive. Executed several times in reaction to requests.
Input	list of systems for sharing, mappings
Output	success or failure
Preconditions	signal to share exists
Actors	Manager or Software Agent
Behaviour	none
Governance	

Table B-5: Activity Descriptions for Management Phase

3. Extensible Information Model

The intention of this section is to provide an overview of the entities that are used in the OISIN process. The entity model has been made extensible, in some cases by not enumerating subclasses and in all cases by not including attribute/type information.

3.1 Entities

This section presents, in Figure B-6, a class diagram for the entities introduced in the activity diagrams of section two and then presents an overview of each of the entities.

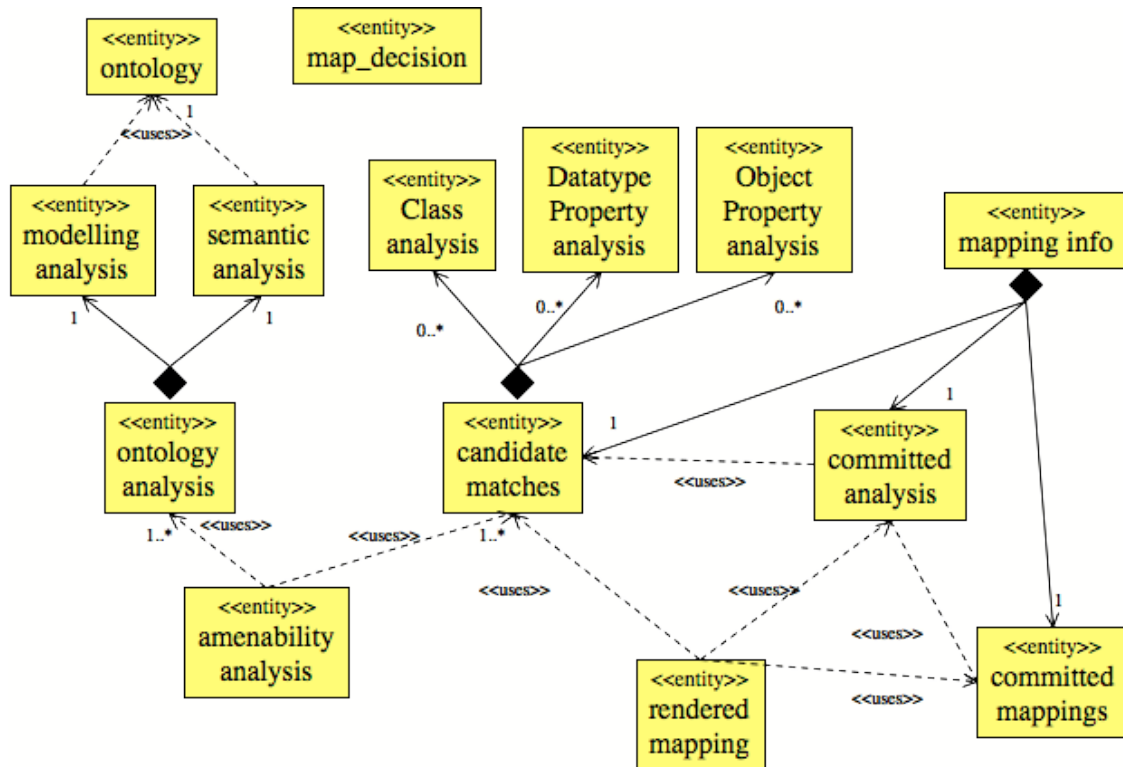


Figure B-6: UML Class Diagram of Entities

ontology

This entity is a canonical form representation of an ontology.

ontology analysis

This entity comprises the semantic analysis and modelling analysis for an ontology.

modelling analysis

This entity represents information about the quality of an ontology and the modelling approach used.

semantic analysis

This entity represents information about the nature of the terms used in the ontology.

candidate matches

This entity comprises the class match, datatype property and object property match analyses for an ontology with respect to the other ontologies in the set.

Class analysis

This entity represents information about identified matches for an ontology (with respect to other ontologies in the set) between classes based on a particular matcher algorithm.

Datatype Property analysis

This entity represents information about identified matches for an ontology (with respect to other ontologies in the set) between datatype properties based on a particular algorithm.

Object Property analysis

This entity represents information about identified matches for an ontology (with respect to other ontologies in the set) between object properties based on a particular matcher algorithm.

amenability analysis

This entity represents information that may be used in the determination of the difficulties that may exist in attempting to map between ontologies in a set.

committed mappings

This entity represents the committed mappings identified for an ontology with respect to the other ontologies in the set.

committed analysis

This entity presents information about candidate matches in light of the committed mappings.

mapping info

This entity comprises the candidate matches, committed mappings and committed mapping analysis for an ontology with respect to the other ontologies in the set.

rendered mapping

This entity represents the mapping information rendered into a particular mapping language format.

map_decision

This entity records the decision for to whether a mapping should proceed between a set of ontologies.

3.2 Behaviour Governance Entities

This section presents a class diagram for the Behaviour Governance entities in Figure B-7 and then presents an overview of each of the entities.

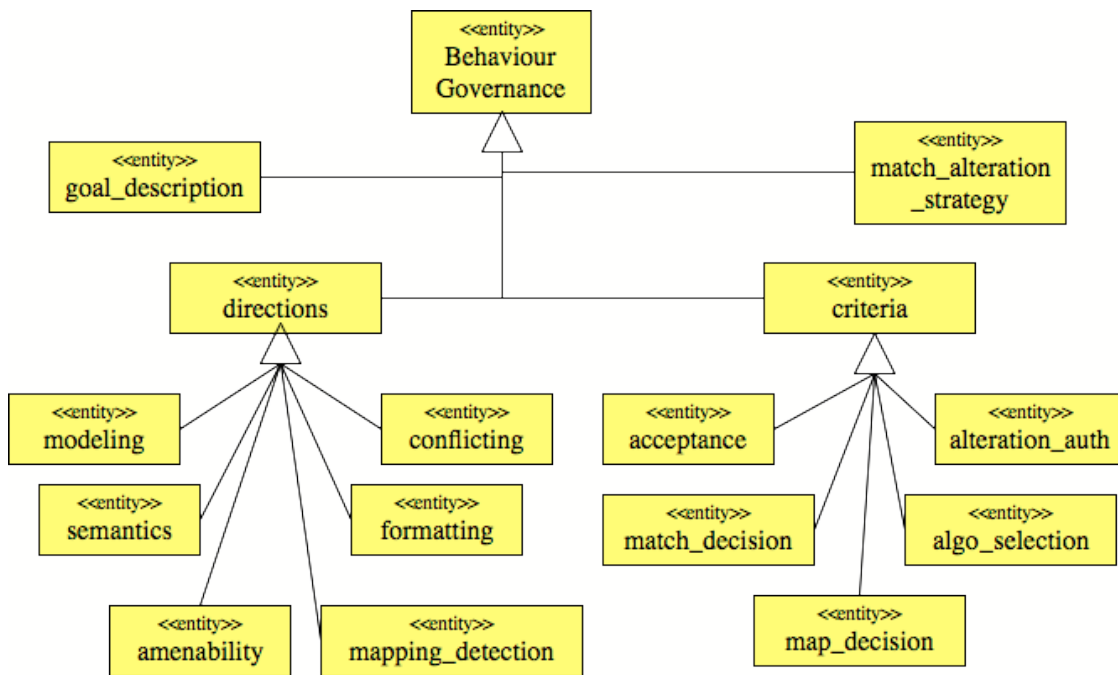


Figure B-7: UML Class Diagram for Behaviour Governance Entities

goal_description

This entity provides an indication of the scope for the discovery of ontologies activity. Depending on the situation, this can be done by constraining the locations to examine, the ontology file types to search for, and so on.

acceptance_criteria

This entity provides an indication of the criteria to be applied in accepting an ontology into the result set. Examples include, the terms/concepts that must or must not be present, and the currency (time) of the ontology in terms of date created or date of last revision. How these criteria are expressed depends on the organisation and situation. For example the expression could range from simple enumerated string value pairs to rules/policies.

modelling_directions

This entity provides an indication of the facets of modelling of an ontology that should be analysed. A `quality` attribute might represent a set of quality metrics to be applied. A `style` attribute might indicate the depth of analysis that should be undertaken for examination of the style. A `dimensions` attribute might indicate the dimensions of the ontology that should be calculated. All three attributes could be expressed as

enumerated types with the precise enumeration sets depending on the organisation and the situation. For example, the quality set might be those quality metrics from Burton et al. 2003 {Lawfulness, Richness, Interpretability, Consistency, Clarity, Comprehensibility, Relevance, Accuracy, Authority, History}; the style set might be {low, medium, high}; and the dimensions set might be {basic-dimensions, relation_based-dimensions, comprehensive-dimensions}. An investigations attribute might indicate how the modelling directions should be undertaken. This could be some enumerated value indicating a fixed workflow to be undertaken or some rule/policy based expression to allow for more dynamic workflows.

semantics_directions

This entity provides an indication of how the contents of the ontology should be analysed from the perspective of the semantics of the terms used. One possible approach (used in the implementation) is to undertake a term complexity analysis. It is possible to apply many other Information Retrieval techniques. The investigations attribute would thus indicate which technique(s) should be applied. This could be some enumerated value indicating a fixed workflow to be undertaken or some rule/policy based expression to allow for more dynamic workflows.

match_decision_criteria

This entity indicates the criteria that should be applied in order to decide whether to attempt to match a set of ontologies or not. It is highly dependent on the attributes of the analyses entities as some of the criteria are likely to be defined in terms of those attributes. The type of the criteria attribute will depend on the specialisation of the process to a particular organisation and the form in which these criteria are likely to be captured and processed. The criteria expression could range from simple named pair values representation to rules/policies.

algo_selection_criteria

This entity indicates the criteria that should be used in order to determine which match algorithms should be applied during a matching problem and in what (if any) order. The criteria depends on the metadata that is available describing the match algorithms, and the relationship of that metadata to the ontology analysis entities that are available. The match algorithm metadata, analyses, and criteria are highly dependent on the organisational and situational deployment of the process. The criteria expression could range from simple named pair values representation to rules/policies.

amenability_directions

This entity, at a minimum, indicates (perhaps through a `summarise` attribute) which parts of the ontology analyses should be gathered as a basic amenability analysis. This could be as simple as a list of attributes from the ontology analysis entities that need to be gathered, or as complex as a set of queries to be applied upon the ontology analysis entities. An optional `investigations` attribute might indicate what further analysis/transformation needs to take place in order to turn the gathered information into an amenability analysis. The types and expressions used for the `summarise` and the `investigations` attributes are dependent on the organisational and situational deployment of the process.

map_decision_criteria

This entity indicates the criteria that should be applied in order to decide whether or not to attempt to map a set of ontologies. It is highly dependent on the attributes of the amenability analysis entity as some of the criteria are likely to be defined in terms of those attributes. The type of the `criteria` attribute will depend on the specialisation of the process to a particular organisation and the form in which these criteria are likely to be captured and processed. The criteria expression could range from simple named pair values representation to rules/policies.

mapping_detection_guidance

This entity provides guidance on how mappings should be detected. This can range from a directed workflow of candidate match investigation (as shown in the OISIN implementation) to the detection of patterns resulting in suggested potential mappings.

match_alteration_strategy

This entity indicates how the candidate match analysis should be changed in light of the identification of committed mappings. The expression language to be used will depend on whether simple or complex strategies are to be deployed. For example, one strategy would be to simply change an attribute in the candidate match, in which case something like `XUpdate` would be sufficient to express the strategy. A more complex strategy may involve some ontology reasoning in which case an expression language such as `SWRL` may be required. Note the expression language is used to specify the strategy and may be implemented in many different ways.

formatting_directions

This entity provides the directions needed to transform the mapping information into a particular mapping format, e.g. `OWL`, `SWRL`.

conflict_directions

This entity provides the directions needed to detect conflicts in mappings and to react to such conflicts. The rules/policies with respect to conflict detection and reaction is dependent on the attributes of the mapping information entities available.

alteration_authorisation_policies

This entity indicates the policies (in terms of event, condition, action) to be applied in how mappings are to be altered and by whom.

3.3 Signals and Events

Table B-6 presents the UML Signals and Table B-7 shows the UML Events used in the OISIN activity diagrams of section two.

Signals	
discover	used to indicate that discovery of ontologies should be undertaken
characterise	used to indicate that characterisation of an ontology needs to be undertaken
amenable	used to indicate that the request has been made to characterise the amenability of a set of ontologies for mapping.
map?	used to indicate that a request has been determine the mapping difficulty involved in mapping a set of ontologies
map	used to indicate that a mapping activity has been requested
render	used to indicate that a new mapping format is required
interpret	used to indicate that interpretation of mappings is required
share	used to indicate that sharing of mapping(s) is required
integrate	used to indicate that integration of new mapping(s) is required
alter	used to indicate that alteration of mapping(s) is required

Table B-6: Description of Signals used in Activity Diagrams

Events	
process_more?	Used to indicate whether there are more entities to be processed. Can be either yes or no.
match_decision	Used to indicate the decision with respect to whether matching should be attempted. Can be either yes or no.
conflict	Used to indicate whether a mapping that has been requested to be integrated into a mapping set is in conflict with one of the existing mappings. Can be either yes or no.
valid	Used to indicate that an alteration to a mapping is valid. Can be either yes or no.
success	Used to indicate that an activity has been undertaken successfully
failure	Used to indicate that an activity has failed

Table B-7: Description of Events used in Activity Diagrams

Appendix C. XML DTDs

1. ANC.dtd – Anchors/Committed Mappings

```
<!ELEMENT owl:Ontology ( #PCDATA ) >
<!ATTLIST owl:Ontology rdf:about CDATA #REQUIRED >

<!ELEMENT owl:equivalentClass ( #PCDATA ) >
<!ATTLIST owl:equivalentClass rdf:resource NMTOKEN #REQUIRED >

<!ELEMENT owl:equivalentProperty ( #PCDATA ) >
<!ATTLIST owl:equivalentProperty rdf:resource NMTOKEN #REQUIRED >

<!ELEMENT rdf:RDF ( owl:Ontology, su:Class+ ) >

<!ELEMENT su:Class ( owl:equivalentClass | su:DatatypeProperty |
su:Name | su:ObjectProperty | su:SubClass | su:SuperClass )* >

<!ELEMENT su:DatatypeProperty ( su:Name, owl:equivalentProperty*)
>
<!ELEMENT su:ObjectProperty ( su:Name, owl:equivalentProperty*) >

<!ELEMENT su:Name ( #PCDATA ) >
<!ELEMENT su:SubClass ( #PCDATA ) >
<!ELEMENT su:SuperClass ( #PCDATA ) >
```

2. CAN.dtd – Canonical Format

```
<!ELEMENT owl:Ontology ( #PCDATA ) >
<!ATTLIST owl:Ontology rdf:about CDATA #REQUIRED >

<!ELEMENT rdf:RDF ( owl:Ontology, su:Class+ ) >

<!ELEMENT su:Class ( su:DatatypeProperty | su:Name |
su:ObjectProperty | su:SubClass | su:SuperClass )* >

<!ELEMENT su:DatatypeProperty ( #PCDATA ) >
<!ELEMENT su:Name ( #PCDATA ) >
<!ELEMENT su:ObjectProperty ( #PCDATA ) >
<!ELEMENT su:SubClass ( #PCDATA ) >
<!ELEMENT su:SuperClass ( #PCDATA ) >
```

3. CMB.dtd – Candidate Matches

```
<!ELEMENT su:Complete ( su:Class+, su:DatatypeProperties,
su:ObjectProperties ) >

<!ELEMENT su:Class ( complete_match | partial_match |
su:DatatypeProperty | su:Name | su:ObjectProperty | su:SubClass |
su:SuperClass )* >

<!ELEMENT complete_match ( local, analysis+ ) >
<!ELEMENT partial_match ( local, analysis+ ) >
<!ELEMENT analysis ( match | target | target_classes | type )* >

<!ELEMENT su:DatatypeProperty ( #PCDATA | name | partial_match
)*>
```

```

<!ELEMENT su:Name ( #PCDATA ) >
<!ELEMENT su:ObjectProperty ( #PCDATA | complete_match | name |
partial_match )* >
<!ELEMENT su:SubClass ( #PCDATA ) >
<!ELEMENT su:SuperClass ( #PCDATA ) >

<!ELEMENT su:DatatypeProperties ( su:DatatypeProperty+ ) >
<!ELEMENT su:ObjectProperties ( su:ObjectProperty+ ) >

<!ELEMENT target_classes ( class* ) >

<!ELEMENT class ( #PCDATA ) >
<!ELEMENT local ( #PCDATA ) >
<!ELEMENT match ( #PCDATA ) >
<!ELEMENT name ( #PCDATA ) >
<!ELEMENT target ( #PCDATA ) >
<!ELEMENT type ( #PCDATA ) >

```

4. DIM.dtd – Dimensions Analysis

```

<!ELEMENT dimensions ( for_ontology, schema_dimensions, tree+ ) >
<!ELEMENT schema_dimensions ( num_concepts,
num_unique_attributes, num_unique_relationships,
min_num_atts_of_concept, avg_num_atts_of_concept,
max_num_atts_of_concept, min_num_rels_of_concept,
avg_num_rels_of_concept, max_num_rels_of_concept ) >
<!ELEMENT tree ( majority_height | max_height | max_width |
min_width | root | width )* >

<!ELEMENT avg_num_atts_of_concept ( #PCDATA ) >
<!ELEMENT avg_num_rels_of_concept ( #PCDATA ) >
<!ELEMENT for_ontology ( #PCDATA ) >
<!ELEMENT level ( number, width_level, percent_level ) >
<!ELEMENT majority_height ( #PCDATA ) >
<!ELEMENT max_height ( #PCDATA ) >
<!ELEMENT max_num_atts_of_concept ( #PCDATA ) >
<!ELEMENT max_num_rels_of_concept ( #PCDATA ) >
<!ELEMENT max_width ( #PCDATA ) >
<!ELEMENT min_num_atts_of_concept ( #PCDATA ) >
<!ELEMENT min_num_rels_of_concept ( #PCDATA ) >
<!ELEMENT min_width ( #PCDATA ) >
<!ELEMENT num_concepts ( #PCDATA ) >
<!ELEMENT num_unique_attributes ( #PCDATA ) >
<!ELEMENT num_unique_relationships ( #PCDATA ) >
<!ELEMENT number ( #PCDATA ) >
<!ELEMENT percent_level ( #PCDATA ) >
<!ELEMENT root ( #PCDATA ) >
<!ELEMENT width ( level* ) >
<!ELEMENT width_level ( #PCDATA ) >

```

5. DOM.dtd – Quality Analysis

```

<!ELEMENT domain_semantics ( for_ontology, uses, swoogle_analysis
) >
<!ELEMENT swoogle_analysis ( rank, used_by ) >
<!ELEMENT uses ( num_onts_imported, num_onts_referenced ) >
<!ELEMENT used_by ( #PCDATA ) >
<!ELEMENT for_ontology ( #PCDATA ) >
<!ELEMENT num_onts_imported ( #PCDATA ) >
<!ELEMENT num_onts_referenced ( #PCDATA ) >

```

```
<!ELEMENT rank ( #PCDATA ) >
```

6. INT.dtd - Candidate Matches plus Anchors

```
<!ELEMENT rdf:RDF ( owl:Ontology, su:Class+,
su:DatatypeProperties, su:ObjectProperties ) >

<!ELEMENT su:Class ( complete_match | owl:equivalentClass |
partial_match | su:DatatypeProperty | su:Name | su:ObjectProperty
| su:SubClass | su:SuperClass )* >

<!ELEMENT owl:Ontology ( #PCDATA ) >
<!ATTLIST owl:Ontology rdf:about CDATA #REQUIRED >
<!ELEMENT owl:equivalentClass ( #PCDATA ) >
<!ATTLIST owl:equivalentClass rdf:resource NMTOKEN #REQUIRED >
<!ELEMENT owl:equivalentProperty ( #PCDATA ) >
<!ATTLIST owl:equivalentProperty rdf:resource NMTOKEN #REQUIRED >

<!ELEMENT partial_match ( local, analysis+ ) >

<!ELEMENT analysis ( match | target | target_classes | type )* >
<!ELEMENT class ( #PCDATA ) >
<!ELEMENT complete_match ( local, analysis+ ) >
<!ELEMENT local ( #PCDATA ) >
<!ELEMENT match ( #PCDATA ) >
<!ELEMENT name ( #PCDATA ) >

<!ELEMENT su:DatatypeProperties ( su:DatatypeProperty+ ) >
<!ELEMENT su:DatatypeProperty ( name | partial_match | su:Name
owl:equivalentProperty)* >
<!ELEMENT su:Name ( #PCDATA ) >
<!ELEMENT su:ObjectProperties ( su:ObjectProperty+ ) >
<!ELEMENT su:ObjectProperty ( complete_match | name |
partial_match | su:Name | owl:equivalentProperty)* >

<!ELEMENT su:SubClass ( #PCDATA ) >
<!ELEMENT su:SuperClass ( #PCDATA ) >
<!ELEMENT target ( #PCDATA ) >
<!ELEMENT target_classes ( class* ) >
<!ELEMENT type ( #PCDATA ) >
```

7. LANG.dtd – Language Analysis

```
<!ELEMENT lang_usage ( keywords_used, percent_keywords_used ) >
<!ELEMENT keyword ( #PCDATA ) >
<!ELEMENT keywords_used ( keyword+ ) >
<!ELEMENT percent_keywords_used ( #PCDATA ) >
```

8. LEX.dtd – Term Analysis

```
<!ELEMENT lex_analysis ( term_analysis+ ) >
<!ELEMENT term_analysis ( term, type, pattern, token* ) >
<!ELEMENT pattern ( #PCDATA ) >
<!ELEMENT term ( #PCDATA ) >
<!ELEMENT token ( #PCDATA ) >
<!ELEMENT type ( #PCDATA ) >
```

9. MA.dtd – Mapping Analysis

```
<!ELEMENT overall_analysis ( source, target, schema_dimensions,
matches_analysis, anchors_analysis, anchor_paths_analysis ) >

<!ELEMENT schema_dimensions ( num_concepts,
num_unique_attributes, num_unique_relationships,
min_num_atts_of_concept, avg_num_atts_of_concept,
max_num_atts_of_concept, min_num_rels_of_concept,
avg_num_rels_of_concept, max_num_rels_of_concept ) >

<!ELEMENT matches_analysis ( complete_matches, partial_matches )>
<!ELEMENT complete_matches ( ClassNames, DataPropertyName,
ObjectPropertyName, PropertyName_ClassName,
ClassName_PropertyName, DataPropertyName_ObjectPropertyName,
ObjectPropertyName_DataPropertyName ) >

<!ELEMENT partial_matches ( ClassNames, DataPropertyName,
ObjectPropertyName, PropertyName_ClassName,
ClassName_PropertyName, DataPropertyName_ObjectPropertyName,
ObjectPropertyName_DataPropertyName ) >

<!ELEMENT anchors_analysis ( number_of_specified_anchors, anchor+
) >
<!ELEMENT anchor ( name, properties_analysis ) >
<!ELEMENT properties_analysis ( num_properties,
num_with_no_matches, num_complete_matches, num_partial_matches,
num_equivalent_properties, percent_equivalent,
num_disjoint_properties ) >

<!ELEMENT anchor_paths_analysis ( anchor_paths,
min_anchored_path_length, max_anchored_path_length,
anchored_paths_analysis, num_distinct_concepts_in_anchor_paths,
num_concepts_not_in_anchor_paths, anchors_not_on_paths,
structural_mismatches, number_of_valid_paths, valid_paths ) >
<!ELEMENT path ( nd+, length ) >
<!ELEMENT paths ( path+ ) >
<!ELEMENT anchor_paths ( paths+ ) >
<!ELEMENT anchored_paths_analysis ( a_path+ ) >
<!ELEMENT match ( kind, types? ) >
<!ELEMENT nd ( #PCDATA | match | name )* >
<!ELEMENT valid_path ( anchor_path, matches, length, nd+ ) >
<!ELEMENT valid_paths ( valid_path+ ) >

<!ELEMENT ClassName_PropertyName ( #PCDATA ) >
<!ELEMENT ClassNames ( #PCDATA ) >
<!ELEMENT DataPropertyName_ObjectPropertyName ( #PCDATA ) >
<!ELEMENT DataPropertyNames ( #PCDATA ) >
<!ELEMENT ObjectPropertyName_DataPropertyName ( #PCDATA ) >
<!ELEMENT ObjectPropertyNames ( #PCDATA ) >
<!ELEMENT PropertyName_ClassName ( #PCDATA ) >
<!ELEMENT a_path ( path_len, freq ) >
<!ELEMENT anchor_path ( #PCDATA ) >
<!ELEMENT anchors_not_on_paths ( #PCDATA ) >
<!ELEMENT avg_num_atts_of_concept ( #PCDATA ) >
<!ELEMENT avg_num_rels_of_concept ( #PCDATA ) >
<!ELEMENT freq ( #PCDATA ) >
<!ELEMENT kind ( #PCDATA ) >
<!ELEMENT length ( #PCDATA ) >
<!ELEMENT matches ( #PCDATA ) >
<!ELEMENT max_anchored_path_length ( #PCDATA ) >
<!ELEMENT max_num_atts_of_concept ( #PCDATA ) >
```

```

<!ELEMENT max_num_rels_of_concept ( #PCDATA ) >
<!ELEMENT min_anchored_path_length ( #PCDATA ) >
<!ELEMENT min_num_atts_of_concept ( #PCDATA ) >
<!ELEMENT min_num_rels_of_concept ( #PCDATA ) >
<!ELEMENT name ( #PCDATA ) >
<!ELEMENT num_complete_matches ( #PCDATA ) >
<!ELEMENT num_concepts ( #PCDATA ) >
<!ELEMENT num_concepts_not_in_anchor_paths ( #PCDATA ) >
<!ELEMENT num_disjoint_properties ( #PCDATA ) >
<!ELEMENT num_distinct_concepts_in_anchor_paths ( #PCDATA ) >
<!ELEMENT num_equivalent_properties ( #PCDATA ) >
<!ELEMENT num_partial_matches ( #PCDATA ) >
<!ELEMENT num_properties ( #PCDATA ) >
<!ELEMENT num_unique_attributes ( #PCDATA ) >
<!ELEMENT num_unique_relationships ( #PCDATA ) >
<!ELEMENT num_with_no_matches ( #PCDATA ) >
<!ELEMENT number_of_specified_anchors ( #PCDATA ) >
<!ELEMENT number_of_valid_paths ( #PCDATA ) >
<!ELEMENT path_len ( #PCDATA ) >
<!ELEMENT percent_equivalent ( #PCDATA ) >
<!ELEMENT source ( #PCDATA ) >
<!ELEMENT structural_mismatches ( #PCDATA ) >
<!ELEMENT target ( #PCDATA ) >
<!ELEMENT type ( #PCDATA ) >
<!ELEMENT types ( type ) >

```

10. MERLIN.dtd – WordNet analysis detail

```

<!ELEMENT wordnet_analysis ( compositeterm | term )* >
<!ELEMENT compositeterm ( name, type, unknown, term+ ) >
<!ELEMENT term ( name | synset | type | unknown )* >
<!ELEMENT synset ( pos, sense+ ) >
<!ELEMENT sense ( word+ ) >
<!ELEMENT name ( #PCDATA ) >
<!ELEMENT pos ( #PCDATA ) >
<!ELEMENT type ( #PCDATA ) >
<!ELEMENT unknown ( #PCDATA ) >
<!ELEMENT word ( #PCDATA ) >

```

11. WNET.dtd – WordNet analysis statistics

```

<!ELEMENT wordnet_stats ( known_terms, composite_terms,
unknown_terms ) >
<!ELEMENT composite_terms ( #PCDATA ) >
<!ELEMENT known_terms ( #PCDATA ) >
<!ELEMENT unknown_terms ( #PCDATA ) >

```

Appendix D. Decide to Map Experiment

1. Decision Sheet

Name of Evaluator:

Background

Please circle your choice for each of the following:

Previous experience of working with ontologies:	none	just theory	some	a lot
Previous experience of mapping between ontologies:	none	very little	some	a lot

Recommendations

Please circle your choice for each of the following:

	Comment			
Would you recommend proceeding to using a mapping capture tool for ontologies A and B ?	No	Probably Not	Possibly	Definitely
Would you recommend proceeding to using a mapping capture tool for ontologies C and T ?	No	Probably Not	Possibly	Definitely
Would you recommend proceeding to using a mapping capture tool for ontologies F and H ?	No	Probably Not	Possibly	Definitely
Would you recommend proceeding to using a mapping capture tool for ontologies P and E ?	No	Probably Not	Possibly	Definitely
Would you recommend proceeding to using a mapping capture tool for ontologies C and D ?	No	Probably Not	Possibly	Definitely

Task Questionnaire

Please circle your choice for each of the following:

How difficult was it to come to an overall decision for each pair of ontologies? Please explain	Difficult	Fairly Difficult	Fairly Easy	Easy
How confident are you in the decisions that you have made? Please explain	Not confident	Sort of Confident	Confident	Very Confident

2. Excel Spreadsheet for AB Mapping

Page 1

Syntax Analysis

a.owl b.owl

compiles?	yes	yes
% OWL keywords used	29	20
Number of Terms	122	188
% Terms which are Composite	77.05	54.26
Number of Hyphenated Composite Terms	94	0
Number of Underscored Composite Terms	0	3
Number of MiddleCapital Composite Terms	0	99

WORDNET Analysis

Terms known directly	28	77
Term Combinations where some constituent terms known	94	102
Unknown Terms	0	9
% of ontology whose terms are unknown	0	5

Ontology References Analysis

Number of Ontologies that it imports	0	1
Number of Ontologies that it references	5	3

Analysis using SWOOGLE

Rank	0.15	0.23
Ontologies which import	0	2
Ontologies which reference	1	4
Ontologies which extend	1	1
Ontologies which map	0	0

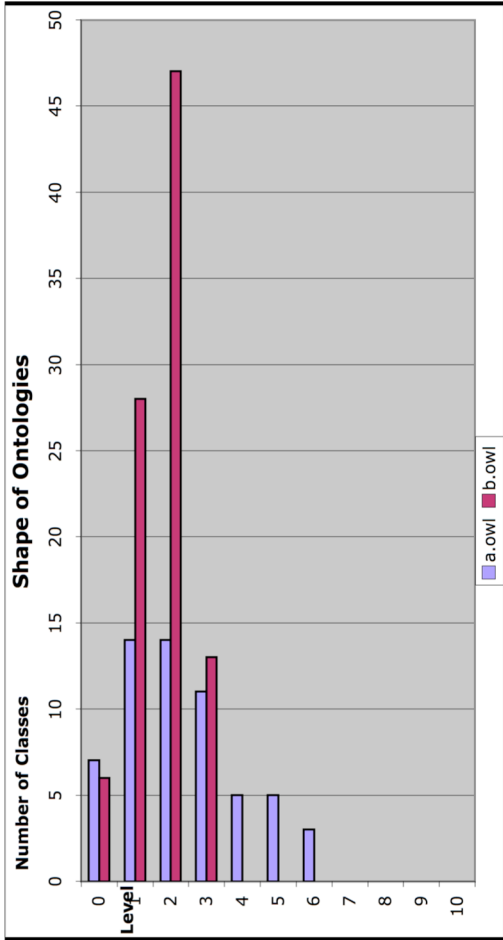
(Top Ten normally in range 0.2 to 0.5

Dimensions Analysis

	a.owl	b.owl
Number of Classes	56	96
Number of Unique Attributes	20	32
Number of Unique Relationships	46	60

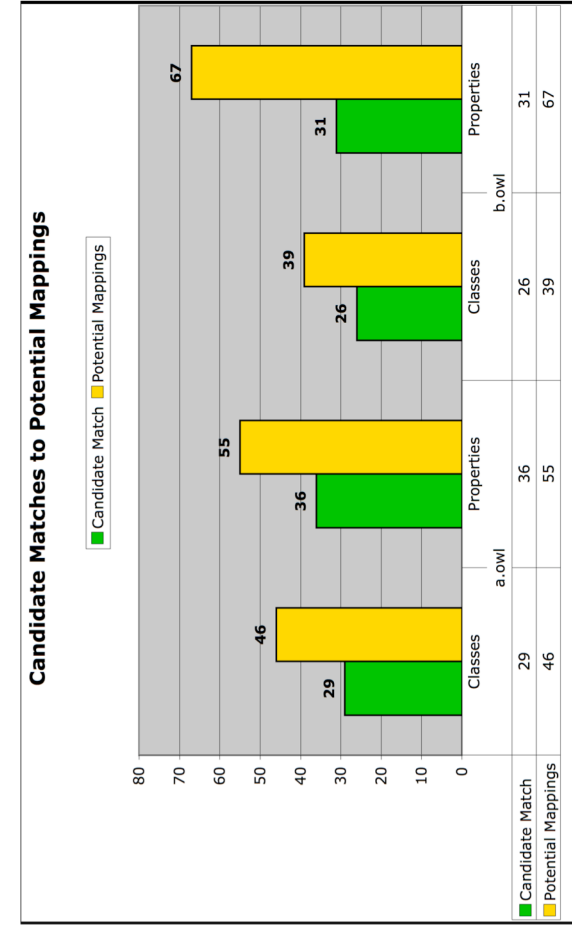
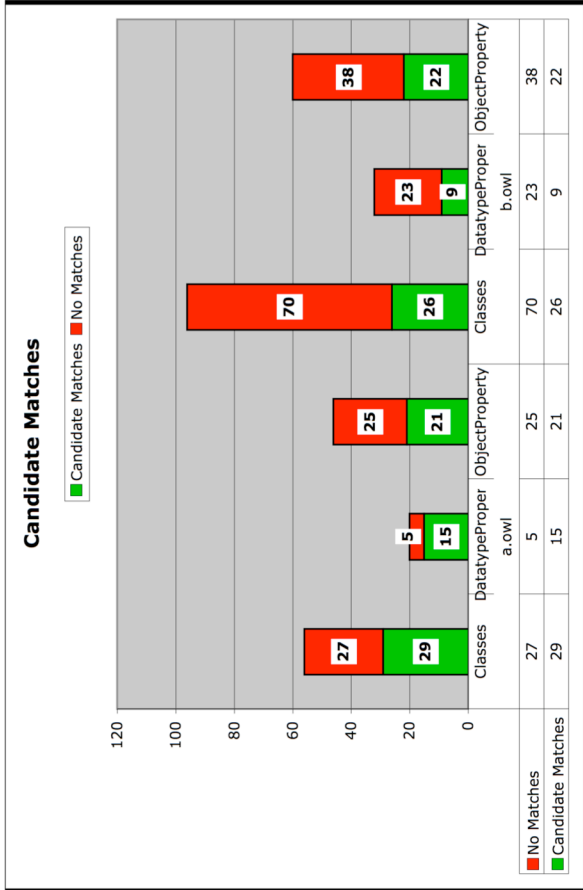
Shape of Ontologies

Level	a.owl Width	b.owl Width
0	7	6
1	14	28
2	14	47
3	11	13
4	5	0
5	5	0
6	3	0
7	0	0
8	0	0
9	0	0
10	0	0



Overall Match Analysis

a.owl				b.owl			
Classes		DatatypePr		ObjectProp		DatatypePr	
		operty	erty	erty	operty	erty	erty
Candidate Matches		15	21		9		22
No Matches		5	25		23		38
a.owl				b.owl			
Classes		Properties	Classes	Properties		Classes	
Candidate Match		36	26				
Potential Mappings		55	39				



Candidate Runtime Matching Algorithms

a.owl		b.owl	
SMU	SMU	SMU	
CUPID	CUPID	CUPID	

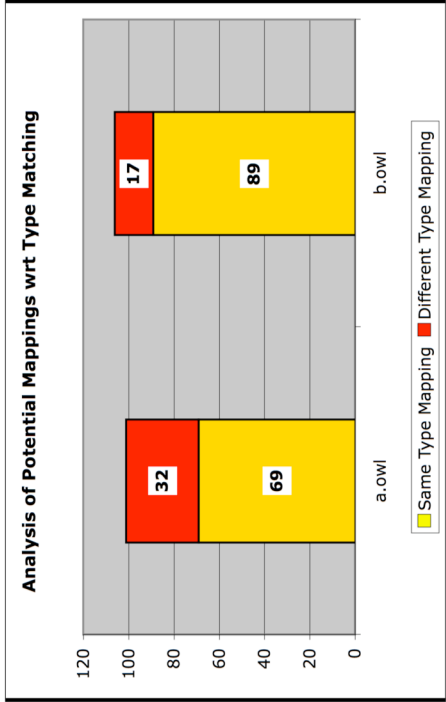
Potential Mappings: Match of Term to Term or Term in Composite Term

	Class to Class	27	26
DatatypeProperty to DatatypeProperty		1	16
ObjectProperty to ObjectProperty		7	34
Property to Class		0	11
Class to Property		8	0
DatatypeProperty to ObjectProperty		0	0
ObjectProperty to DatatypeProperty		0	0

Same Type Mapping	69	89
Different Type Mapping	32	17

Potential Mappings: Match of Term in Composite Term to Term or Composite Term

	Class to Class	8	13
DatatypeProperty to DatatypeProperty		15	0
ObjectProperty to ObjectProperty		11	0
Property to Class		13	6
Class to Property		3	0
DatatypeProperty to ObjectProperty		1	0
ObjectProperty to DatatypeProperty		7	0



3. Decision Guidance Sheet

Evaluator:

Evaluation for
Ontologies:

☐ & ☐

Q1	If on page 3 the candidate matches for classes in each ontology is not greater than 10 then you should immediately proceed to the decision of NO
----	--



Otherwise answer the following questions and then follow instructions in the MAPPING DECISION section below

Ref	Question	yes	no	comment
Q2	On page 1 syntax analysis section, does each of the ontologies compile and is the percentage of OWL keywords used in each ontology > 0.2 ?			
Q3	On page 1 syntax analysis section, does each ontology have greater than 90% of composite terms predominantly of one type (I.e. hyphenated, underscored or MiddleCapital) ?			
Q4	On page 1 WordNet analysis section, does each ontology have a low percentage (<10%) of unknown terms ?			
Q5	On page 1 Ontology References analysis section, does each ontology either import or reference other ontologies?			
Q6	On page 1 SWOOGLE analysis section, is there information for each ontology? If so, comment on what the information might suggest			
Q8	On page 2 Shape of Ontologies section, does each ontology have greater than 2 levels ?			
Q9	On page 3, is the majority of the ratios of Candidate Matches to Potential Mappings less than 1:2			
Q10	On page 4, for each ontology, is the percentage of Potential Mappings that do not match with respect to type mapping less than 33%			

MAPPING DECISION

Depending on the number of "No"s to the above questions, the following decisions are recommended

0 or 1 no then you should decide **YES**

2 or 3 nos then you should decide **PROBABLY**

4 or 5 nos then you should decide **POSSIBLY NOT**

6 or more nos then you should decide **NO**

☐
☐
☐
☐

If you decide to ignore the advice above then state what your decision would be and your reason

4. Task Questionnaire

Experiment#2b Task Questionnaire

Name of Evaluator:

1	Were the guidance questions provided helpful in coming to a decision for each pair of ontologies? Please explain.	
2	Which guidance questions (if any) were unclear? Please explain.	
3	Which (if any) numeric information or chart was particularly helpful in reaching your recommendations? Please explain.	
4	Which (if any) numeric information or chart was not particularly helpful in reaching your recommendations? Please explain.	
5	Do you have any suggestions for numeric information, charts or guidance that might be added to help in this task?	
6	After using the guidance, how difficult was it to come to an overall decision for each pair of ontologies ? Please explain	Difficult Fairly Difficult Fairly Easy Easy
7	After using the guidance, how confident are you in the decisions that you have made? Please explain	Not confident Sort of Confident Confident Very Confident

Additional Comments

5. MINITAB analysis output

01/03/2005 14:40:55

Results for: OSullivan_Declan_#2_ABS.MTW

Paired T-Test and CI: SCORE 1, SCORE 2A

Paired T for SCORE 1 - SCORE 2A

	N	Mean	StDev	SE Mean
SCORE 1	50	0.480000	0.579937	0.082015
SCORE 2A	50	0.620000	0.602376	0.085189
Difference	50	-0.140000	0.670364	0.094804

95% CI for mean difference: (-0.330515, 0.050515)

T-Test of mean difference = 0 (vs not = 0): T-Value = -1.48 P-Value = 0.146

Paired T-Test and CI: SCORE 1, SCORE 2B

Paired T for SCORE 1 - SCORE 2B

	N	Mean	StDev	SE Mean
SCORE 1	50	0.480000	0.579937	0.082015
SCORE 2B	50	0.140000	0.350510	0.049570
Difference	50	0.340000	0.592814	0.083837

95% CI for mean difference: (0.171524, 0.508476)

T-Test of mean difference = 0 (vs not = 0): T-Value = 4.06 P-Value = 0.000

Paired T-Test and CI: SCORE 2A, SCORE 2B

Paired T for SCORE 2A - SCORE 2B

	N	Mean	StDev	SE Mean
SCORE 2A	50	0.620000	0.602376	0.085189
SCORE 2B	50	0.140000	0.350510	0.049570
Difference	50	0.480000	0.614120	0.086850

95% CI for mean difference: (0.305469, 0.654531)

T-Test of mean difference = 0 (vs not = 0): T-Value = 5.53 P-Value = 0.000

General Linear Model: Score1A versus Person_, Question_, Group_

Factor	Type	Levels	Values
Person_	random	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Question_	random	5	a, b, c, d, e
Group_	fixed	2	1, 2

Analysis of Variance for Score1A, using Adjusted SS for Tests

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Person_	9	2.6500	2.6500	0.2944	1.21	0.301
Question_	4	10.9000	10.9000	2.7250	11.18	0.000
Group_	1	0.4900	0.4900	0.4900	2.01	0.160
Error	85	20.7100	20.7100	0.2436		
Total	99	34.7500				

S = 0.493606 R-Sq = 40.40% R-Sq(adj) = 30.59%

Unusual Observations for Score1A

Obs	Score1A	Fit	SE Fit	Residual	St Resid
17	2.00000	0.98000	0.19117	1.02000	2.24 R
72	0.00000	0.92000	0.19117	-0.92000	-2.02 R
76	2.00000	0.77000	0.19117	1.23000	2.70 R
86	2.00000	1.07000	0.19117	0.93000	2.04 R
87	0.00000	1.32000	0.19117	-1.32000	-2.90 R
90	0.00000	0.97000	0.19117	-0.97000	-2.13 R

R denotes an observation with a large standardized residual.

General Linear Model: Score1B versus Person_, Question_, Group_

Factor	Type	Levels	Values
Person_	random	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Question_	random	5	a, b, c, d, e

Group_ fixed 2 1, 2

Analysis of Variance for Score1B, using Adjusted SS for Tests

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Person_	9	2.8900	2.8900	0.3211	1.93	0.059
Question_	4	5.4400	5.4400	1.3600	8.16	0.000
Group_	1	2.8900	2.8900	2.8900	17.34	0.000
Error	85	14.1700	14.1700	0.1667		
Total	99	25.3900				

S = 0.408296 R-Sq = 44.19% R-Sq(adj) = 35.00%

Unusual Observations for Score1B

Obs	Score1B	Fit	SE Fit	Residual	St Resid
17	2.00000	0.86000	0.15813	1.14000	3.03 R
37	2.00000	1.06000	0.15813	0.94000	2.50 R
40	0.00000	0.76000	0.15813	-0.76000	-2.02 R
75	1.00000	0.22000	0.15813	0.78000	2.07 R

R denotes an observation with a large standardized residual.

General Linear Model: ScoreAB versus Person_, Question_, Group_

Factor	Type	Levels	Values
Person_	random	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Question_	random	5	a, b, c, d, e
Group_	fixed	2	1, 2

Analysis of Variance for ScoreAB, using Adjusted SS for Tests

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Person_	9	2.7600	2.7600	0.3067	1.52	0.155
Question_	4	3.8600	3.8600	0.9650	4.77	0.002
Group_	1	5.7600	5.7600	5.7600	28.50	0.000
Error	85	17.1800	17.1800	0.2021		
Total	99	29.5600				

S = 0.449575 R-Sq = 41.88% R-Sq(adj) = 32.31%

Unusual Observations for ScoreAB

Obs	ScoreAB	Fit	SE Fit	Residual	St Resid
2	2.00000	0.96000	0.17412	1.04000	2.51 R
26	2.00000	0.81000	0.17412	1.19000	2.87 R
36	2.00000	0.71000	0.17412	1.29000	3.11 R
75	1.00000	0.13000	0.17412	0.87000	2.10 R

R denotes an observation with a large standardized residual.

General Linear Model: Score3 versus Person3, Question3, Group3

Factor	Type	Levels	Values
Person3	random	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Question3	random	5	a, b, c, d, e
Group3	fixed	3	1, 2A, 2B

Analysis of Variance for Score3, using Adjusted SS for Tests

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Person3	9	3.0400	3.0400	0.3378	1.61	0.118
Question3	4	9.1733	9.1733	2.2933	10.95	0.000
Group3	2	6.0933	6.0933	3.0467	14.55	0.000
Error	134	28.0667	28.0667	0.2095		
Total	149	46.3733				

S = 0.457660 R-Sq = 39.48% R-Sq(adj) = 32.70%

Unusual Observations for Score3

Obs	Score3	Fit	SE Fit	Residual	St Resid
17	2.00000	0.82000	0.14947	1.18000	2.73 R
37	2.00000	0.95333	0.14947	1.04667	2.42 R
52	2.00000	1.02667	0.14947	0.97333	2.25 R
72	0.00000	0.89333	0.14947	-0.89333	-2.07 R
76	2.00000	0.66000	0.14947	1.34000	3.10 R

86	2.00000	0.86000	0.14947	1.14000	2.64	R
87	0.00000	1.09333	0.14947	-1.09333	-2.53	R

R denotes an observation with a large standardized residual.

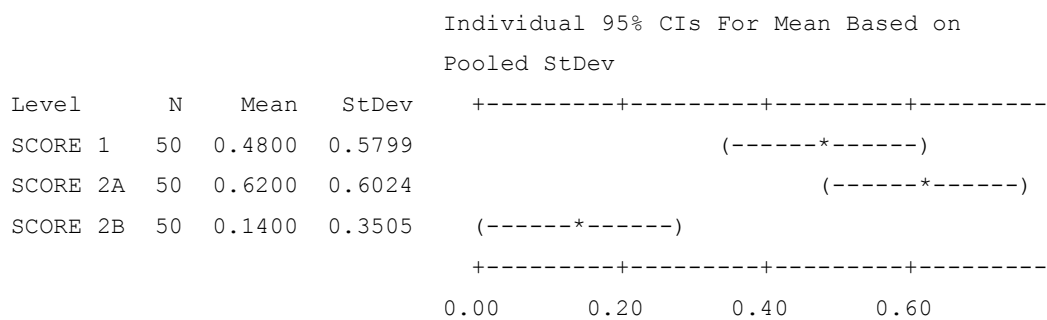
Least Squares Means for Score3

Question3	Mean
a	0.5333
b	0.7667
c	0.1000
d	0.1667
e	0.5000
Group3	
1	0.4800
2A	0.6200
2B	0.1400

One-way ANOVA: SCORE 1, SCORE 2A, SCORE 2B

Source	DF	SS	MS	F	P
Factor	2	6.093	3.047	11.12	0.000
Error	147	40.280	0.274		
Total	149	46.373			

S = 0.5235 R-Sq = 13.14% R-Sq(adj) = 11.96%



Pooled StDev = 0.5235

Appendix E. Mapping Experiment

1. First Task Questionnaire

Students in Group:

Date:

1. This question explores how difficult you found the different parts of the task
Please rate the difficulty you encountered overall according to the following
ratings by circling your chosen number:

1. Very Easy 2. Straightforward 3. Difficult 4. Very Difficult

- Identifying the mappings according to the suggested mapping workflow
- Identifying concepts that might be a candidate anchor pair
- Identifying longest possible valid anchor paths
 - Identifying whether anchor pairs lie on similar ISA paths
 -
 - Identifying mismatch anchor pairs
- Identifying properties of a candidate anchor pair considered equivalent
- Documenting the anchors and anchor paths

Explanatory Comments

2. List the reasons why your group did or did not use the PROMPT tab throughout the task.

Questions 3 and 4 need only be answered if you did use the PROMPT tab throughout the task.

Considering the following parts of the task:

- a. Identifying the mappings according to the suggested mapping workflow
- b. Identifying concepts that might be a candidate anchor pair
- c. Identifying longest possible valid anchor paths
 - i. Identifying whether anchor pairs lie on similar ISA paths
 - ii. Identifying mismatch anchor pairs
- d. Identifying properties of a candidate anchor pair considered equivalent
- e. Documenting the anchors and anchor paths

3. For what parts of the task did you find the PROMPT tab useful?

4. For what parts of the task did you find the PROMPT tab was **not** useful?

2. Second Task Questionnaire

Students in Group:

Date:

1. The OISIN tools allowed ontology mappings to be identified and documented easily

Strongly Agree

Agree

Neutral

Disagree

Strongly Disagree

Comment

2. Some of the difficulties experienced in undertaking the mapping task documented during the last assignment were eased due to the use of the OISIN tools

Strongly Agree

Agree

Neutral

Disagree

Strongly Disagree

Comment

3. Which tool did the group use in the first assignment?

Protégé Tool on its own

Protégé tool with Prompt

4. It was easier to identify mappings with the OISIN tools than with the Protégé/PROMPT tool

Strongly Agree

Agree

Neutral

Disagree

Strongly Disagree

Comment

Considering the following parts of the task:

- f. Identifying the mappings according to the suggested mapping workflow
- g. Identifying concepts that might be a candidate anchor pair
- h. Identifying longest possible valid anchor paths
 - i. Identifying whether anchor pairs lie on similar ISA paths
 - ii. Identifying mismatch anchor pairs
- i. Identifying properties of a candidate anchor pair considered equivalent
- j. Documenting the anchors and anchor paths

5. For which parts of the task did you find the OISIN tools useful?

6. For which parts of the task did you find the OISIN tools **not** useful?

7. Suggested improvements for the OISIN tools to better support the task of ontology mapping identification and documentation

Appendix F. SMU Trial

1. Example XML instance document based on Ontology “C”

```
<ORI xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#">

<Course>

  <Postgraduate_Course>
    <Taught_Postgraduate_Course rdf:Course_Code="CS_UBC"
rdf:Course_Title="Ubiquitous Computing" Duration="2 years"
Semesterised="YES" Student_Quota="20">
      <Student_on_course rdf:resource="00001" />
      <Student_on_course rdf:resource="00002" />
      <Student_on_course rdf:resource="00003" />

    <Associated_Website>www.ubicomm.com</Associated_Website>

  </Taught_Postgraduate_Course>
  <Taught_Postgraduate_Course rdf:Course_Code="UBC_111"
rdf:Course_Title="Interactive Storytelling" Duration="2 years"
Semesterised="YES" Student_Quota="20">
    <Student_on_course rdf:resource="00001" />
    <Student_on_course rdf:resource="00002" />
    <Student_on_course rdf:resource="00003" />
    <attached_lab rdf:resource="Lab_11" />
    <exam_for_course rdf:resource="" />

  <Associated_Website>www.ubicomm.com/UBC_111</Associated_Website>
  <Course_Coordinator rdf:resource="101"/>

  <Course_Lecture rdf:resource="Interactive
Storytelling Intro" />
  <Course_Tutorial rdf:resource="Interactive
Storytelling Tutorial" />

  </Taught_Postgraduate_Course>
  <Taught_Postgraduate_Course rdf:Course_Code="UBC_112"
rdf:Course_Title="Video Technologies" Duration="2 years"
Semesterised="YES" Student_Quota="20">
    <Student_on_course rdf:resource="00002" />
    <Student_on_course rdf:resource="00003" />
    <attached_lab rdf:resource="Lab_10" />

  <Associated_Website>www.ubicomm.com/UBC_112</Associated_Website>

  <Course_Coordinator rdf:resource="102"/>
  <Course_Lecture rdf:resource="Video Technologies
Intro" />
  <Course_Tutorial rdf:resource="Video Technologies
Tutorial" />

  </Taught_Postgraduate_Course>
  <Taught_Postgraduate_Course rdf:Course_Code="UBC_113"
rdf:Course_Title="Context Awareness" Duration="2 years"
Semesterised="YES" Student_Quota="20">
    <Student_on_course rdf:resource="00001" />
    <Student_on_course rdf:resource="00002" />
```

```

        <attached_lab rdf:resource="Lab_12" />

    <Associated_Website>www.ubicomm.com/UBC_113</Associated_Website>

        <Course_Coordinator rdf:resource="103"/>
        <Course_Lecture rdf:resource="Context Awareness
Intro" />

    </Taught_Postgraduate_Course>
</Postgraduate_Course>
</Course>

<Event>
    <Conference rdf:Topic="What is Interactive Storytelling">
        <Contact_Person rdf:resource="#101" />
        <Associated_Website>www.ubicomm.com/Interactive_Storytelling</As
sociated_Website>

        <Location rdf:resource="Venue_10" />
        <Start_Time>24/06/2004 10:00am</Start_Time>
        <Event_Duration>2h</Event_Duration>
    </Conference>

    <Lab_Session rdf:Topic="Video Technologies Lab">
        <Lab_Demonstrator rdf:resource="#102" />
        <Contact_Person rdf:resource="#102" />

        <Location rdf:resource="Lab_10" />
        <Start_Time>21/06/2004 10:00am</Start_Time>
        <Event_Duration>2h</Event_Duration>
    </Lab_Session>

    <Lecture rdf:Topic="Interactive Storytelling Intro">
        <Contact_Person rdf:resource="#101" />
        <course_for_lecture rdf:resource="UBC_111" />
        <Lecture_lecturer rdf:resource="#101" />

        <Location rdf:resource="Venue_12" />
        <Start_Time>22/06/2004 10:00am</Start_Time>
        <Event_Duration>2h</Event_Duration>
    </Lecture>

    <Lecture rdf:Topic="Video Technologies Intro">
        <Contact_Person rdf:resource="#102" />
        <course_for_lecture rdf:resource="UBC_112" />

        <Lecture_lecturer rdf:resource="#102" />
        <Location rdf:resource="Venue_11" />
        <Start_Time>23/06/2004 10:00am</Start_Time>
        <Event_Duration>2h</Event_Duration>
    </Lecture>

    <Lecture rdf:Topic="Context Awareness Intro">
        <Contact_Person rdf:resource="#103" />

        <course_for_lecture rdf:resource="UBC_113" />
        <Lecture_lecturer rdf:resource="#103" />
        <Location rdf:resource="Venue_12" />
        <Start_Time>25/06/2004 10:00am</Start_Time>
        <Event_Duration>2h</Event_Duration>
    </Lecture>

    <Tutorial rdf:Topic="Interactive Storytelling Tutorial">

```

```

        <course_for_tutorial rdf:resource="UBC_111"/>
        <Tutorial_Tutor rdf:resource="#101"/>
        <Location rdf:resource="Tut_11"/>
        <Start_Time>26/06/2004 10:00am</Start_Time>
        <Event_Duration>2h</Event_Duration>
    </Tutorial>

    <Tutorial rdf:Topic="Video Technologies Tutorial">

        <course_for_tutorial rdf:resource="UBC_112"/>
        <Tutorial_Tutor rdf:resource="#102"/>
        <Location rdf:resource="Tut_10"/>
        <Start_Time>27/06/2004 10:00am</Start_Time>
        <Event_Duration>2h</Event_Duration>
    </Tutorial>

    <Exam rdf:Topic="Interactive Storytelling Exam">

        <examined_course rdf:resource="UBC_111" />
        <Exam_Invigilator rdf:resource="#101" />
        <Exam_Invigilator rdf:resource="#102" />
        <Location rdf:resource="Venue_11" />
        <Start_Time>23/06/2004 10:00am</Start_Time>
        <Event_Duration>2h</Event_Duration>
    </Exam>

    <Exam rdf:Topic="Video Technologies Exam">
        <examined_course rdf:resource="UBC_112" />
        <Exam_Invigilator rdf:resource="#103" />
        <Exam_Invigilator rdf:resource="#104" />
        <Location rdf:resource="Venue_12" />
        <Start_Time>23/06/2004 2:00pm</Start_Time>
        <Event_Duration>1h 45min</Event_Duration>
    </Exam>

    <Exam rdf:Topic="Context Awareness Exam">
        <examined_course rdf:resource="UBC_113" />
        <Exam_Invigilator rdf:resource="#101" />
        <Location rdf:resource="Venue_13" />
        <Start_Time>20/06/2004 10:00am </Start_Time>
        <Event_Duration>45min</Event_Duration>
    </Exam>

</Event>

<Facility>
    <Computer Asset_Tag_Number="9991" Monitor_Size="#21"
    OS_Installed="Linux">
        <Current_Location rdf:resource="Lab_10"/>
    </Computer>
    <Computer Asset_Tag_Number="9992" Monitor_Size="#18"
    OS_Installed="Windows">
        <Current_Location rdf:resource="Venue_12"/>
    </Computer>
    <Computer Asset_Tag_Number="9993" Monitor_Size="#29"
    OS_Installed="Windows">

        <Current_Location rdf:resource="Lab_11"/>
    </Computer>
    <Printer Asset_Tag_Number="9994" Printer_Type="Inkjet"
    Is_working="True" Is_Colour="True" >
        <Current_Location rdf:resource="Lab_10"/>

```

```

    </Printer>
    <Printer Asset_Tag_Number="9995" Printer_Type="Laser"
Is_working="True" Is_Colour="False" >
        <Current_Location rdf:resource="Lab_10"/>
    </Printer>
    <Printer Asset_Tag_Number="9996" Printer_Type="BubbleJet"
Is_working="True" Is_Colour="False" >
        <Current_Location rdf:resource="Lab_11"/>
    </Printer>
    <Projector Asset_Tag_Number="9997" Input_Devices="RGB">
        <Current_Location rdf:resource="Venue_12"/>
    </Projector>
</Facility>

<Person>
    <Student_Person>
        <Postgrad_Student_Person rdf:Student_ID="00001"
Second_Name="Bernardi" >
            <Postgraduate_Course_Taken rdf:resource="CS_UBC"/>
            <Postgraduate_Course_Taken rdf:resource="UBC_111"/>
            <Postgraduate_Course_Taken rdf:resource="UBC_113"/>
        </Postgrad_Student_Person>
    </Student_Person>

    <Student_Person>
        <Postgrad_Student_Person rdf:Student_ID="00002"
Second_Name="Morrissey">
            <Postgraduate_Course_Taken rdf:resource="CS_UBC"/>
            <Postgraduate_Course_Taken rdf:resource="UBC_111"/>
            <Postgraduate_Course_Taken rdf:resource="UBC_112"/>
            <Postgraduate_Course_Taken rdf:resource="UBC_113"/>
        </Postgrad_Student_Person>
    </Student_Person>

    <Student_Person>
        <Postgrad_Student_Person rdf:Student_ID="00003"
Second_Name="Madej">
            <Postgraduate_Course_Taken rdf:resource="CS_UBC"/>
            <Postgraduate_Course_Taken rdf:resource="UBC_111"/>
            <Postgraduate_Course_Taken rdf:resource="UBC_112"/>
        </Postgrad_Student_Person>
    </Student_Person>

    <Staff_Person>
        <Lecturer_Staff_Person rdf:Employee_Number="101"
Second_Name="Barrett" Current_Contact_Hours="">
            <exam_invigilated rdf:resource="Interactive
Storytelling Exam" />
            <exam_invigilated rdf:resource="Context Awareness
Exam" />
        </Lecturer_Staff_Person>

        <Lecturer_Staff_Person rdf:Employee_Number="102"
Second_Name="Dowling" Current_Contact_Hours="">

```

```

        <exam_invigilated rdf:resource="Context Awareness
Exam" />
        </Lecturer_Staff_Person>

        <Lecturer_Staff_Person rdf:Employee_Number="103"
Second_Name="Harris" Current_Contact_Hours="">
        <exam_invigilated rdf:resource="Video Technologies
Exam" />
        </Lecturer_Staff_Person>

        <Lecturer_Staff_Person rdf:Employee_Number="104"
Second_Name="Upton" Current_Contact_Hours="">
        <exam_invigilated rdf:resource="Video Technologies
Exam" />

        </Lecturer_Staff_Person>

    </Staff_Person>

</Person>

<Place>
    <Space>
        <Academic_Space>
            <Lecture_Theatre rdf:Room_Code="Venue_10"
Capacity="30" GPS_Location="22x10">
                <Theatre_Name>Swift</Theatre_Name>

            </Lecture_Theatre>

            <Lecture_Theatre rdf:Room_Code="Venue_11"
Capacity="145" GPS_Location="321X321">
                <Theatre_Name>MacNeill</Theatre_Name>
            </Lecture_Theatre>

            <Lecture_Theatre rdf:Room_Code="Venue_12"
Capacity="130" GPS_Location="654X74">
                <Theatre_Name>Goldsmith Hall</Theatre_Name>
            </Lecture_Theatre>

            <Tutorial_Room rdf:Room_Code="Tut_10" Capacity="10"
GPS_Location="999X474">
                </Tutorial_Room>

            <Tutorial_Room rdf:Room_Code="Tut_11" Capacity="10"
GPS_Location="998X475">
                </Tutorial_Room>

            <Laboratory rdf:Room_Code="Lab_10" Capacity="30"
GPS_Location="922X575">
                <Attached_Course rdf:resource="UBC_112"/>
            </Laboratory>

            <Laboratory rdf:Room_Code="Lab_11" Capacity="27"
GPS_Location="978X599">
                </Laboratory>

            <Laboratory rdf:Room_Code="Lab_12" Capacity="20"
GPS_Location="87x980" />

        </Academic_Space>

```

```

    </Space>
</Place>

<Research_Group Name="Information Architecture Group"
rdf:ID="Group_01">
  <Associated_Website rdf:resource="#Web_01"/>
  <attached_lab rdf:resource="#Lab_12" />

  <Attaced_Research_Project rdf:resource="#Proj_01" />
  <member rdf:resource="#101" />
  <member rdf:resource="#00001" />
  <member rdf:resource="#00003" />
</Research_Group>

<Research_Paper rdf:ID="Paper_01">
  <Abstract>There are many ways to query XML documents using XPath
and XQuery. In this
  Article we'll focus on the advanced use of optimization
functions.</Abstract>
  <Lenght>32</Lenght>

  <Title>XML querying</Title>
  <Topic>XML</Topic>
  <Topic>Ontologies</Topic>
  <Topic>XPath</Topic>
  <paper_published_by rdf:resource="#101" />
</Research_Paper>

<Research_Project rdf:ID="Proj_01">

  <Associated_Website rdf:resource="#Web_01"/>
  <Attached_Research_Group rdf:resource="#Group_01"/>
</Research_Project>

<Thesis rdf:ID="Thesis_01">
  <Title>Advanced XML</Title>
  <Abstract>An advanced method to generate and query XML
documents</Abstract>
  <Author rdf:resource="#00001"/>
  <Author rdf:resource="#101"/>

  <Lenght>123</Lenght>
  <Topic>XML</Topic>
</Thesis>

<Thesis rdf:ID="Thesis_02">
  <Title>XML for Dummies</Title>
  <Abstract>Basic XML concept</Abstract>
  <Author rdf:resource="#101"/>

  <Lenght>86</Lenght>
  <Topic>XML</Topic>
</Thesis>

<Webpage Page_Title="Welcome!" rdf:ID="Page_01">
  <Author rdf:resource="#00001" />
  <Full_Url>http://www.cs.tcd.ie/test/welcome.html</Full_Url>
</Webpage>

<Webpage Page_Title="Contact us!" rdf:ID="Page_02">

  <Author rdf:resource="#00002" />

```

```

    <Full_Url>http://www.cs.tcd.ie/test/email.html</Full_Url>
</Webpage>

<Webpage Page_Title="Projects" rdf:ID="Page_03">
  <Author rdf:resource="#101" />
  <Full_Url>http://www.cs.tcd.ie/test/projects.html</Full_Url>
</Webpage>

<Website rdf:ID="Web_01">
  <Index_Page>index.php</Index_Page>

  <Base_Url>http://www.cs.tcd.ie/test/</Base_Url>
  <Associated_Pages rdf:resource="#Page_01" />
  <Associated_Pages rdf:resource="#Page_02" />
  <Associated_Pages rdf:resource="#Page_03" />
</Website>

</ORI>

```

2. Example Queries from Group 3

```
declare namespace rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#";
```

(: UBICOM APPLICATION :- **E-Nameplate (Electronic Board):**

Designed to hang on the wall at the entrance doors.

Shows all information connected with specific room, like
timetable or students list for an exam

Query:= Timetable for specific room (Events ordered by date) :

```
declare function getEvents($data){
<Room_Event>
{
for $r in doc("A2/Finished.xml")/ORI/Event/*
    order by $r/Start_Time
    return
        if (contains(string($r/Location/@rdf:resource),$data))
        then <Event>{$r/@rdf:Topic,$r/Start_Time}</Event>
        else ""
    }
</Room__Event>
};
```

```
declare namespace rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#";
```

(: UBICOM APPLICATION :- **Campus wide computer services search Facility:**

Executes spoken user queries about available facilities :)

(: **Query:= Are there any lecture theatres with computers and Projectors :**

(: returns a listing of all the rooms ID numbers:)

```
declare function getLectureRoomID()
{
    for $a in
        doc("Ontology_C.xml")/ORI/Place/Space/Academic_Space/Lecture_Theatre
```

```

    return
    <id>
    {string($a/@rdf:Room_Code)}
    </id>
};

```

(: Returns a list of rooms with computers:)

```

declare function getRoomsWithComp()
{
for $b in getLectureRoomID()
for $c in doc("Ontology_C.xml")/ORI/Facility/Computer
    where $c/Current_Location/@rdf:resource = $b
    return <With_Computer>{$b}</With_Computer>
};

```

(: Returns a list of rooms with projectors:)

```

declare function getRoomsWithProjector()
{
for $b in getLectureRoomID()
for $c in doc("Ontology_C.xml")/ORI/Facility/Projector
    where $c/Current_Location/@rdf:resource = $b
    return <With_Projector>{$b}</With_Projector>
};

```

(:returns a list of all rooms with both computers and projectors:)

```

for $d in getRoomsWithComp()
for $e in getRoomsWithProjector()
    where string($d/With_Computer/id) = string($e/With_Projector/id)
return
<Room_With_Comp_and_Projector>
    {concat("Room_ID = ",string($d))}
</Room_With_Comp_and_Projector>

```

```
declare namespace rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#";
```

```
(: UBICOM APPLICATION :- Automatic Student Calendar:
```

```
Gathers information and events of interest to that student  
and generates a personalised calendar for that student :)
```

```
(: Query := List which courses students are enrolled in, and which they are not  
enrolled in :)
```

```
(:returns all courses which students could have enrolled in :)
```

```
declare function getCoursesAvailable()
```

```
{  
for $b in  
doc("Ontology_C.xml")/ORI/Course/Postgraduate_Course/Taught_Postgraduate_Cour  
se  
return  
  <available>  
    {$b}  
  </available>  
};
```

```
(: for each student and each possible course check whether or not the student is  
enrolled :)
```

```
for $d in
```

```
doc("Ontology_C.xml")/ORI/Person/Student_Person/Postgrad_Student_Person
```

```
for $e in getCoursesAvailable()
```

```
let $f:= $e/Taught_Postgraduate_Course/@rdf:Course_Code
```

```
let $g:= $d/Postgraduate_Course_Taken/@rdf:resource
```

```
return
```

```
<status>
```

```
{if ($f = $g)
```

```
  then concat("Student ",concat(string($d/@Second_Name),concat(" is  
enrolled in ",string($f))))
```

```
  else concat("Student ",concat(string($d/@Second_Name),concat(" is  
not enrolled in ",string($f))))
```

```
}
```

```
</status>
```