

The MOUSE approach: Mapping Ontologies using UML for System Engineers

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

Seung-Hwa Chung
Knowledge and Data Engineering Group,
Department of Computer Science,
Trinity College, Dublin

Supervisor: Dr. Declan O'Sullivan
Assistant Supervisor: Dr. Wei Tai
Industrial Supervisor: Dr. Aidan Boran

2014

Declaration

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

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

Seung-Hwa Chung

December 2014

Acknowledgements

Above all, I would like to thank to God for giving me this opportunity in my life. During these past four years, I not only had the opportunity to study my chosen field of semantics but also became an academic researcher all the while experiencing a new foreign culture. Studying in Ireland was just as important part of my education as the subject matter itself. I leave Trinity College Dublin with a very valuable education and a lifetime of memories.

I would like to thank my supervisor Prof. Declan O'Sullivan. Without his guidance and support I would not have come this far. I would also like to express my gratitude to Dr. Wei Tai and Dr. John Keeney for their helpful contributions to this research. I also give thanks to Dr. Aidan Boran at Bell Labs Ireland/Alcatel-Lucent for his support and contribution. Additionally, I express my gratitude to my colleagues in KDEG Lab who have always been kind and supportive during my studies.

Finally, I would like to thank my family (Myung-Bok, Sook-Hee, Ji-Eun, Ha-Eun, Yeon-Soo, Shin-Hyung, Yong-Kyoo, and Faith) and the member of my church for their prayers and words of encouragement to keep me studying and finishing my PhD. Although, I am uncertain about my life, I am conscious that all things are working together for good to those who have love for God in His purpose. I believe this doctoral experience has laid a strong foundation for a future where I can complete any task the Lord has given me to testify to the Gospel of the grace of God, so that I may finish my race with joy.

Abstract

To address the problem of semantic heterogeneity, there has been a large body of research directed toward the study of semantic mapping technologies. Although various semantic mapping technologies have been investigated, facilitating the process for domain experts to perform a semantic data integration task is still not easy. This is because one is required not only to possess domain expertise but also to have a good understanding of knowledge engineering. This thesis proposes an approach that automatically transforms an abstract semantic mapping syntax into a concrete executable mapping syntax, I call this approach **MOUSE (Mapping Ontologies using UML for System Engineers)**. In order to evaluate MOUSE, an implementation of this approach for a semantic data integration use case has been developed (called **SDI, Semantic Data Integration**). The aim is to enable domain experts, particularly system engineers, to undertake mappings using a technology that they are familiar with (UML), while ensuring the created mappings are accurate and the approach is easy to use. The proposed UML-based abstract mapping syntax is evaluated through usability experiments conducted in a lab environment by participants who have skills equivalent to real life system engineers using the SDI tool. Results from the evaluations show that the participants could correctly undertake the semantic data integration task using the MOUSE approach while maintaining accuracy and usability (in terms of ease of use).

Table of Contents

Declaration	i
Acknowledgements	ii
Abstract	iii
Table of Contents	iv
Table of Figures.....	ix
Table of Tables	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Research Question	3
1.3 Research Objectives	4
1.4 Contribution.....	7
1.5 Thesis Overview	8
2 Related Research	10
2.1 Ontology mapping categorization	11
2.2 Ontology mapping application.....	12
2.3 Abstract syntaxes for semantic mapping	14
2.3.1 <i>Alignment Format</i>	15
2.3.2 <i>EDOAL</i>	16
2.3.3 <i>C-OWL</i>	18
2.3.4 <i>SSE</i>	18
2.3.5 <i>SQWRL</i>	19
2.3.6 <i>RIF</i>	20
2.3.7 <i>UML/ODM</i>	21

2.3.8	<i>UML/OCL</i>	23
2.4	Concrete syntaxes for semantic mapping	24
2.4.1	<i>Using OWL for semantic mapping</i>	25
2.4.2	<i>SWRL</i>	26
2.4.3	<i>SPARQL</i>	29
2.5	Summary Tables	30
3	MOUSE Approach	36
3.1	The Core Mapping Types	36
3.1.1	<i>Industry Use Case</i>	36
3.1.2	<i>Core mapping type example</i>	40
3.2	UML Notations Development	43
3.3	UML Notations for Ontology Elements	46
3.3.1	<i>Ontology Document</i>	46
3.3.2	<i>Ontology Class</i>	47
3.3.3	<i>Ontology Individual</i>	48
3.3.4	<i>Ontology Object Property</i>	49
3.3.5	<i>Ontology Datatype Property</i>	50
3.4	UML Notations for Semantic Mappings	51
3.4.1	<i>Direct Mapping Type</i>	52
3.4.2	<i>Data Range Mapping Type</i>	53
3.4.3	<i>Unit Transformation Mapping Type</i>	57
3.5	A walked through example.....	62
3.5.1	<i>Direct Mapping</i>	64
3.5.2	<i>Data Range Mapping</i>	64

3.5.3	<i>Unit Transformation Mapping</i>	64
3.5.4	<i>Limitation</i>	65
4	SDI Tool Development	66
4.1	Conversion Process	66
4.2	UML to RIF Conversion	67
4.2.1	<i>Direct Mapping Type</i>	67
4.2.2	<i>Data Range Mapping Type</i>	69
4.2.3	<i>Unit Transformation Mapping Type</i>	72
4.3	RIF to SPARQL Conversion	75
4.4	Implementation	79
4.4.1	<i>SDI Tool Modules</i>	81
4.4.2	<i>Structured Information Models</i>	82
4.5	A walked through example.....	86
5	System Engineers Experiment	90
5.1	Experiment Question.....	90
5.2	Experiment Hypothesis	90
5.3	Experiment Method.....	91
5.3.1	<i>Participants</i>	91
5.3.2	<i>Materials</i>	91
5.3.3	<i>Procedure</i>	93
5.3.4	<i>Metrics</i>	94
5.4	Experiment Analysis	95
5.4.1	<i>Accuracy</i>	96
5.4.2	<i>Ease of use</i>	103

6	Conclusion.....	109
6.1	Future Works	111
	REFERENCES.....	113
APPENDIX A	Ontologies from the industry use case.....	127
Appendix A.1	Femto level ontology - femtoclassifier.owl	127
Appendix A.2	Raw level ontology - femtoinstances.owl.....	132
APPENDIX B	SDI tool user manual in GUI mode with examples.....	161
Appendix B.1	Example of how to create direct mapping	162
Appendix B.2	Example of how to create data range mapping	165
Appendix B.3	Example of how to create unit transformation mapping.....	168
Appendix B.4	Example of how to generate mapping outputs	171
APPENDIX C	SDI tool java archive file usage in command line mode	176
Appendix C.1	Usage of “sditool.jar” in command line mode	176
Appendix C.2	Composition of the sditool.jar file.....	177
APPENDIX D	Mapping design worksheets	178
Appendix D.1	Mapping design worksheets for direct mapping type.....	178
Appendix D.2	Mapping design worksheets for data range mapping type.....	182
Appendix D.3	Mapping design worksheets for unit transformation mapping type	186
APPENDIX E	OWL view of initial experiment ontologies in Protégé tool	190
Appendix E.1	NC ontology	190
Appendix E.2	TR069 ontology.....	191
Appendix E.3	Femto ontology.....	192
APPENDIX F	OWL view of experiment ontologies in Protégé tool.....	193
Appendix F.1	Source conference ontology	193

Appendix F.2	Target conference ontology	194
APPENDIX G	Potential mapping candidates in experiment ontologies	195
Appendix G.1	Potential mapping candidates in direct mapping type	196
Appendix G.2	Potential mapping candidates in data range mapping type.....	197
Appendix G.3	Potential mapping candidates in unit transformation mapping type	198
APPENDIX H	Brief mapping guide for core mapping types	199
Appendix H.1	Brief mapping guide for direct mapping type	199
Appendix H.2	Brief mapping guide for data range mapping type.....	200
Appendix H.3	Brief mapping guide for unit transformation mapping type	201
APPENDIX I	Questionnaire sheets based on the System Usability Scale	202
APPENDIX J	Experiment overall task sheet	204
APPENDIX K	Scanned paper result from the experiment	205
Appendix K.1	Experiment result from participant ID02.....	205
Appendix K.2	Experiment result from participant ID13.....	222
APPENDIX L	Accompanying DVD: Table of Contents	240

Table of Figures

Figure 2-1 Screenshot of PROMPT by N. F. Noy and M. A. Musen	13
Figure 2-2 Screenshot of CoGZ by S. Falconer and M. A. Storey	14
Figure 2-3 Alignment Format syntax example for equivalent mapping type.....	15
Figure 2-4 EDOAL syntax example for conditional mapping type	17
Figure 2-5 EDOAL syntax example for transformation mapping type.....	17
Figure 2-6 C-OWL syntax example for equivalent mapping type.....	18
Figure 2-7 C-OWL syntax example for subsumption mapping type	18
Figure 2-8 SSE syntax example for conditional mapping type	19
Figure 2-9 SSE syntax example for transformation mapping type	19
Figure 2-10 SQWRL syntax example for conditional mapping type.....	20
Figure 2-11 SQWRL syntax example for transformation mapping type	20
Figure 2-12 RIF syntax example for conditional mapping type	21
Figure 2-13 RIF syntax example for transformation mapping type.....	21
Figure 2-14 UML/ODM example for ontology class and property	22
Figure 2-15 UML/ODM example for ontology datatype property	22
Figure 2-16 UML/ODM example for equivalent mapping type.....	23
Figure 2-17 UML/OCL example for conditional mapping type.....	24
Figure 2-18 UML/OCL example for transformation mapping type	24
Figure 2-19 OWL syntax example for conditional mapping type	26
Figure 2-20 SWRL syntax example for conditional mapping type	27
Figure 2-21 SWRL syntax example for transformation mapping type	29
Figure 2-22 SPARQL syntax example for conditional mapping type	30

Figure 2-23 SPARQL syntax example for transformation mapping type	30
Figure 2-24 Enumerated and structured type expression	34
Figure 3-1 Logical example of the direct mapping type	40
Figure 3-2 Logical example of the data range mapping type	41
Figure 3-3 Logical example of the unit transformation type	42
Figure 3-4 ODM and OCL combined abstract mapping syntax for conditional mapping type	44
Figure 3-5 ODM and OCL combined abstract mapping syntax for transformation mapping type	45
Figure 3-6 UML notation example for ontology document	47
Figure 3-7 UML notation example for ontology class	48
Figure 3-8 UML notation example for RDF statement	48
Figure 3-9 UML notation example for ontology individual	49
Figure 3-10 UML notation example for RDF statement	49
Figure 3-11 UML notation example for ontology object property	50
Figure 3-12 UML notation example for RDF statement	50
Figure 3-13 UML notation example for ontology datatype property	51
Figure 3-14 UML notation example for RDF statement	51
Figure 3-15 UML notation example for direct mapping type	52
Figure 3-16 BNF style syntax expression for data range mapping	54
Figure 3-17 UML notation example for data range mapping type with single ontology datatype property in constraint	55
Figure 3-18 UML notation example for data range mapping type with multiple ontology datatype properties in constraint	55

Figure 3-19 OCL syntax translated from UML constraint expression for data range mapping type	56
Figure 3-20 OCL syntax translated from UML constraint expression for data range mapping type	57
Figure 3-21 BNF style syntax expression for unit transformation mapping	58
Figure 3-22 UML notation example for unit transformation mapping type with single ontology datatype property operation in constraint	60
Figure 3-23 UML notation example for unit transformation mapping type with multiple ontology datatype properties operation in constraint	60
Figure 3-24 OCL syntax translated from UML constraint expression for unit transformation mapping type	61
Figure 3-25 OCL syntax translated from UML constraint expression for unit transformation mapping type	62
Figure 3-26 Example of the UML-based abstract mapping syntax	63
Figure 4-1 SDI tool generation process overview	67
Figure 4-2 Translation example of UML notations to RIF syntax for direct mapping type...	69
Figure 4-3 Translation example of UML notations to RIF syntax for data range mapping type	71
Figure 4-4 Translation example of UML notations to RIF syntax for data range mapping type	72
Figure 4-5 Translation example of UML notations to RIF syntax for unit transformation mapping type	74
Figure 4-6 Translation example of UML notations to RIF syntax for unit transformation mapping type	75
Figure 4-7 Translation example of RIF document to SPARQL query for direct mapping type	77
Figure 4-8 Translation example of RIF document to SPARQL query for data range mapping type	78

Figure 4-9 Translation example of RIF document to SPARQL query for unit transformation mapping type	79
Figure 4-10 SDI tool architecture overview	80
Figure 4-11 UML Class diagram for structured UML information model	83
Figure 4-12 UML Class diagram for structured RIF information model	84
Figure 4-13 UML Class diagram for structured SPARQL information model	85
Figure 4-14 ScreenShot of the StarUML graphic user interface	86
Figure 4-15 RIF document generated from the UML notations	88
Figure 4-16 SPARQL query generated from the RIF document	88
Figure 5-1 A sample abstract mapping syntax for ontology mappings from the experiment.	96
Figure 5-2 Auto-generated SPARQL query syntax for the data range mapping	97
Figure 5-3 Auto-generated SPARQL query syntax for the unit transformation mapping	97
Figure 5-4 A sample mappings on the SDI tool from the experiment	101
Figure 5-5 A sample mappings on the SDI tool from the experiment	102
Figure 5-6 SUS scores bar graph for the abstract mapping syntax and the tool	105
Figure 5-7 A simple ease of use indication question scores bar graph for the abstract mapping syntax.....	107

Table of Tables

Table 2-1 Ontology mapping categorization by F. Scharffe and D. Fensel	11
Table 2-2 Summary table of the supported mapping types for concrete and abstract mapping syntaxes	31
Table 2-3 Summary table of the usability and type of expression for concrete and abstract mapping syntaxes.....	32
Table 3-1 Relationship between the semantic intergration situations and the core mapping types	38
Table 3-2 Relationship between the generic mapping types and the core mapping types	39
Table 3-3 Correspondence between UML constraint expression and OCL syntax for data range mapping type.....	56
Table 3-4 Correspondence between UML constraint expression and OCL syntax for data range mapping type.....	57
Table 3-5 Correspondence between UML constraint expression and OCL syntax for unit transformation mapping type.....	61
Table 3-6 Correspondence between UML constraint expression and OCL syntax for unit transformation mapping type.....	62
Table 4-1 Correspondence between UML notation and RIF syntax for direct mapping type	68
Table 4-2 Correspondence between UML notation and RIF syntax for data range mapping type	70
Table 4-3 Correspondence between UML notation and RIF syntax for unit transformation mapping type	73
Table 4-4 Correspondence between RIF syntax and SPARQL syntax	76
Table 5-1 Number of correctly auto-generated mappings corresponding to abstract mapping syntax	98
Table 5-2 Number of correctly captured mappings corresponding to participant's mapping intention	100

Table 5-3 SUS scores for the proposed abstract mapping syntax and the implemented tool	104
Table 5-4 A simple ease of use indication question scores for the abstract mapping syntax	106

1 Introduction

1.1 Motivation

There has always been a need to integrate an increasing set of diverse devices and information systems within an enterprise, e.g. sensors, mobile devices, inventory tracking system, sales system, finance system, human resource system, and so on [Rho 2007]. Typically these devices and information systems are produced by a variety of developers with different data schemas, and information integration of such heterogeneous data is a specialized and brittle process, such that changing the structure and/or addition of new data sources can force an integration redesign [Bernstein 2008]. This heterogeneity makes it difficult for third-party enterprises to manipulate the data of distinct formats and to extract information from the separated data. However, adding semantics to the data can ease these difficulties caused by data heterogeneity [Waters 2009]. For this reason, there has been an increasing effort to embed semantics along with the data in order to ease the integration effort [Duo 2006].

To enable the encoding of semantics with the data, well-known technologies are RDF (Resource Description Framework) [RDF Syntax] and OWL (Web Ontology Language) [OWL Overview]. These technologies formally represent the meaning involved in information. For example, ontology can describe concepts, relationships between things, and categories of things. These embedded semantics with the data offer significant advantages such as reasoning over data and dealing with heterogeneous data sources [Gruber 1993][Fensel 2004]. However, the ontology that represents the knowledge within a certain domain may be developed within different engineering processes resulting in heterogeneous ontologies in both conceptualization and terminology. Conceptual heterogeneity can occur due to the natural human diversity involved in ontology development of a particular domain or due to the differences between the ways in modelling and interpreting entities that depend on differing viewpoints and different portions of the domain [O'Sullivan 2007][Thomas 2009]. Terminological heterogeneity can occur when mismatches

relating to the naming process of the ontology entities use different words to name the same entity (synonymy), the same word is used to name different entities (polysemy), words from different languages (multilingualism) and syntactic variations of the same word (different acceptable spellings, abbreviations, use of optional prefixes or suffixes, and so on) [Bouquet 2005][Amrouch 2012].

Ontology heterogeneity typically requires mappings to exchange information in a semantically sound manner [Kalfoglou 2003]. There have been investigations into ontology mapping technologies to resolve the ontology heterogeneity issues that are often encountered during the integration of ontology data from various sources. The existing ontology mapping approaches [Falconer 2007a] usually require mapping practitioners to have a considerable amount of expertise in knowledge engineering in order to perform the mapping process, and ontology mappings are often performed by knowledge engineers. Hence in most of the current situations, a mapping practitioner also needs to be a knowledge engineer. In this research, it is argued that performing a semantic data integration task by system engineers such as telecommunications system engineers that are considered domain experts is more realistic because of the complexity in designing semantic mappings for non-trivial cases in the system. However, since the creation of ontology mappings requires considerable effort [Cruz 2012] and considerable amount of expertise in knowledge engineering [Falconer 2007b], it is understandable that not all the system engineers are able to perform semantic data integration tasks. The research presented in this thesis focuses on supporting those system engineers who are expected to have insufficient ontological knowledge or lack knowledge engineering experience.

Considering that system engineers may have little background in ontology techniques, the semantic mapping conceptualization needs to be abstract, meaning that the syntax needs to be more natural to manipulate than a concrete syntax [Miller 1992]. A concrete mapping syntax means that the syntax can be directly executable in a system to perform integration [Fondement 2005].

Thus in this research an approach is proposed - MOUSE (Mapping Ontologies using UML for System Engineers) - that will support system engineers expressing the

mappings that they require, and automatically transform these abstract mapping expressions into executable expressions of the mappings within the system.

1.2 Research Question

The question being addressed in this research is thus:

“To what extent will the proposed MOUSE approach: (i) allow the creation of mappings using an abstract syntax familiar and usable to system engineers; (ii) allow the accurate transformation of the abstract mappings into concrete executable mappings?”

In this question, an abstract mapping syntax needs to be able to represent ontology mapping relationships accurately using an abstract conceptualization without ontological knowledge. This abstract mapping syntax also needs to be easy to understand for system engineers in order to be used without any considerable effort of learning.

A mapping syntax needs to be automatically transformed from abstract to concrete. A concrete mapping syntax means that the syntax can be directly executable in a system to perform integration, and an abstract mapping syntax means the syntax is more natural to manipulate than a concrete mapping syntax by using abstract conceptualization that fades away the concepts of ontology and is more intuitive to represent mappings.

This research identified core mapping types from an industrial ontology integration situation. These core mapping types are used to examine the expressive capacity of abstract mapping syntaxes, and they are required to be captured correctly for the abstract mapping syntax to be usable in a practical integration situation. These identified core mapping types were observed through empirical derivation from an industry use case and dataset [Boran 2011a], which is related to the research on semantic up-lift of the ontology data for the classification of telecommunications network performance; using real network data collected from an Alcatel-Lucent femtocell [Small Cell] test bed. These mapping types had more priority than other

mapping types in the given integration situation, and this research focused on how these mapping types could be captured correctly by the abstract mapping syntax. The core mapping types were also observed in the literature on correspondence patterns for ontology alignment by Scharffe [Scharffe 2008], which is well-known research in ontology mapping field.

1.3 Research Objectives

The following research objectives have been derived from the research question.

1. *Survey and review the related research about (i) ontology mapping relationships; (ii) ontology mapping applications; (iii) abstract mapping syntaxes that can describe ontology mapping relationships; and (iv) concrete mapping syntaxes for semantic data integration.*

This objective was to identify the gap in current research. Up-to date literature was reviewed and examined related to (i) correspondence patterns¹ to categorize ontology mapping relationships; (ii) existing ontology mapping applications that supports ontology mapping processes assisting a mapping practitioner to perform a semantic data integration task; (iii) abstract mapping syntaxes that are capable of describing ontology mapping relationships using an abstract conceptualization in order to develop an abstract mapping syntax that is usable by system engineers - who are expected to be not familiar with ontology - to perform ontology mappings; and (iv) concrete mapping syntaxes that can be directly executable in a system to perform an integration.

2. *Define core mapping types that can be used to examine what types of ontology mapping relationships need to be captured correctly.*

¹ Correspondence patterns inspired from patterns in software engineering. The correspondence patterns capture regularities recurring when aligning ontologies and facilitate the ontology alignment process. [Scharffe 2008]

This objective was to identify the scope of the expressivity that a developed abstract syntax is required to capture correctly to be practically usable. These core mapping types were used to examine the expressive capacity of abstract mapping syntaxes. They were derived from an industrial ontology integration situation that classifies the telecommunications network performance data. The industry use case from an Alcatel-Lucent industrial test dataset - real network data collected from the Alcatel-Lucent femtocell test bed - was provided to identify and to prioritize the ontology mapping types. The core mapping types were also observed from the correspondence patterns for ontology alignment literature.

3. *Develop an abstract mapping syntax that can represent the core mapping types using technology familiar to system engineers.*

This objective facilitated system engineers to describe ontology mappings without considerable amount of expertise in knowledge engineering. An approach using UML for an abstract mapping syntax was proposed. In the first stage, an abstract mapping syntax was developed by combining ODM [ODM 1.0] and OCL [OCL 2.0] standards. ODM was used to represent ontology vocabularies in UML notations, and OCL was used to fill the mapping expressiveness gap of ODM. However, the usability of this initially proposed abstract mapping syntax was shown, through the feedback of an initial experiment on the tool that implements this proposed mapping syntax, to be unsatisfactory. It was discovered that, to be usable by system engineers, the syntax had to be more abstract. Consequently, a more abstract UML-based mapping syntax was developed without using stereotypes for ontology vocabularies and by using syntactic sugar² of the OCL standard. This abstract mapping syntax was shown to be capable of describing the core mapping types.

4. *Develop a tool that automatically generates a concrete mapping syntax from the abstract mapping syntax.*

² In computer science, syntactic sugar means features added to a language or other formalism to make it easier to read or to express for humans to use, while it does not affect the expressiveness [Raymond 2012]. Syntactic sugar offers the user an alternative way of expressing that is more succinct or familiar notation [Howe 2014].

In order to demonstrate the proposed MOUSE approach and facilitate its evaluation, a tool (called SDI, Semantic Data Integration) was developed that automatically generates a concrete mapping syntax from the proposed abstract mapping syntax. The tool allows the automatic transformation of UML notations into the Rule Interchange Format (RIF) [RIF Core] and subsequently into SPARQL [SPARQL 2008] queries which are used as a concrete mapping syntax that is executable within the system. In this research, RIF was chosen as the intermediate format, as RIF has the potential to interoperate with other concrete mapping syntaxes. For example, there is an existing RIF syntax specification for RDF and OWL Compatibility [RIF SWC]. This strategy potentially enables the transformation between the intermediate format and other concrete mapping syntaxes such as ontology (for axiom-based integration) in the future. SPARQL was chosen as the initial concrete mapping syntax, because the Query-based integration approach using SPARQL has been shown [Keeney 2011] as the most practical approach among the three different semantic data integration approaches: Axiom-based integration using ontology modelling [Fürst 2005], Rule-based integration using the Semantic Web Rule Language SWRL [Horrocks 2004], and Query-based integration using SPARQL [Euzenat 2008].

5. *Evaluate the accuracy and usability (in terms of ease of use) of the MOUSE approach.*

Using the SDI tool as an example instance of the proposed MOUSE approach, it was shown that (i) the proposed abstract mapping syntax can correctly capture the mapping intention of participants for the core mapping types, (ii) the tool can accurately transform automatically the abstract mapping syntax into the concrete mapping syntax, (iii) the abstract mapping syntax and the SDI tool using the MOUSE approach is easy to use for system engineers. User based experiments were conducted in order to undertake these evaluations.

Analysis of the results suggest that the abstract mapping syntax (i) was able to capture correctly (for the core mapping types) the mappings without distorting the modeller's mapping intention and (ii) was able to accurately generate concrete mapping syntax from the abstract mapping syntax. The ease of use was evaluated

through a questionnaire, and the answers suggested that the abstract mapping syntax and the tool were usable for system engineers.

1.4 Contribution

The major contribution of this research is the MOUSE approach, providing a UML-based abstract syntax for mappings that abstracts away the concepts of ontology vs. existing semantic mapping syntaxes that tend to rely on a mapping practitioner to understand the idea of ontological concepts (examined by this research). The MOUSE approach is easy to use by system engineers and accurately captures their mapping intentions and automatically transforms their abstract mappings into mappings executable by the system. This approach lowers the barrier for system engineers to conduct a semantic data integration task that used to require considerable amount of ontological knowledge. The MOUSE approach allows system engineers without sufficient ontological knowledge or knowledge engineering experience to describe semantic mappings using UML notations familiar to them [Gasevic 2004].

Minor contributions are (i) defining core mapping types: direct mapping type, data range mapping type and unit transformation mapping type; and (ii) the development of SDI (Semantic Data Integration) tool. The core mapping types were identified and prioritized from mapping relationships in the industry use case [Boran 2011a], i.e. real network data collected from the Alcatel-Lucent femtocell test bed [Small Cell]. To generalize the use of core mapping types (in other words, to prove that the core mapping types are usable not only in the telecommunications domain), this research applied them in another domain (conference domain), and the core mapping types were proven usable through the examples and experiments conducted by this research. These core mapping types were used to define the scope of the mapping expressivity in order to correctly capture the usability of the abstract syntax in a practical integration situation.

The SDI tool supports precise mappings such as a data range mapping type (a mapping that requires constraint on the data value of property) and a unit transformation mapping type (a mapping that transforms the data value of property), while most applications are designed to only support a direct mapping type (an one-to-one equivalent/subsumption mapping) [Shvaiko 2005][Thomas 2009]. Moreover the SDI tool supports automated transformation of the abstract mapping syntax to the concrete mapping syntax without knowing the intricacy in writing a concrete mapping syntax for the core mapping types and also, automatically generates an integrated ontology by computing the concrete mapping syntax on the source and target ontologies. This eases the semantic data integration task for system engineers.

Publications to date related to the research:

Seung-Hwa Chung, Wei Tai, Aidan Boran and Declan O’Sullivan, “A Semantic Mapping Representation and Generation Tool using UML for System Engineers”, the 8th IEEE International Conference on Semantic Computing (ICSC 2014), California, USA, June 16-18, 2014.

1.5 Thesis Overview

Chapter 2 surveys and reviews: (i) ontology mapping relationships; (ii) ontology mapping applications; (iii) abstract mapping syntaxes that can describe ontology mapping relationships; and (iv) concrete mapping syntaxes for semantic data integration.

Chapter 3 describes three core mapping types: direct mapping type, data range mapping type and unit transformation mapping type, and the development of an abstract mapping syntax that can represent the core mapping types.

Chapter 4 describes the development of a tool that implements the proposed abstract mapping syntax. The tool is called SDI (Semantic Data Integration) tool.

Chapter 5 describes the experiment conducted by system engineers to evaluate the accuracy and ease of use of the proposed abstract mapping syntax and the developed SDI tool.

Chapter 6 discusses how well the research question was addressed by evaluating the experiment's results and suggesting future work.

2 Related Research

This chapter presents research on ontology mapping relationships in order to categorize the mapping types that define the scope of the mapping relationships supported by the abstract mapping syntax. This categorization is done in order to consider the part of the research question about to what extent will the proposed MOUSE approach allow the creation of mappings using an abstract syntax. For the part of the research question about the transformation of the abstract mappings into concrete executable mappings, this chapter also reviews existing abstract and concrete mapping syntaxes that can describe ontology mapping relationships. There is a certain ambiguity between ‘Abstract Syntax’ and ‘Concrete Syntax’ that also needs to be defined. This research defines these two categorical syntaxes as following: (1) abstract syntax is the format that is independent of particular representation, and it is more convenient and natural to manipulate than concrete syntax [Miller 1992]; and (2) concrete syntax is the format that can be derived from the abstract syntax, is ready to be used in a system, or is the target of a specific machine representation or encoding [Fondement 2005].

The evaluation criteria used in this research are their capacity to express the ontology mapping relationships discussed in the categorization of the mapping types and their usability to generate and manipulate mapping relationships in the first place by human perspective. At the end of this chapter, each discussed abstract and concrete mapping syntax is summarized in a table that describes each mapping syntax’s capability to support the mapping relationships and usability to generate mapping relationships from a human perspective.

2.1 Ontology mapping categorization

There are various kinds of patterns³ for classification in the ontology engineering field [Blomqvist 2005]. This section specifically reviews the work in Francois Scharffe and Dieter Fensel's research [Scharffe 2008], which is a renowned research study in the ontology mapping field. The research study is well-known because the researchers surveyed patterns in ontology mapping relationships and published detailed correspondence patterns that are at the top level of abstraction of the ontology alignment⁴ representation stack [Scharffe 2008]. Table 2-1 shows the categorization of generic correspondence patterns from their research.

Table 2-1 Ontology mapping categorization by F. Scharffe and D. Fensel

Generic Patterns	Equivalent correspondence pattern	
	Subsumption correspondence pattern	
	Conditional pattern e.g. a restriction on the scope of a class based on:	The occurrence of a property
		The value of a property (Data Range)
		The type of a property
	Transformation pattern e.g. a transformation of a property value about:	Datatype Conversion
		Unit Transformation
		Currency Conversion
	Union and intersection patterns	

Scharffe's research of correspondence patterns defines mapping relationships in six generic patterns: (1) Equivalent correspondence pattern: the pattern usually used to solve a terminological mismatch; (2) Subsumption correspondence pattern: the

³ In software engineering, patterns are an accepted way to facilitate and support reuse [Blomqvist 2005].

⁴ When two ontologies overlap, they can be linked together in order to enable exchange of their underlying knowledge and this link is called an alignment. An alignment between two ontologies specifies a set of correspondences, and each correspondence forms a bridge between a set of ontology entities. [Scharffe 2008]

pattern typically solves a granularity mismatch; (3) Conditional pattern: the pattern requires a restriction to narrow down the scope of an entity in an ontology to match the scope of an entity in the another ontology, e.g. a restriction on the scope of a class based on 1. the occurrence, 2. the value or 3. the type of an attribute; (4) Transformation pattern: the pattern requires a transformation of a property value to fit the corresponding property in the another ontology, e.g. the transformation about 1. data type conversion, 2. unit transformation and 3. currency conversion; and (5) Union and (6) Intersection patterns: these patterns are used to relate entities modelled at a different granularity.

In the following sections, the categorization of correspondence patterns will be used in the discussion of the abstract and concrete syntaxes for semantic mapping to illustrate their relative capabilities and limitations. In this thesis, instead of using the terminology ‘correspondence pattern’, this research uses the term ‘mapping type’ in order to express the category of mapping relationships more plainly.

2.2 Ontology mapping application

There have been investigations into ontology mapping applications that support the mapping process [Shvaiko 2005][Granitzer 2010], and many of the existing mapping applications: PROMPT [Noy 2003], CoGZ [Falconer 2007b], COMA++ [Aumuller 2005], OLA [Euzenat 2004a] and Schema Mapper [Raghavan 2005] are designed to support a matching activity of mapping process.

For example, PROMPT [Noy 2003] application offers ontology entities in a list form which separates source and target data using different lists. Fig. 2-1 shows the screenshot of PROMPT tool. In this figure, it can be observed the source and target ontology entities are listed separately as mapping candidates. This application is useful to discover many one-to-one mappings. However, the application is designed to support a matching activity and does not support actual semantic data integration using the source and target ontologies.

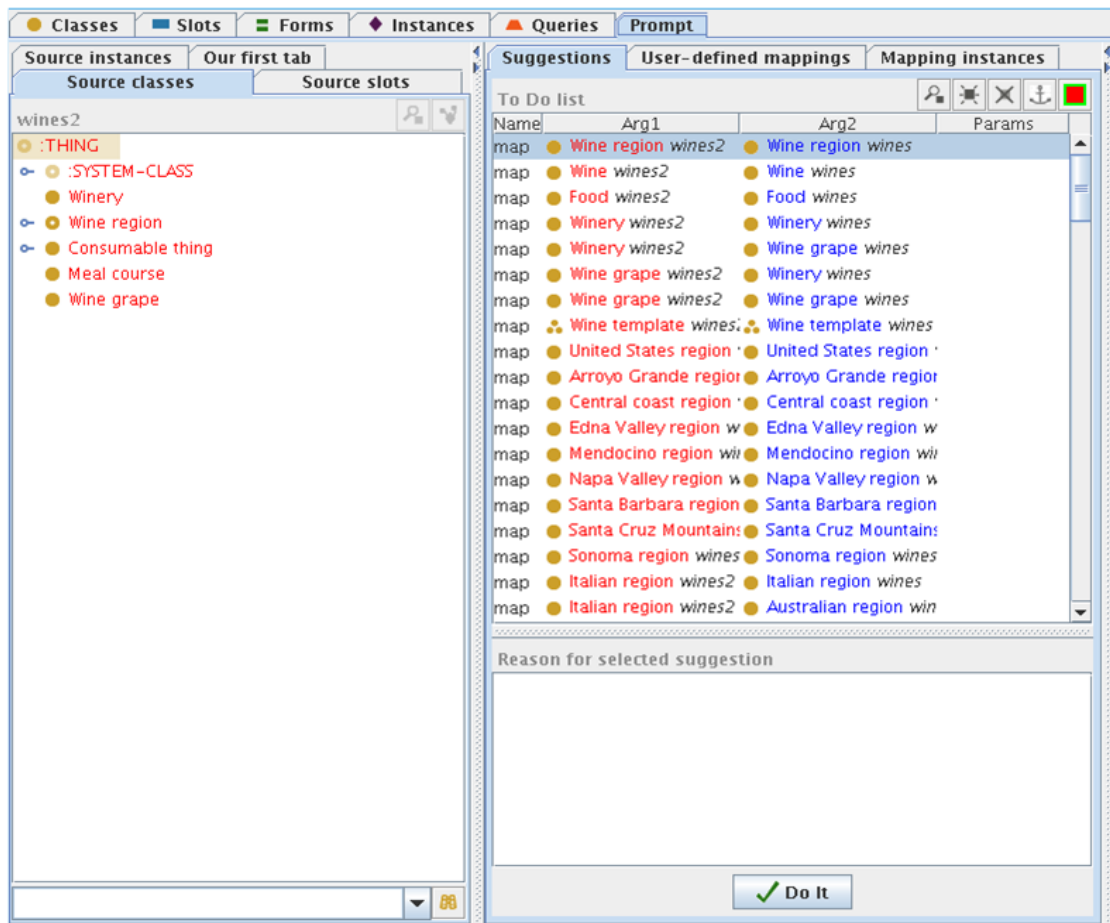


Figure 2-1 Screenshot of PROMPT by N. F. Noy and M. A. Musen

CoGZ [Falconer 2007b] application offers ontology entities in a tree form which shows the hierarchy. Fig. 2-2 shows the screenshot of CoGZ tool. In the figures, it can be observed the mappings between source and target ontologies are expressed using a line to point between ontology mapped entities. This application is useful as it expresses the ontology mappings very explicitly among other entities. However, the application supports only one-to-one mappings, and is designed to support a matching activity.

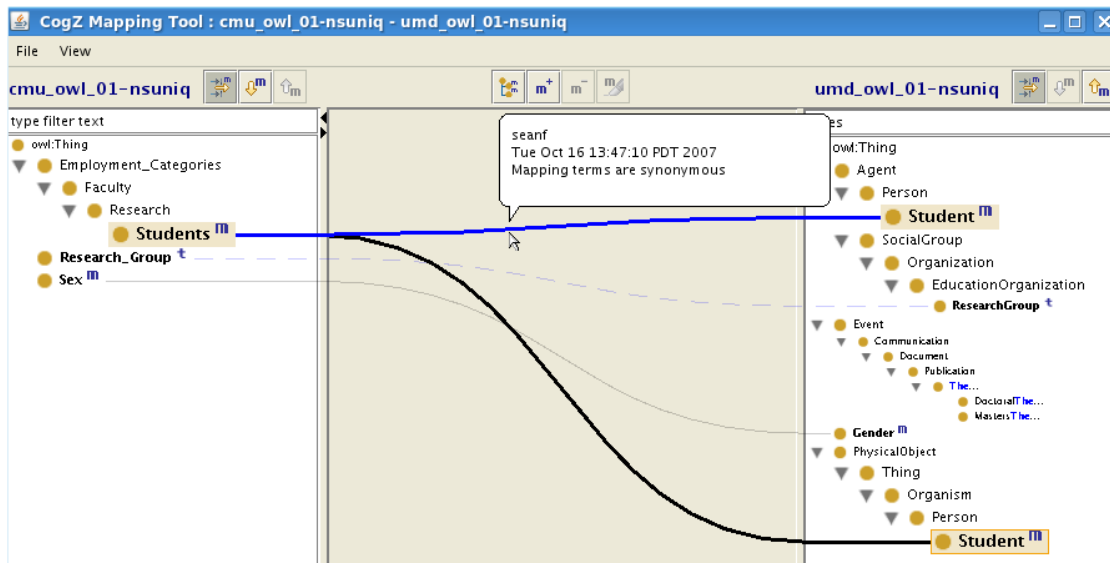


Figure 2-2 Screenshot of CoGZ by S. Falconer and M. A. Storey

This research reviewed two ontology mapping application samples according to their limitations about the mapping type supports and automated process to facilitate the semantic data integration task. There are other ontology mapping applications [Granitzer 2010]. Nevertheless, most applications are designed to only support an one-to-one equivalent/subsumption mapping [Shvaiko 2005][Thomas 2009], and the main focus of the current mapping applications is in assisting to discover the possible mappings between two ontologies [Grau 2013][Shvaiko 2013]. A matching activity is one of important mapping process. However, more automated process - to generate a concrete mapping syntax from abstract and to perform the actual integration of semantic data - is required to facilitate the semantic data integration task for non-ontology background users, and the mapping applications need to support more precise mappings such as a mapping that requires constraint on the data value of property and a mapping that transforms the data value of property to be more usable in a practical integration situation.

2.3 Abstract syntaxes for semantic mapping

This research surveyed the existing abstract syntaxes: Alignment Format, EDOAL, C-OWL, SSE, SQWRL, RIF, UML/ODM and UML/OCL, which are able to

describe semantic mapping relationships at an abstract level. These abstract syntaxes may require less ontological knowledge than concrete syntaxes, because abstract syntaxes use a more convenient and natural representation of mapping rules. This section reviews these abstract syntaxes in order to determine their characteristics and examine their capability of describing ontology mapping relationships.

2.3.1 Alignment Format

Alignment Format [Euzenat 2010] was developed to express a set of pairs of mapping elements from source and target ontologies in an XML format. This syntax describes the general mapping information with the properties: xml, level, type, onto1, onto2, and map. The detail mapping information between two entities is described by the properties: entity1, entity2, measure, and relation. This Alignment Format syntax is well known for its practical use in the Ontology Alignment Evaluation Initiative (OAEI) Campaign [OAEI 2012] as it is the formal mapping document format for this initiative. Fig. 2-3 shows an example of an equivalent mapping type using Alignment Format syntax, i.e. ‘MEDTESTFEMTO’ is equivalent to ‘MediumHandoverFemto’. The measure property refers to the accuracy of the mapping relationship, e.g. in the figure, measure value 1.0 means the equivalent relationship is very accurate. As shown in the figure, this syntax uses simple vocabularies. However, it focuses primarily on one to one mappings and does not offer much expressiveness. For example, this syntax cannot be used to describe the conditional mapping type and the transformation mapping type.

```
<map>
  <Cell>
    <entity1 rdf:resource="http://www.owl-ontologies.com/femto10.owl#MEDTESTFEMTO"/>
    <entity2 rdf:resource="http://www.owl-
ontologies.com/FemtoOntology.owl#MediumHandoverFemto"/>
    <relation>=</relation>
    <measure rdf:datatype="xsd:float">1.0</measure>
  </Cell>
</map>
```

Figure 2-3 Alignment Format syntax example for equivalent mapping type

2.3.2 EDOAL

Expressive and Declarative Ontology Alignment Language (EDOAL) [Euzenat 2011] extends the Alignment Format by extending the expressiveness. Particularly, it is designed for the representation of complex mappings. This syntax enables more precise relationships between entities, more than equivalence or subsumption relationships, by using the operators: and, or, not, compose, inverse, transitive, reflexive, and symmetric. This enables more mapping expressiveness than the Alignment Format syntax. One example of this expressiveness is the conditional mapping type. Fig. 2-4 shows an example of a conditional mapping type that has a constraint on the value of a datatype property (data range) using EDOAL syntax, i.e. ‘Femto’ is equivalent to ‘MediumHandoverFemto’ with the data range constraint ($40 < \text{‘MTS_Handover_Failure_Rate’} < 60$ and $40 < \text{‘UMTS_Handover_Failure_Rate’} < 60$).

```
<map>
  <cell>
    <entity1>
      <edoal:Class>
        <edoal:and rdf:parseType="Collection">
          <edoal:Class rdf:about="&ont1;Femto"/>
          <edoal:AttributeValueRestriction>
            <edoal:onAttribute>
              <edoal:Property rdf:about="&ont1; MTS_Handover_Failure_Rate"/>
            </edoal:onAttribute>
            <edoal:comparator rdf:resource="&xsd;lower-than"/>
            <edoal:value>60</edoal:value>
          </edoal:AttributeValueRestriction>
          <edoal:AttributeValueRestriction>
            <edoal:onAttribute>
              <edoal:Property rdf:about="&ont1; MTS_Handover_Failure_Rate"/>
            </edoal:onAttribute>
            <edoal:comparator rdf:resource="&xsd;greater-than"/>
            <edoal:value>40</edoal:value>
          </edoal:AttributeValueRestriction>
          <edoal:AttributeValueRestriction>
            <edoal:onAttribute>
              <edoal:Property rdf:about="&ont1; UMTS_Handover_Failure_Rate"/>
            </edoal:onAttribute>
            <edoal:comparator rdf:resource="&xsd;lower-than"/>
            <edoal:value>60</edoal:value>
          </edoal:AttributeValueRestriction>
          <edoal:AttributeValueRestriction>
            <edoal:onAttribute>
              <edoal:Property rdf:about="&ont1; UMTS_Handover_Failure_Rate"/>
            </edoal:onAttribute>
            <edoal:comparator rdf:resource="&xsd;greater-than"/>
            <edoal:value>40</edoal:value>
        </edoal:and>
      </edoal:Class>
    </entity1>
  </cell>
</map>
```

```

        </edoal:AttributeValueRestriction>
    </edoal:and>
</edoal:Class>
</entity1>
<entity2>
    <edoal:Class rdf:about="&ont2;MediumHandoverFemto"/>
</entity2>
<relation>Equivalence</relation>
<measure rdf:datatype="&xsd;float">1.0</measure>
</cell>
</map>

```

Figure 2-4 EDOAL syntax example for conditional mapping type

Fig. 2-5 shows an example of a transformation mapping type that transforms the data value of a property (unit transformation) using EDOAL syntax, i.e. the value of ‘VS_NumUserBits_EDCH_PS’ is equivalent to the value of ‘UL_E-DCH_Data_Volume_on_RLC_SDU’ with the unit transformation (‘VS_NumUserBits_EDCH_PS’ * 900). However, this syntax supports a limited transformation mapping type, which transforms only the value of datatype properties expressed in the entity clause. It is not possible to express the transformation mapping that requires the combination of multiple datatype properties which belong to a certain class instance and are not expressed as a source or target entity.

```

<map> <Cell>
  <entity1>
    <edoal:Property rdf:about="&ont1;VS_NumUserBits_EDCH_PS"/>
  </entity1>
  <entity2>
    <edoal:Property rdf:about="&ont2;UL_E-DCH_Data_Volume_on_RLC_SDU"/>
  </entity2>
  <measure rdf:datatype='&xsd;float'>1.0</measure>
  <relation>Equivalence</relation>
  <edoal:transformation>
    <edoal:Transformation edoal:type="-o">
      <edoal:entity1>
        <edoal:Apply edoal:operator="multiply">
          <edoal:arguments rdf:parseType="Collection">
            <edoal:Property rdf:about="&ont1;VS_NumUserBits_EDCH_PS"/>
            <edoal:Literal edoal:string="900"/>
          </edoal:arguments>
        </edoal:Apply>
      </edoal:entity1>
      <edoal:entity2>
        <edoal:Property><edoal:compose rdf:parseType="Collection"/></edoal:Property>
      </edoal:entity2>
    </edoal:Transformation>
  </edoal:transformation>
</Cell> </map>

```

Figure 2-5 EDOAL syntax example for transformation mapping type

2.3.3 C-OWL

Contextualized OWL (C-OWL) [Bouquet 2003] has been developed to map contextualized ontologies (local models encoding the view of a group of people on a domain) to a shared ontology (a shared model encoding a common view of different parties on some domains). C-OWL keeps mappings explicitly and extends the OWL syntax. It uses bridge rules to express the mapping. These rules have five different attributes, i.e. equiv, onto, into, compat, and incompat. Each of these attributes describes a particular relationship, respectively: equivalent, less specific (subsumes), more specific (subsumedBy), compatible, and disjoint. Fig. 2-6 shows an example of an equivalent mapping type using C-OWL syntax, i.e. ‘User’ is equivalent to ‘Account’.

```
<owl:bridgRule owl:br-type="equiv">
  <owl:sourceConcept
rdf:resource="http://www.semanticweb.org/ontologies/SampleTelcoOnt2.owl#User"/>
  <owl:targedConcept
rdf:resource="http://www.semanticweb.org/ontologies/SampleTelcoOnt1.owl#Account"/>
</owl:bridgRule>
```

Figure 2-6 C-OWL syntax example for equivalent mapping type

Fig. 2-7 shows an example of a subsumption mapping type using C-OWL syntax, i.e. ‘WiFi’ is subsumed by ‘Packet_Network’. As shown in the figures, this syntax represents the mappings in a simple and explicit way between source and target ontologies. However, because of limited vocabularies for mapping expressions, this syntax cannot express the conditional mapping type and the transformation mapping type.

```
<owl:bridgRule owl:br-type="into">
  <owl:sourceConcept
rdf:resource="http://www.semanticweb.org/ontologies/SampleTelcoOnt2.owl#WiFi"/>
  <owl:targedConcept
rdf:resource="http://www.semanticweb.org/ontologies/SampleTelcoOnt1.owl#Packet_Network"/>
</owl:bridgRule>
```

Figure 2-7 C-OWL syntax example for subsumption mapping type

2.3.4 SSE

SPARQL Syntax Expressions (SSE) [JENA 2011] has been developed by the JENA Apache incubator project to abstractly express SPARQL algebra expressions. This

syntax is capable of describing mappings that may require expression of data range constraints or unit transformations. Fig. 2-8 shows an example of a conditional mapping type that has a constraint on the value of a datatype property (data range) using SSE syntax, i.e. an instance that satisfies the data range condition ($40 < \text{'MTS_Handover_Failure_Rate'} < 60$ and $40 < \text{'UMTS_Handover_Failure_Rate'} < 60$) is the instance of 'MediumHandoverFemto'.

```
(prefix ( (ftoc: <http://sample.org/ontology/test/femto> )
(distinct
  (project ( [triple ?femto a ftoc:MediumHandoverFemto] )
    (leftjoin
      (BGP
        [triple ?femto ftoc:MTS_Handover_Failure_Rate ?v1]
        [triple ?femto ftoc:UMTS_Handover_Failure_Rate ?v2]
      )
      ( FILTER (<= 40 ?v1) )
      ( FILTER (>= 60 ?v1) )
      ( FILTER (<= 40 ?v2) )
      ( FILTER (>= 60 ?v2) )
    )
  )))
```

Figure 2-8 SSE syntax example for conditional mapping type

Fig. 2-9 shows an example of a transformation mapping type that transforms the data value of a property (unit transformation) using SSE syntax, i.e. the value of 'Handover_Failure_Gauge' is the result of unit transformation operation ($\text{'MTS_Handover_Failure_Rate'} + \text{'UMTS_Handover_Failure_Rate'} / 100$).

```
(prefix ( (ftoc: <http://sample.org/ontology/test/femto> )
(distinct
  (project ( [triple ?femto ftoc:Handover_Failure_Gauge ?value] )
    (leftjoin
      (BGP
        [triple ?femto ftoc:MTS_Handover_Failure_Rate ?v1]
        [triple ?femto ftoc:UMTS_Handover_Failure_Rate ?v2]
      )
      ( BIND (?value ( (?v1 + ?v2) / 100) )
    )
  )))
```

Figure 2-9 SSE syntax example for transformation mapping type

2.3.5 SQWRL

Semantic Query-Enhanced Web Rule Language (SQWRL) [SQWRL 2009] has been developed by the Stanford Center for Biomedical Information Research Lab in order to express rules abstractly and to query OWL ontologies. This syntax has been used

in the Protégé tool [Protégé Project] to facilitate the creation and modification of the SWRL syntax. Its query structure is based on SWRL and it supports SWRL expressivity with more abstracted expressions. Fig. 2-10 shows an example of a conditional mapping type that has a constraint on the value of a datatype property (data range) using SQWRL syntax, i.e. ‘Femto’ is equivalent to ‘MediumHandoverFemto’ with the data range constraint ($40 \leq \text{‘MTS_Handover_Failure_Rate’} \leq 60$ and $40 \leq \text{‘UMTS_Handover_Failure_Rate’} \leq 60$).

Femto(?x), MTS_Handover_Failure_Rate(?x, ?v2), UMTS_Handover_Failure_Rate(?x, ?v1), greaterThanOrEqualTo(?v1, "40"^^long), greaterThanOrEqualTo(?v2, "40"^^long), lessThanOrEqualTo(?v1, "60"^^long), lessThanOrEqualTo(?v2, "60"^^long) → MediumHandoverFemto(?x)
--

Figure 2-10 SQWRL syntax example for conditional mapping type

Fig. 2-11 shows an example of a transformation mapping type that transforms the data value of a property (unit transformation) using SQWRL syntax, i.e. the value of ‘BSR_cluster_to_MTS_underlay_Intra_Frequency_Hard_Handover_Failure_Rate’ is the result of unit transformation operation ($(1 - (\text{‘VS_HHO_SuccBsrUmtsIntraFreq’} / \text{‘VS_HHO_AttBsrUmtsIntraFreq’}) * 100)$).

Femto(?x), hasNPMDData(?x, ?pm1), VS_HHO_AttBsrUmtsIntraFreq(?pm1, ?v0), VS_HHO_SuccBsrUmtsIntraFreq(?pm1, ?v1), divide(?div1, ?v1, ?v0), subtract(?sub1, "1"^^long, ?div1), multiply(?ans, "100"^^long, ?sub1) → BSR_cluster_to_MTS_underlay_Intra_Frequency_Hard_Handover_Failure_Rate(?x, ?ans)
--

Figure 2-11 SQWRL syntax example for transformation mapping type

2.3.6 RIF

Rule Interchange Format (RIF) [RIF 2010] has been developed by the W3C RIF Working Group to create a standard for exchanging rules among rule systems. This syntax uses many externally defined functions, and these functions offer good expressiveness to describe mappings that may require expressing data range constraints or unit transformation. Fig. 2-12 shows an example of a conditional mapping type that has a constraint on the value of a datatype property (data range) using RIF syntax, i.e. ‘Femto’ is equivalent to ‘MediumHandoverFemto’ with the data range constraint ($40 < \text{‘MTS_Handover_Failure_Rate’} < 60$ and $40 < \text{‘UMTS_Handover_Failure_Rate’} < 60$).

```

Document (
  Prefix ( ftoc <http://sample.org/ontology/test/femto#> )
  Group (
    Forall ?Femto ?MediumHandoverFemto ?MTS_Handover_Failure_Rate
    ?UMTS_Handover_Failure_Rate (
      ftoc:equal( ?Femto ?MediumHandoverFemto ) :-
      And (
        ftoc:Femto( ?femto )
        ftoc:MTS_Handover_Failure_Rate( ?femto ?v1 )
        ftoc:UMTS_Handover_Failure_Rate( ?femto ?v2 )
        External( pred:numeric-greater-than(?v1 40) )
        External( pred:numeric-less-than(?v1 60) )
        External( pred:numeric-greater-than(?v2 40) )
        External( pred:numeric-less-than(?v 60) )
      )))
)

```

Figure 2-12 RIF syntax example for conditional mapping type

Fig. 2-13 shows an example of a transformation mapping type that transforms the data value of a property (unit transformation) using RIF syntax, i.e. the value of ‘Handover_Failure_Gauge’ is the result of unit transformation operation (‘MTS_Handover_Failure_Rate’ + ‘UMTS_Handover_Failure_Rate’ / 100).

```

Document (
  Prefix ( ftoc <http://sample.org/ontology/test/femto#> )
  Group (
    Forall ?Femto ?Handover_Failure_Gauge ?MTS_Handover_Failure_Rate
    ?UMTS_Handover_Failure_Rate (
      ftoc:Handover_Failure_Gauge( ?Femto ?value ) :-
      And (
        ftoc:Femto( ?femto )
        ftoc:MTS_Handover_Failure_Rate( ?femto ?v1 )
        ftoc:UMTS_Handover_Failure_Rate( ?femto ?v2 )
        ?v3 = External( func:add(?v1 ?v2) )
        ?value = External( func:divide(?v3 100) )
      )))
)

```

Figure 2-13 RIF syntax example for transformation mapping type

2.3.7 UML/ODM

Unified Modeling Language (UML) [UML 2011] is widely used in many areas to model data structure, application structure, application behaviour, and business processes in a standardized and easy-to-understand manner [Kogut 2002]. It offers a graphical representation of entities and relationships, which is more readable to non-technical users than a text based expression. Attempts have been made [Baclawski 2001][Brockmans 2004][Brockmans 2006] to create stereotypes in UML to enable the representation of ontologies using UML. Subsequently, the OMG group [OMG

2010] has standardized ontology definition metamodel specification, called the ODM [ODM 1.0]. This standard describes how ontological terms can be mapped to UML notations. For example, the UML generalization with `<<rdfsSubClassOf>>` stereotype is used to represent “subClassOf” vocabulary in RDFS syntax [RDF Schema]. Fig. 2-14 shows an example of UML notations that represent an ontology class and property. In this figure, the UML notations represent a ‘Network Device’ ontology class that has ‘hasPort’ ontology property whose value is an instance of ‘Interface’ ontology class.

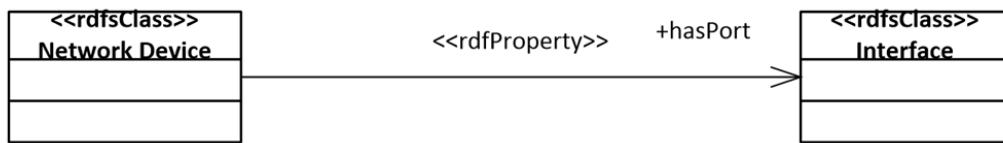


Figure 2-14 UML/ODM example for ontology class and property

Fig. 2-15 shows an example of UML notations that represent an ontology datatype property. In this figure, the UML notations represent a ‘Byte’ ontology datatype property that has the value type ‘Integer’.

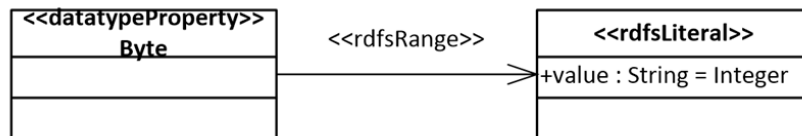


Figure 2-15 UML/ODM example for ontology datatype property

Fig. 2-16 shows an example of an equivalent mapping type using UML notations according to ODM specification, i.e. ‘Router’ is equivalent to ‘Layer3Switch’.

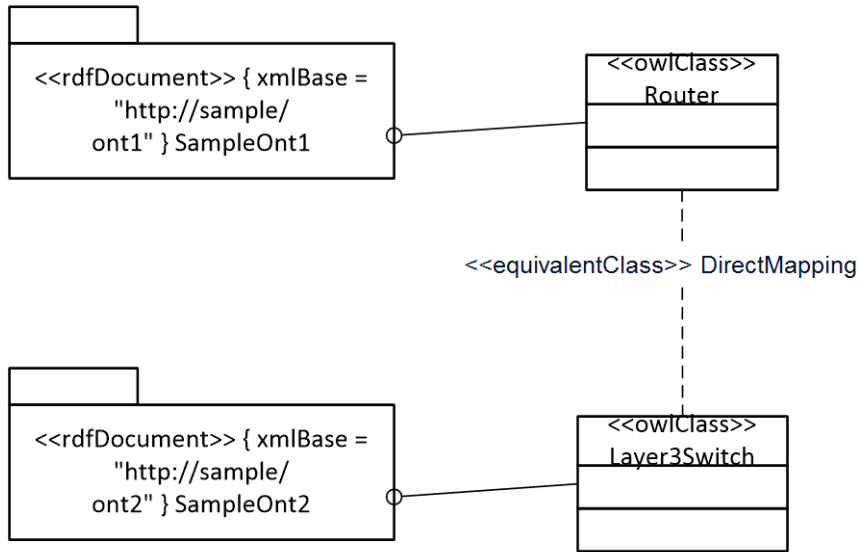


Figure 2-16 UML/ODM example for equivalent mapping type

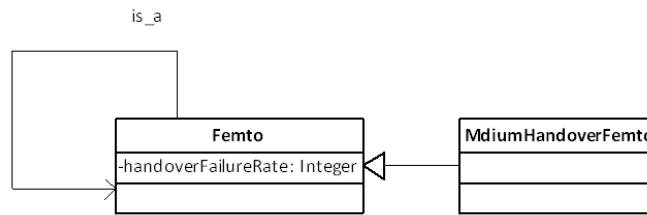
There are limitations on describing mapping relationships using ODM. Firstly this ODM specification only supports OWL version 1 and shares the limitations of OWL version 1, i.e. the lack of the ability to describe the data range constraints (this expression is supported from OWL version 2 [Horrocks 2010]), and secondly this specification does not support arithmetic expressions.

2.3.8 UML/OCL

Object Constraint Language (OCL) [OCL 2.0] is a formal language that enables UML constraints to be modelled in an unambiguous means. It avoids contradictions in the intended UML semantics by describing rules between UML notations. For example, OCL can describe the mapping that constrains property values using the comparison or arithmetic operation between UML notations. This language is expressed in UML constraint⁵ notation. Fig. 2-17 shows an example of a conditional mapping type that has a constraint on the value of a datatype property (data range) using OCL syntax, i.e. an instance in 'Femto' class that satisfies the data range

⁵ A UML constraint is a condition or restriction expressed in natural language text or in a machine readable language for the purpose of declaring some of the semantics of an element. The constraint can be predefined (e.g. OCL) in UML or user-defined. The syntax and interpretation of a user-defined UML constraint is a tool responsibility. [UML 2.0]

condition $(50 < \text{'handoverFailureRate'} < 90)$ is the instance of 'MediumHandoverFemto'.

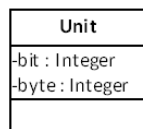


```

CONTEXT Femto INV:
  SELF.is_a -> forAll( f:Femto | f.handoverFailureRate < 90 ) AND
  SELF.is_a -> forAll( f:Femto | f.handoverFailureRate > 50 )
IMPLIES
  SELF.oclIsKindOf(MediumHandoverFemto)
  
```

Figure 2-17 UML/OCL example for conditional mapping type

Fig. 2-18 shows an example of a transformation mapping type that transforms the data value of a property (unit transformation) using OCL syntax, i.e. the value of 'byte' is the result of unit transformation operation ('bit' * 8).



```

CONTEXT Unit
INV: SELF.bit.oclIsTypeOf( Integer )
INV: SELF.byte.oclIsTypeOf( Integer )
IMPLIES
  SELF.byte = SELF.bit * 8
  
```

Figure 2-18 UML/OCL example for transformation mapping type

2.4 Concrete syntaxes for semantic mapping

This section reviews three concrete syntaxes: OWL, SWRL and SPARQL for their ability to represent mappings that are directly executable by the system such as the JENA query execution engine [JENA 2013], which enables the automation of data processing through the defined functions. These concrete syntaxes can be derived from the abstract syntax as an executable output, and these three syntaxes represent

three different approaches of integration that are popular: Axiomatic, Rule and Query [Keeney 2011].

2.4.1 Using OWL for semantic mapping

Web Ontology Language (OWL) [Bechhofer 2004] is a semantic markup language for publishing and sharing ontologies on the World Wide Web. This language has XML elements such as `equivalentClass` and `equivalentProperty` that can be used to describe mappings. If the application is ontologically based, this syntax is generally considered first to use as a concrete syntax to perform the integration task. However, the syntax has some limitations. First, mappings can be more manifest and easier to modify in a separate document. Using OWL for the mappings mixes the definitions of the concepts and mapping information [Keeney 2011]. Second, this syntax does not support data value transformation (arithmetic operations) that might be required for mapping. Fig. 2-19 shows an example of a conditional mapping type that has a constraint on the value of a datatype property (data range) using OWL syntax. In this example, ‘Femto’ class with the property ‘MTS_Handover_Failure_Rate’ value between 40 and 60, and the property ‘UMTS_Handover_Failure_Rate’ value between 40 and 60 is equivalent to ‘MediumHandoverFemto’ class.

```

<owl:Class rdf:about="http://www.owl-ontologies.com/femto.owl#MediumHandoverFemto">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="http://www.owl-ontologies.com/femto.owl#Femto"/>
        <owl:Restriction>
          <owl:onProperty rdf:resource="http://www.owl-ontologies.com/femto.owl#MTS_Handover_Failure_Rate"/>
          <owl:someValuesFrom>
            <rdfs:Datatype>
              <owl:onDatatype rdf:resource="&xsd;integer"/>
              <owl:withRestrictions rdf:parseType="Collection">
                <rdf:Description>
                  <xsd:maxInclusive rdf:datatype="&xsd;integer">60</xsd:maxInclusive>
                </rdf:Description>
                <rdf:Description>
                  <xsd:minInclusive rdf:datatype="&xsd;integer">40</xsd:minInclusive>
                </rdf:Description>
              </owl:withRestrictions>
            </rdfs:Datatype>
          </owl:someValuesFrom>
        </owl:Restriction>
        <owl:Restriction>
          <owl:onProperty rdf:resource="http://www.owl-ontologies.com/femto.owl#UMTS

```

```

_Handover_Failure_Rate"/>
  <owl:someValuesFrom>
    <rdfs:Datatype>
      <owl:onDatatype rdf:resource="&xsd;integer"/>
      <owl:withRestrictions rdf:parseType="Collection">
        <rdf:Description>
          <xsd:maxInclusive rdf:datatype="&xsd;integer">60</xsd:maxInclusive>
        </rdf:Description>
        <rdf:Description>
          <xsd:minInclusive rdf:datatype="&xsd;integer">40</xsd:minInclusive>
        </rdf:Description>
      </owl:withRestrictions>
    </rdfs:Datatype>
  </owl:someValuesFrom>
</owl:Restriction>
</owl:intersectionOf>
</owl:Class>
</owl:equivalentClass>
</owl:Class>

```

Figure 2-19 OWL syntax example for conditional mapping type

2.4.2 SWRL

Semantic Web Rule Language (SWRL) [Horrocks 2004] is a rule language used to express the rules in an ontology in addition to concept definitions. This syntax is based on a combination of OWL and RuleML [Boley 2011], which is a sublanguage of the Rule Markup Language. The syntax includes a high-level abstract syntax for Horn-like rules [Horn 1951] and uses the following vocabularies: XSD, OWLX, RuleML, SWRLX, and SWRLB. These vocabularies offer good expressiveness to describe mappings that may require expressing data range constraints or data value transformations (arithmetic operation). However, this syntax is difficult to generate for its rigidity that often requires domain experts to have intensive training on writing rules. Fig. 2-20 shows an example of a conditional mapping type that has a constraint on the value of a datatype property (data range) using SWRL syntax. In this example, ‘Femto’ class instances that have the property ‘UMTS_Handover_Failure_Rate’ value greater than or equal to 40 and less than or equal to 60, and the property ‘MTS_Handover_Failure_Rate’ value greater than or equal to 40 and less than or equal to 60 are the instances of ‘MediumHandoverFemto’ class.

```

<swrl:Imp>
  <swrl:body>
    <swrl:AtomList><rdf:first>
      <swrl:ClassAtom>
        <swrl:classPredicate rdf:resource="#Femto"/>

```

```

<swrl:argument1 rdf:resource="#x"/>
</swrl:ClassAtom></rdf:first><rdf:rest><swrl:AtomList><rdf:first>
<swrl:DatavaluedPropertyAtom>
<swrl:propertyPredicate rdf:resource="#UMTS_Handover_Failure_Rate"/>
<swrl:argument1 rdf:resource="#x"/>
<swrl:argument2 rdf:resource="#v1"/>
</swrl:DatavaluedPropertyAtom></rdf:first><rdf:rest><swrl:AtomList><rdf:first>
<swrl:DatavaluedPropertyAtom>
<swrl:propertyPredicate rdf:resource="#MTS_Handover_Failure_Rate"/>
<swrl:argument1 rdf:resource="#x"/>
<swrl:argument2 rdf:resource="#v2"/>
</swrl:DatavaluedPropertyAtom></rdf:first><rdf:rest><swrl:AtomList><rdf:first>
<swrl:BuiltinAtom>
<swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#greaterThanOrEqual"/>
<swrl:arguments><rdf:List><rdf:first rdf:resource="#v1"/><rdf:rest><rdf:List>
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#long">40</rdf:first>
<rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
</rdf:List></rdf:rest></rdf:List>
</swrl:arguments>
</swrl:BuiltinAtom></rdf:first><rdf:rest><swrl:AtomList><rdf:first>
<swrl:BuiltinAtom>
<swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#lessThanOrEqual"/>
<swrl:arguments>
<rdf:List><rdf:rest><rdf:List>
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#long">60</rdf:first>
<rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
</rdf:List></rdf:rest><rdf:first rdf:resource="#v1"/></rdf:List></swrl:arguments>
</swrl:BuiltinAtom></rdf:first><rdf:rest><swrl:AtomList><rdf:first>
<swrl:BuiltinAtom>
<swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#greaterThanOrEqual"/>
<swrl:arguments><rdf:List><rdf:first rdf:resource="#v2"/><rdf:rest><rdf:List>
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#long">40</rdf:first>
<rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
</rdf:List></rdf:rest></rdf:List>
</swrl:arguments>
</swrl:BuiltinAtom></rdf:first><rdf:rest><swrl:AtomList><rdf:rest
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/><rdf:first><swrl:BuiltinAtom>
<swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#lessThanOrEqual"/>
<swrl:arguments><rdf:List><rdf:first rdf:resource="#v2"/><rdf:rest><rdf:List>
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#long">60</rdf:first>
<rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
</rdf:List></rdf:rest></rdf:List>
</swrl:arguments>
</swrl:BuiltinAtom>
</rdf:first></swrl:AtomList></rdf:rest></swrl:AtomList></rdf:rest></swrl:AtomList></rdf:rest></swrl:AtomList>
</swrl:body>
<swrl:head>
<swrl:AtomList><rdf:first>
<swrl:ClassAtom>
<swrl:classPredicate rdf:resource="#MediumHandoverFemto"/>
<swrl:argument1 rdf:resource="#x"/>
</swrl:ClassAtom></rdf:first><rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"/></swrl:AtomList>
</swrl:head>
</swrl:Imp>

```

Figure 2-20 SWRL syntax example for conditional mapping type

Fig. 2-21 shows an example of a transformation mapping type that transforms the data value of a property (unit transformation) using SWRL syntax. In this example, the property ‘MTS_Handover_Failure_Rate’ value of the instance in ‘Femto’ class will be assigned by the transformation of values of the ‘hasNPMDData’ instance properties ‘VS_HHO_AttBsrUmtsIntraFreq’ and ‘VS_HHO_SuccBsrUmtsIntraFreq’ by the arithmetic operation ((1 subtract by (‘VS_HHO_SuccBsrUmtsIntraFreq’ value divide by ‘VS_HHO_AttBsrUmtsIntraFreq’ value)) multiply by 100).

```

<swrl:Imp>
<swrl:body>
<swrl:AtomList><rdf:first><swrl:ClassAtom><swrl:classPredicate rdf:resource="http://www.owl-
ontologies.com/Ontology1304066878.owl#Femto"/><swrl:argument1 rdf:resource="http://www.owl-
ontologies.com/Ontology1304066878.owl#x"/></swrl:ClassAtom></rdf:first><rdf:rest><swrl:AtomL
ist><rdf:rest><swrl:AtomList><rdf:first><swrl:DatavaluedPropertyAtom><swrl:propertyPredicate
rdf:resource="http://www.owl-
ontologies.com/Ontology1304066878.owl#VS_HHO_AttBsrUmtsIntraFreq"/><swrl:argument1
rdf:resource="http://www.owl-ontologies.com/Ontology1304066878.owl#pm1"/><swrl:argument2
rdf:resource="http://www.owl-
ontologies.com/Ontology1304066878.owl#v0"/></swrl:DatavaluedPropertyAtom></rdf:first><rdf:res
t><swrl:AtomList><rdf:rest><swrl:AtomList><rdf:first>
<swrl:BuiltinAtom><swrl:builtin rdf:resource="&swrlb;divide"/>
<swrl:arguments rdf:parseType="Collection"><rdf:Description rdf:about="http://www.owl-
ontologies.com/Ontology1304066878.owl#div1"/><rdf:Description rdf:about="http://www.owl-
ontologies.com/Ontology1304066878.owl#v1"/><rdf:Description rdf:about="http://www.owl-
ontologies.com/Ontology1304066878.owl#v0"/></swrl:arguments></swrl:BuiltinAtom></rdf:first><
rdf:rest><swrl:AtomList><rdf:rest>
<swrl:BuiltinAtom><swrl:builtin rdf:resource="&swrlb;multiply"/>
<swrl:arguments><rdf:Description><rdf:type rdf:resource="&rdf;List"/><rdf:first
rdf:resource="http://www.owl-
ontologies.com/Ontology1304066878.owl#ans"/><rdf:rest><rdf:Description><rdf:type
rdf:resource="&rdf;List"/><rdf:first rdf:datatype="&xsd;long">100</rdf:first><rdf:rest
rdf:parseType="Collection"><rdf:Description rdf:about="http://www.owl-
ontologies.com/Ontology1304066878.owl#sub1"/></rdf:rest></rdf:Description></rdf:rest></rdf:Desc
ription></swrl:arguments></swrl:BuiltinAtom></rdf:first><rdf:rest><swrl:AtomList><rdf:rest
rdf:resource="&rdf:nil"/><rdf:rest>
<swrl:BuiltinAtom><swrl:builtin rdf:resource="&swrlb;subtract"/>
<swrl:arguments><rdf:Description><rdf:type rdf:resource="&rdf;List"/><rdf:first
rdf:resource="http://www.owl-
ontologies.com/Ontology1304066878.owl#sub1"/><rdf:rest><rdf:Description><rdf:type
rdf:resource="&rdf;List"/><rdf:first rdf:datatype="&xsd;long">1</rdf:first><rdf:rest
rdf:parseType="Collection"><rdf:Description rdf:about="http://www.owl-
ontologies.com/Ontology1304066878.owl#div1"/></rdf:rest></rdf:Description></rdf:rest></rdf:Desc
ription></swrl:arguments></swrl:BuiltinAtom></rdf:first></swrl:AtomList>
</rdf:rest></swrl:AtomList>
</rdf:rest></swrl:AtomList>
</rdf:rest><rdf:first><swrl:DatavaluedPropertyAtom><swrl:propertyPredicate
rdf:resource="http://www.owl-
ontologies.com/Ontology1304066878.owl#VS_HHO_SuccBsrUmtsIntraFreq"/><swrl:argument1
rdf:resource="http://www.owl-ontologies.com/Ontology1304066878.owl#pm1"/><swrl:argument2
rdf:resource="http://www.owl-
ontologies.com/Ontology1304066878.owl#v1"/></swrl:DatavaluedPropertyAtom></rdf:first></swrl:
AtomList>

```

```

</rdf:rest></swrl:AtomList>
</rdf:rest><rdf:first><swrl:IndividualPropertyAtom><swrl:propertyPredicate
rdf:resource="http://www.owl-
ontologies.com/Ontology1304066878.owl#hasNPMDData"/><swrl:argument2
rdf:resource="http://www.owl-ontologies.com/Ontology1304066878.owl#pm1"/><swrl:argument1
rdf:resource="http://www.owl-
ontologies.com/Ontology1304066878.owl#x"/></swrl:IndividualPropertyAtom></rdf:first></swrl:At
omList>
</rdf:rest></swrl:AtomList>
</swrl:body>
<swrl:head>
<swrl:AtomList><rdf:rest
rdf:resource="&rdf:nil"/><rdf:first><swrl:DatavaluedPropertyAtom><swrl:propertyPredicate
rdf:resource="http://www.owl-
ontologies.com/Ontology1304066878.owl#MTS_Handover_Failure_Rate"/><swrl:argument2
rdf:resource="http://www.owl-ontologies.com/Ontology1304066878.owl#ans"/><swrl:argument1
rdf:resource="http://www.owl-
ontologies.com/Ontology1304066878.owl#x"/></swrl:DatavaluedPropertyAtom></rdf:first></swrl:A
tomList>
</swrl:head>
</swrl:Imp>

```

Figure 2-21 SWRL syntax example for transformation mapping type

2.4.3 SPARQL

SPARQL Protocol and RDF Query Language (SPARQL) [SPARQL 2008] is a query language for RDF [RDF Concepts], which is a directed and labelled graph data format for representing information. SPARQL can transform semantic data at query time using a CONSTRUCT statement that can generate the desired RDF graph based on the results of the query. This syntax supports conditional and transformation mapping types that may require expressing data range constraints or data value transformations (arithmetic operation). However, this syntax requires ontological knowledge such as RDF statements to generate a query and specific query writing skills including built-in functions [SPARQL 2008]. Fig. 2-22 shows an example of a conditional mapping type that has a constraint on the value of a datatype property (data range) using SPARQL syntax. In this example, ‘Femto’ class instances that have the property ‘MTS_Handover_Failure_Rate’ and ‘UMTS_Handover_Failure_Rate’ values within the comparison operation ($40 \leq \text{‘UMTS_Handover_Failure_Rate’} \leq 60$ and $40 \leq \text{‘MTS_Handover_Failure_Rate’} \leq 60$) are the instances of ‘MediumHandoverFemto’ class.


```

CONSTRUCT { $femto a :MediumHandoverFemto . }
WHERE {
  $femto :MTS_Handover_Failure_Rate $v2 ;
  :UMTS_Handover_Failure_Rate $v1 .
  FILTER ( 40 <= $v1 && $v1 <= 60 && 40 <= $v2 && $v2 <= 60 )
}

```

Figure 2-22 SPARQL syntax example for conditional mapping type

Fig. 2-23 shows an example of a transformation mapping type that transforms the data value of a property (unit transformation) using SPARQL syntax. In this example, the ‘Femto’ instance will have the property ‘MTS_Handover_Failure_Rate’ value that is the transformation result of the values of ‘hasNPMDData’ instance properties ‘VS_HHO_AttBsrUmtsIntraFreq’ and ‘VS_HHO_SuccBsrUmtsIntraFreq’ by the arithmetic operation $((1 - (\text{‘VS_HHO_SuccBsrUmtsIntraFreq’ value} / \text{‘VS_HHO_AttBsrUmtsIntraFreq’ value})) * 100)$.

```

CONSTRUCT { $femto :MTS_Handover_Failure_Rate $kpi . }
WHERE {
  $femto a :Femto ;
  :hasNPMDData $npm .
  $npm :VS_HHO_AttBsrUmtsIntraFreq $v0 .
  $npm :VS_HHO_SuccBsrUmtsIntraFreq $v1 .
  BIND ( (1 - ($v1 / $v0)) * 100 as $kpi )
}

```

Figure 2-23 SPARQL syntax example for transformation mapping type

2.5 Summary Tables

This section summarizes each discussed mapping syntax’s capability to support mapping types and their relative usability to generate mapping relationships in the first place by human perspective in the opinion of the author of this thesis as shown in table 2-2 and table 2-3.

Table 2-2 Summary table of the supported mapping types for concrete and abstract mapping syntaxes

Concrete Syntax	Supported Mapping Types					
	E. Type	S. Type	C. Type	T. Type	U. Type	I. Type
OWL 2	X	X	X		X	X
SWRL	X	X	X	X	X	X
SPARQL	X	X	X	X	X	X

Abstract Syntax	Supported Mapping Types					
	E. Type	S. Type	C. Type	T. Type	U. Type	I. Type
Alignment Format	X	X				
EDOAL	X	X	X	X	X	X
C-OWL	X	X				
SSE	X	X	X	X	X	X
SQWRL	X	X	X	X	X	X
RIF	X	X	X	X	X	X
UML/ODM	X	X			X	X
UML/OCL	X	X	X	X	X	X

E. Type: Equivalent mapping type **C. Type:** Conditional mapping type **U. Type:** Union mapping type
S. Type: Subsumption mapping type **T. Type:** Transformation mapping type **I. Type:** Intersection mapping type

Table 2-3 Summary table of the usability and type of expression for concrete and abstract mapping syntaxes

Concrete Syntax	Usability	Type of Expression
OWL 2	Difficult to Generate, Very Difficult to Update	Structured and Text-based Expression
SWRL	Difficult to Generate, Difficult to Update	Structured and Text-based Expression
SPARQL	Difficult to Generate, Easy to Update	Structured and Text-based Expression
Abstract Syntax	Usability	Type of Expression
Alignment Format	Very Easy to Generate, Very Easy to Update	Structured and Text-based Expression
EDOAL	Difficult to Generate, Difficult to Update	Structured and Text-based Expression
C-OWL	Very Easy to Generate, Very Easy to Update	Structured and Text-based Expression
SSE	Very Difficult to Generate, Easy to Update	Structured and Text-based Expression
SQWRL	Easy to Generate, Difficult to Update	Enumerated and Text-based Expression
RIF	Difficult to Generate, Easy to Update	Structured and Text-based Expression
UML/ODM	Very Easy to Generate, Easy to Update	Graphic-based Expression
UML/OCL	Difficult to Generate, Easy to Update	Graphic and Text (Structured) based Expression

Usability: Indication of how difficult to generate in the first place and to update (manipulate) by human perspective
 - Level: Very Easy, Easy, Difficult, Very Difficult

Type of Expression: The expression is classified into four types.

1. Enumerated: Expressing relationship in line formation
2. Structured: Expressing relationship in structure formation
3. Text-based Expression: If the expression is represented based on text
4. Graphic-based Expression: If the expression is represented graphically such as diagram

Table 2-3 shows the summary of usability and the type of expression for each discussed mapping syntax. The expression of the mapping relationships of each syntax can be classified into two categories, and this affects the usability of how difficult to generate and to manipulate. One category is enumerated type expression. This means mapping relationships are expressed in line formation such as SQWRL. This form of expression is syntactically simple to generate as the author can create the mapping relationships just by listing them. However, this form may not be easy to manipulate. For example, the discovery of the mapping relationships and the

modification of the mapped entities are not quite obvious because the result of each consequent of the rules is not explicit in this expression.

The other category is structured type expression. This means mapping relationships are expressed in structured formation such as RIF. They are relatively more difficult to generate in the first place by human perspective than enumerated expression because this expression separates each rule or mapped entity with order. However, the discovery of the mapping relationships and the modification of the mapped entities are quite obvious because the result of each consequent of the rules is explicitly grouped in a structured form.

Fig. 2-24 shows the contrast between enumerated type expression (SQWRL syntax) and structured type expression (RIF syntax). For example, an arithmetic operation ‘add’, that uses SWRL math built-ins, in SQWRL is described with other properties in a line, and there is ambiguity of the operation process and the usage of the parameters (?v1, ?v2, ?v3). In RIF, the same arithmetic operation ‘func:add’, that uses RIF functional built-ins, is described in a structured form with the discrimination between the result (?v3) of the operation and the parameters (?v1, ?v2), and this facilitates the introspection for the manipulation.

```

SQWRL: Enumerated Expression (Expressing relationship in line formation)
Femto(?x), MTS_Handover_Failure_Rate(?x, ?v1), UMTS_Handover_Failure_Rate(?x, ?v2),
add(?v3, ?v1, ?v2), divide(?value, ?v3, "100"^^long) → Handover_Failure_Gauge(?x, ?
value)

RIF: Structured Expression (Expressing relationship in structure formation)
Document (
  Prefix ( floc <http://sample.org/ontology/test/femto#> )
  Group (

Forall ?Femto ?Handover_Failure_Gauge ?MTS_Handover_Failure_Rate ?UMTS_Handover
_Failure_Rate (
  floc:Handover_Failure_Gauge( ?Femto ?value ) :-
  And (
    floc:Femto( ?femto )
    floc:MTS_Handover_Failure_Rate( ?femto ?v1 )
    floc:UMTS_Handover_Failure_Rate( ?femto ?v2 )
    ?v3 = External( func:add(?v1 ?v2) )
    ?value = External( func:divide(?v3 100) )
  )))

```

Figure 2-24 Enumerated and structured type expression

In the case of usability, Alignment Format, C-OWL and UML have been observed as the most suitable representations. However, part of their (excluding UML) simplicity is derived from their lack of capacity to support certain mapping types. SQWRL uses an enumerated type expression and this makes SQWRL easy to generate in the first place. However, it is difficult to manipulate the mapping relationships using SQWRL, e.g. the discovery of the mapping relationships and the modification of the mapped entities are not obvious because the result of each rule is not explicit in this expression. In the case of SSE and RIF, they both use a structured expression. Consequently, they are more difficult to generate than SQWRL. However, the source for the manipulation is easier. For example, the discovery of the mapping relationships and the modification of the mapped entities are quite obvious in a structured form. If generating this expression type of syntax can be automated by a tool, then it is more suitable than a syntax that uses an enumerated type expression.

Generally, text-based expression is less preferred than graphic-based expression for a novice user who does not have enough knowledge on the specific technical engineering skill to work on mappings. Graphic-based expression offers better interaction to a novice user because a novice user can work with the graphical

assistance. UML uses a graphic-based expression that is traditionally proven as a better approach than a text-based approach. In addition, UML [UML 2011] is widely used in many areas to model data structure, application structure, application behaviour, and business processes in a standardized and easy-to-understand manner [Kogut 2002]. It is the dominant language becoming the de facto standard for describing system applications, and system engineers are familiar with UML notations [Gasevic 2004]. It was thus decided in our research to use UML as the means to allow a system engineer to express mappings in an abstract manner.

This chapter presented related research about ontology mapping categorization and used their categorization of mapping types in the discussion of existing abstract and concrete syntaxes for semantic mapping. Through the examination of each syntax, this research summarized their capability to support the mapping types and their relative usability to generate in the first place by human perspective. The next chapter further discusses about mapping relationships in order to define the scope of the mapping types that will be supported by a developed abstract mapping syntax.

3 MOUSE Approach

This chapter describes three core mapping types derived from the industry use case, and the development of a UML-based abstract mapping syntax that can represent defined core mapping types.

3.1 The Core Mapping Types

This section describes three core mapping types: (1) Direct Mapping Type, (2) Data Range Mapping Type and (3) Unit Transformation Mapping Type that need to be captured correctly for the abstract syntax to be usable in a practical integration situation.

3.1.1 Industry Use Case

SDA⁶ (Semantic Data Access) is a research program initiated by Alcatel-Lucent Bell Labs Ireland [Boran 2011b]. This program constitutes several modules of industry and PhD research that aim to create a semantic data access plane within the telecommunications network. This access plane will facilitate the modelling of, integration of, and reasoning of heterogeneous data sources and will support third-party enterprises accessing data in a unified approach for lifting rough refined raw semantic data to rich semantic data. Because parts of this study fit within this large industrial research program inside Alcatel-Lucent Bell Labs Ireland, which provided the semantic mapping representation to support the SDA, there was collaboration with Bell Labs Ireland throughout this research; mainly supplying their industry use case.

This industry use case was the result of the research [Boran 2011a] conducted by Bell Labs Ireland about empirical analysis for network performance classification.

⁶ The SDA program research area includes: semantic mapping representation by KDEG in TCD, ontology development for sensor devices by CLARITY in UCD, data transformation from raw data into RDF triples using XSPARQL by DERI in NUIG, optimizing the query on semantic data for the integration in Oxford, performance evaluation of the instruments and deployment.

The use case is about the classification of network performance by semantic up-lift using real network ontology data collected from an Alcatel-Lucent femtocell [Small Cell] test bed. The performance management (PM) data is periodically collected and stored as an instance in raw-level ontology and then processed to generate key performance indicators (KPI) from the calculation over the PM data. The performance of the femtocell network is evaluated according to the KPIs - in total 19 rules were required to compute the set of 19 KPIs - and classified the femtocells into 15 performance-related classifications in femto-level ontology. APPENDIX A shows femto-level (femtoclassifier.owl) ontology and raw-level (femtoinstances.owl) ontology from the industry use case. In the appendix, all the ontology individuals are removed because of the large volume.

This process requires semantic mappings to integrate a femtocell instance in raw-level ontology to femto-level ontology. For example, a femtocell is classified as 'HighHandoverFemto' in femto ontology if the generated KPI value of 'BSR_cluster_to_MTS_underlay_Intra_Frequency_Hard_Handover_Failure_Rate' is higher or equal to 90. The KPI values are generated by processing the PM data collected from the femtocell in the telecommunications network, i.e. 'BSR_cluster_to_MTS_underlay_Intra_Frequency_Hard_Handover_Failure_Rate' = $(1 - ('VS_HHO_SuccBsrUmtsIntraFreq'$ value / 'VS_HHO_AttBsrUmtsIntraFreq' value)) * 100).

From the industry use case, i.e. 1000 raw-level femto instances that were collected from the femtocell test bed with the 15 minutes sampling interval, this research has observed that the semantic mappings were performed only within the scope of the ontology class or the datatype property (it is understandable considering the industry use case is from telecommunications network performance data). All the semantic data integrations for the mappings were tested both by the query-based integration approach using SPARQL and the rule-based integration approach using SWRL for the performance evaluation purpose. The semantic data integration using SPARQL has been shown to demonstrate better performance in terms of integration process speed. Table 3-1 shows the mapping between the identified semantic integration situations and the defined three core mapping types.

Table 3-1 Relationship between the semantic intergration situations and the core mapping types

Integration situations	Mapping relationships	Mapping types
Integration of ontology class instances into another ontology class without any constraint	Equivalent or subsumption mapping relationship	Direct mapping type
Integration of selected instances in one ontology class into another ontology class by the range constraint of the property value	Conditional mapping relationship that constraints the value of property (data range)	Data range mapping type
Integration of ontology class instances into another ontology class with the value transformation of the instance property	Transformation mapping relationship that transforms the value of property (unit transformation)	Unit transformation mapping type

As shown in the table 3-1, these identified semantic integrations are (1) integration of ontology class instances into another ontology class without any constraint, (2) integration of selected instances in one ontology class into another ontology class by the range constraint of the property value and (3) integration of ontology class instances into another ontology class with the value transformation of the instance property. Each of them requires a particular mapping relationship, which is respectively: (1) equivalent or subsumption mapping relationship, (2) conditional mapping relationship that constraints the value of property (data range) and (3) transformation mapping relationship that transforms the value of property (unit transformation). This research defines each identified mapping relationship as either (1) direct mapping type, (2) data range mapping type or (3) unit transformation mapping type. These mapping types had more priority than others because, they were the only three semantic mapping types from the supplied industry use case in the telecommunications network performance domain. To satisfy the needs of the industry use case, a proposed abstract mapping syntax must express these core mapping types in order to be usable in a practical integration situation.

In addition, the core mapping types identified from the industry use case were also observed in the state of the art section 2.1 ontology mapping categorization. Table 3-2 shows the relationship between these core mapping types and the correspondence patterns.

Table 3-2 Relationship between the generic mapping types and the core mapping types

Core Mapping Types	Link	Generic Patterns
Direct Mapping Type	$=$ equivalent to	Equivalent correspondence pattern
		Subsumption correspondence pattern
Data Range Mapping Type	\subset subset of	Conditional pattern e.g. a restriction on the scope of a class based on: <ul style="list-style-type: none"> - The occurrence of a property - The value of a property (Data Range) - The type of a property
Unit Transformation Mapping Type	\subset subset of	Transformation pattern e.g. a transformation of a property value about: <ul style="list-style-type: none"> - Datatype Conversion - Unit Transformation - Currency Conversion
	void	Union and intersection patterns

As shown in the table 3-2, (1) direct mapping type can be used for the semantic integration that requires equivalent or subsumption mapping relationships; (2) data range mapping type can be used for the semantic integration that requires conditional mapping relationships that constrain the value of a property (data range); and (3) unit transformation mapping type can be used for the semantic integration that requires transformation mapping relationships that transform the value of a property (unit transformation).

It was determined that these core mapping types would be required to be captured correctly for the abstract syntax to be usable in a practical integration situation. Therefore, the initial focus was placed on these three mapping types in order to develop the SDI (Semantic Data Integration) tool to implement the proposed abstract mapping syntax; allowing it to be used in experiments to evaluate the proposed MOUSE approach.

3.1.2 Core mapping type example

This section explains each core mapping type identified in the industry use case with a logical example.

1) *Direct Mapping Type*

This mapping type represents a mapping relationship, i.e. a class is equivalent to or subset of another class. This type has the integration property corresponding to “Unite the source class individuals to the target class”. Fig. 3-1 shows a logical example of this mapping type. As shown in the figure, Router class and Layer 3 Switch class are equivalent. So individuals in the Router class can be integrated into the Layer 3 Switch class.

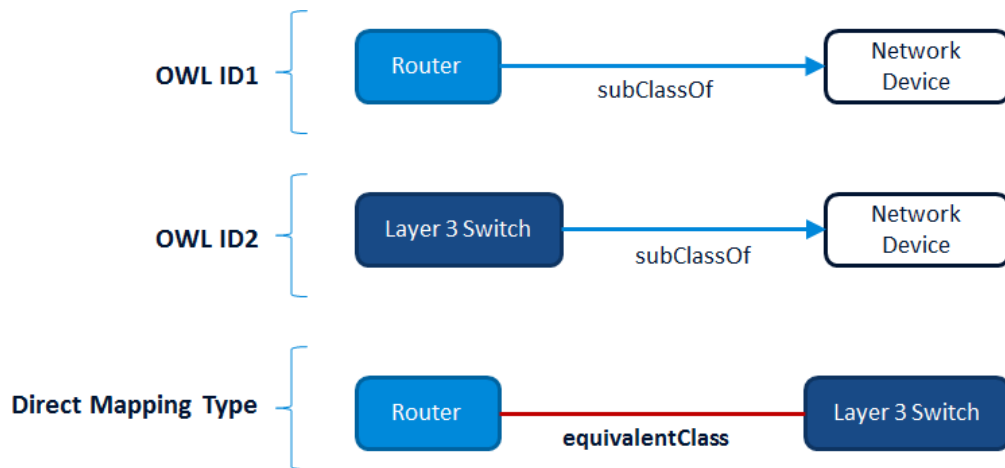


Figure 3-1 Logical example of the direct mapping type

2) *Data Range Mapping Type*

This mapping type represents a mapping relationship, i.e. a source ontology class constrained by the data range of single or multiple ontology datatype properties in the class and is equivalent to the target ontology class. This type has the integration property corresponding to “Unite the source ontology class individuals, which satisfy the data range constraint on single or multiple datatype properties on the individuals, to the target ontology class”. Fig. 3-2 shows a logical example of this mapping type. As shown in the figure, Femto class and Medium Handover Femto class are equivalent only if the data range constraint on

the Handover Failure Rate property is satisfied. Only individuals in the Femto class that satisfy this condition can be integrated into the Medium Handover Femto class.

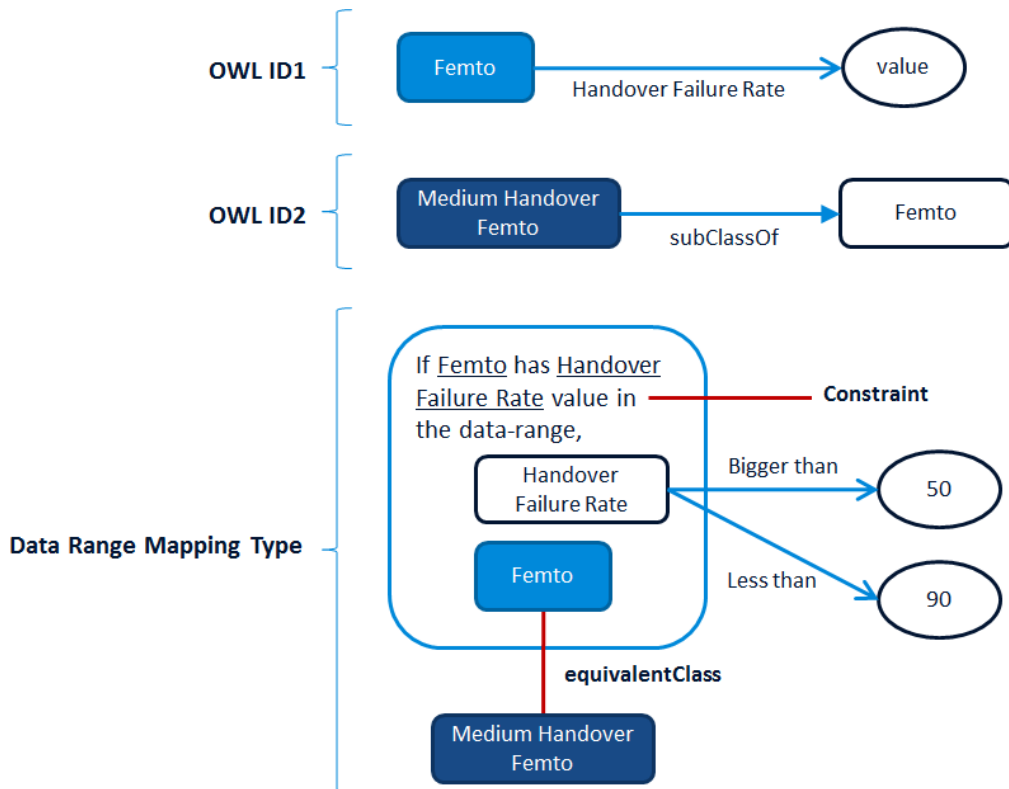


Figure 3-2 Logical example of the data range mapping type

3) Unit Transformation Mapping Type

This mapping type represents a mapping relationship, i.e. a source ontology class constrained by the arithmetic transformation of one or more ontology datatype properties in the class and is equivalent to the target ontology class. This type has the integration property corresponding to “Unite the source ontology class individuals to the target ontology class with the new assigned data value in the constrained datatype property by the arithmetic transformation of single or multiple datatype properties of the source ontology class”. Fig. 3-3 shows a logical example of this mapping type. As shown in the figure, the Byte property can be equivalent to the Bit property only if there is an arithmetic transformation on the value of the Byte property.

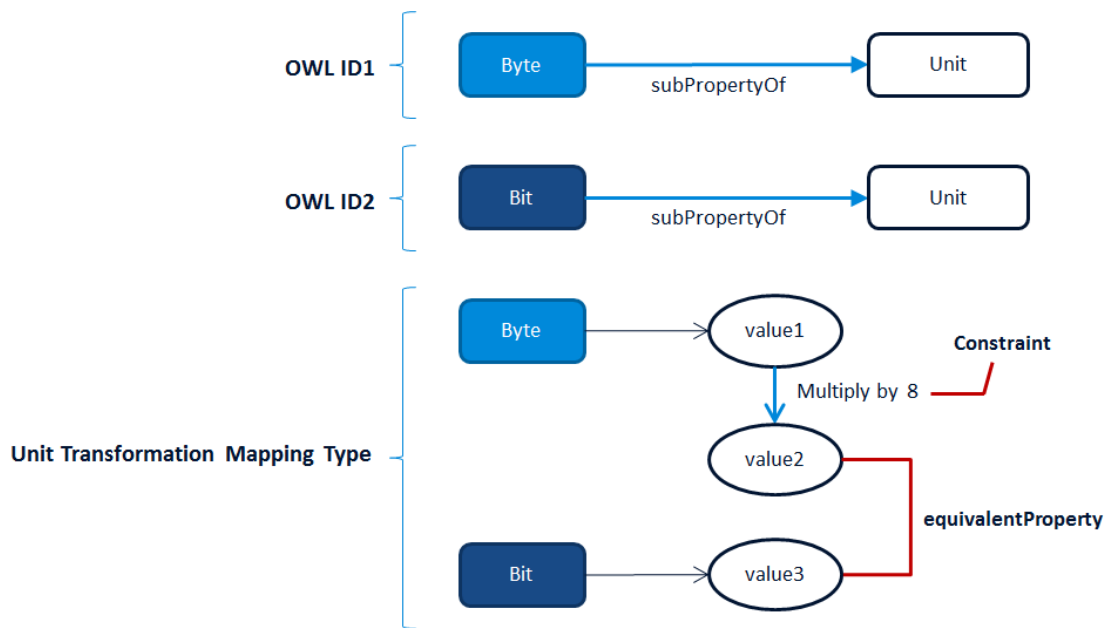


Figure 3-3 Logical example of the unit transformation type

This section explained the industry semantic integration situation within the telecommunications domain and defined the core mapping types (direct mapping type, data range mapping type and unit transformation mapping type) from the industry use case. These core mapping types are required to be captured correctly for the abstract syntax to be usable in a practical integration situation. The following section describes the development of a UML-based abstract mapping syntax that supports the core mapping types.

3.2 UML Notations Development

This section describes the development of a UML-based abstract mapping syntax that can represent the core mapping types. From the related research, UML/ODM supports expressing ontological terms using UML notations and UML/OCL supports more precise mappings by describing rules between UML notations. This research had initially developed a UML-based abstract mapping syntax that combines ODM and OCL standards because using the existing standards can reduce the learning curve of a new syntax and comply with the tools that already support them. ODM has limitations in describing some of the core mapping types (data range mapping type and unit transformation mapping). This research tried to cope with these deficiencies in expressing mapping relationships by using the OCL standard with ODM notations. This research conducted the experiment on this ODM and OCL combined abstract mapping syntax with experts. Fig. 3-4 and Fig. 3-5 show this initially developed UML-based abstract mapping syntax for different ontology mapping types: data range mapping type and unit transformation mapping type. As shown in the figures, ODM represents ontology vocabularies using stereotyped UML notations, and OCL is used to fill the mapping expressiveness gap of ODM. Fig. 3-4 shows ODM and OCL combined abstract mapping syntax that describes an ontology mapping in data range mapping type, i.e. ‘Femto’ is equivalent to ‘MediumHandoverFemto’ with the data range constraint ($50 < \text{handoverFailureRate} < 90$).

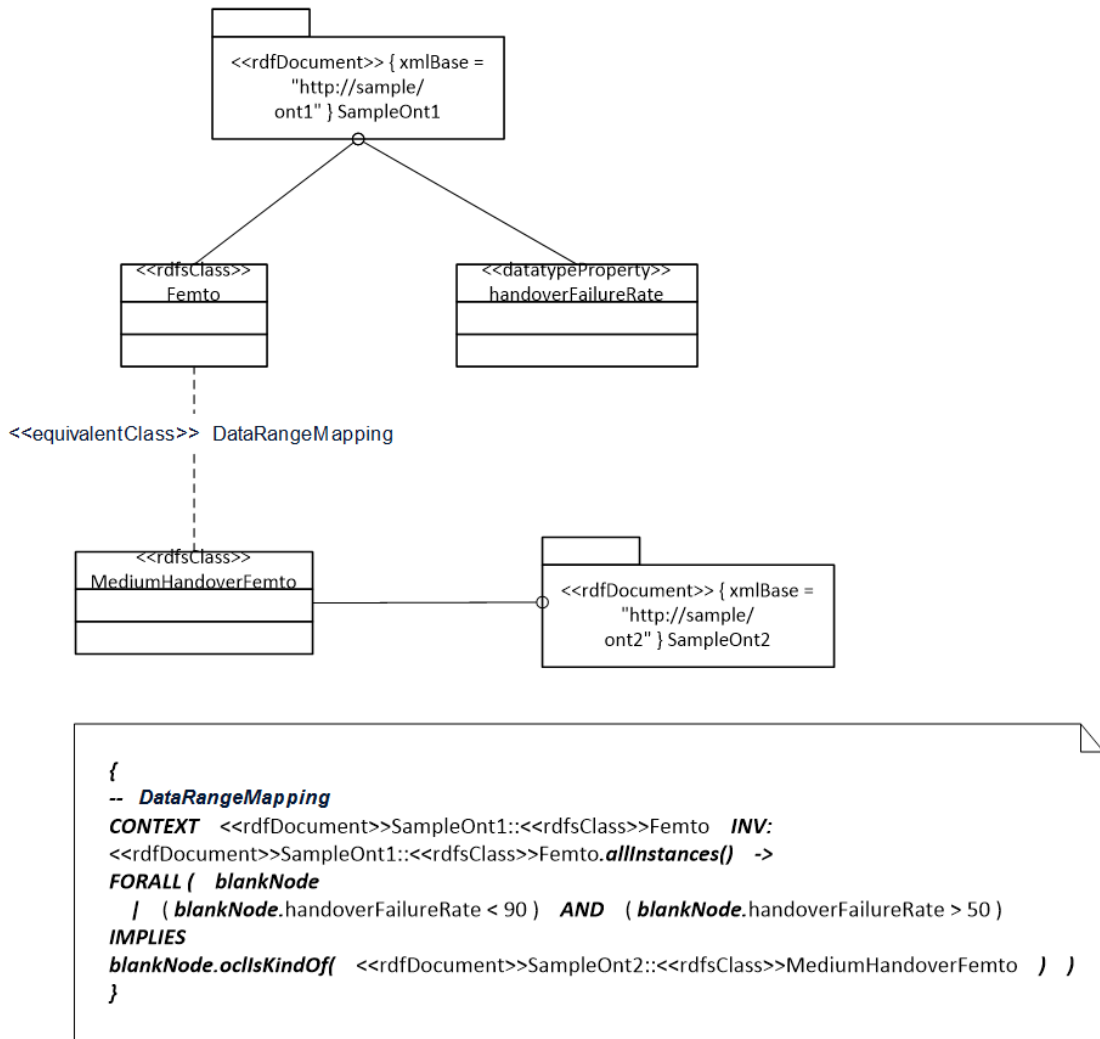


Figure 3-4 ODM and OCL combined abstract mapping syntax for conditional mapping type

Fig. 3-5 shows an ODM and OCL combined abstract mapping syntax that describes an ontology mapping in unit transformation mapping type, i.e. the value of 'FailureGauge' is equivalent to the result value of arithmetic operation ('FailureRate1' + 'FailureRate2').

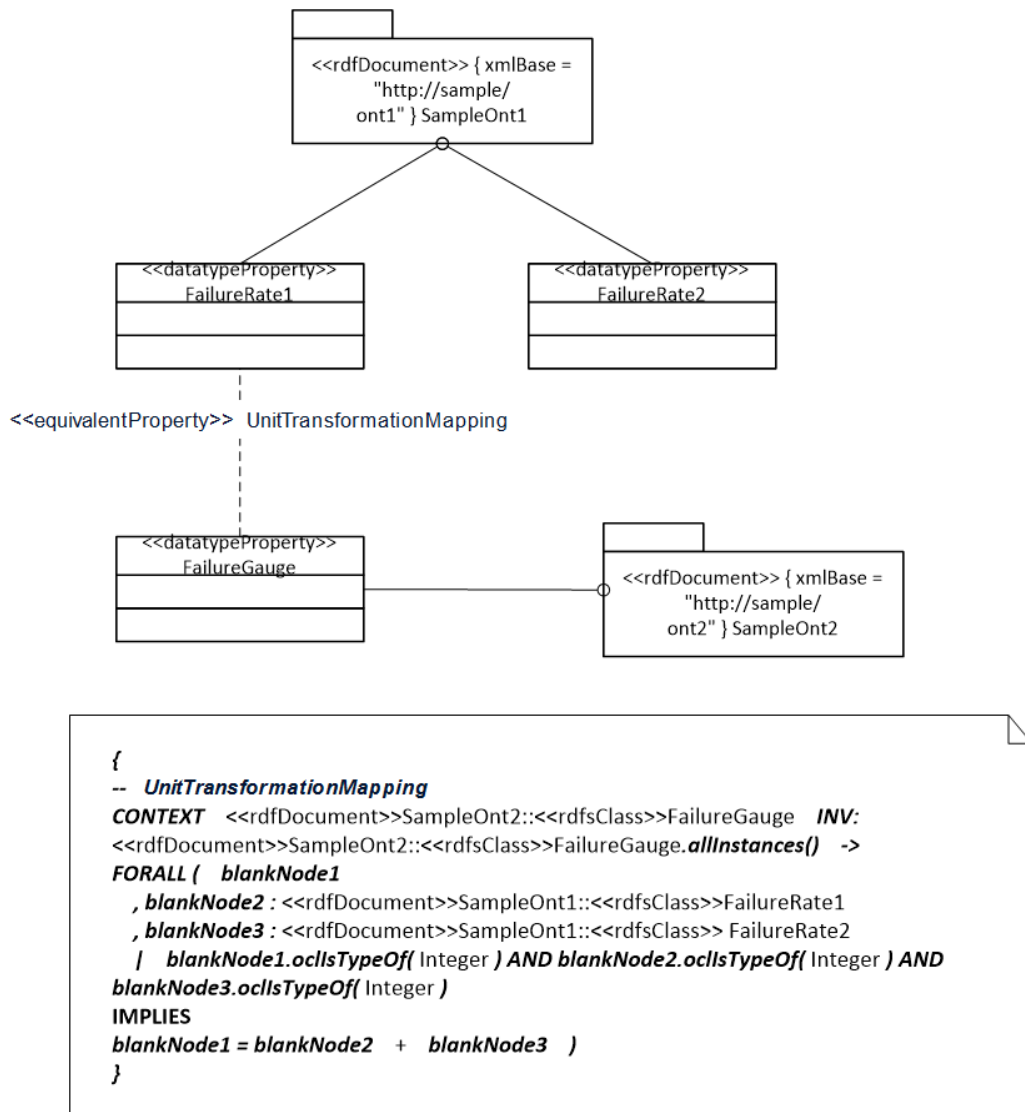


Figure 3-5 ODM and OCL combined abstract mapping syntax for transformation mapping type

An initial experiment was conducted to evaluate the accuracy of the ODM and OCL combined abstract mapping syntax. From the experiment, the usability of this initially proposed abstract mapping syntax was shown to be unsatisfactory through the feedback of the experiment on the tool that implements this proposed mapping syntax. For example, participant ID2 commented: *“Overall it is very difficult to write the OCL constraint for data range mappings. It seems that this tool cannot parse the constraint with multiple datatype properties. This tool can parse AND relationship well but it cannot parse the OR operator in the constraint”*. APPENDIX E shows an OWL view of the initial experiment ontologies using the Protégé tool [Protégé

Project]. The tool source code and initial experiment results from each participant are available on the accompanying DVD for this thesis, see APPENDIX L.

It was discovered that the ODM and OCL combined abstract mapping syntax needed to be more abstract to be usable by non-ontology experts because the ODM standard uses customized stereotypes in UML to define terms in ontology. Unfortunately, the user of the combined notation must be a knowledge engineer who understands ontologies in the first place to use these customized stereotypes for ontology vocabularies. The syntax had to be more abstract to be usable by system engineers. This research further developed a more abstract UML-based mapping syntax without using stereotypes for ontology vocabularies and even without a deep understanding of UML constraint language such as OCL than the initially developed UML-based abstract mapping syntax.

The following sections describe the detailed specification of the more abstract UML-based mapping syntax that uses UML notations without stereotypes for ontology vocabularies and UML constraint expressions, which are the syntactic sugar to facilitate the use of OCL constraints. UML constraint expression is a user-defined expression using usual programming arithmetic and logical expressions [Banahan 1991] in UML constraint notation using the UML 2.0 standard [UML 2.0] to express more precise semantic mappings. This more abstract syntax should improve the usability for system engineers.

3.3 UML Notations for Ontology Elements

3.3.1 Ontology Document

Semantic mapping requires one or more ontology documents that describe vocabularies [OWL Overview] of a specific domain. This ontology document notation distinguishes the ownership of the ontology elements.

UML Notation

- Package::ownedMember classifier - represents an ontology document and a vocabulary that is described in the document.
- First part of the classifier separated by a double colon (‘::’) - represents the name of the ontology document.

Graphical Representation

Package::ownedMember classifier in the name field of UML::Class and UML::Association is used to represent an ontology document. Fig. 3-6 shows the UML notations that represent ‘Device’, ‘Cable’, and ‘hasInterface’ ontology vocabularies, which are described in ‘Network’ ontology document.

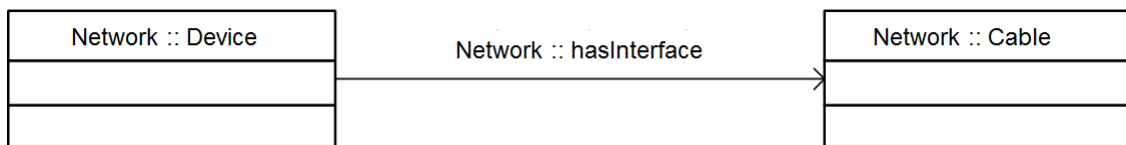


Figure 3-6 UML notation example for ontology document

3.3.2 Ontology Class

Ontology class defines a group of individuals that belong together because they share some properties [OWL Overview]. For example, ‘femtocell1’ and ‘femtocell2’ are both members of the class ‘Femto’. Ontology vocabularies can be aligned into a collection of triples, i.e. RDF statement [RDF Concepts], each consisting of a subject, a predicate and an object. An ontology class can be a subject or an object in the statement.

UML Notation

- UML::Class - represents an ontology class.
- Second part of the classifier in the name field of UML::Class separated by a double colon (‘::’) - represents the name of the ontology class.

Graphical Representation

UML::Class is used to represent an ontology class. Fig. 3-7 shows the UML notation that represents the ontology class ‘Femto’.

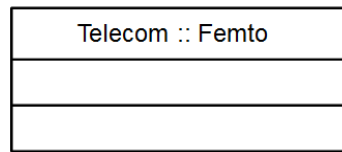


Figure 3-7 UML notation example for ontology class

Fig. 3-8 shows the UML notations which each notation corresponds to each node in RDF statement, i.e. Subject (Femto) - Predicate (hasNPM) - Object (NPM).

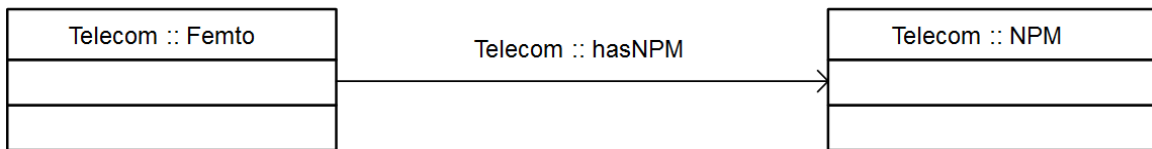


Figure 3-8 UML notation example for RDF statement

3.3.3 Ontology Individual

An individual is an instance of a class, and properties may be used to relate one individual to another [OWL Overview]. For example, an individual named ‘femtocell1’ may be described as an instance of the class ‘Femto’ and the property ‘hasNPM’ may be used to relate the individual ‘femtocell1’ to the individual ‘networkdata1’. An ontology individual can be a subject or an object in a RDF statement.

UML Notation

- UML::InstanceSpecification - represents an ontology individual.
- First part of the classifier in the name field of UML::InstanceSpecification separated by a colon (‘:’) - represents the name of the individual.
- Second part of the classifier in the name field of UML::InstanceSpecification separated by a colon (‘:’) - represents the ontology document and class name

of the individual that has this class extension. The individuals in the class extension are called the instances of the class.

Graphical Representation

UML::InstanceSpecification is used to represent an ontology individual. Fig. 3-9 shows the UML notation that represents the ontology individual 'femtocell1' of the ontology class 'Femto'.

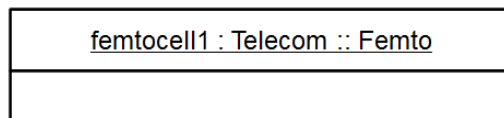


Figure 3-9 UML notation example for ontology individual

Fig. 3-10 shows the UML notations which each notation corresponds to each node in a RDF statement, i.e. Subject (femtocell1) - Predicate (hasNPM) - Object (networkdata1).

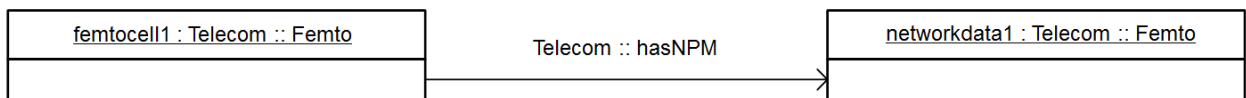


Figure 3-10 UML notation example for RDF statement

3.3.4 Ontology Object Property

Ontology object property can be used to state relationships between instances of two classes [OWL Guide]. For example, the object property 'hasInterface' can be used to relate an instance of the class 'Device' to an instance of the class 'Cable'. An ontology object property can be a predicate in a RDF statement, and there should be classes or individuals as a subject and an object.

UML Notation

- UML::Association - represents an ontology object property. The property links from a subject to an object with unidirectional navigation.

- Second part of the classifier in the name field of UML::Association separated by a double colon (‘::’) - represents the name of the ontology object property.

Graphical Representation

UML::Association is used to represent an ontology object property. Fig. 3-11 shows the UML notation that represents the ontology object property ‘hasInterface’.

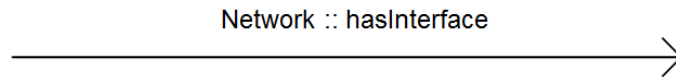


Figure 3-11 UML notation example for ontology object property

Fig. 3-12 shows the UML notations which each notation corresponds to each node in a RDF statement, i.e. Subject (router1) - Predicate (hasInterface) - Object (rj45).

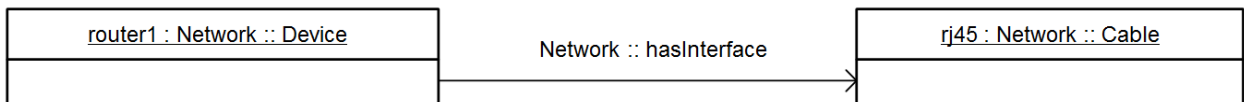


Figure 3-12 UML notation example for RDF statement

3.3.5 Ontology Datatype Property

Ontology datatype property can be used to state relationships between instances of classes and RDF literals and XML schema data types [XML Datatype][OWL Guide]. For example, the datatype property ‘successRate’ can be used to relate an instance of the class ‘Femto’ to a value of the data type ‘double’. An ontology datatype property can be a predicate in RDF statement, and there should be a RDF literal or XML Schema data type as an object.

UML Notation

- UML::Property - represents an ontology datatype property.
- Second part of the classifier in the property name separated by a double colon (‘::’) - represents the name of the ontology datatype property.
- Property type - represents the data type of the ontology datatype property.

Graphical Representation

UML::Property is used to represent an ontology datatype property. Fig. 3-13 shows the UML notation that represents the ontology datatype property ‘successRate’.

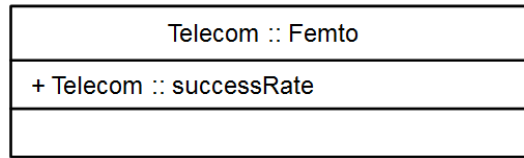


Figure 3-13 UML notation example for ontology datatype property

Fig. 3-14 shows the UML notation which each element: Instance name, Property name and Property value, corresponds to each node in a RDF statement, i.e. Subject (femtocell1) - Predicate (successRate) - Value (90.0)

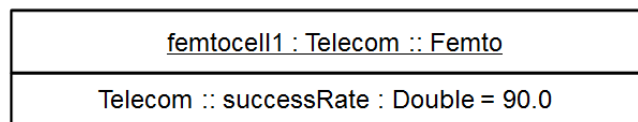


Figure 3-14 UML notation example for RDF statement

3.4 UML Notations for Semantic Mappings

UML notations for ontology elements use the standard way of naming UML notations [UML 2.0]. For semantic mappings, this research uses <<Unite>> stereotyped UML::AssociationClass to represent the semantic mapping that integrates class instances. UML::AssociationClass can be used in a UML diagram without the encoded meaning of a semantic mapping. To distinguish UML::AssociationClass that encodes the meaning of semantic mapping from the general use of UML::AssociationClass, <<Unite>> stereotype is used with the classifier: DirectMapping, DataRangeMapping, and UnitTransformationMapping. A numeric identification label is used following each mapping type classifier in order to track the sequence of semantic mappings for the semantic data integration. For example, semantic mapping <<Unite>> DataRangeMapping 3 will be executed after

<<Unite>> UnitTransformationMapping 2. In case the same sequence number appears, the order of mapping executions is dependent on the implementation.

3.4.1 Direct Mapping Type

This mapping type represents a mapping relationship, i.e. a class is equivalent to or subset of another class. This type has the integration property corresponding to “Unite the source class individuals to the target class”.

UML Notation

- <<Unite>> with base class of UML::AssociationClass - represents the semantic mapping to integrate between two associated classes with unidirectional navigation.
- ‘DirectMapping’ label with numeric identification in the postfix in the name field of <<Unite>> stereotyped UML::AssociationClass - represents that the mapping is a direct mapping type.

Graphical Representation

<<Unite>> stereotyped UML::AssociationClass is used to represent a direct mapping type. Fig. 3-15 shows the UML notations that represent the integration property corresponding to “Unite the Layer3Switch class individuals to the Router class”.

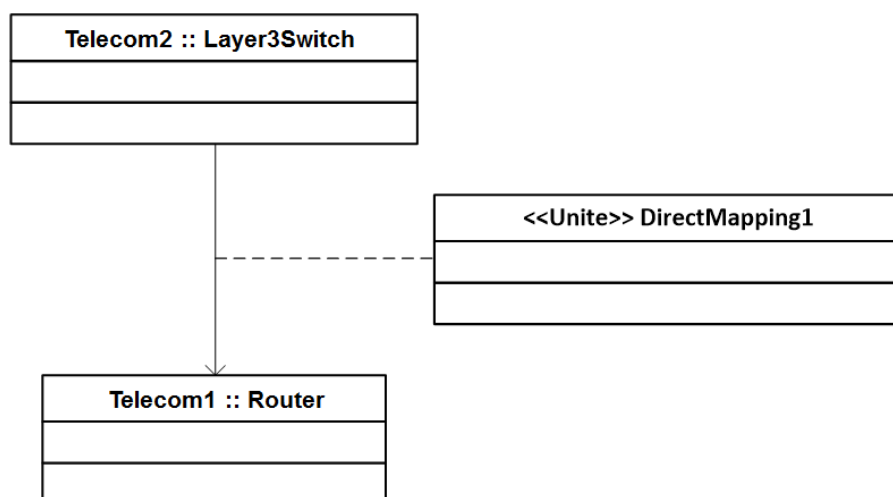


Figure 3-15 UML notation example for direct mapping type

3.4.2 Data Range Mapping Type

This mapping type represents a mapping relationship, i.e. a source ontology class, which is constrained by the data range of single or multiple ontology datatype properties in the class and is equivalent to the target ontology class. This type has the integration property corresponding to “Unite the source ontology class individuals, which satisfy the data range constraint on single or multiple datatype properties on the individuals, to the target ontology class”.

UML Notation

- <<Unite>> with base class of UML::AssociationClass - represents the semantic mapping to integrate between two associated classes with unidirectional navigation.
- ‘DataRangeMapping’ label with numeric identification in the postfix in the name field of <<Unite>> stereotyped UML::AssociationClass - represents that the mapping is a data range mapping type.
- UML::Constraint at the end of the UML::Property entry which has the property name ‘constraint’ - represents the specification of this mapping relationship.

Note: UML constraint expression is used in the constraint following the property ‘constraint’. <<Unite>> stereotyped association class can only have the ‘constraint’ named property followed by the constraint.

UML constraint expression

Fig. 3-16 shows the BNF (Backus Normal Form) style syntax expression for the UML constraint expression in data range mapping type.


```

Var ::= Classifier
Operator ::= '=' | '<>' | '<' | '<=' | '>' | '>='
Atomic ::= ( Var Operator Constant )
Formula ::= Atomic | Atomic ( 'AND' | 'OR' ) Formula *

For example: Var1 > 90 AND Var2 > 90 OR Var3 < 10

```

Figure 3-16 BNF style syntax expression for data range mapping

- Comparison operation - represents a mapping relationship that the individuals of the source ontology class can be united to the target ontology class, only if the comparison operation is satisfied.
- Comparison operator between a property classifier and a data value - can be equal ('=='), not equal ('<>'), less than ('<'), less than or equal ('<='), greater than ('>'), or greater than or equal ('>=').
- AND or OR operator between sets of a comparison operation - represents the Boolean logic. For example, in the case of 'AND', both comparison operations need to be satisfied, and in the case of 'OR', at least one of the comparison operation needs to be satisfied.

Graphical Representation

<<Unite>> stereotyped UML::AssociationClass with UML::Constraint in UML::Property is used to represent a data range mapping type. Fig. 3-17 shows the UML notations that represent the integration property corresponding to “Unite the Femto class individuals that satisfies the constraint of comparison operation with the successRate1 datatype property, to the HighHandoverFemto class”.

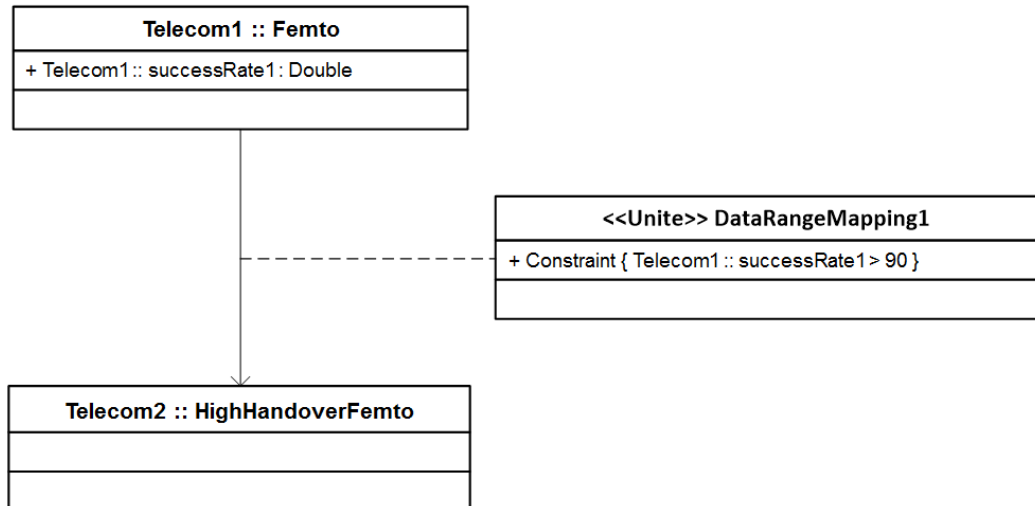


Figure 3-17 UML notation example for data range mapping type with single ontology datatype property in constraint

Fig. 3-18 shows the UML notations that represent an integration property corresponding to “Unite the Femto class individuals that satisfies the constraint of comparison operations with the successRate1 and successRate2 datatype properties, to the HighHandoverFemto class”.

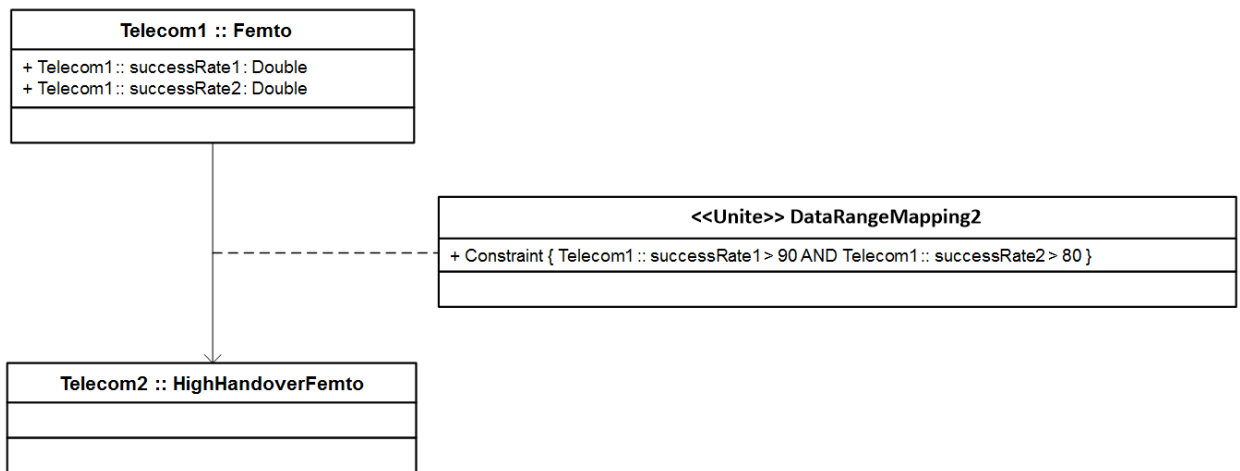


Figure 3-18 UML notation example for data range mapping type with multiple ontology datatype properties in constraint

UML notations and the constraint expression shown in Fig. 3-17 and Fig. 3-18 can be corresponded to the OCL constraint language. Table 3-3 shows how a UML

constraint expression can be translated to the OCL syntax, and Fig. 3-19 shows the OCL syntax result translated from the UML constraint expression used in Fig. 3-17.

Table 3-3 Correspondence between UML constraint expression and OCL syntax for data range mapping type

UML constraint expression	OCL Syntax	Comment
'Constraint' UML::Property used in <<Unite>> stereotyped UML::AssociationClass with the name 'DataRangeMapping'	CONTEXT <<Unite>> DataRangeMapping INV:	Specifying the UML context
The constraint is scoped for all the Telecom1::Femto class instances at the association from	Telecom1::Femto.allInstances() -> FORALL	Indicating the scope of the constraint to apply
Telecom1::successRate1 > 90	blankNode (blankNode.Telecom1::successRate1 > 90)	The comparison operation
Unite the instances that satisfy the constraint to the Telecom2::HighHandoverFemto class at the association to	IMPLIES blankNode.oclsKindOf(Telecom2::HighHandoverFemto)	Indicating the target class

```

{
  CONTEXT <<Unite>> DataRangeMapping      INV:
    Telecom1::Femto .allInstances() -> FORALL
    (
      blankNode | ( blankNode. Telecom1::successRate1 > 90 )
      IMPLIES
      blankNode.oclsKindOf( Telecom2::HighHandoverFemto )
    )
}

```

Figure 3-19 OCL syntax translated from UML constraint expression for data range mapping type

Table 3-4 shows how a UML constraint expression can be translated to the OCL syntax, and Fig. 3-20 shows the OCL syntax result translated from the UML constraint expression used in Fig. 3-18.

Table 3-4 Correspondence between UML constraint expression and OCL syntax for data range mapping type

UML constraint expression	OCL Syntax	Comment
'Constraint' UML::Property used in <<Unite>> stereotyped UML::AssociationClass with the name 'DataRangeMapping'	CONTEXT <<Unite>> DataRangeMapping INV:	Specifying the UML context
The constraint is scoped for all the Telecom1::Femto class instances at the association from	Telecom1::Femto.allInstances() -> FORALL	Indicating the scope of the constraint to apply
Telecom1::successRate1 > 90 AND Telecom1::successRate2 > 80	blankNode (blankNode.Telecom1::successRate1 > 90) AND (blankNode.Telecom1::successRate2 > 80)	The comparison operation
Unite the instances that satisfy the constraint to the Telecom2::HighHandoverFemto class at the association to	IMPLIES blankNode.oclsKindOf(Telecom2::HighHandoverFemto)	Indicating the target class

```

{
  CONTEXT <<Unite>> DataRangeMapping      INV:
    Telecom1::Femto .allInstances() ->  FORALL
  (
    blankNode | ( blankNode.Telecom1::successRate1 > 90 ) AND ( blankNode.Telecom1::successRate2 > 80 )
    IMPLIES
    blankNode.oclsKindOf( Telecom2::HighHandoverFemto )
  )
}

```

Figure 3-20 OCL syntax translated from UML constraint expression for data range mapping type

3.4.3 Unit Transformation Mapping Type

This mapping type represents a mapping relationship, i.e. a source ontology class, which is constrained by the arithmetic transformation of one or more ontology datatype properties in the class and is equivalent to the target ontology class. This type has the integration property corresponding to “Unite the source ontology class individuals to the target ontology class with the new assigned data value in the

constrained datatype property by the arithmetic transformation of single or multiple datatype properties of the source ontology class”.

UML Notation

- <<Unite>> with base class of UML::AssociationClass - represents the semantic mapping to integrate between two associated classes with unidirectional navigation.
- ‘UnitTransformationMapping’ label with numeric identification in the postfix in the name field of <<Unite>> stereotyped UML::AssociationClass - represents that the mapping is a unit transformation mapping type.
- UML::Constraint at the end of the UML::Property entry which has the property name ‘constraint’ - represents the specification of this mapping relationship.

Note: UML constraint expression is used in the constraint following the property ‘constraint’. <<Unite>> stereotyped association class can only have the ‘constraint’ named property followed by the constraint.

UML constraint expression

Fig. 3-21 shows the BNF (Backus Normal Form) style syntax expression for the UML constraint expression in unit transformation mapping type.

```
Var ::= Classifier
Operator ::= '+' | '-' | '*' | '/'
Calc ::= Var | ( Var | Constant ) Operator Calc *
Formula ::= Var '=' Calc

For example: Var3 = Var1 + Var2
```

Figure 3-21 BNF style syntax expression for unit transformation mapping

- Arithmetic equation - represents a mapping relationship in which the individuals of the source ontology class can be united to the target ontology class, only if the data value of the source ontology datatype property of the

individual is transformed by the arithmetic equation and the result value is assigned to the target ontology datatype property.

- A property classifier in the left side of an equals sign ('=') of the arithmetic equation - represents the target ontology datatype property that the data value of the property will have the result value by the arithmetic operation in the right side of the equation.
- Left side of an equals ('=') sign in the arithmetic equation - can only be placed by single target ontology datatype property.
- Property classifiers in the arithmetic operation in the right side of an equals sign ('=') of the equation - represents the source ontology datatype properties that the data values need to be transformed by the operation.
- Right side of an equals ('=') sign in the arithmetic equation - can have combinations of numeric and source ontology datatype properties with add ('+'), subtraction ('-'), multiply ('*'), and divide ('/') operation.

Graphical Representation

<<Unite>> stereotyped UML::AssociationClass with UML::Constraint in UML::Property is used to represent a unit transformation mapping type. Fig. 3-22 shows the UML notations that represent the integration property corresponding to “Unite the Femto class individuals that have the FailureRate datatype property to the Device class that has the FailureGauge datatype property with the new assigned data value for the FailureGauge datatype property by the arithmetic operation of the FailureRate datatype property of the Femto class individual”.

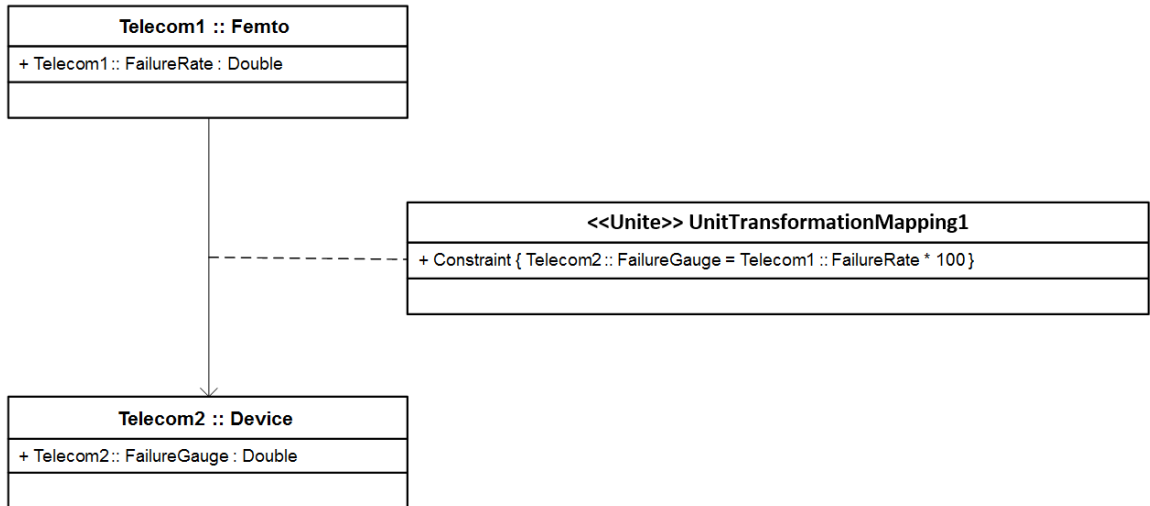


Figure 3-22 UML notation example for unit transformation mapping type with single ontology datatype property operation in constraint

Fig. 3-23 shows the UML notations that represent an integration property corresponding to “Unite the Femto class individuals that have the FailureRate1 and FailureRate2 datatype properties to the Device class that has the FailureGauge datatype property with the new assigned data value for the FailureGauge datatype property by the arithmetic operation of the FailureRate1 and FailureRate2 datatype properties of the Femto class individual”.

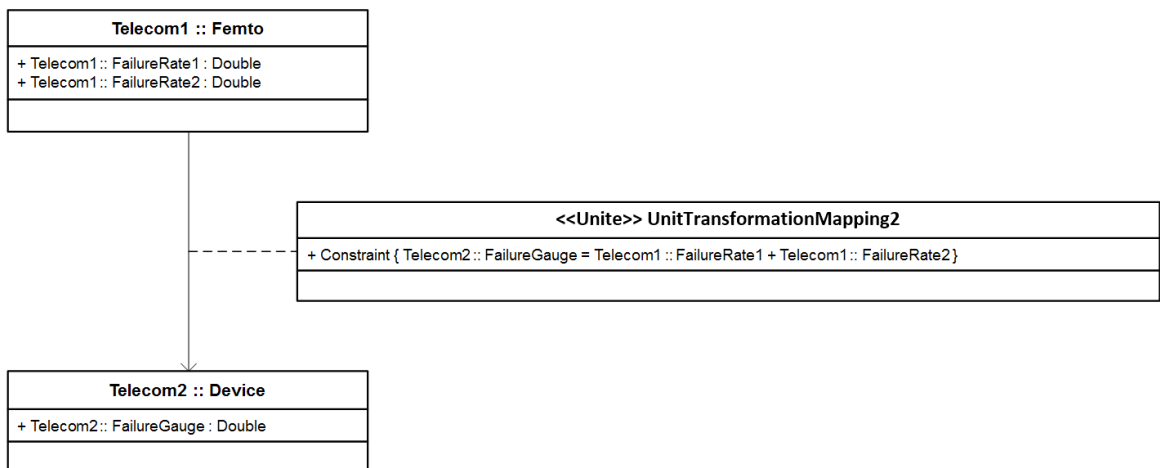


Figure 3-23 UML notation example for unit transformation mapping type with multiple ontology datatype properties operation in constraint

UML notations and the constraint expression shown in Fig. 3-22 and Fig. 3-23 can be corresponded to the OCL constraint language. Table 3-5 shows how a UML constraint expression can be translated to the OCL syntax, and Fig. 3-24 shows the OCL syntax result translated from the UML constraint expression used in Fig. 3-22.

Table 3-5 Correspondence between UML constraint expression and OCL syntax for unit transformation mapping type

UML constraint expression	OCL Syntax	Comment
'Constraint' UML::Property used in <<Unite>> stereotyped UML::AssociationClass with the name 'UnitTransformationMapping'	CONTEXT <<Unite>> UnitTransformationMapping INV:	Specifying the UML context
The constraint is scoped for all the Telecom1::Femto class instances at the association from	Telecom1::Femto.allInstances() -> FORALL	Indicating the scope of the constraint to apply
Telecom2::FailureGauge = Telecom1::FailureRate * 100	blankNode1 , blankNode2 : Telecom2::Device blankNode2.FailureGauge = blankNode1.FailureRate * 100	The arithmetic operation
Unite the instances that satisfy the constraint to the Telecom2::Device class at the association to	IMPLIES blankNode1.oclsKindOf(Telecom2::Device)	Indicating the target class

```

{
  CONTEXT <<Unite>> UnitTransformationMapping      INV:
    Telecom1::Femto.allInstances() -> FORALL
  (
    blankNode1 , blankNode2 : Telecom2::Device
    | blankNode2.FailureGauge = blankNode1.FailureRate * 100
    IMPLIES
    blankNode1.oclsKindOf( Telecom2::Device )
  )
}

```

Figure 3-24 OCL syntax translated from UML constraint expression for unit transformation mapping type

Table 3-6 shows how a UML constraint expression can be translated to the OCL syntax, and Fig. 3-25 shows the OCL syntax result translated from the UML constraint expression used in Fig. 3-23.

Table 3-6 Correspondence between UML constraint expression and OCL syntax for unit transformation mapping type

UML constraint expression	OCL Syntax	Comment
'Constraint' UML::Property used in <<Unite>> stereotyped UML::AssociationClass with the name 'UnitTransformationMapping'	CONTEXT <<Unite>> UnitTransformationMapping INV:	Specifying the UML context
The constraint is scoped for all the Telecom1::Femto class instances at the association from	Telecom1::Femto.allInstances() -> FORALL	Indicating the scope of the constraint to apply
Telecom2::FailureGauge = Telecom1::FailureRate1 + Telecom1::FailureRate2	blankNode1 , blankNode2 : Telecom2::Device blankNode2.FailureGauge = blankNode1.FailureRate1 + blankNode1.FailureRate2	The arithmetic operation
Unite the instances that satisfy the constraint to the Telecom2::Device class at the association to	IMPLIES blankNode1.oclsKindOf(Telecom2::Device)	Indicating the target class

```

{
  CONTEXT <<Unite>> UnitTransformationMapping      INV:
    Telecom1::Femto.allInstances() ->  FORALL
  (
    blankNode1 , blankNode2 : Telecom2::Device
    | blankNode2.FailureGauge = blankNode1.FailureRate1 + blankNode1.FailureRate2
    IMPLIES
    blankNode1.oclsKindOf( Telecom2::Device )
  )
}

```

Figure 3-25 OCL syntax translated from UML constraint expression for unit transformation mapping type

3.5 A walked through example

This subsection describes the use of the proposed abstract mapping syntax explained in previous sections. To generalize the use of the core mapping types (not limited to the telecommunications domain), this research tried to apply these core mapping types in another domain, and the conference domain was selected because this domain is more generalized than the telecommunications domain (that normally

requires deep understanding of the domain specific knowledge) and is generally understandable by more people with a system engineering background.

Fig. 3-26 shows two conference ontologies turned into UML notations, e.g. “SourceOnt|Article” class with ‘hasSize’ and ‘hasScore’ properties in the UML model represents the ontology class ‘Article’ with datatype property ‘hasSize’ and ‘hasScore’ in SourceOnt ontology. This figure also shows how the conference UML models can be mapped together using UML notations outlined earlier. In the figure, there are examples of applications for the three core mapping types: direct mapping, data range mapping and unit transformation mapping. For example, the “DataRangeMapping2” association class with the constraint in property represents the semantic mapping of data range mapping type.

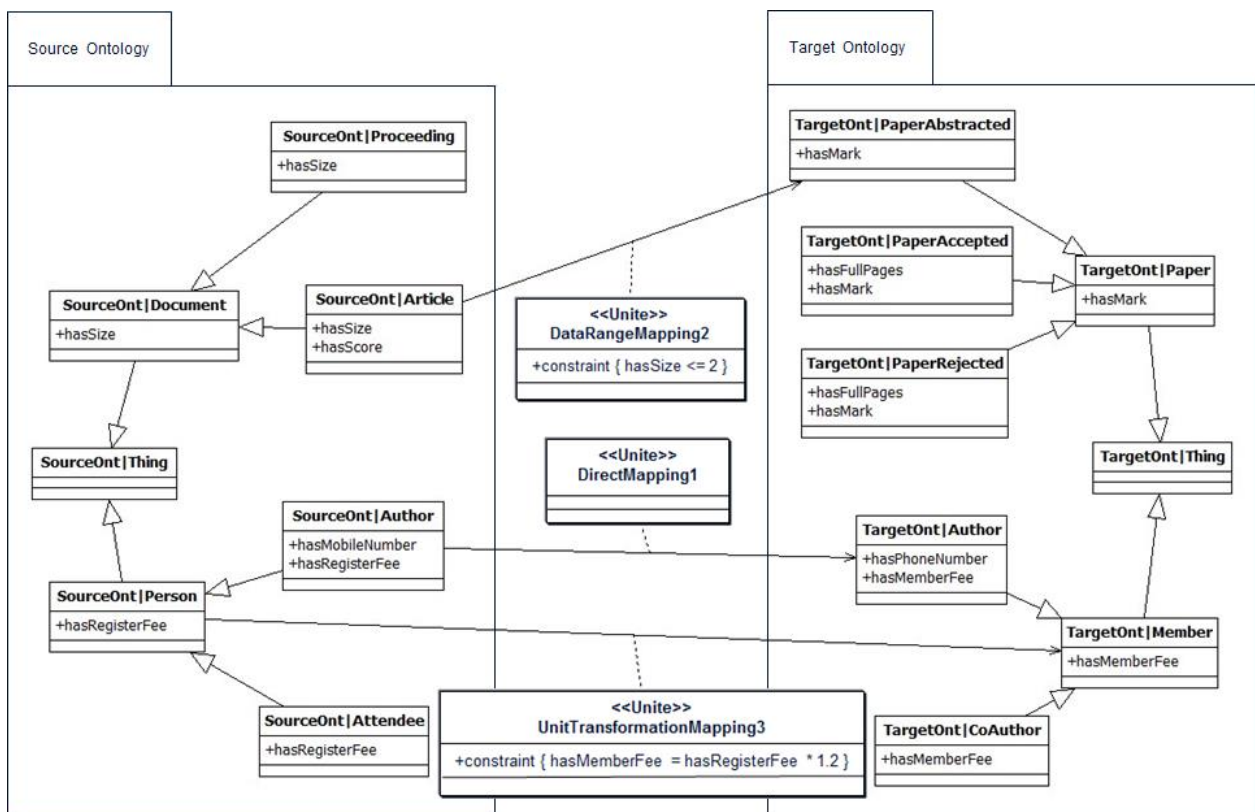


Figure 3-26 Example of the UML-based abstract mapping syntax

3.5.1 Direct Mapping

UML::AssociationClass ‘<<Unite>> DirectMapping1’ will be executed first among the three semantic mappings in the UML diagram. Author class (which has hasMobileNumber and hasRegisterFee properties) in the source ontology and Author class (which has hasPhoneNumber and hasMemberFee properties) in the target ontology are connected via a DirectMapping association class indicating that the Author class in the source ontology is mapped to the other Author class in the target ontology with the integration property corresponding to “Unite the source class (Author) individuals to the target class (Author)”.

3.5.2 Data Range Mapping

UML::AssociationClass ‘<<Unite>> DataRangeMapping2’ will be executed second among the three semantic mappings in the UML diagram. Article class (which has hasSize and hasScore properties) in the source ontology and PaperAbstracted class (which has hasMark property) in the target ontology are connected via a DataRangeMapping association class indicating that the Article class in the source ontology is mapped to the PaperAbstracted class in the target ontology with the integration property corresponding to “Unite the source ontology class (Article) individuals, that satisfies the data range constraint (hasSize <= 2), to the target ontology class (PaperAbstracted)”. This means that a paper with less than or equal to 2 pages in the Article class corresponds to an abstract paper in the PaperAbstracted class.

3.5.3 Unit Transformation Mapping

UML::AssociationClass ‘<<Unite>> UnitTransformationMapping3’ will be executed third among the three semantic mappings in the UML diagram. Person class (which has hasRegisterFee) in the source ontology and Member class (which has hasMemberFee) in the target ontology are connected via a UnitTransformationMapping association class indicating that the Person class in the ontology is mapped to the Member class in the target ontology with the integration property corresponding to “Unite the source ontology class (Person) individuals to the target ontology class (Member) with the new assigned data value (hasMemberFee

= $\text{hasRegisterFee} * 1.2$) for the datatype property (hasMemberFee) by the arithmetic transformation of the datatype properties (hasRegisterFee)". This means the registration fee is multiplied by 1.2 (e.g. because of the currency difference) paid by a person in the Person class corresponds to the amount of the member fee paid by a person in the Member class.

3.5.4 Limitation

There is a limitation in the developed UML notations for semantic mappings. When using the proposed UML notations, it is not possible to perform a semantic data integration that requires both a unit transformation mapping and a data range mapping without the use of a spare class. The spare class is an intermediate class to facilitate the mapping. For example, if integration requires a data range mapping after a unit transformation mapping, there must be an existence of a spare class that will temporarily save the result instances of the unit transformation mapping. The spare class will allow these result instances to be integrated by the data range mapping. This research does not offer a mapping type that combines a unit transformation mapping and a data range mapping.

This chapter described the development of a UML-based abstract mapping syntax that supports the core mapping types. This developed abstract mapping syntax will allow a system engineer (without sufficient ontological knowledge or knowledge engineering experience) to describe semantic mappings using familiar technology to them. The following chapter describes the development of a tool that supports the core mapping types using the developed UML-based abstract mapping syntax.

4 SDI Tool Development

This chapter describes the development of a tool that implements the proposed abstract mapping syntax and the automatic transformation process that constitutes the MOUSE approach. The tool, called SDI (Semantic Data Integration), allows a system engineer to express mappings using UML and then, automatically generates the corresponding executable semantic mappings. It also generates an integrated ontology as a computation result of executing the generated executable mappings on the source and target ontologies. This tool is designed to make it easier for system engineers to perform a semantic data integration task.

4.1 Conversion Process

The SDI tool enables the transformation of UML notations into the Rule Interchange Format (RIF) [RIF Core] and subsequently into SPARQL [SPARQL 2008] queries which is a concrete mapping syntax and ready to be executable. In this research, RIF was chosen as the intermediate format because it has the potential to interoperate with other concrete mapping syntaxes. For example, there is an existing RIF syntax specification [RIF SWC] for RDF and OWL Compatibility. This strategy potentially enables the transformation between the intermediate format and other concrete mapping syntaxes such as the ontology for axiom-based integration in future. SPARQL was chosen as a target concrete mapping syntax for Query-based integration approach. Using SPARQL has proven to be the most practical approach - in terms of integration process speed - among three different semantic data integration approaches [Keeney 2011], i.e. Axiom-based integration using ontology, Rule-based integration using SWRL and Query-based integration using SPARQL. There is no loss of expressivity in using RIF as an intermediate syntax between UML notations and SPARQL queries for the core mapping types because the proposed UML notations are designed to support the core mapping types. RIF and SPARQL support direct, conditional and transformation mapping types that may require expressing data range constraints or data value transformations (arithmetic operation) as discussed in related research. Fig. 4-1 shows the overall SDI tool generation

process transforming the UML notations into concrete executable mappings in SPARQL queries through the intermediate RIF syntax.

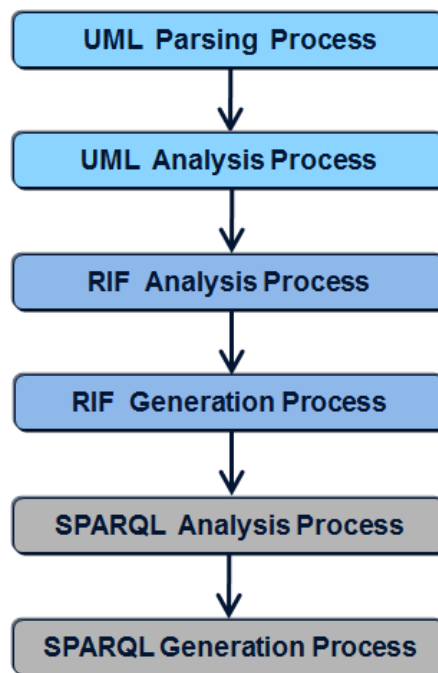


Figure 4-1 SDI tool generation process overview

4.2 UML to RIF Conversion

The UML notations (UML and UML constraint expressions) are interpreted into the RIF syntax. The following sections describe the translation method from UML notations into the RIF syntax for each core mapping type.

4.2.1 Direct Mapping Type

This section describes the translation method and an example from UML notations into the RIF syntax for the direct mapping type. Table 4-1 shows the RIF syntax according to UML notations with corresponding ontology syntax for the notation.

Table 4-1 Correspondence between UML notation and RIF syntax for direct mapping type

UML Notation	OWL Syntax	RIF Syntax	Comment
Package::ownedMember Classifier	Ontology Document	Import with OWL RDF-Based entailment	Package name is the document name. Multiple documents can be imported.
First part of the classifier separated by a double colon ('::')	A prefix that represents the namespace of document	Prefix (<Name> < <Namespace> >)	<Name> is a string entity that represents <Namespace> in IRI/URI form.
UML::Class with the classifier 'Ont1::Class1' in the name field	<owl:Class rdf:about="Ont1:Class1"/>	?blankNode [rdf:type -> Ont1:Class1]	This type of syntax is called RIF frame formula of the RDF triple correspondent form. Each '?blankNode', 'rdf:type', and 'Ont1:Class1' corresponds to subject, predicate, and object in RDF triple.
<<Unite>> stereotyped UML::AssociationClass with the name 'DirectMapping'	<owl:Class rdf:about="#Class2"> <owl:equivalentClass rdf:resource="#Class1"/> </owl:Class>	?blankNode [rdf:type -> <UML::Class1>] :- ?blankNode [rdf:type -> <UML::Class2>]	" ϕ :- ψ " is a rule implication. Consequent (ϕ) is true, when the antecedent (ψ) is true.

Fig. 4-2 shows a translation example of UML notations into the RIF syntax according to the table 4-1. In this figure, bold and italic keywords in the RIF syntax are a template which means these keywords will not be altered by the UML notation changes in this mapping type. UML notations in Fig. 4-2 represent the mapping corresponding to "Integrate 'Layer3Switch' class individuals to 'Router' class".

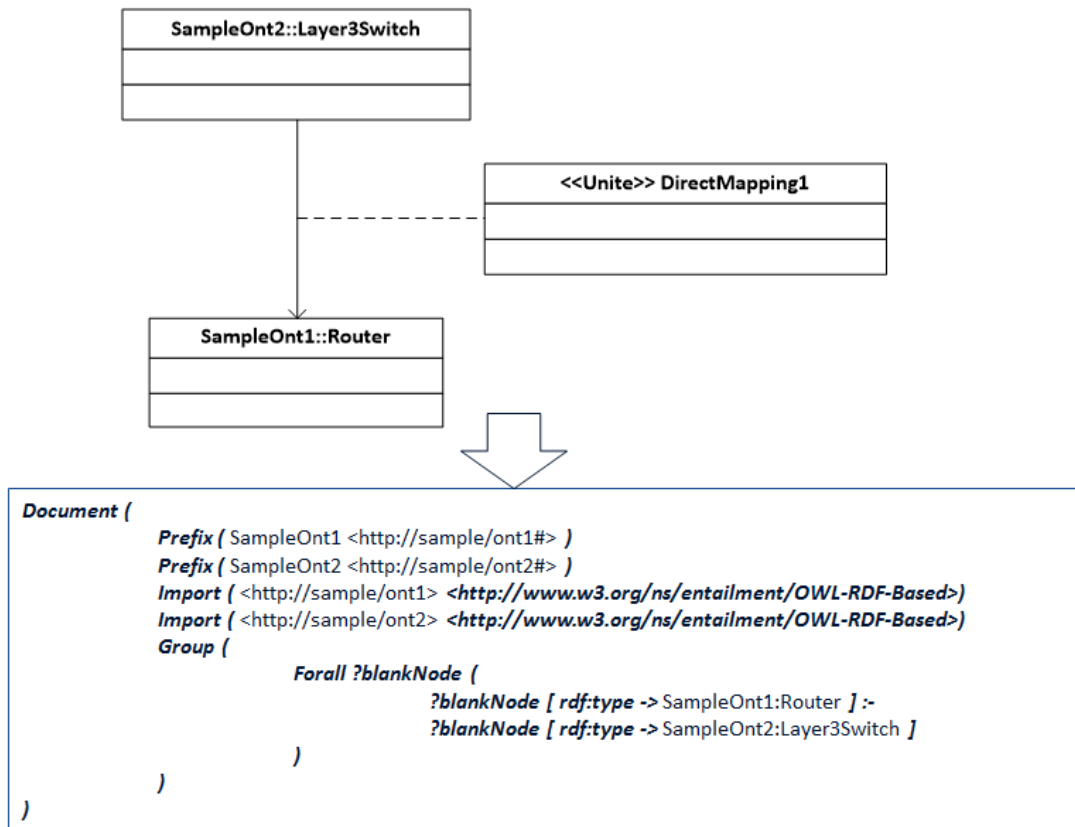


Figure 4-2 Translation example of UML notations to RIF syntax for direct mapping type

4.2.2 Data Range Mapping Type

This section describes the translation method and examples from UML notations into the RIF syntax for the data range mapping type. Table 4-2 shows the RIF syntax according to UML notations with corresponding ontology syntax for the notation.

Table 4-2 Correspondence between UML notation and RIF syntax for data range mapping type

UML Notation	OWL Syntax	RIF Syntax	Comment
<<Unite>> stereotyped UML::AssociationClass with the name 'DataRangeMapping'		Prefix (pred <http://www.w3.org/2007/rif-builtin-predicate#>)	Prefix for RIF predicate built-ins for comparison operation
Comparison operator in UML::Constraint such as '<', '>', '<=', '>=', '==', or '<>'		External (pred: <RIF predicate built-in for comparison operation>)	<RIF predicate built-in for comparison operation> can be one of following: numeric-equal(), numeric-not-equal(), numeric-less-than(), numeric-less-than-or-equal(), numeric-greater-than(), or numeric-greater-than-or-equal(). [RIF Built-Ins]
Classifier used in UML::Constraint of the property 'Constraint' represents UML::Property of a certain UML::Class.	<owl:DatatypeProperty rdf:ID="A" />	?blankNode [A -> ?value]	?value can be used as a variable bounded in a RULE.
Classifier used in UML::Constraint of the property 'Constraint' represents UML::Property of a certain UML::Class.	<owl:DatatypeProperty rdf:ID="A" /> <owl:DatatypeProperty rdf:ID="B" />	?blankNode [A -> ?v1] ?blankNode [B -> ?v2]	Adding the RIF frame formula in FORMULA part of RULE within 'And' or 'Or' relation depending on the number of datatypeProperty variables. ?v1 and ?v2 can be used as a variable bounded in a RULE.
UML::Class at the association to represents a target class.	<rdf:Description rdf:ID="blankNode"> <rdf:type rdf:resource="#object" /> </rdf:Description>	?blankNode [rdf:type -> ?object]	This represents the constrained class instances, i.e. blankNode, are member of the ?object class.

Fig. 4-3 and Fig. 4-4 show translation examples of UML notations into the RIF syntax according to the table 4-2. In these figures, bold and italic keywords in the RIF syntax are a template which means these keywords will not be altered by the UML notation changes in this mapping type. UML notations in Fig. 4-3 represent the mapping corresponding to “Integrate ‘Femto’ class individuals that are constrained by the data range (successRate1 > 90) of single datatype property ‘successRate1’ to ‘HighHandoverFemto’ class”.

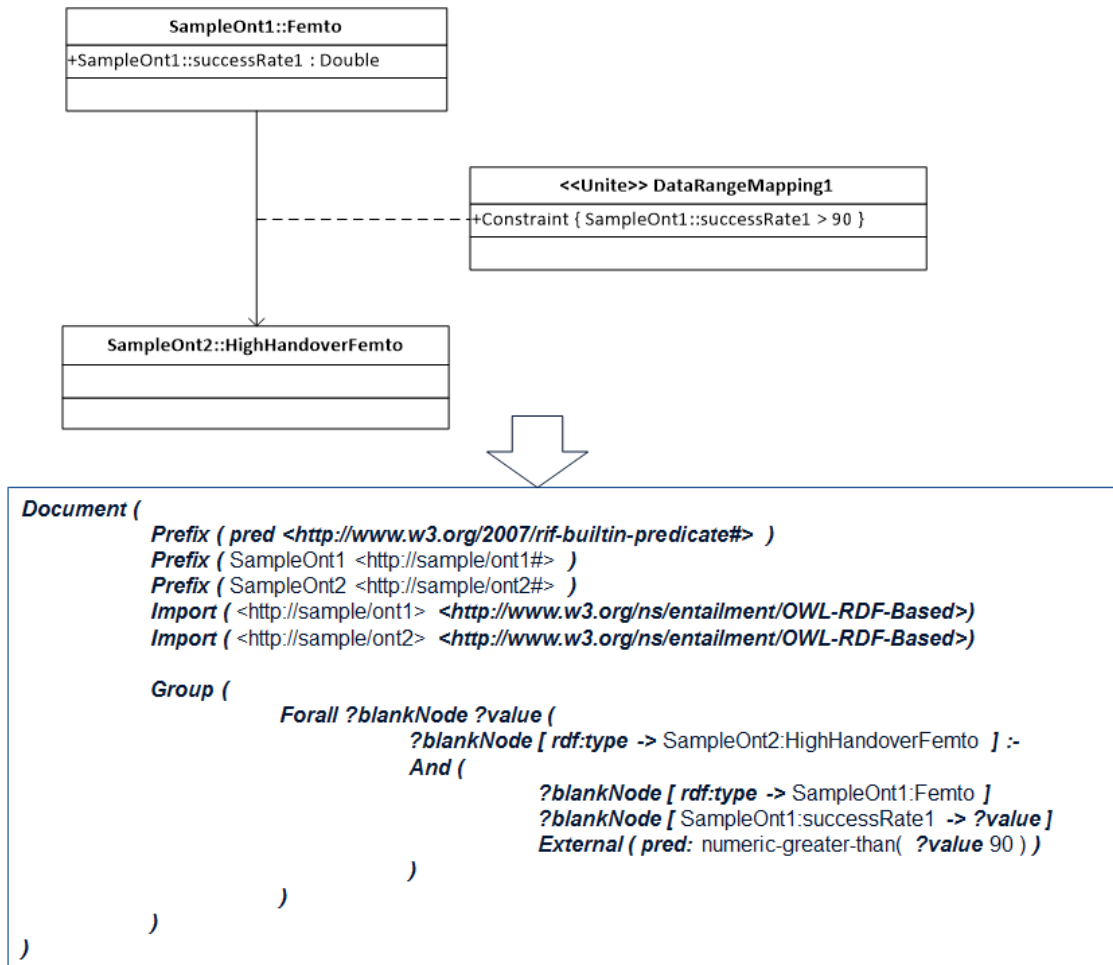


Figure 4-3 Translation example of UML notations to RIF syntax for data range mapping type

UML notations in Fig. 4-4 represent the mapping corresponding to “Integrate ‘Femto’ class individuals that are constrained by the data range (successRate1 > 90 AND successRate2 > 80) of multiple datatype properties ‘successRate1’ and ‘successRate2’ to ‘HighHandoverFemto’ class”.

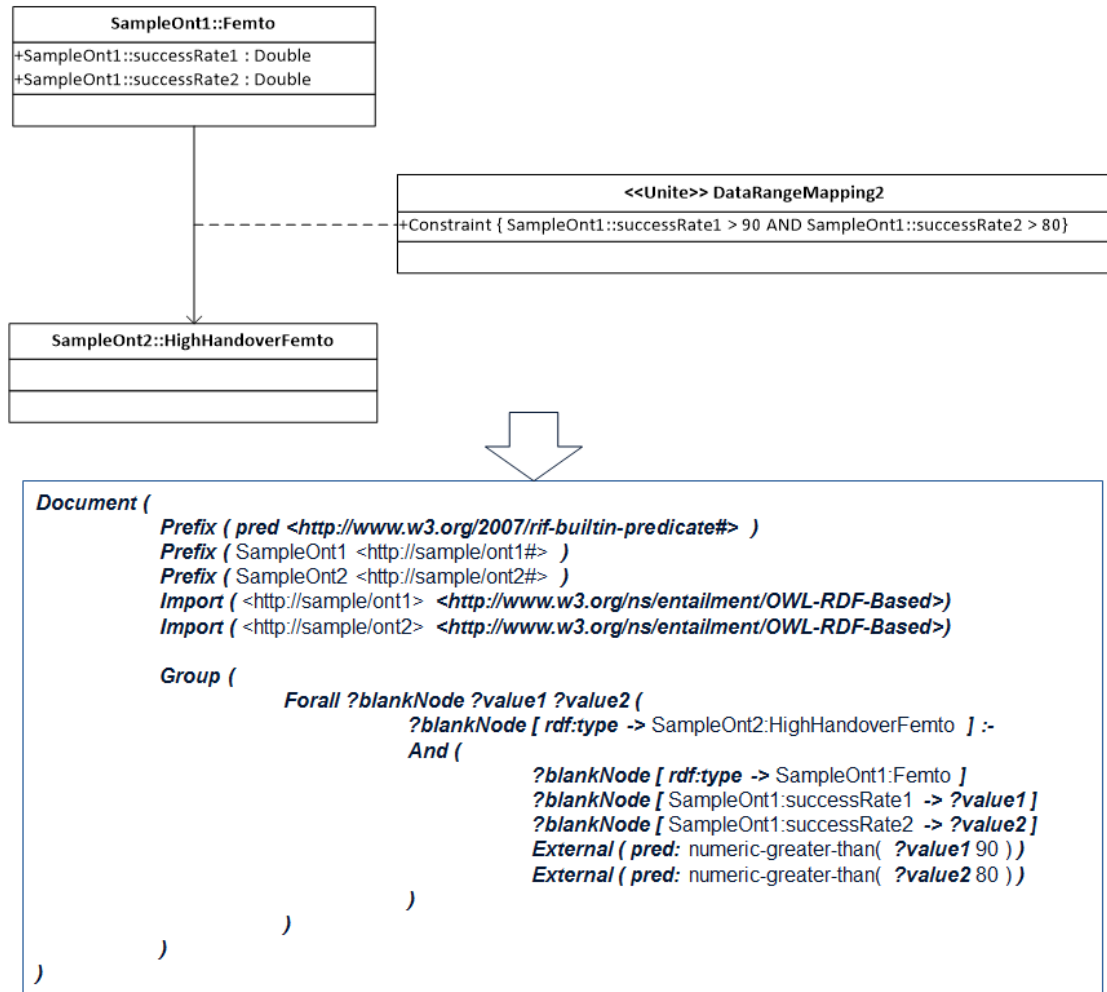


Figure 4-4 Translation example of UML notations to RIF syntax for data range mapping type

4.2.3 Unit Transformation Mapping Type

This section describes the translation method and examples from UML notations into the RIF syntax for the unit transformation mapping type. Table 4-3 shows the RIF syntax according to UML notations with corresponding ontology syntax for the notation.

Table 4-3 Correspondence between UML notation and RIF syntax for unit transformation mapping type

UML Notation	OWL Syntax	RIF Syntax	Comment
<<Unite>> stereotyped UML::AssociationClass with the name 'UnitTransformationMapping'		Prefix (func <http://www.w3.org/2007/rif -builtin-function#>)	Prefix for RIF functional built-ins for arithmetic operation
Arithmetic operator in UML::Constraint such as '*', '/', '+', or '-'		External (pred: <RIF functional built-in for mathematical operation>)	<RIF functional built-in for mathematical operation> can be one of following: numeric-add(), numeric-subtract(), numeric-multiply(), numeric-divide(). [RIF Built-Ins]
Classifier used in UML::Constraint of the property 'Constraint' represents UML::Property of a certain UML::Class.	<owl:DatatypeProperty rdf:ID="A" /> <owl:DatatypeProperty rdf:ID="B" />	?blankNode [A -> ?value1] ?blankNode [B -> ?value2]	?value1 and ?value2 can be used as a variable bounded in a RULE.
UML::Class at the association to represents a target class.	<rdf:Description rdf:ID="blankNode"> <rdf:type rdf:resource="#object" /> </rdf:Description>	?blankNode [rdf:type -> ?object]	This represents the constrained class instances, i.e. blankNode, are member of the ?object class.

Fig. 4-5 and Fig. 4-6 show translation examples of UML notations into the RIF syntax according to the table 4-3. In these figures, bold and italic keywords in the RIF syntax are a template which means these keywords will not be altered by the UML notation changes in this mapping type. UML notations in Fig. 4-5 represent the mapping corresponding to "Integrate 'Femto' class individuals that have the datatype property 'FailureRate' to 'Device' class that has the datatype property 'FailureGauge' with the transformed value by the arithmetic operation (FailureGauge = FailureRate * 100)".

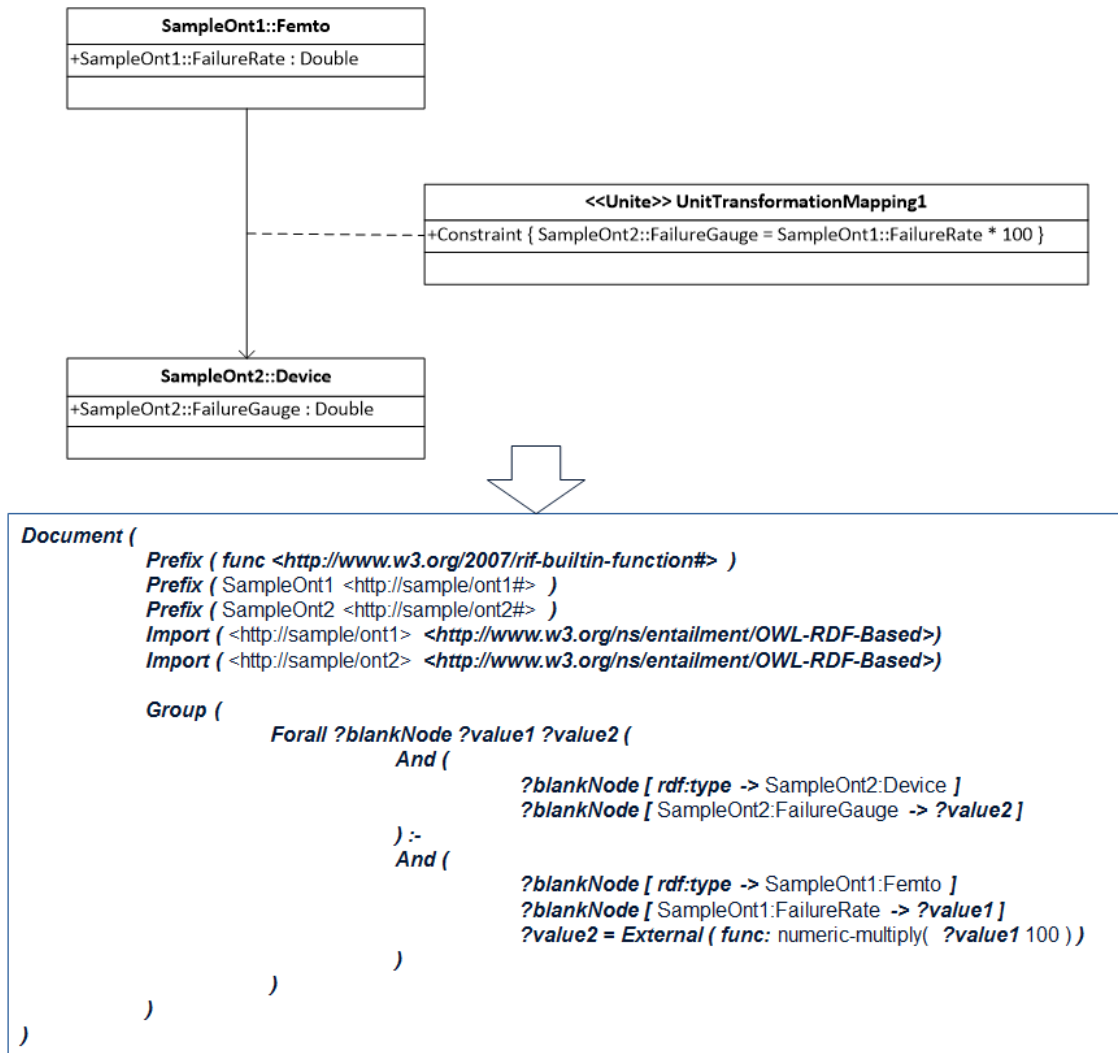


Figure 4-5 Translation example of UML notations to RIF syntax for unit transformation mapping type

UML notations in Fig. 4-6 represent the mapping corresponding to “Integrates ‘Femto’ class individuals that have the datatype properties: ‘FailureRate1’ and ‘FailureRate2’ to ‘Device’ class that has the datatype property ‘FailureGauge’ with the transformed value by the arithmetic operation (FailureGauge = FailureRate1 + FailureRate2)”.

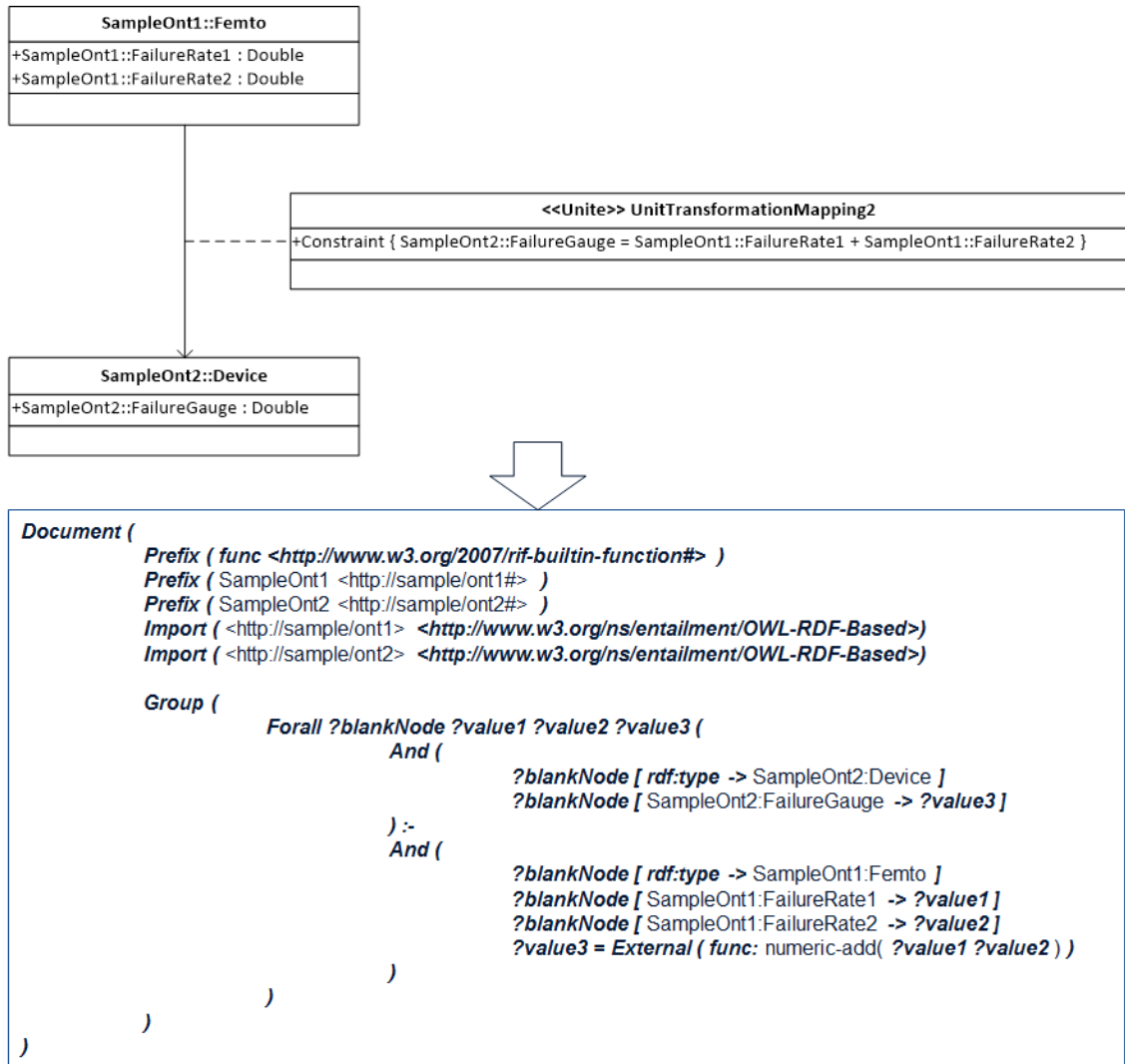


Figure 4-6 Translation example of UML notations to RIF syntax for unit transformation mapping type

4.3 RIF to SPARQL Conversion

The RIF syntax interpreted from the UML notations is translated into SPARQL queries. This section describes the translation method from the RIF syntax to SPARQL queries and examples for each core mapping type. Table 4-4 shows the SPARQL syntax based on the RIF syntax which is based on the UML notations.

Table 4-4 Correspondence between RIF syntax and SPARQL syntax

RIF Syntax	SPARQL Syntax	Comment
Prefix	PREFIX	Prefix places in the beginning of RIF document.
Group	A query consists of CONSTRUCT and WHERE clauses.	RIF document can contain several groups, and each group represents one complete SPARQL query.
?blankNode	\$instance	?blankNode is used in a RULE. RULE is the term used in RIF for a rule implication ($\phi :- \psi$). Consequent (ϕ) is true, when the antecedent (ψ) is true.
rdf:type	a	rdf:type places in Frame used in RULE. Frame is the term used in RIF for "TERM '[' (TERM '->' TERM)* ']' ", where TERM can be "constant, variable, list or (external) expression".
ATOMIC Frame	CONSTRUCT clause	ATOMIC in RIF means consequent part (ϕ) of the rule.
FORMULA	WHERE clause	FORMULA in RIF means antecedent part (ψ) of the rule.
RIF predicate built-in for comparison operation	FILTER	FILTER in SPARQL is used to express the constraint. In this case, comparison operation is expressed in this filter.
RIF functional built-in for arithmetic operation	BIND	BIND in SPARQL allows a value to be assigned to a variable from a graph pattern.

Fig. 4-7 shows a translation example of a RIF document to SPARQL query according to Table 4-4 for the direct mapping type. SPARQL query in Fig. 4-7 represents the mapping that all the instances in Layer3Switch class by WHERE clause will be an instance of Router class by CONSTRUCT clause.

```

Document (
  Prefix ( SampleOnt1 <http://sample/ont1#> )
  Prefix ( SampleOnt2 <http://sample/ont2#> )
  Import ( <http://sample/ont1> <http://www.w3.org/ns/entailment/OWL-RDF-Based> )
  Import ( <http://sample/ont2> <http://www.w3.org/ns/entailment/OWL-RDF-Based> )
  Group (
    Forall ?blankNode (
      ?blankNode [ rdf:type -> SampleOnt1:Router ] :-
      ?blankNode [ rdf:type -> SampleOnt2:Layer3Switch ]
    )
  )
)

```



```

PREFIX SampleOnt1: <http://sample/ont1#>
PREFIX SampleOnt2: <http://sample/ont2#>

CONSTRUCT {
  $instance a SampleOnt1:Router .
}

WHERE {
  $instance a SampleOnt2:Layer3Switch .
}

```

Figure 4-7 Translation example of RIF document to SPARQL query for direct mapping type

Fig. 4-8 shows a translation example of a RIF document to SPARQL query according to Table 4-4 for the data range mapping type. SPARQL query in Fig. 4-8 represents the mapping that the instances in Femto class, which the property successRate1 is constrained by FILTER (value of the property > 90), will be an instance of HighHandoverFemto class by CONSTRUCT clause.


```

Document (
  Prefix ( pred <http://www.w3.org/2007/rif-builtin-predicate#> )
  Prefix ( SampleOnt1 <http://sample/ont1#> )
  Prefix ( SampleOnt2 <http://sample/ont2#> )
  Import ( <http://sample/ont1> <http://www.w3.org/ns/entailment/OWL-RDF-Based> )
  Import ( <http://sample/ont2> <http://www.w3.org/ns/entailment/OWL-RDF-Based> )

  Group (
    Forall ?blankNode ?value (
      ?blankNode [ rdf:type -> SampleOnt2:HighHandoverFemto ] :-
      And (
        ?blankNode [ rdf:type -> SampleOnt1:Femto ]
        ?blankNode [ SampleOnt1:successRate1 -> ?value ]
        External ( pred: numeric-greater-than( ?value 90 ) )
      )
    )
  )
)

```



```

PREFIX SampleOnt1: <http://sample/ont1#>
PREFIX SampleOnt2: <http://sample/ont2#>

CONSTRUCT{
  $instance a SampleOnt2:HighHandoverFemto .
}

WHERE{
  $instance a SampleOnt1:Femto .
  $instance SampleOnt1:successRate1 $value .
  FILTER ( $value > 90 )
}

```

Figure 4-8 Translation example of RIF document to SPARQL query for data range mapping type

Fig. 4-9 shows a translation example of a RIF document to SPARQL query according to Table 4-4 for the unit transformation mapping type. SPARQL query in Fig. 4-9 represents the mapping that the instances in Femto class, which the value of property FailureRate is assigned to a variable with arithmetic operation by BIND (value of the property * 100), will be an instance of Device class by CONSTRUCT clause with a new value of property FailureGauge from the variable in BIND clause.

```

Document (
  Prefix ( func <http://www.w3.org/2007/rif-builtin-function#> )
  Prefix ( SampleOnt1 <http://sample/ont1#> )
  Prefix ( SampleOnt2 <http://sample/ont2#> )
  Import ( <http://sample/ont1> <http://www.w3.org/ns/entailment/OWL-RDF-Based> )
  Import ( <http://sample/ont2> <http://www.w3.org/ns/entailment/OWL-RDF-Based> )

  Group (
    Forall ?blankNode ?value1 ?value2 (
      And (
        ?blankNode [ rdf:type -> SampleOnt2:Device ]
        ?blankNode [ SampleOnt2:FailureGauge -> ?value2 ]
      ) :-
      And (
        ?blankNode [ rdf:type -> SampleOnt1:Femto ]
        ?blankNode [ SampleOnt1:FailureRate -> ?value1 ]
        ?value2 = External ( func: numeric-multiply( ?value1 100 ) )
      )
    )
  )
)

```



```

PREFIX SampleOnt1: <http://sample/ont1#>
PREFIX SampleOnt2: <http://sample/ont2#>

CONSTRUCT {
  $instance a SampleOnt2:Device .
  $instance SampleOnt2:FailureGauge $value2 .
}
WHERE {
  $instance a SampleOnt1:Femto .
  $instance SampleOnt1:FailureRate $value1 .
  BIND ( ( $value1 * 100 ) AS $value2 )
}

```

Figure 4-9 Translation example of RIF document to SPARQL query for unit transformation mapping type

4.4 Implementation

The SDI tool was developed in order to evaluate the MOUSE approach. This tool allows a system engineer to express mappings using UML and then automatically generates the corresponding executable semantic mappings. It also generates an integrated ontology as a computation result of executing the generated semantic mappings on the source and target ontologies. To select a UML modelling tool, this research surveyed open source tools because the implementation of the SDI tool may require a modification of the tool's source codes for customization. In the initial experiment, the Modelio modelling tool [Modelio Tool] was selected because there

were few open source tools that support expressing OCL constraint language. Modelio tool is a heavy-weight application that is equipped with unnecessary functions such as supporting BPMN (Business Process Model and Notation) which is not used in this research, and the tool is not user-friendly in the case of UML notations alignment and creation. StarUML⁷ modelling tool [StarUML Project] version 5.0 was selected to draw UML notations for the mappings in this implementation. StarUML tool functions are modularized and any unnecessary modules can be disabled. This tool offers plugin programming that enables an external application to be executed inside the tool for the customization. The development was undertaken using Java language with the version 1.6.0.20 for the SDI tool, and the developed tool is embedded into the StarUML by using plugin programming. Fig. 4-10 shows the overall architecture of the tool.

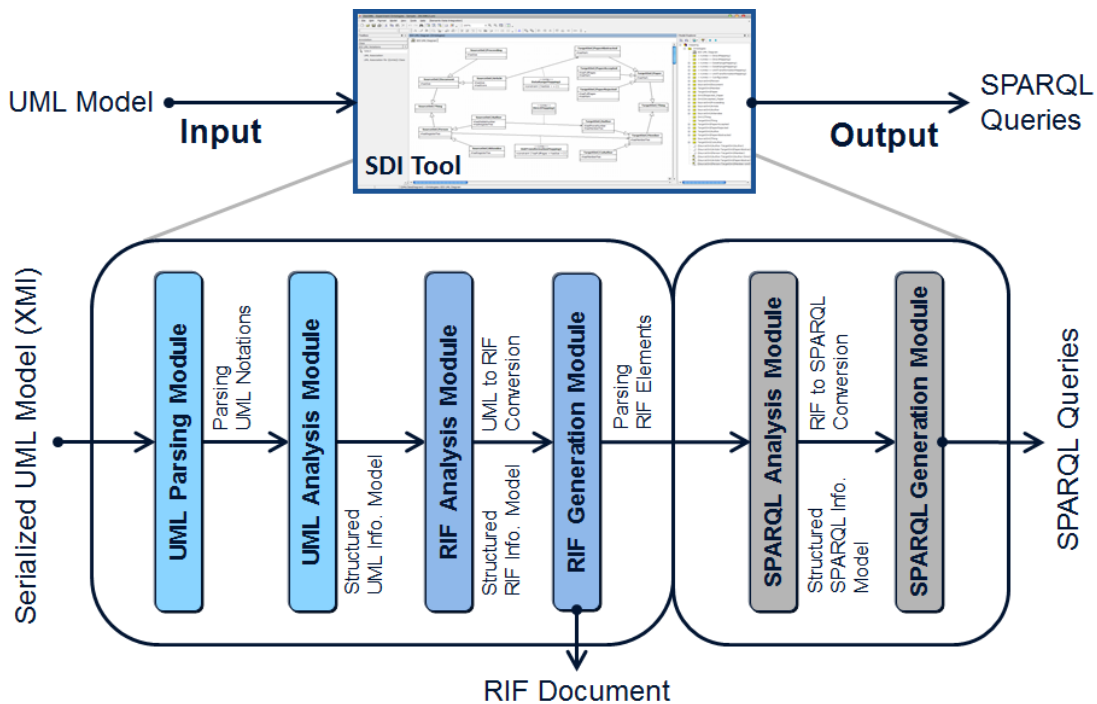


Figure 4-10 SDI tool architecture overview

⁷ StarUML is an open source project to develop a fast, flexible, extensible, featureful, and freely-available UML tool. The goal of the StarUML project is to build a software modelling tool that is a compelling replacement of commercial UML tools. [StarUML Project]

4.4.1 SDI Tool Modules

This section describes each module in the tool architecture shown in Fig. 4-10.

- 1) **UML Parsing Module:** This module extracts UML elements from the serialized UML model in XML format. For parsing XML, the module uses java DOM(Document Object Model) core 2.0 version library that is based on org.w3c.dom.Node interface [DOM Core]. This library treats all XML elements as nodes and parses them into a tree structured model. The memory holds the whole XML document using this library. This enables fast processing by allowing random access to the required information of the XML document. This module returns parsed UML notations that are designed by the user using the graphic user interface of the tool.
- 2) **UML Analysis Module:** This module creates data in the structured UML information model from the parsed UML notations drawn by the user.
- 3) **RIF Analysis Module:** This module creates data in the structured RIF information model from the data in the structured UML information model using the translation method described in the section 4.2 UML to RIF Conversion.
- 4) **RIF Generation Module:** This module generates a RIF document from the data in the structured RIF information model. This module shows the output file information and the process time information to generate a RIF document.
- 5) **SPARQL Analysis Module:** This module creates data in the structured SPARQL information model from the data in the structured RIF information model using the translation method described in the section 4.3 RIF to SPARQL Conversion.
- 6) **SPARQL Generation Module:** This module generates SPARQL queries from the data in the structured SPARQL information model. This module shows the output files information and the process time information to generate SPARQL queries.

4.4.2 Structured Information Models

This tool uses three different structured information models. The data in these information models are saved in memory during the process time and deleted after the conversion process finishes. Each structured information model holds the elements and relationships between the elements of UML notations, RIF document, and SPARQL queries.

The first structured information model holds the UML notations information. Fig. 4-11 shows a UML class diagram for this structured information model. This model saves and keeps the relationships of UML notations drawn by the user. For example, in the diagram, UMLClass, UMLProperty, UMLAssociation and UMLAssociationClass can be mapped to UML class, property, association and association class notations.

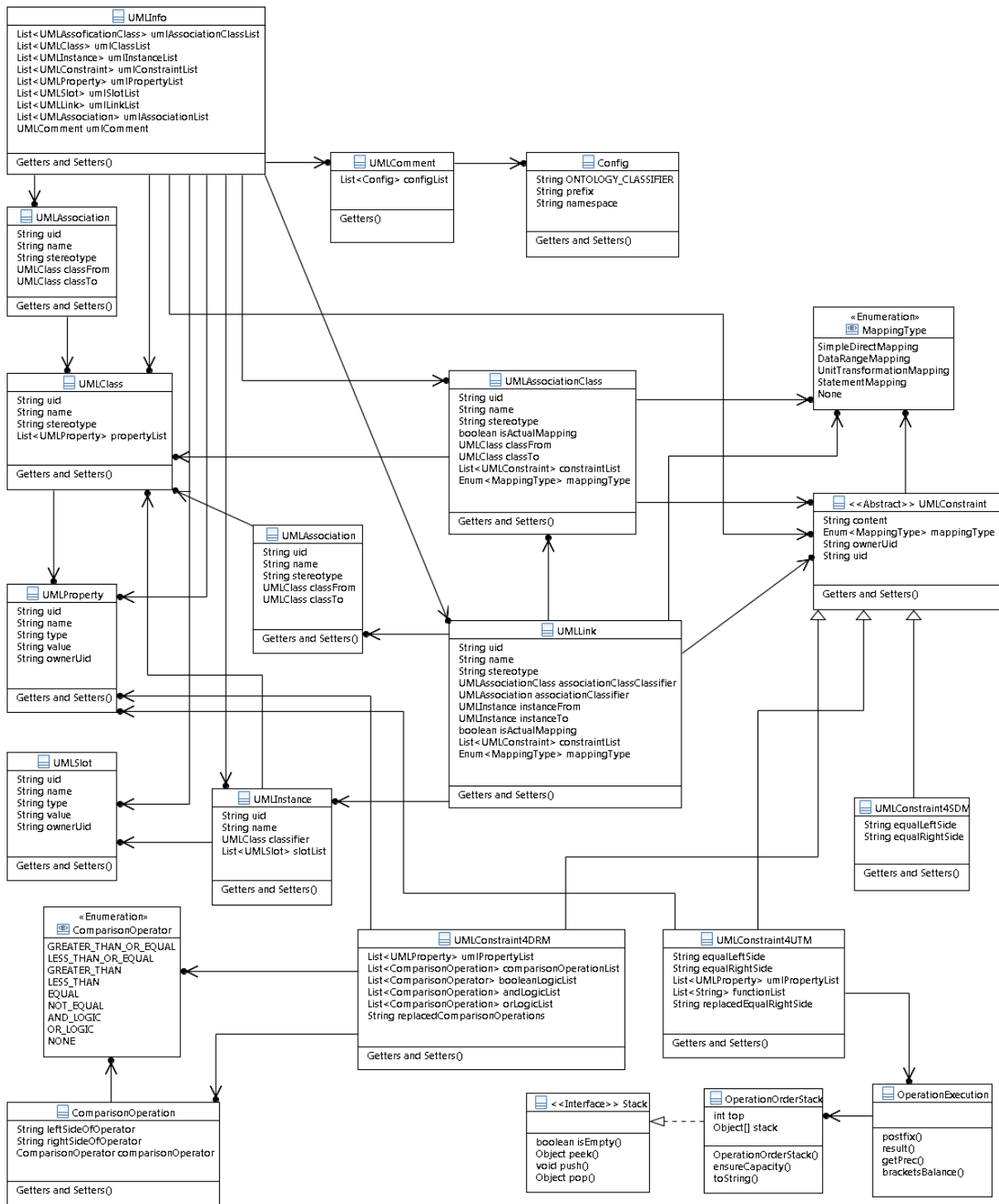


Figure 4-11 UML Class diagram for structured UML information model

The second structured information model holds the RIF document information. Fig. 4-12 shows a UML class diagram for this structured information model. This model saves RIF information with relationships for rule implications.

For example, in the diagram, RIFGroup has rdfGraphList4Consequent and rdfGraphList4Antecedent properties. Each property can be mapped to the consequent and antecedent part of the RIF rule implication.

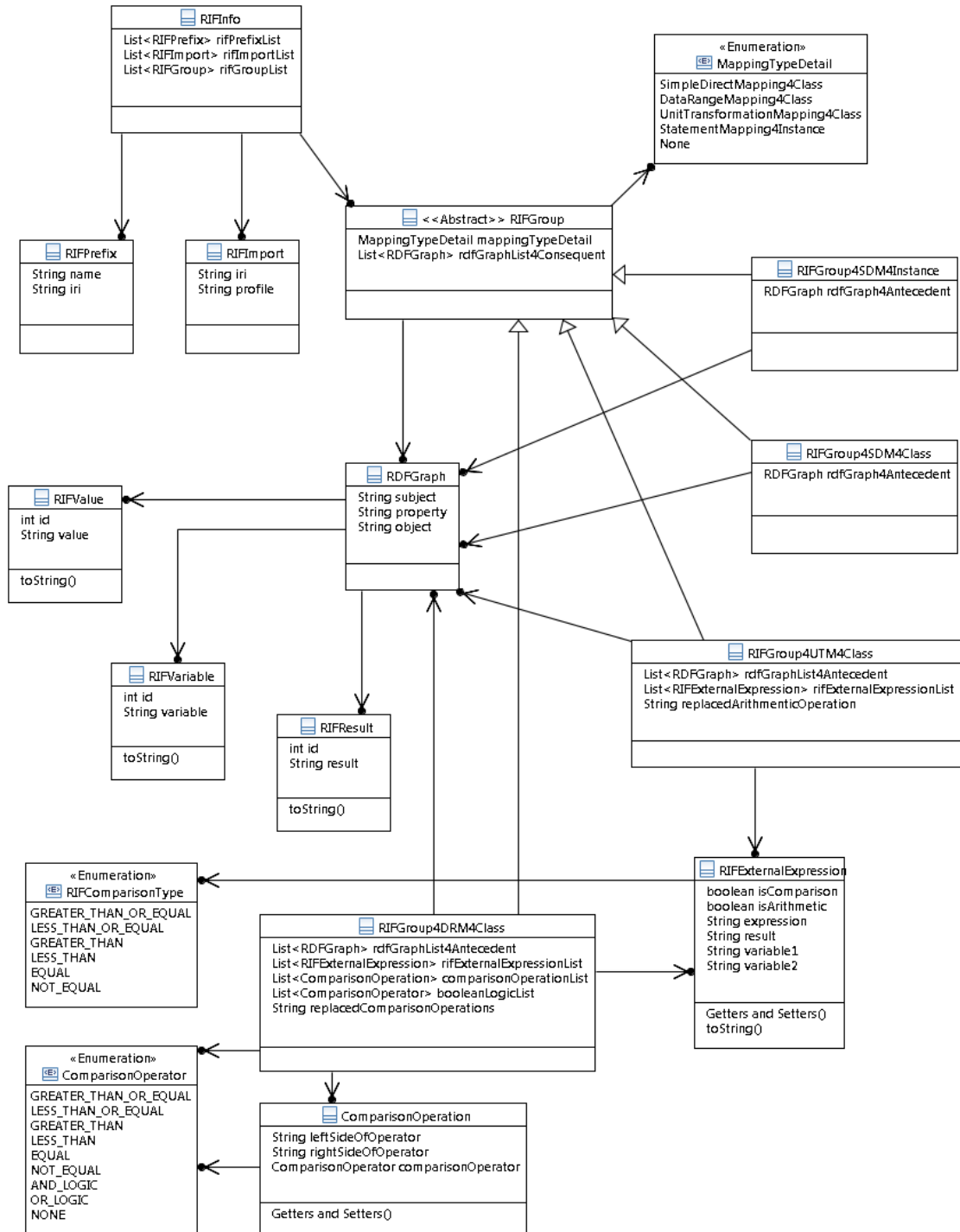


Figure 4-12 UML Class diagram for structured RIF information model

The third structured information model holds the SPARQL queries information. Fig. 4-13 shows a UML class diagram for this structured information model. This model saves SPARQL information to generate SPARQL queries. For example, in the diagram, SPARQLConstructClause, SPARQLWhereClause, SPARQLFilterClause and SPARQLBindClause can be mapped to SPARQL construct, where, filter and bind clauses in the query syntax.

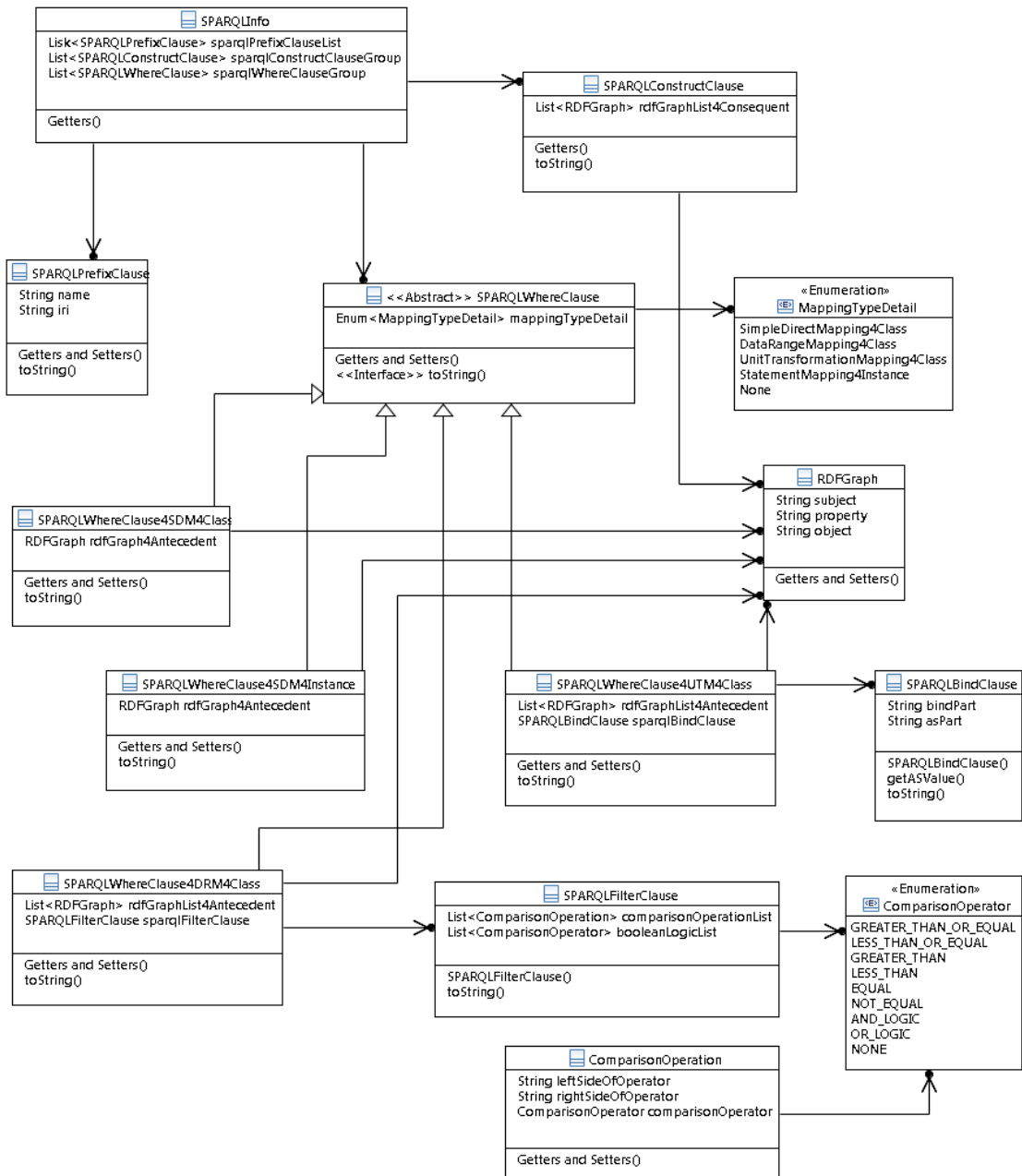


Figure 4-13 UML Class diagram for structured SPARQL information model

4.5 A walked through example

This subsection describes a running example of the SDI tool. The tool does not automatically import ontologies and represent them in UML notations. This auto-representation of UML notations for ontologies will be available in the future. In the current version of the tool, a manual drawing of the UML notations that represent source and target ontologies must be used. This research assumes that the ontologies are first represented in the proposed UML notations as a prerequisite to conduct the mappings. Fig. 4-14 shows a mapping represented by using UML notations in the StarUML graphic user interface. There is a hidden UML class that encodes the mapping information about ontology prefix and namespace [OWL Guide]. For example, this hidden class has the information about the actual namespace ‘http://kdeg.scss.tcd.ie/ontology/test/source’ for prefixed ontology classifier ‘SourceOnt’. In this figure, ‘ClassSB’ class is mapped to ‘ClassTB’ class with the data range constraint ($20 < \text{AttributeSB1} < 50$ or $\text{AttributeSB2} > 50$) on the properties in ‘ClassSB’ class.

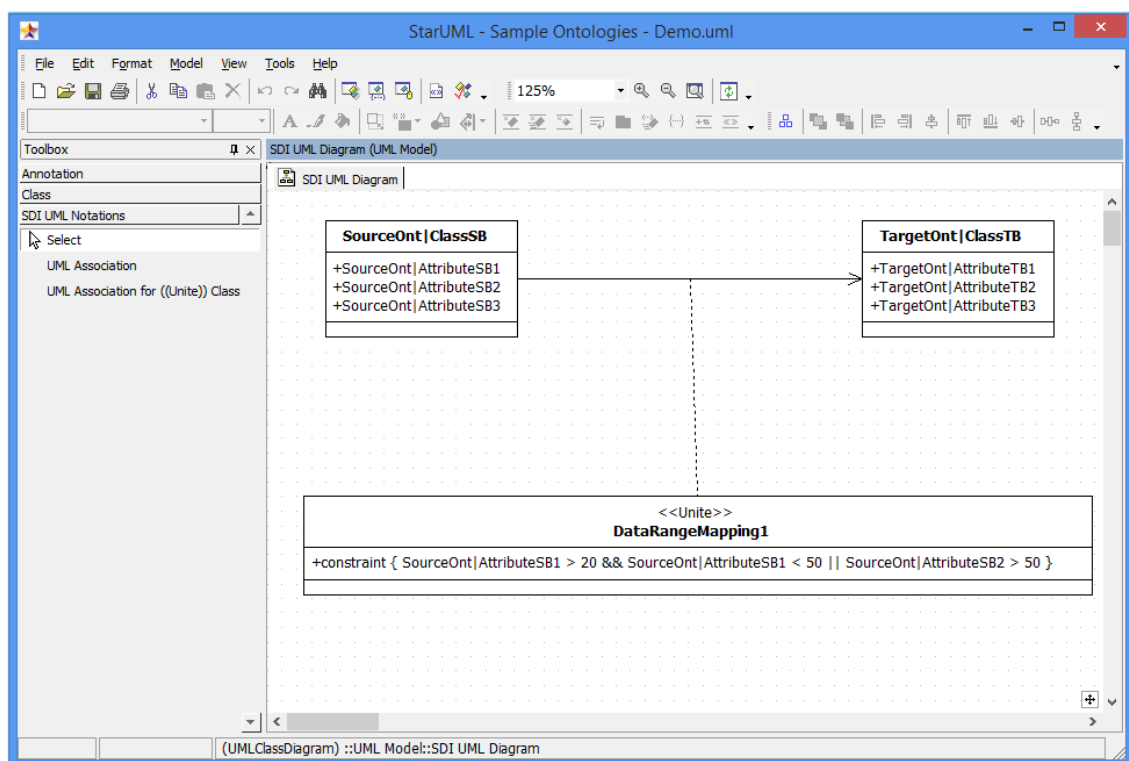


Figure 4-14 ScreenShot of the StarUML graphic user interface

The Boolean expressions in the implemented SDI tool are tested up to two combinations of Boolean logics. This means the tool supports up to 4 Boolean expression combinations by two Boolean logics, i.e. AND(&&) and OR(||). For example, in the case of ‘comparison operation 1 && comparison operation 2 && comparison operation 3’, all the comparison operations need to be satisfied to be integrated. In the case of ‘comparison operation 1 || comparison operation 2 || comparison operation 3’, at least one of the comparison operation needs to be satisfied to be integrated. In the case of ‘comparison operation 1 && comparison operation 2 || comparison operation 3’, either both comparison operations 1 and 2 need to be satisfied or comparison operation 3 needs to be satisfied to be integrated. In the case of ‘comparison operation 1 || comparison operation 2 && comparison operation 3’, either comparison operation 1 needs to be satisfied or both comparison operations 2 and 3 need to be satisfied to be integrated.

The SDI tool automatically transforms UML notations shown in Fig. 4-14 to SPARQL queries through intermediate RIF document. Fig. 4-15 shows the RIF document generated from the UML notations. In the figure, it can be seen that RIF predicate built-in functions (pred:numeric-greater-than and pred:numeric-less-than) are used to express the data range mapping.

```

1 Document (
2
3 Prefix ( TargetOnt <http://kdeg.scss.tcd.ie/targetOnt#> )
4 Prefix ( SourceOnt <http://kdeg.scss.tcd.ie/sourceOnt#> )
5
6 Import ( <http://kdeg.scss.tcd.ie/targetOnt> <http://www.w3.org/ns/entailment/OWL-RDF-Based> )
7 Import ( <http://kdeg.scss.tcd.ie/sourceOnt> <http://www.w3.org/ns/entailment/OWL-RDF-Based> )
8
9 Group (
10   Forall ?blankNode2 ?value1 ?value2 (
11     ?blankNode2 [ rdf:type -> TargetOnt:ClassTB ] :-
12     And (
13       ?blankNode2 [ rdf:type -> SourceOnt:ClassSB ]
14       ?blankNode2 [ SourceOnt:AttributeSB1 -> ?value1 ]
15       ?blankNode2 [ SourceOnt:AttributeSB2 -> ?value2 ]
16       Or (
17         And (
18           External ( pred:numeric-greater-than( ?value1 20 ) )
19           External ( pred:numeric-less-than( ?value1 50 ) )
20         )
21         External ( pred:numeric-greater-than( ?value2 50 ) )
22       )
23     )
24   )
25 )
26 )
27 )
28 )

```

Figure 4-15 RIF document generated from the UML notations

Fig. 4-16 shows a SPARQL query generated from the RIF document in Fig. 4-15. In Fig. 4-16, it can be seen that FILTER is used in the WHERE clause to express the data range constraint and CONSTRUCT is used to generate the desired RDF graph according to the mapping result.

```

1 PREFIX TargetOnt: <http://kdeg.scss.tcd.ie/targetOnt#>
2 PREFIX SourceOnt: <http://kdeg.scss.tcd.ie/sourceOnt#>
3
4 CONSTRUCT {
5   ?instance2 a TargetOnt:ClassTB .
6 }
7
8 WHERE {
9   ?instance2 a SourceOnt:ClassSB .
10  ?instance2 SourceOnt:AttributeSB1 ?value1 .
11  ?instance2 SourceOnt:AttributeSB2 ?value2 .
12  FILTER ( ?value1 > 20 && ?value1 < 50 || ?value2 > 50 )
13 }
14

```

Figure 4-16 SPARQL query generated from the RIF document

For more details about the tool, APPENDIX B shows the SDI tool user manual in GUI mode with examples, and APPENDIX C shows the SDI tool java archive file [JAR Spec] usage in command line mode.

This chapter described the development of the SDI tool that implements the proposed UML-based abstract mapping syntax and supports the core mapping types. The tool accurately and automatically transforms the abstract mapping syntax into a concrete mapping syntax. This developed tool will ease the performance of semantic data integration by a system engineer. The following chapter describes the user based experiments that evaluated the MOUSE approach by conducting a semantic data integration task using the SDI tool.

5 System Engineers Experiment

This chapter describes the experiment undertaken by system engineers to evaluate the proposed MOUSE approach that incorporates the abstract mapping syntax and automatic translation into executable mappings. The subsections of this chapter describe the experiment question, hypothesis, method and analysis.

5.1 Experiment Question

The question investigated through this experiment was “*Can a participant undertake ontology mappings using the UML-based abstract mapping syntax in order to successfully perform a semantic data integration task?*”

In this question, a participant means a system engineer who will not be an ontology expert or have knowledge engineering experience. To evaluate success of the semantic data integration task, the mappings that were auto-generated by the tool on behalf of each participant were executed in order to create an integrated ontology which in turn was then examined.

5.2 Experiment Hypothesis

From the experiment question stated above, four hypotheses were formulated to be evaluated.

1. *The SDI tool can correctly capture the mapping intention of a participant for the core mapping types.*
2. *The SDI tool can accurately transform the abstract mappings of each participant into SPARQL executable mappings.*
3. *The proposed UML-based abstract mapping syntax is easy to use for the participants.*
4. *The SDI tool is easy to use for participants.*

5.3 Experiment Method

This section explains how the experiment was carried out. First, the details of the participants in the experiment are explained. Second, the materials used in the experiment are described. Finally, the procedure of the experiment is outlined.

5.3.1 Participants

The experiment was conducted by 15 participants who were postgraduate students or academics in the School of Computer Science and Statistics in Trinity College Dublin or were an employee of Alcatel-Lucent Bell Labs Ireland. They were selected by email distribution using the mailing lists of the department. 15 participants can be considered a reasonable number to evaluate the SDI tool; considering that in ontology mapping research [Noy 2003][Raghavan 2005][Aumuller 2005][Falconer 2007b] the number of participants typically used for evaluation has been less than 10 participants.

The participants are computer science researchers with varying levels of experience as software engineering practitioners, and were all expected to have basic UML knowledge. Participants had at least basic knowledge to draw essential UML notations as a system engineer. The expertise about ontology knowledge varied for each participant. Most of them did not have sufficient ontological knowledge that would be needed to perform a semantic data integration task directly. The ontology expertise distribution for the participants was the following: one participant from Alcatel-Lucent was an ontology expert, 3 participants had knowledge engineering experience but were not familiar with ontology mappings, and 11 participants had no background in ontology knowledge.

5.3.2 Materials

The experiment used mapping design worksheets in order to capture a participant's mapping intentions on paper. APPENDIX D shows blank mapping design worksheets. The mapping design worksheet showed source and target ontologies using UML notations and the participants were asked to describe the mappings using natural language and drew lines if necessary. The selected domain for the experiment

was **conference** because this domain is generally understandable by most people with a system engineering background. The experiment ontologies used were carefully created by the author of this thesis in order to have many potential variable mappings, and all of the ontology class names about the conference, such as AcademicEvent, NonAcademicEvent, Tutorial, Workshop, MealEvent, SocialEvent, Accepted_Paper, Rejected_Paper, Undecided_Paper, Abstract_of_Paper, Short_Paper, Poster, CommitteeMember, Primary_Reviewer, Secondary_Reviewer, Author, Participant, Early_Paid_Applicant, Late_Paid_Applicant, Registered_Applicant, Important_Dates, Date_of_Conference, Abstract_Submission_Date, and Full_Paper_Submission_Date, in the experiment ontologies are from the OAEI2013 conference ontologies dataset [OAEI 2013] in order to be more realistic because the OAEI dataset is widely used in conference domain for experimental purposes. APPENDIX F shows an OWL view of the experiment ontologies using the Protégé tool [Protégé Project], and APPENDIX G shows potential mapping candidates using these experiment ontologies that were derived by the author of this thesis and can be considered a gold standard for comparison purposes. There were more than 40 different mappings available, and each mapping type had more than 10 potential mappings to be tested.

Two videos were used to explain the research and experiment. To minimize the risk of giving different amounts of information about the experiment to each participant, the experiment used videos to inform the participants. These two videos are accessible on the accompanying DVD for this thesis, see APPENDIX L. The first video explained what the MOUSE approach is, and the second video showed an example of how to use the SDI tool developed by this study.

The SDI tool was used by participants to draw mappings on the computer using UML notations from the mapping design worksheets. A brief mapping guide for the core mapping types was provided to the participants as a reference while using the tool during the mapping process. APPENDIX H shows this brief mapping guide. The tool automatically generated a concrete mapping syntax (SPARQL queries) from the abstract mapping syntax using UML. The last experiment output by the tool was an

integrated ontology, which was generated by executing auto-generated concrete mapping syntaxes on the source and target ontologies.

A questionnaire was provided to the participants to evaluate user satisfaction. APPENDIX I shows some blank questionnaire sheets. This questionnaire was based on the System Usability Scale (SUS) [Brooke 1996] questions to assess the ease of use for the abstract mapping syntax and the SDI tool. In addition, this questionnaire also had a simple ease of use indication (scale from 1 to 5) question such as “I think UML representation of semantic data and mapping is easy to understand for system engineers.” to get an overall instant user satisfaction level for the abstract mapping syntax.

5.3.3 Procedure

Ethical approval was granted for the experiment in late August 2013 by the Research Committee of the School of Computer Science and Statistics in Trinity College Dublin. Participants were invited by making a general call from the School of Computer Science and Statistics and Alcatel-Lucent Bell Labs Ireland. Each experiment was conducted with one participant at a time. The experiment was carried out over a period of three months which started in late August 2013 and finished in early October 2013.

Each participant was asked to complete five tasks in the course of the experiment, and the process of each experiment followed the task sheet as shown in APPENDIX J. First, a participant watched a video giving an overview of the MOUSE approach and the SDI tool. The video was about 2 minutes long. Second, the participant described mappings (a total of six different mappings) on mapping design worksheets. This task took about 25 minutes. Third, the participant watched another video showing an example of how to use the tool. The video lasted about 7 minutes. Fourth, the participant conducted a semantic data integration task using the SDI tool. The participant used the tool to draw mappings using UML notations from the mapping design worksheets. This task took about 10 minutes. Subsequently, the tool automatically generated a concrete mapping syntax (SPARQL queries) and an integrated ontology that was generated by executing auto-generated concrete

mapping syntaxes on the source and target ontologies. This auto-generation time lasted about two seconds. Finally, the participant answered a questionnaire. Answering the questionnaire took about 5 minutes. The total experiment time took about an hour per participant.

After the experiment finished, the collected results were analysed and evaluated not just by the author of this thesis but also the assistant supervisor of this research in order to be objective and transparent. During the analysis phase, if there was an unclear description of a mapping intention on the mapping design worksheets and/or UML notations in the SDI tool, feedback was obtained through a follow-up interview with the participant in question.

5.3.4 Metrics

1) Accuracy

There were two types of accuracy to measure. The first one was whether the tool could accurately auto-generate the mapping syntax from abstract to concrete. The ontology mapping representation using the MOUSE approach needed to be accurately and automatically transformed to a concrete mapping syntax in order to correctly perform the semantic data integration task. This accuracy was measured rigorously by manual syntax check by the author of this thesis and his assistant supervisor and using the JENA library syntax checker [JENA 2013] for the generated SPARQL queries corresponding to the abstract mapping syntax.

The second type of accuracy was whether the mapping intention of the participant was correctly captured and performed in the semantic data integration task. This accuracy was measured by the author of this thesis and his assistant supervisor by comparing the mapping design worksheets against the integrated ontology produced as a result of the auto-generated SPARQL queries from the tool. If the integrated ontology correctly corresponded to the mapping intention, then it was considered that the semantic data integration task accurately captured the participant's mapping intention.

2) Ease of use

There were two aspects to measure for ease of use. The first aspect was whether the ontology mapping using the MOUSE approach was easy to use for the participants. The second aspect was whether the tool was easy to use to facilitate the semantic data integration task. The semantic data integration task using the developed tool that implements the MOUSE approach needs to be easily performed by system engineers without any considerable learning effort. Because there is yet to be a comparable tool that supports semantic data integration by system engineers and that does not require the knowledge of ontological concepts and is assisted by the auto-generation from abstract to a concrete mapping syntax, the ease of use for the abstract mapping syntax and the tool were measured using a questionnaire. This questionnaire is based on the System Usability Scale (SUS) questions. The SUS has become an industry standard with references in over 1,300 articles and publications [Usability 2014], and the SUS questions in the questionnaire gave us the assessment of usability by a score that scales from 0 to 100. In addition, this questionnaire also had a simple ease of use indication (scale from 1 to 5) question such as “I think UML representation of semantic data and mapping is easy to understand for system engineers.” to get an overall instant user satisfaction level for the abstract mapping syntax using the MOUSE approach.

5.4 Experiment Analysis

This section outlines the results of the experiment. All the paper outputs from the experiment such as signed informed consent forms of the ethical application, mapping design worksheets, and questionnaires are kept in a paper document folder (total 255 pages, 17 pages per participant). APPENDIX K shows a sample of a scanned full paper record documentation from participant ID02 and ID13. In the appendix, participant’s name and signature are removed for anonymity. The source code for the SDI tool and experiment input/output files such as serialized UML model, RIF document, SPARQL queries, integrated OWL, and log of the tool generated by each participant are available on the accompanying DVD for this thesis,

see APPENDIX L. As the experiment metrics stated in the previous subsection, the analysis is split into the two areas: (1) Accuracy and (2) Ease of use.

5.4.1 Accuracy

There were two types of accuracy to measure. The first was the accuracy of the auto-generation of the mapping syntax from abstract to concrete. This accuracy was measured by both a manual and mechanical syntax check according to the UML-based abstract mapping syntax. Fig. 5-1, Fig. 5-2, and Fig. 5-3 show an example of the accuracy analysis of the mapping syntaxes, and these figures are extracted from actual results of the experiment conducted by participant ID02.

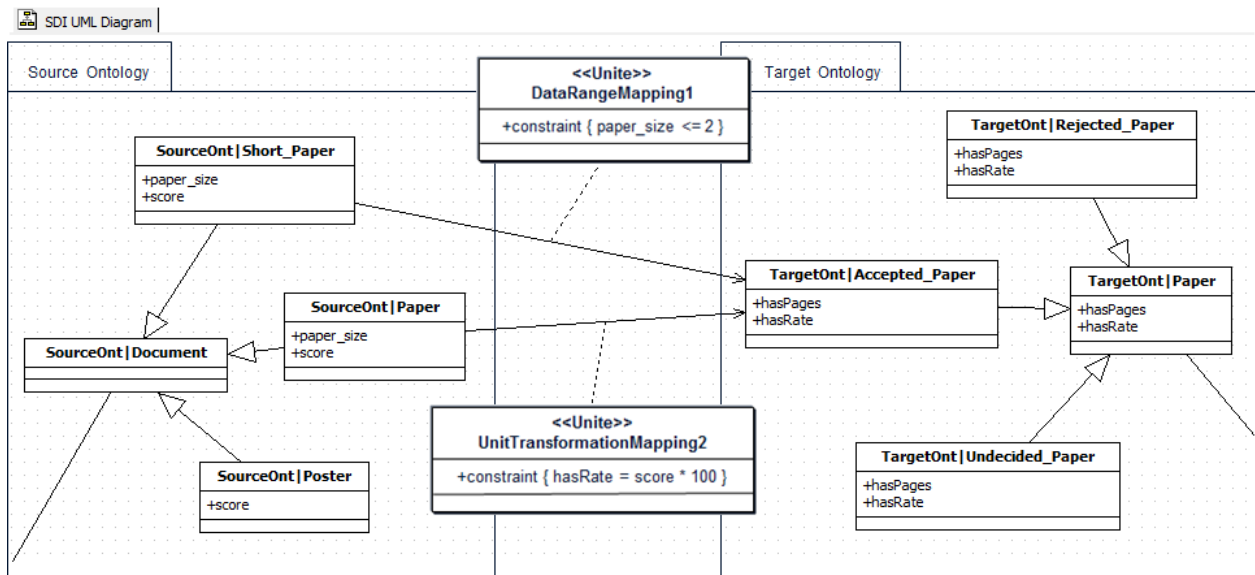


Figure 5-1 A sample abstract mapping syntax for ontology mappings from the experiment

In Fig. 5-1, ontology mappings (data range mapping and unit transformation mapping) are described using the SDI tool by a participant using the proposed UML-based abstract mapping syntax. SPARQL queries were auto-generated according to these mappings by the tool for the semantic data integration. Fig. 5-2 shows the SPARQL query syntax corresponding to the data range mapping shown in Fig. 5-1. Fig. 5-3 shows the SPARQL query syntax corresponding to the unit transformation mapping shown in Fig. 5-1. If the auto-generated SPARQL queries had correct syntax corresponding to the mappings described in the abstract mapping syntax, then

this study considered auto-generation of the mapping syntax from abstract to concrete was accurate.

```
1 PREFIX TargetOnt: <http://experiment/ontology1/conference/targetOnt#>
2 PREFIX SourceOnt: <http://experiment/ontology2/conference/sourceOnt#>
3
4 CONSTRUCT {
5   ?instance3 a TargetOnt:Accepted_Paper .
6 }
7
8 WHERE {
9   ?instance3 a SourceOnt:Short_Paper .
10  ?instance3 SourceOnt:paper_size ?value1 .
11   FILTER ( ?value1 <= 2 )
12 }
13
```

Figure 5-2 Auto-generated SPARQL query syntax for the data range mapping

```
1 PREFIX TargetOnt: <http://experiment/ontology1/conference/targetOnt#>
2 PREFIX SourceOnt: <http://experiment/ontology2/conference/sourceOnt#>
3
4 CONSTRUCT {
5   ?instance6 a TargetOnt:Accepted_Paper .
6   ?instance6 TargetOnt:hasRate ?value2 .
7 }
8
9 WHERE {
10  ?instance6 a SourceOnt:Paper .
11  ?instance6 SourceOnt:score ?value1 .
12  BIND ( ?value1 * 100 AS ?value2 )
13 }
14
```

Figure 5-3 Auto-generated SPARQL query syntax for the unit transformation mapping

There were a total of 90 ontology mappings conducted by the participants during the experiments: 30 mappings in the direct mapping type, 30 mappings in the data range mapping type and 30 mappings in the unit transformation mapping type. Table 5-1

shows the results of the accuracy for the auto-generation of the mapping syntax from abstract to concrete.

Table 5-1 Number of correctly auto-generated mappings corresponding to abstract mapping syntax

Participant ID	Ontology mappings described using the abstract mapping syntax			# of correctly auto-generated mappings
	# of Direct Mappings	# of Data Range Mappings	# of Unit Transformation Mappings	
P01	2	2	2	6
P02	2	2	2	6
P03	2	2	2	6
P04	2	2	2	6
P05	2	2	2	6
P06	2	2	2	6
P07	2	2	2	6
P08	2	2	2	6
P09	2	2	2	6
P10	2	2	2	6
P11	2	2	2	6
P12	2	2	2	6
P13	2	2	2	6
P14	2	2	2	6
P15	2	2	2	6
Total #	30	30	30	90

In the table 5-1, it can be seen that the total number of ontology mappings described using the proposed abstract mapping syntax was 90, and the total number of correctly auto-generated mappings was 90 as well. The 90 mappings generated by the participants were not all distinct from each other. However, at least 46 different mappings in the gold standard for experiment ontologies, see APPENDIX G, were accurately auto-generated, and the tool accurately auto-generated most of the cases of semantic mappings conducted by the participants. This means the tool can correctly auto-generate the ontology mappings from abstract to concrete at least for the typical mapping cases in the experiment ontologies. The accuracy rate of the auto-generation was satisfactory.

The second accuracy measured by this experiment was the accuracy of capturing the mapping intention of the participant in the semantic data integration task. This accuracy was measured by comparing the mapping design worksheets and the integrated ontology from the computation result of the auto-generated SPARQL queries by the tool. If the integrated ontology correctly corresponded to the mapping intentions, then it was considered that the participant's mapping intention had been correctly captured.

For example, from the actual result of the experiment, participant ID02 described his mapping intention "Short papers are published if they have less than 3 pages" between classes (Short Paper in source ontology and Accepted Paper in target ontology) on the mapping design worksheet. The participant meant the instances of the Short Paper class need to be integrated into the Accepted Paper class only if the number of pages of a short paper is less than 3 pages. This study examined the instances in the Accepted Paper class in the integrated ontology performed by this participant, and the Accepted Paper class included the instances from the Short Paper class that satisfied the constraint of the number of pages less than 3. In this case, the semantic data integration task using the MOUSE approach correctly performed according to the participant's mapping intention. Table 5-2 shows the results of the accuracy of capturing participants' mapping intentions.

Table 5-2 Number of correctly captured mappings corresponding to participant's mapping intention

Participant ID	Mappings described in mapping design worksheets			# of correctly captured mappings in the integrated ontology		
	# of Direct Mappings	# of Data Range Mappings	# of Unit Transformation Mappings	# of Direct Mappings	# of Data Range Mappings	# of Unit Transformation Mappings
P01	2	2	2	2	2	2
P02	2	2	2	2	2	2
P03	2	2	2	2	2	2
P04	2	2	2	2	2	2
P05	2	2	2	2	2	2
P06	2	2	2	2	2	2
P07	2	2	2	0	2	2
P08	2	2	2	2	2	2
P09	2	2	2	2	2	2
P10	2	2	2	2	2	1
P11	2	2	2	2	2	2
P12	2	2	2	2	2	2
P13	2	2	2	1	2	2
P14	2	2	2	2	2	2
P15	2	2	2	2	2	2
Total #	30	30	30	27	30	29

In the table 5-2, it can be seen that the total number of mappings described in mapping design worksheets was 90, and the total number of correctly performed mappings during the semantic data integration task according to the participant's mapping intention was 86. There were 4 mappings that could not correctly capture the participant's mapping intention.

For example, participant ID13 described his mapping intention "Instances in ConferenceEvent in source ontology class need to be integrated into Event in target ontology class. Because they have the same level of abstraction and their subclasses are comparable." on the mapping design worksheet. This study examined the instances in Event class in the integrated ontology performed by this participant, and the Event class did not include the instances from the ConferenceEvent class. So

there was further investigation in this case. Fig. 5-4 shows the mappings drawn by this participant on the SDI tool.

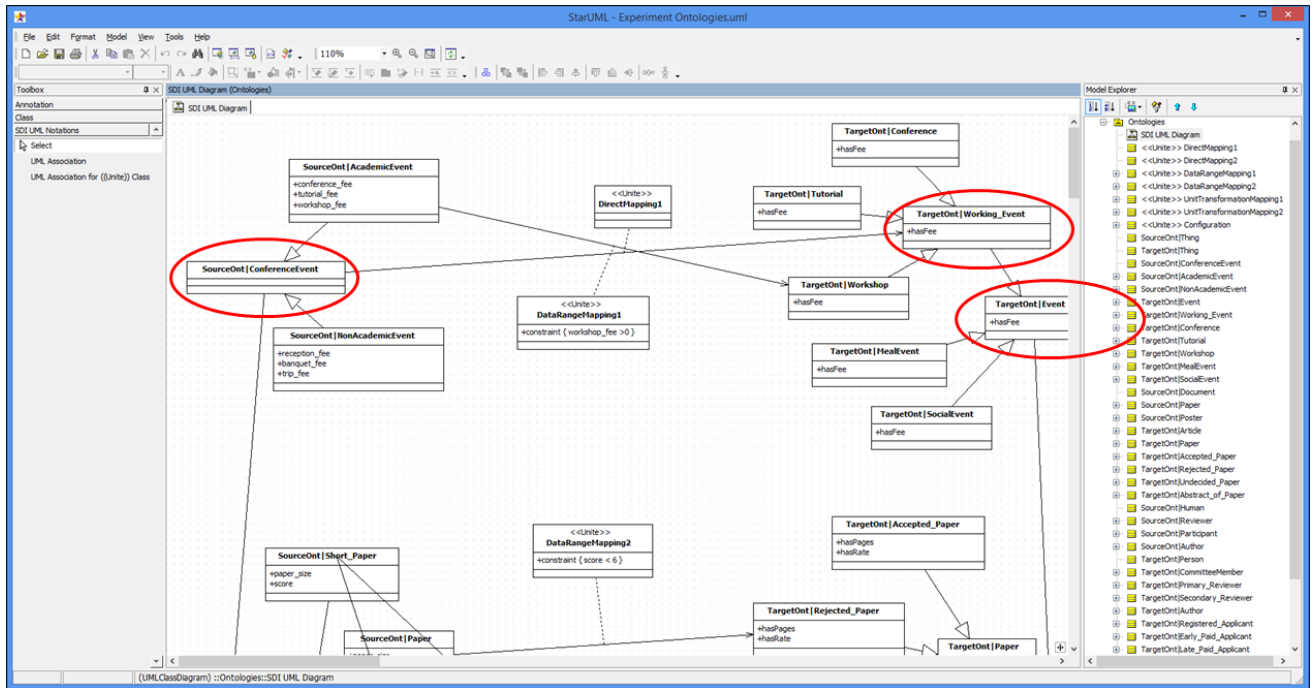


Figure 5-4 A sample mappings on the SDI tool from the experiment

In Fig. 5-4, it can be observed that the participant made the mapping between the ConferenceEvent class and the Working_Event class, not with the Event class. This study interviewed this participant for feedback, and he mentioned that he made a mistake. Because the target UML class (Event) was partially out of screen, he drew this mapping unintentionally. There were further interviews with the participants (participant ID07, ID10, and ID13), whose mapping intentions were not captured correctly in the semantic integration task; asking why their semantic data integration task could not correctly capture their mapping intentions. From the 4 inaccurate ontology mappings, three of these mappings were the result of mis-drawn UML notations by the participant's unintentional mistake and the other one was the result of an intended mis-drawing of UML notation; meaning the participant changed his mind to make a different mapping during the semantic data integration task on the tool but did not apply his change to the mapping design worksheet. From the feedback of the participants, it has been identified that there can be graphical intricacy of UML notations.

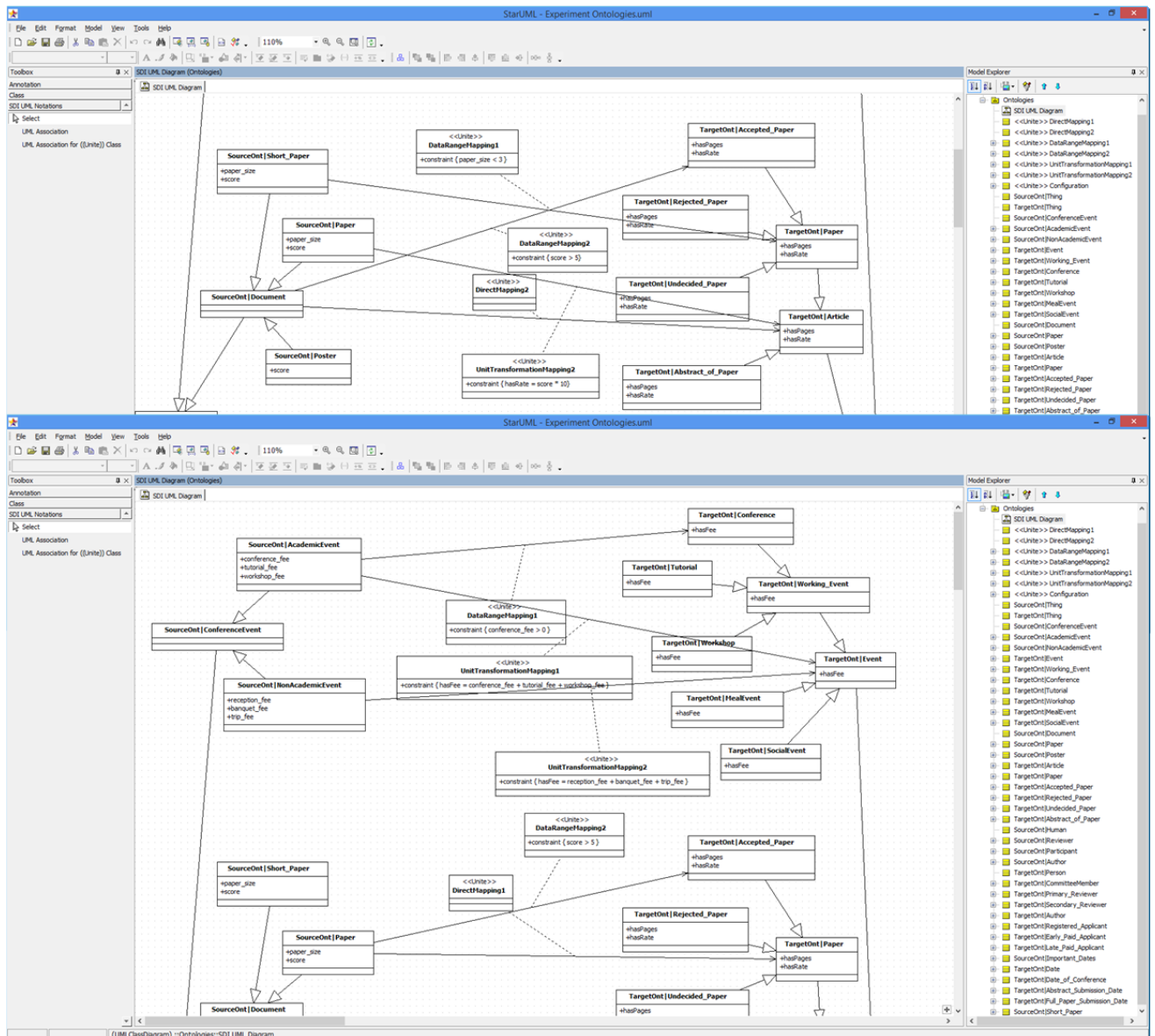


Figure 5-5 A sample mappings on the SDI tool from the experiment

Fig. 5-5 shows the mappings drawn on the tool by the participant ID06 and ID12. In this figure, it can be observed that there is a potential graphical intricacy on the drawing of UML notations due to crossing and overlapping lines and arrows between classes and associations for representing multiple mappings in complex ontology relationships.

From the experiments, the number of correctly captured mapping intentions during the semantic data integration tasks was 86 out of 90, and this result suggests that the

semantic data integration task using the MOUSE approach can correctly capture each participant's mapping intention; at least for the core mapping types.

5.4.2 Ease of use

There were two things to measure for ease of use. The first was to assess the ease of use of the abstract mapping syntax using the MOUSE approach to represent ontology mappings. The second was to assess the ease of use of the SDI tool that facilitates the semantic data integration task by supporting the auto-generation of a concrete mapping syntax from the proposed abstract mapping syntax. The ease of use for the abstract mapping syntax and the tool were measured using two SUS based questionnaires. One questionnaire was used for the proposed abstract mapping syntax and the other questionnaire was used for the tool. These SUS questions gave the assessment of usability by a score that scales from 0 to 100. Table 5-3 and Fig. 5-6 show the analysis results of the ease of use from the SUS scores for the proposed abstract mapping syntax and the implemented tool using the MOUSE approach.

Table 5-3 SUS scores for the proposed abstract mapping syntax and the implemented tool

Participant ID	System Usability Scale Score	
	Abstract Mapping Syntax	SDI Tool
P01	87.5	87.5
P02	70	77.5
P03	85	92.5
P04	92.5	95
P05	97.5	97.5
P06	85	85
P07	77.5	85
P08	70	70
P09	95	100
P10	92.5	100
P11	72.5	85
P12	80	77.5
P13	62.5	52.5
P14	92.5	90
P15	60	95
Average	81.3	86

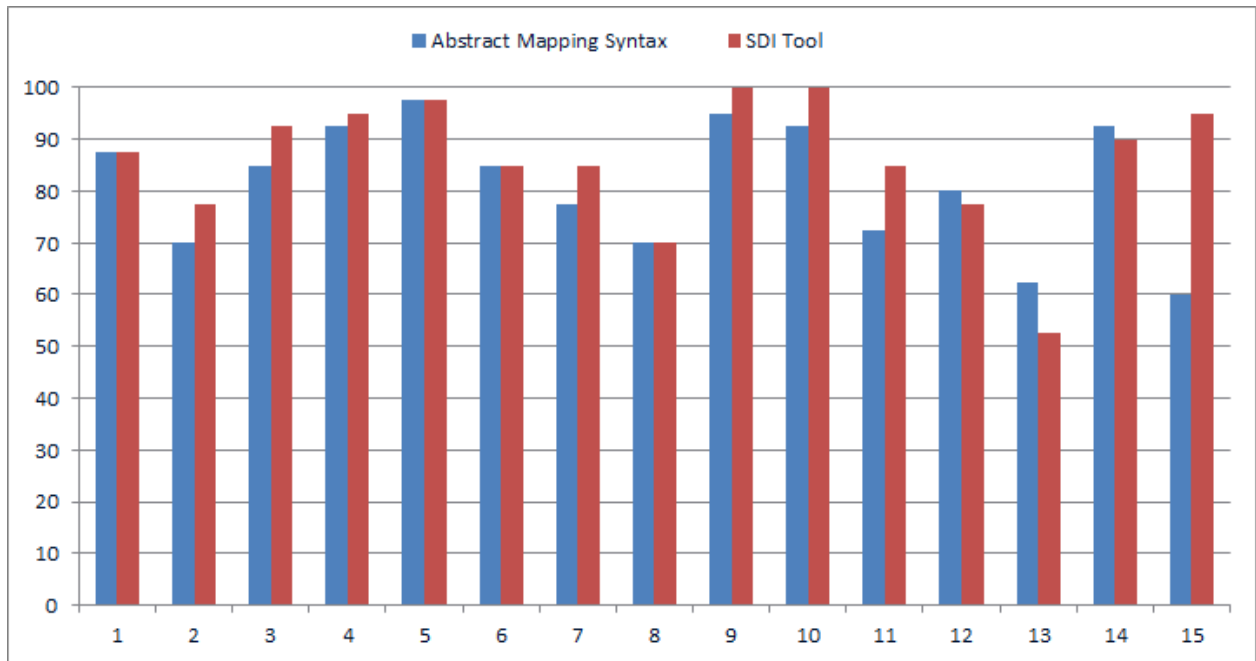


Figure 5-6 SUS scores bar graph for the abstract mapping syntax and the tool

In the table 5-3 and Fig. 5-6, it can be seen that the average SUS score for the abstract mapping syntax was 81.3 and the tool was 86. In the case of SUS scores, if it is higher than 80.3, which is the top 10% of scores by the percentile ranks associated with SUS scores [Sauro 2011], then it indicates “A” grade usability meaning the users are more likely to recommend the representation [Sauro 2010]. Both SUS scores for the abstract mapping syntax and the tool using the MOUSE approach were assessed “A” grades by the SUS score percentile rank. These results suggest that they were easy to use for the participants (system engineers). This research also recalculated the ease of use result values without participant ID06, ID08, ID09, and ID10 because these participants are ontology experts or have knowledge engineering experience. The average SUS score - by the 11 participants who had no background in ontology knowledge - for the abstract mapping syntax was 79.7 and the tool was 85. There was little decrease on the average SUS score values. However, both SUS scores still suggest that the proposed abstract mapping syntax and the developed tool were easy to use for the participants.

There was also a simple question asking the participant’s opinion about whether the ontology mapping using UML representation was easy to use from a system

engineer’s perspective. Table 5-4 and Fig. 5-7 show the analysis results of this simple ease of use indication question.

Table 5-4 A simple ease of use indication question scores for the abstract mapping syntax

Participant ID	Simple ease of use indication for the abstract mapping syntax	
	Range 1 (Very difficult) to 5 (very easy)	Percentage (%)
P01	5	100
P02	5	100
P03	5	100
P04	5	100
P05	5	100
P06	5	100
P07	4	80
P08	5	100
P09	5	100
P10	5	100
P11	5	100
P12	4	80
P13	4	80
P14	5	100
P15	3	60
Average	4.66	93

1 = Strongly Disagree, 5 = Strongly Agree

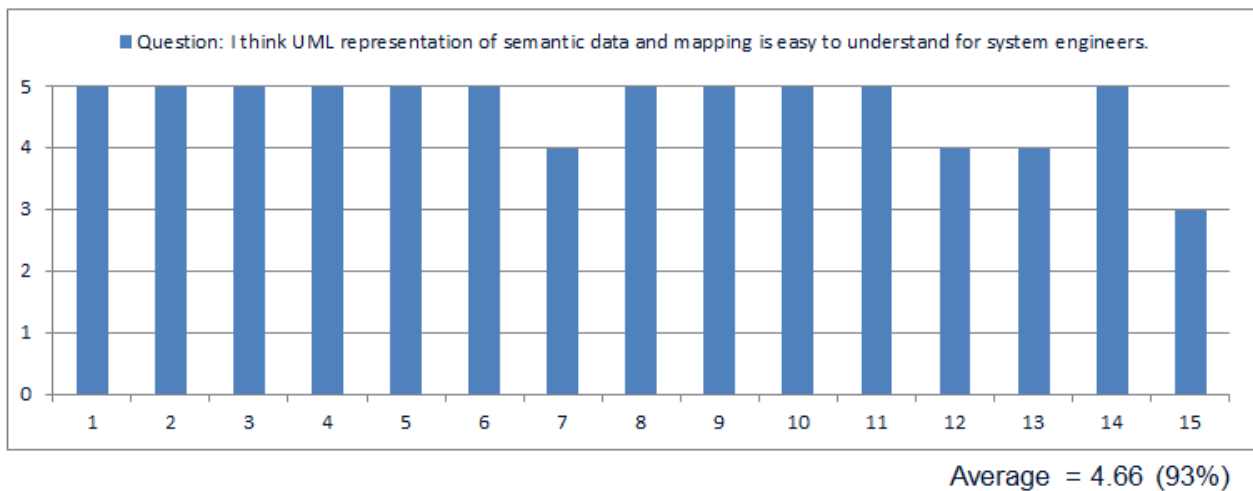


Figure 5-7 A simple ease of use indication question scores bar graph for the abstract mapping syntax

From this question, this study could get an overall intuitive user satisfaction level for the proposed abstract mapping syntax. In the table 5-4 and Fig. 5-7, it can be seen that the average score for this indication question was 4.66 (93%). This average score was also recalculated without participant ID06, ID08, ID09, and ID10 because these participants are ontology experts or have knowledge engineering experience. The recalculated average score for this indication question was 4.55 (91%). Both calculated average scores were very high. This means the participants strongly agreed that the abstract mapping syntax was easy to use for the system engineers, and overall, the participants were highly satisfied with the abstract mapping syntax using the MOUSE approach.

Some participants did criticise the tool. For example, participants ID13 who gave the lowest SUS score from the experiment commented: *“The interface for the tool presents several UI challenges. Firstly there are lots of opportunities to make mistakes, more constraint in the actions the user can perform is necessary. Allowing users to enter free text is liable to error that could be avoided by using dropdowns. Secondly there are too many options in the right-hand column when I’m only using 6 of the many available. Thirdly there is an issue with association arrows overlap and it becomes hard to distinguish associations.”*

In addition, from the feedback, this study could also observe that the participants did not like to go through several steps in the tool for the semantic data integration task. For example, the steps in the sequence the participants needed to follow in the tool to perform the semantic data integration included “Exporting XMI”, “Initializing SDI Tool”, “Generating RIF document and SPARQL queries”, “Executing SPARQL queries on the ontologies” and “Archiving generated files”. Instead of these steps, most of users wanted an automated complete work with one-click after drawing their mappings. They wanted the interaction with the tool to be as simple (few) as possible for their task.

This chapter explained and analysed the user based experiments undertaken by system engineers (in a lab environment by participants who have skills equivalent to real life system engineers) to evaluate the proposed MOUSE approach that incorporates the abstract mapping syntax and automatic translation into a concrete mapping syntax. The following chapter describes how well the research question was addressed by evaluating the experiment’s results and suggests future work.

6 Conclusion

This chapter gives an answer to the research question:

“To what extent will the proposed MOUSE approach: (i) allow the creation of mappings using an abstract syntax familiar and usable to system engineers; (ii) allow the accurate transformation of the abstract mappings into concrete executable mappings?”

by evaluating the experiment results according to the experiment hypotheses and by suggesting future work.

1. The SDI tool can correctly capture the mapping intention of a participant for the core mapping types.

True: For each core mapping type (direct mapping, data range mapping and unit transformation mapping), 30 mappings were analysed. For the direct mapping type, 27 mappings correctly captured the participant’s mapping intention. For the data range mapping, 30 mappings correctly captured the participant’s mapping intention. For the unit transformation mapping, 29 mappings correctly captured the participant’s mapping intention. There were 4 mappings out of total 90 mappings that did not correctly capture the participant’s mapping intention. From the feedback, these inaccurate ontology mappings were identified as the result of mis-drawing UML notations. From the analysis of the results, this hypothesis can be considered true.

2. The SDI tool can accurately transform the abstract mappings of each participant into SPARQL executable mappings.

True: The total number of ontology mappings conducted by the participants was 90, and the total number of correctly auto-generated mappings was 90 as well. The accuracy rate of the auto-generation was satisfactory. From the analysis of the results, this hypothesis is proven true at least for the core mapping types.

3. The proposed UML-based abstract mapping syntax is easy to use for the participants.

True: To evaluate the ease of use of the proposed abstract mapping syntax, two assessment methods were used. The first was a SUS based questionnaire and the second was a simple overview indication question asking the participant's opinion about whether the ontology mapping using UML-based mapping syntax was easy to use in a system engineer's perspective. Both assessments for the ease of use of the UML-based abstract mapping syntax scored very high. From the analysis of the results, this hypothesis can be considered true.

4. The SDI tool is easy to use for participants.

True: To evaluate the ease of use for the tool, a SUS based questionnaire was used. The SUS assessment for the ease of use of the tool to perform a semantic data integration task scored very high. From the analysis of the results, this hypothesis can be considered true.

This research has identified the core mapping types to be correctly captured by the abstract mapping syntax and reviewed existing semantic mapping syntaxes that tend to rely on a mapping practitioner to understand the idea of ontological concepts. In our research, it is proposed to have a UML-based abstract semantic mapping representation that tries to abstract away the concepts of ontology and is more intuitive to use in order to represent the mapping intentions of domain experts who have system engineering backgrounds. The abstract mapping syntax and tool has been developed into the MOUSE approach, which is usable for system engineers to accomplish their semantic data integration tasks. From the experiment, it is suggested that system engineers can perform the semantic data integration task including ontology mappings without knowing the ontological knowledge by using the proposed UML-based abstract mapping syntax and a tool that they are familiar with, while maintaining accuracy and usability (in terms of ease of use).

6.1 Future Works

The current MOUSE approach is assessed by two criteria. One criterion is the evaluation of the MOUSE approach within the core mapping types, and the other criterion is the evaluation of the MOUSE approach supporting only query-based semantic integration. Since there can be too many possible mapping relationships for different semantic integration situations, there was a need to first scope the mapping relationships and second, create tangible practical semantic integration solutions. Therefore, this research identified and prioritized the mapping relationships that mostly occurred in the industry use case. Three core mapping types were defined to realize the proposed MOUSE approach. This research tried to generalize the use of these core mapping types - in other words, to prove that the core mapping types are usable not only in the telecommunications domain. The core mapping types were applied in another domain (conference) and were proven usable through the examples and experiments conducted by this research.

The MOUSE approach supports only query-based semantic integration. There are three different semantic data integration approaches: (1) Query-based integration using SPARQL, (2) Rule-based integration using SWRL, and (3) Axiom-based integration using ontology. This research selected query-based integration approach to begin the tool development that implements the MOUSE approach because from the related research, query-based integration using SPARQL has proven to be the most practical approach. This does not mean that this research abandoned the other integration approaches. The SDI tool generates an intermediate RIF document because RIF has the potential to interoperate with another integration approach. For example, there is an existing RIF specification for RDF and OWL Compatibility. This strategy potentially enables the transformation between RIF and OWL for axiom-based integration in future.

There are some suggested future works to improve the current MOUSE approach.

- 1) Further experiments on the MOUSE approach in another domain*

In this research, the telecommunications domain and conference domain were tested to prove the approach can be generalized. Experiments on two different ontology domain datasets to generalize the theory can be considered a reasonable number in light of ontology mapping research [Brockmans 2006][Keeney 2011][Lukichev 2008][Scharffe 2008] stating that the number of ontology domains typically used to argue a theory was between 1 and 3. However, it is always better to test a theory with more varied domain datasets in order to support the argument for the generalization of the approach.

2) Extending the core mapping types to cover more mapping relationships

In this research, three core mapping types: direct mapping, data range mapping, and unit transformation mapping, were derived from the industry use case. It has been proven that the core mapping types are usable outside the telecommunications domain. However, to cover more integration situations, there is a need to extend the current core mapping types.

3) Generating another concrete mapping syntaxes

In this research, only the query-based integration approach using SPARQL queries was implemented. The generation of other concrete mapping syntaxes from the abstract mapping syntax will enable different semantic data integration approaches such as the axiom-based integration approach using OWL or rule-based integration approach using SWRL.

4) Resolving the graphical intricacy of the drawing UML notations

Because of this problem, some mapping intentions of the experiment participants - to evaluate the accuracy of the SDI tool - could not be captured correctly. Graphical intricacy occurs when there are several crossings and overlappings between UML classes and associations for representing multiple mappings in complex ontology relationships. This graphical intricacy existed in UML long before this research. Normally, the graphical intricacy was treated in the tool level such as the auto-alignment function for UML notations. The SDI tool can be improved and made to be more user-friendly with such a function.

REFERENCES

- [Amrouch 2012] S. Amrouch and S. Mostefai, "Survey on the literature of ontology mapping, alignment and merging", In IEEE International conference on Information Technology and e-Services (ICITeS), pp. 1-5, 2012.
- [Aumuller 2005] D. Aumuller, H. Do, S. Massmann and E. Rahm, "Schema and Ontology matching with COMA++", In Proceedings of the ACM SIGMOD conference on management of data (SIGMOD), pp. 906-908, 2005.
- [Baclawski 2001] K. Baclawski, M. K. Kokar, P. A. Kogut, L. Hart, J. Smith, W. S. Holmes III, J. Letkowski and M. L. Aronson, "Extending UML to support ontology engineering for the Semantic Web", In Proceedings of the 4th International Conference on the Unified Modeling Language, Lecture Notes in Computer Science, Springer, 2001.
- [Banahan 1991] M. Banahan, D. Brady and M. Doran "The C Book: Featuring the ANSI C Standard (Instruction Set)", Addison-Wesley Pub (Sd), ISBN13- 9780201544336, Aug. 1991.
- [Bechhofer 2004] Sean Bechhofer et al., "OWL Web Ontology Language Reference", In M. Dean and G. Schreiber (Eds.), W3C Recommendation, <http://www.w3.org/TR/owl-ref/>, 10 Feb. 2004.
- [Bernstein 2008] P. A. Bernstein and L. M. Haas, "Information Integration in the Enterprise", Communications of the ACM, 51(9), Sep. 2008.
- [Blomqvist 2005] E. Blomqvist and K. Sandkuhl, "Patterns in ontology engineering: Classification of ontology patterns", In

- Proceedings of International Conference on Enterprise Information Systems (ICEIS), pp. 413-416, 2005.
- [Boley 2011] Harold Boley, “The Rule Markup Initiative: Schema Specification of RuleML 1.0”, <http://ruleml.org/1.0/>, Retrieved 09 Feb. 2011.
- [Boran 2010] Aidan Boran, “A Dependency Modelling Approach for the Management of Ontology-Based Integration Systems”, PhD thesis, Trinity College Dublin, Ireland, 2010.
- [Boran 2011a] A. Boran, C. Matheus, I. Bedini and P. F. Patel-Schneider, “Empirical Analysis of Semantic Techniques applied to a Classification Problem involving Network Performance Data”, The 10th International Semantic Web Conference (ISWC), Bonn, Germany, 23-27 Oct. 2011.
- [Boran 2011b] A. Boran, I. Bedini, C. Matheus, P. F. Patel-Schneider and J. Keeney, “A smart campus prototype for demonstrating the semantic integration of heterogeneous data”, In Web Reasoning and Rule Systems, pp. 238-243, Springer Berlin Heidelberg, 2011.
- [Bouquet 2003] P. Bouquet, F. Giunchiglia, F. V. Harmelen, L. Serafini and H. Stuckenschmidt, “C-OWL: Contextualizing Ontologies”, The Semantic Web - ISWC 2003, volume 2870 of Lecture Notes in Computer Science (LNCS), pp. 164-179, FL, USA, Oct. 2003.
- [Bouquet 2005] P. Bouquet, M. Ehrig and J. Euzenat, “D2.2.1 Specification of a common framework for characterizing alignment”, <http://www.inrialpes.fr/exmo/cooperation/kweb/heterogeneity/deli/kweb-221v1.pdf>, 2005.
- [Brockmans 2004] S. Brockmans, R. Volz, A. Eberhart and P. Löffler, “Visual modeling of OWL DL ontologies using UML”, In Proceedings

of the 3rd International Semantic Web Conference, Hiroshima, Japan, 2004.

- [Brockmans 2006] S. Brockmans, P. Haase, P. Hitzler and R. Studer, “A Metamodel and UML Profile for Rule-Extended OWL DL Ontologies”, In Proceedings of 3rd European Semantic Web Conference (ESWC), volume 4011 of LNCS, pp. 303-316, Springer, 2006.
- [Brooke 1996] John Brooke, “SUS: A quick and dirty usability scale”, In P. W. Jordon, B. A. Weerdmeester and I. L. McClelland (Eds.), Usability evaluation in industry, pp. 189-194, London, England, 1996.
- [Bruijn 2004] J. D. Bruijn, F. Martin-Recuerda, D. Manov and M. Ehrig, “D4.2.1 state-of-the-art-survey on ontology merging and aligning v1”, Semantically Enabled Knowledge Technologies (SEKT) Project Deliverable D4.2.1, 2004.
- [Cruz 2012] I. F. Cruz, C. Stroe and M. Palmonari, “Interactive User Feedback in Ontology Matching Using Signature Vectors”, In: Proc. of the 28th Int. Conference on Data Engineering, pp. 1321-1324, 2012.
- [DOM Core] World Wide Web Consortium (W3C), “Document Object Model Core”, W3C Recommendation, <http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/core.html>, 13 Nov. 2000.
- [Duo 2006] D. Dou and P. LePendu, “Ontology-based integration for relational databases”, Proceedings of the 2006 ACM symposium on Applied computing, Dijon, France, 23-27 Apr. 2006.
- [Euzenat 2004a] J. Euzenat, D. Loup, M. Touzani and P. Valtchev, “Ontology alignment with OLA”, In Proceedings of the 3rd EON

- Workshop, 3rd International Semantic Web Conference (ESWC), 2004.
- [Euzenat 2004b] Jerome Euzenat et al., “D2.2.3: State of the art on ontology alignment”, <ftp://ftp.inrialpes.fr/pub/exmo/reports/kweb-223.pdf>, 2004.
- [Euzenat 2006a] J. Euzenat, F. Scharffe and L. Serafin, “D2.2.6: Specification of the delivery alignment format”, <ftp://ftp.inrialpes.fr/pub/exmo/reports/kweb-226.pdf>, 2006.
- [Euzenat 2006b] Jerome Euzenat, “An API for ontology alignment (version 2.1)”, INRIA Rhône-Alpes, Montbonnot, France, 2 Feb. 2006.
- [Euzenat 2007] J. Euzenat and P. Shvaiko, “Ontology Matching”, Springer-Verlag Berlin Heidelberg, ISBN13-9783540496113, 2007.
- [Euzenat 2008] J. Euzenat, A. Polleres and F. Scharffe, “Processing Ontology Alignments with SPARQL”, International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2008), Barcelona, Spain, 4-7 Mar. 2008.
- [Euzenat 2010] Jerome Euzenat, “A format for ontology alignment”, <http://alignapi.gforge.inria.fr/format.html>, Retrieved 09 Mar. 2010.
- [Euzenat 2011] Jerome Euzenat, “EDOAL: Expressive and Declarative Ontology Alignment Language”, <http://alignapi.gforge.inria.fr/edoal.html>, Retrieved 13 May 2011.
- [Falconer 2007a] S. M. Falconer, N. F. Noy and M. A. D. Storey, “Ontology Mapping - a User Survey”, In Proceedings of the Workshop on Ontology Matching (OM 2007), ISWC/ASWC 2007, Busan, Korea, Nov. 2007.

- [Falconer 2007b] S. Falconer and M. A. Storey, “A cognitive support framework for ontology mapping”, In Processings of the 6th International Semantic Web Conference (ISWC), pp. 114-127, 2007.
- [Fensel 2004] Dieter Fensel, “Ontologies: a silver bullet for knowledge management and electronic commerce”, Springer, Heidelberg (DE), 2nd edition, 2004.
- [Fikes 2004] R. Fikes, P. Hayes, and I. Horrocks. “OWL-QL: a language for deductive query answering on the Semantic Web”, Journal of Web Semantics, 2(1):19-29, 2004.
- [FOAF 2010] The Friend of a Friend (FOAF) project, “FOAF Vocabulary Specification 0.98”, Namespace Document 9 August 2010 - Marco Polo Edition, <http://xmlns.com/foaf/spec/>, 2010.
- [Fondement 2005] F. Fondement and T. Baar, “Making Metamodels Aware of Concrete Syntax”, European Conference on Model Driven Architecture (ECMDA), LNCS 3748, pp. 190-204, 2005.
- [Fürst 2005] F. Fürst and F. Trichet, “Axiom-based ontology matching”, In Proceedings of the 3rd international conference on Knowledge capture, ACM, pp. 195-196, Oct. 2005.
- [Gasevic 2004] D. Gasevic, D. Djuric, V. Devedzic and V. Damjanovi, “Converting UML to OWL ontologies”, In Proceedings of the 13th international World Wide Web conference on Alternate track papers and posters, New York, NY, USA, 19-21 May 2004.
- [Granitzer 2010] M. Granitzer, V. Sabol, K. W. Onn, D. Lukose and K. Tochtermann, “Ontology Alignment - A Survey with Focus on Visually Supported Semi-Automatic Techniques.” Future Internet 2, no. 3: 238-258, 2010.

- [Grau 2013] B. C. Grau, Z. Dragisic, K. Eckert, J. Euzenat, A. Ferrara, R. Granada, V. Ivanova et al. “Results of the ontology alignment evaluation initiative 2013”, In Proceedings of the 8th ISWC workshop on ontology matching (OM), pp. 61-100, 2013.
- [Gruber 1993] Thomas R. Gruber, “A Translation Approach to Portable Ontology Specifications”, Knowledge Acquisition, 5, pages 199-220, 1993.
- [Horn 1951] Alfred Horn, “On sentences which are true of direct unions of algebras.” Journal of symbolic logic, pp. 14-21, 1951.
- [Horrocks 2004] Ian Horrocks et al., “SWRL: A Semantic Web Rule Language Combining OWL and RuleML”, W3C Member Submission, <http://www.w3.org/Submission/SWRL/>, 21 May 2004.
- [Horrocks 2010] I. Horrocks and P. F. Patel-schneider, “Knowledge Representation and Reasoning on the Semantic Web: OWL.” Handbook of Semantic Web Technologies, J. Domingue, D. Fensel and J. A. Hendler (Eds.), pp. 365-398, 2011.
- [Howe 2014] Denis Howe, “Dictionary: Syntactic Sugar”, In The Free On-line Dictionary of Computing, http://dictionary.reference.com/browse/syntactic_sugar, Retrieved 04 Jul. 2014.
- [JAR Spec] ORACLE - Java SE Documentation, “The Java™ Archive (JAR) File Specification”, <http://docs.oracle.com/javase/6/docs/technotes/guides/jar/jarGuide.html>, 18 Nov. 1996.
- [JENA 2011] JENA Apache incubator project, “SPARQL Syntax Expressions”, <http://openjena.org/wiki/SSE>, Aug. 2011.

- [JENA 2013] JENA Apache incubator project, “ARQ - A SPARQL Processor for Jena”,
<http://jena.apache.org/documentation/query/>, Library File:
 Apache Jena - ARQ (SPARQL 1.1 Query Engine) -
<http://central.maven.org/maven2/org/apache/jena/jena-arq/2.10.0/jena-arq-2.10.0.jar>, 20 Feb. 2013.
- [Kalfoglou 2003] Y. Kalfoglou and M. Schorlemmer, “Ontology Mapping: The State of The Art”, *The Knowledge Engineering Review Journal (KER)*, DOI: 10.1017/S0269888903000651, 18(1):1-31, Dec. 2003.
- [Kalibatiene 2010] D. Kalibatiene and O. Vasilecas, “On OWL/SWRL mapping to OCL/UML“, In *Proceedings of the 11th International Conference on Computer Systems and Technologies (CompSysTech)*, pp. 58-63, Sofia, Bulgaria, Jun. 2010.
- [Keeney 2011] J. Keeney, A. Boran, I. Bedini, C. J. Matheus and P. F. Patel-Schneider, “Approaches to Relating and Integrating Semantic Data from Heterogeneous Sources”, In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT 2011)*, Lyon, France, 22-27 Aug. 2011.
- [Knublauch 2011] H. Knublauch, J. A. Hendler, K. Idehen, “SPINMap: SPARQL-based Ontology Mapping with a Graphical Notation”, *Composing the Semantic Web*, Retrieved from [http:// http://composing-the-semantic-web.blogspot.co.uk/2011/04/spinmap-sparql-based-ontology-mapping.html](http://http://composing-the-semantic-web.blogspot.co.uk/2011/04/spinmap-sparql-based-ontology-mapping.html), 21 Apr. 2011.
- [Kogut 2002] P. Kogut, S. Cranefield, L. Hart, M. Dutra, K. Baclawski, M. Kokar and J. Smith, “UML for ontology development”, *The Knowledge Engineering Review*, 17 (1), pp. 61-64, 2002.

- [Lukichev 2008] Sergey Lukichev, “Defining a subset of OCL for expressing SWRL rules”, Proceedings of the 2nd East European Workshop on Rule-Based Applications (RuleApps) at the 18th European Conference on Artificial Intelligence, Patras, Greece, 23 Jul. 2008.
- [Miller 1992] Dale Miller, “Abstract Syntax and Logic Programming”, In Proceedings of the First and Second Russian Conferences on Logic Programming, pp. 322-337, Springer-Verlag LNAI 592, Irkutsk and St. Petersburg, Russia, 1992.
- [Modelio Tool] Modelio Soft, “Enterprise solutions based on the leading open source modeling tool”, <http://www.modeliosoft.com/en/technologies/uml.html>, Jan. 2012.
- [Noy 2003] N. F. Noy and M. A. Musen, “The PROMPT suite: Interactive tools for ontology merging and mapping”, International Journal of Human computing Studies, Elsevier, 6(59), 2003.
- [OAEI 2012] Ontology Alignment Evaluation Initiative (OAEI), “Ontology Alignment Evaluation Initiative Campaign”, <http://oaei.ontologymatching.org/>, 19 Jun. 2012.
- [OAEI 2013] Ontology Alignment Evaluation Initiative (OAEI), “Ontology Alignment Evaluation Initiative - OAEI2013 Campaign - Conference track”, <http://oaei.ontologymatching.org/2013/conference/index.html>, 15 Jun. 2013.
- [OCL 2.0] Object Management Group (OMG), “Object Constraint Language (OCL) - version 2.0”, <http://www.omg.org/spec/OCL/>, Released May 2006.

- [ODM 1.0] Object Management Group (OMG), “Documents associated with Ontology Definition Metamodel (ODM) Version 1.0”, <http://www.omg.org/spec/ODM/1.0/>, Released May 2009.
- [OMG 2010] Object Management Group (OMG), “Who’s OMG”, <http://www.omg.org/marketing/about-omg.htm>, Dec. 2010.
- [OWL Guide] World Wide Web Consortium (W3C), “OWL Web Ontology Language Guide”, W3C Recommendation, <http://www.w3.org/TR/owl-guide/>, 10 Feb. 2004.
- [OWL Overview] World Wide Web Consortium (W3C), “OWL Web Ontology Language Overview”, W3C Recommendation, <http://www.w3.org/TR/owl-features/>, 10 Feb. 2004.
- [O’Sullivan 2007] D. O’Sullivan, V. Wade and D. Lewis, “Understanding as We Roam”, in IEEE Internet Computing, 11, DOI: 10.1109/MIC.2007.50, p26 - 33, 2007.
- [Perez 1999] A. Gomez Perez and V. Richard Benjamins, “Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods”, Proceedings of IJCAI-99 Workshop on Ontologies and Problem Solving Methods, Stockholm, Sweden, 1999.
- [Pfenning 1988] F. Pfenning and C. Elliot, “Higher-order abstract syntax”, In proceedings of the ACM SIGPLAN, Conference on Programming Language Design and Implementation, pages 199-208, ACM Press, 1988.
- [Protégé Project] Protégé Project, “A free, open-source ontology editor and framework for building intelligent systems”, Developed at the Stanford Center for Biomedical Informatics Research (BMIR) at the Stanford University School of Medicine, <http://protege.stanford.edu/>.

- [Raghavan 2005] A. Raghavan, D. Rangarajan, R. Shen, M. A. Gonçalves, N. S. Vemuri, W. Fan and E. A. Fox, “Schema Mapper: A visualization tool for DL integration”, In Proceedings of the 5th ACM/IEEE-CS Joint Conference on (JCDL’05), pp. 414-414, Jun. 2005.
- [Raymond 2012] E. S. Raymond, “The New Hacker’s Dictionary 3rd Edition: Syntactic Sugar”, MIT Press, p. 432, ISBN 978-0-262-68092-9, http://books.google.co.uk/books?id=g80P_4v4QbIC&pg=PA432, Retrieved 5 Aug. 2012.
- [Rho 2007] S. Rho and J. Park, “Ontology: The key of Internet Evolution, from Web 2.0 to Web 3.0”, Gods’ Toy Business, Chapter 1, ISBN13-9788995919118, Korea, 2007.
- [RDF Concepts] World Wide Web Consortium (W3C), “Resource Description Framework (RDF): Concepts and Abstract Syntax”, W3C Recommendation, <http://www.w3.org/TR/rdf-concepts/>, 10 Feb. 2004.
- [RDF Schema] World Wide Web Consortium (W3C), “RDF Vocabulary Description Language 1.0: RDF Schema”, W3C Recommendation, <http://www.w3.org/TR/rdf-schema/>, 10 Feb. 2004.
- [RDF Syntax] World Wide Web Consortium (W3C), “RDF/XML Syntax Specification (Revised)”, W3C Recommendation, <http://www.w3.org/TR/rdf-syntax-grammar/>, 10 Feb. 2004.
- [RIF 2010] World Wide Web Consortium (W3C), “The Rule Interchange Format”, RIF Working Group, http://www.w3.org/2005/rules/wiki/RIF_Working_Group, Oct. 2010.

- [RIF Built-Ins] World Wide Web Consortium (W3C), “RIF Datatypes and Built-Ins 1.0”, W3C Recommendation, <http://www.w3.org/TR/rif-dtb/>, 22 Jun. 2010.
- [RIF BLD] World Wide Web Consortium (W3C), “RIF Basic Logic Dialect”, W3C Recommendation, <http://www.w3.org/TR/rif-bld/>, 22 Jun. 2010.
- [RIF Core] World Wide Web Consortium (W3C), “RIF Core Dialect”, W3C Recommendation, <http://www.w3.org/TR/rif-core/>, 22 Jun. 2010.
- [RIF Impl] World Wide Web Consortium (W3C), “RIF Implementations”, RIF Working Group, <http://www.w3.org/2005/rules/wiki/Implementations>, Retrieved 10 May 2011.
- [RIF SWC] World Wide Web Consortium (W3C), “RIF RDF and OWL Compatibility”, W3C Recommendation, <http://www.w3.org/TR/rif-rdf-owl/>, 22 Jun. 2010.
- [Sauro 2011] Jeff Sauro, “Measuring Usability with the System Usability Scale (SUS)”, Retrieved from <http://www.measuringusability.com/sus.php>, 11 Jul. 2011.
- [Sauro 2010] Jeff Sauro, “Measuring Usability: The Net Promoter Score and the System Usability Scale (SUS)”, Retrieved from <http://www.measuringusability.com/usability-loyalty.php>, 07 Jan. 2010.
- [SAX 2004] The SAX Project, “SAX 2.0.2: Simple API for XML”, <http://www.saxproject.org/>, 27 Apr. 2004.
- [Scharffe 2008] F. Scharffe and D. Fensel, “Correspondence Patterns for Ontology Alignment”, In Proceedings of the 16th International

- Conference on Knowledge Engineering (EKAW 2008), pp. 83-92, 2008.
- [Shvaiko 2005] P. Shvaiko and J. Euzenat, “A Survey of Schema-based Matching Approaches”, *Journal of Data Semantics IV*, pages 146-171, 2005.
- [Shvaiko 2013] P. Shvaiko and J. Euzenat, “Ontology matching: state of the art and future challenges”, *Knowledge and Data Engineering, IEEE Transactions on Volume 25-1*, pp. 158-176, 2013.
- [Siricharoen 2006] W. V. Siricharoen, “Ontologies and Object models in Object Oriented Software Engineering”, In *International Association of Engineers (IAENG) Journal of Computer Science (JICS)*, Volume 33, pp. 1151-1162, 2006.
- [Small Cell] Alcatel-Lucent Femtocell Test bed, “<http://www.alcatel-lucent.com>”, Alcatel-Lucent 9360 Small Cell.
- [SPARQL 2008] World Wide Web Consortium (W3C), “SPARQL Query Language for RDF”, *W3C Recommendation*, <http://www.w3.org/TR/rdf-sparql-query/>, 15 Jan. 2008.
- [SQWRL 2009] M. J. O’Connor and A. Das, “SQWRL: a Query Language for OWL” *OWL: Experiences and Directions (OWLED)*, 6th International Workshop, Chantilly, VA, 2009.
- [StarUML Project] StarUML Project, “StarUML - The Open Source UML/MDA Platform”, <http://staruml.sourceforge.net/en/>, StarUML 5.0 Official Stable released 30 Dec. 2005.
- [Thomas 2009] H. Thomas, D. O’Sullivan and R. Brennan, “Evaluation of Ontology Mapping Representations: a Pragmatic Evaluation”, In *Workshop on Matching and Meaning, Part of the AISB 2009 Convention*, April 9th 2009. Edinburgh, Scotland, 2009.

- [UML 2011] UML® Resource Page, “Unified Modeling Language”, <http://www.uml.org>, Retrieved Jul. 2011.
- [UML 2.0] Object Management Group (OMG), “Unified Modeling Language: Superstructure version 2.0”, <http://www.omg.org/spec/UML/2.0/>, Released Jul. 2010.
- [Unicode] The Unicode Standard, “Unicode Normalization Forms version 6.1.0”, Technical Reports, <http://www.unicode.org/reports/tr15/>, 13 Jan. 2012.
- [Usability 2014] Improving the User Experience (usability.gov), “System Usability Scale (SUS)”, Methods in How To & Tools, <http://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html/>, Form Approved OMB#0990-0379 Exp-date, 6 Mar. 2014.
- [Waters 2009] J. Waters, B. J. Powers and M.G. Ceruti, “Global Interoperability Using Semantics, Standards, Science & Technology (GIS3T),” J. Computer Stands and Interfaces, Vol. 31-6, pp. 1158-1166, 2009.
- [XML Datatype] World Wide Web Consortium (W3C), “XML Schema Part 2: Datatypes Second Edition”, W3C Recommendation, <http://www.w3.org/TR/xmlschema-2/>, 28 Oct. 2004.
- [XML Namespace] World Wide Web Consortium (W3C), “Namespaces in XML 1.0 (Third Edition)”, W3C Recommendation, <http://www.w3.org/TR/REC-xml-names/>, 8 Dec. 2009.
- [XSLT 1999] World Wide Web Consortium (W3C), “XSL Transformations (XSLT) Version 1.0”, W3C Recommendation, <http://www.w3.org/TR/xslt>, 16 Nov. 1999.
- [XSPARQL 2009] World Wide Web Consortium (W3C), “XSPARQL Language Specification”, W3C Member Submission,

<http://www.w3.org/Submission/2009/SUBM-xsparql-language-specification-20090120/>, 20 Jan. 2009.

[XQuery 2010] World Wide Web Consortium (W3C), “XQuery 1.0: An XML Query Language (Second Edition)”, W3C Recommendation, <http://www.w3.org/TR/xquery/>, 14 Dec. 2010.

[Yan 2003] B. Yan, M. Frank and P. Szekely, “WebScripter: Grass-root Ontology Alignment via End-User Report Creation”, In Proceedings of the 2nd International Semantic Web Conference (ISWC 2003), LCNS, vol. 2870, Springer, Heidelberg, 2003.

APPENDIX A Ontologies from the industry use case

Appendix A.1 Femto level ontology - femtoclassifier.owl

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<rdf:RDF xmlns="http://femto.testbed.alcatel-lucent.com/femtoclassifier#"
  xml:base="http://femto.testbed.alcatel-lucent.com/femtoclassifier"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace">
  <owl:Ontology rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier.owl"/>

  <!--
  ////////////////////////////////////////////////////////////////////
  //
  // Classes
  //
  ////////////////////////////////////////////////////////////////////
  -->

  <!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#Accessibility -->

  <owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Accessibility">
  <rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkHealthData"/>
  </owl:Class>

  <!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#AccessibilityLevels -->

  <owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#AccessibilityLevels">
  <rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Femto"/>
  </owl:Class>

  <!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#Femto -->

  <owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Femto"/>

  <!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#Handover -->

  <owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Handover">
  <rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkHealthData"/>
  </owl:Class>

  <!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#HandoverLevels -->
```

```

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#HandoverLevels">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Femto"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#HighHandoverFemto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#HighHandoverFemto">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#HandoverLevels"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#HighPSDataFemto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#HighPSDataFemto">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#PSDataLevels"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#HighRABFemto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#HighRABFemto">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#RABLevels"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#HighlyAccessibilyFemtos -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#HighlyAccessibilyFemtos">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#AccessabilityLevels"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#HighlyReliableFemto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#HighlyReliableFemto">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#ReliabilityLevels"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#HomeNetwork -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#HomeNetwork">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkHealthData"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#LowAccessibilityFemto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#LowAccessibilityFemto">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#AccessabilityLevels"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#LowHandoverFemto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#LowHandoverFemto">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#HandoverLevels"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#LowPSDataFemto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#LowPSDataFemto">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#PSDataLevels"/>
</owl:Class>

```

```

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#LowRABFemto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#LowRABFemto">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#RABLevels"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#LowReliabilibiltyFemto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#LowReliabilibiltyFemto">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#ReliabilityLevels"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#MediumAccessibilityFemto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#MediumAccessibilityFemto">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#AccessabilityLevels"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#MediumHandoverFemto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#MediumHandoverFemto">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#HandoverLevels"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#MediumPSDataFemto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#MediumPSDataFemto">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#PSDataLevels"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#MediumRABFemto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#MediumRABFemto">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#RABLevels"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#MediumReliabilityFemto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#MediumReliabilityFemto">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#ReliabilityLevels"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#NPM -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NPM">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkData"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkData -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkData"/>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkHealthData -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkHealthData">
<rdfs:comment rdf:datatype="&xsd:string">these subclasses come for the KPI categories.</rdfs:comment>
</owl:Class>

```

```

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#PSData -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#PSData">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkHealthData"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#PSDataLevels -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#PSDataLevels">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Femto"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#RAB -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#RAB">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkHealthData"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#RABLevels -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#RABLevels">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Femto"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#Reliability -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Reliability">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkHealthData"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#ReliabilityLevels -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#ReliabilityLevels">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Femto"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#Security -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Security">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkHealthData"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#System -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#System">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkHealthData"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#Traffic -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Traffic">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkHealthData"/>
</owl:Class>

<!--
////////////////////////////////////
//
// Object Properties

```

```

//
////////////////////////////////////
-->

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasAccesabilityKPI -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasAccesabilityKPI">
<rdfs:range rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Accessibility"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Femto"/>
<rdfs:subPropertyOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasKPI"/>
</owl:ObjectProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasHandoverKPI -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasHandoverKPI">
<rdfs:range rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Handover"/>
<rdfs:subPropertyOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasKPI"/>
</owl:ObjectProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasHomeNetworkKPI -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasHomeNetworkKPI">
<rdfs:range rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#HomeNetwork"/>
<rdfs:subPropertyOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasKPI"/>
</owl:ObjectProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasKPI -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasKPI">
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Femto"/>
<rdfs:range rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NetworkHealthData"/>
</owl:ObjectProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasNPMDData -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasNPMDData">
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Femto"/>
<rdfs:range rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#NPM"/>
</owl:ObjectProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasPMDData -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasPMDData"/>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasPSDataKPI -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasPSDataKPI">
<rdfs:range rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#PSData"/>
<rdfs:subPropertyOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasKPI"/>
</owl:ObjectProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasRABKPI -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasRABKPI">
<rdfs:range rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#RAB"/>
<rdfs:subPropertyOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasKPI"/>
</owl:ObjectProperty>

```

```

<!-- http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasTrafficKPI -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasTrafficKPI">
<rdfs:range rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#Traffic"/>
<rdfs:subPropertyOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoclassifier#hasKPI"/>
</owl:ObjectProperty>

</rdf:RDF>

```

Appendix A.2 Raw level ontology - femtoinstances.owl

```

<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
<!ENTITY owl "http://www.w3.org/2002/07/owl#" >
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
<!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<rdf:RDF xmlns="http://femto.testbed.alcatel-lucent.com/femtoinstances#"
xml:base="http://femto.testbed.alcatel-lucent.com/femtoinstances"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xml="http://www.w3.org/XML/1998/namespace">
<owl:Ontology rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances.owl"/>

<!--
////////////////////////////////////
//
// Classes
//
////////////////////////////////////
-->

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM">
<rdfs:subClassOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NetworkData"/>
</owl:Class>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#NetworkData -->

<owl:Class rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#NetworkData"/>

<!--
////////////////////////////////////
//

```

```

// Object Properties
//
////////////////////////////////////
-->

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#hasAccesabilityKPI -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasAccesabilityKPI">
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:subPropertyOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasKPI"/>
</owl:ObjectProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#hasHandoverKPI -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasHandoverKPI">
<rdfs:subPropertyOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasKPI"/>
</owl:ObjectProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#hasHomeNetworkKPI -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasHomeNetworkKPI">
<rdfs:subPropertyOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasKPI"/>
</owl:ObjectProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#hasKPI -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasKPI">
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
</owl:ObjectProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#hasNPMDData -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasNPMDData">
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
</owl:ObjectProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#hasPMDData -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasPMDData"/>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#hasPSDataKPI -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasPSDataKPI">
<rdfs:subPropertyOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasKPI"/>
</owl:ObjectProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#hasRABKPI -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasRABKPI">
<rdfs:subPropertyOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasKPI"/>
</owl:ObjectProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#hasTrafficKPI -->

<owl:ObjectProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasTrafficKPI">
<rdfs:subPropertyOf rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#hasKPI"/>
</owl:ObjectProperty>

```



```

<!--
////////////////////////////////////
//
// Data properties
//
////////////////////////////////////
-->

<!-- http://femto.testbed.alcatel-
lucent.com/femtoinstances#BSR_cluster_to_GSM_Hard_Handovers_CS_Voice_Failure_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#BSR_cluster_to_GSM_Hard_Handovers_CS_Voice_Failure_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-
lucent.com/femtoinstances#BSR_cluster_to_MTS_underlay_Intra_Frequency_Hard_Handover_Failure_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#BSR_cluster_to_MTS_underlay_Intra_Frequency_Hard_Handover_Failure_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-
lucent.com/femtoinstances#BSR_cluster_to_UMTS_underlay_Hard_Handover_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#BSR_cluster_to_UMTS_underlay_Hard_Handover_Success_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-
lucent.com/femtoinstances#BSR_cluster_to_UMTS_underlay_Inter_Frequency_Hard_Handover_Failure_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#BSR_cluster_to_UMTS_underlay_Inter_Frequency_Hard_Handover_Failure_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#CSD_RAB_Establishment_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#CSD_RAB_Establishment_Success_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

```

```

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#CS_Data_Call_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#CS_Data_Call_Success_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#CS_Speech_Call_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#CS_Speech_Call_Success_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#CS_Successful_RRC_Connection_Establishment_Rate
-->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#CS_Successful_RRC_Connection_Establishment_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#CS_Voice_RAB_Establishment_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#CS_Voice_RAB_Establishment_Success_Rate">
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#DCH_PS_RAB_Establishment_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#DCH_PS_RAB_Establishment_Success_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#DCH_PS_RAB_Success -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#DCH_PS_RAB_Success">
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#DatatypeProperty_2 -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#DatatypeProperty_2">
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

```

```

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#E-DCH_Call_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#E-
DCH_Call_Success_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#E-DCH_PS_RAB_Establishment_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#E-
DCH_PS_RAB_Establishment_Success_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto-BSR_to_Femto-
BSR_CS_Hard_Handover_in_same_Femto-BSR_Cluster_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto-BSR_to_Femto-
BSR_CS_Hard_Handover_in_same_Femto-BSR_Cluster_Success_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#HHO_AttBsrGSMCSVoice -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#HHO_AttBsrGSMCSVoice">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#HHO_SuccBsrGSMCSVoice -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#HHO_SuccBsrGSMCSVoice">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#HSDPA_Call_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#HSDPA_Call_Success_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#HSDPA_PS_RAB_Establishment_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-

```

```

lucent.com/femtoinstances#HSDPA_PS_RAB_Establishment_Success_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#HSDPA_PS_RAB_Success -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#HSDPA_PS_RAB_Success">
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#HomeNetwork_Local_Routing_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#HomeNetwork_Local_Routing_Success_Rate">
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#KPI_2G_Macro_to_Femto-
BSR_CS_Handover_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#KPI_2G_Macro_to_Femto-BSR_CS_Handover_Success_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#KPI_3G_Macro_to_Femto-
BSR_CS_Handover_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#KPI_3G_Macro_to_Femto-BSR_CS_Handover_Success_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#MM_AttCsPagingProc -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#MM_AttCsPagingProc">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#MM_AttPsPagingProc -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#MM_AttPsPagingProc">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

```

```

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#MM_SuccCsPagingProc -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#MM_SuccCsPagingProc">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#MM_SuccPsPagingProc -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#MM_SuccPsPagingProc">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#NumCellUpdate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#NumCellUpdate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#NumRBReconfAtt_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#NumRBReconfAtt_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#NumRBReconfFail_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#NumRBReconfFail_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#NumSuccCellUpdate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#NumSuccCellUpdate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#NumUserBits_HSDPADL -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#NumUserBits_HSDPADL">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>

```

```

<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#NumUserBits_PS128DL -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#NumUserBits_PS128DL">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#NumUserBits_PS128UL -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#NumUserBits_PS128UL">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#NumUserBits_PS384DL -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#NumUserBits_PS384DL">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#NumUserBits_PS384UL -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#NumUserBits_PS384UL">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#NumUserBits_PS64DL -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#NumUserBits_PS64DL">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#NumUserBits_PS64UL -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#NumUserBits_PS64UL">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

```

```

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Num_PreEmptCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#Num_PreEmptCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Number_of_Full_EAP-
AKA_Authentication_Attempts -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#Number_of_Full_EAP-
AKA_Authentication_Attempts">
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-
lucent.com/femtoinstances#Number_of_Transmission_octets_on_Signalling_type_traffic -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Number_of_Transmission_octets_on_Signalling_type_traffic">
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Oam_Bootup_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#Oam_Bootup_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Oam_CellUnavailable_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Oam_CellUnavailable_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Oam_HardwareRfFailure_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Oam_HardwareRfFailure_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Oam_SoftwareError_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Oam_SoftwareError_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

```

```

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#PS_RAB_Establishment_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#PS_RAB_Establishment_Success_Rate">
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#PS_Successful_RRC_Connection_Establishment_Rate
-->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#PS_Successful_RRC_Connection_Establishment_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Proportional_UTRAN_E-DCH_RAB_Drop_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Proportional_UTRAN_E-DCH_RAB_Drop_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#R99_PS_Call_Success_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#R99_PS_Call_Success_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_AttEstabCS_Conv -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_AttEstabCS_Conv">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_AttEstabCS_Data -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_AttEstabCS_Data">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_AttEstabPS_DCH_DCH -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_AttEstabPS_DCH_DCH">

```



```

<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_AttEstabPS_DCH_HSDSCH -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_AttEstabPS_DCH_HSDSCH">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_AttEstabPS_EDCH -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_AttEstabPS_EDCH">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_AttEstabPS_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_AttEstabPS_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_AttEstabSecondPS_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_AttEstabSecondPS_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_FailEstabCS_Conv -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_FailEstabCS_Conv">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_FailEstabCS_Data -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_FailEstabCS_Data">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

```

```

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_FailEstabPS_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_FailEstabPS_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_Rel_DropCSData_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_Rel_DropCSData_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_Rel_DropCSVoice_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_Rel_DropCSVoice_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_Rel_DropPS_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_Rel_DropPS_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_SuccEstabCS_Conv -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_SuccEstabCS_Conv">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_SuccEstabCS_Data -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_SuccEstabCS_Data">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_SuccEstabPSNoQueuing_DCH_DCH -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-

```

```

lucent.com/femtoinstances#RAB_SuccEstabPSNoQueuing_DCH_DCH">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_SuccEstabPSNoQueuing_DCH_HSDSCH -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_SuccEstabPSNoQueuing_DCH_HSDSCH">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_SuccEstabPS_EDCH -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_SuccEstabPS_EDCH">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RAB_SuccEstabPS_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RAB_SuccEstabPS_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_CallReestab -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_CallReestab">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_Detach -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_Detach">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_Emergency -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_Emergency">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>

```

```

</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_OrigBgrdCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_OrigBgrdCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_OrigConvCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_OrigConvCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_OrigHiPrioSign -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_OrigHiPrioSign">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_OrigIntactCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_OrigIntactCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_OrigLoPrioSign -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_OrigLoPrioSign">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_OrigStrmCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_OrigStrmCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_OrigSubscrTrafficCall -->

```

```

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_OrigSubscrTrafficCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_Registration -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_Registration">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_TermBgrdCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_TermBgrdCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_TermCauseUnknown -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_TermCauseUnknown">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_TermConvCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_TermConvCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_TermHiPrioSign -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_TermHiPrioSign">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_TermIntactCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_TermIntactCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>

```

```

<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_TermLoPrioSign -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_TermLoPrioSign">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_TermStrmCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_TermStrmCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_AttConnEstab_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_AttConnEstab_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_ConnReject_Congestion -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_ConnReject_Congestion">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_ConnSRBDrop_CS -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_ConnSRBDrop_CS">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_ConnSRBDrop_PS -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_ConnSRBDrop_PS">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_CallReestab -->

```

```

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_CallReestab">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_Detach -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_Detach">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_Emergency -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_Emergency">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_OrigBgrdCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_OrigBgrdCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_OrigConvCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_OrigConvCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_OrigHiPrioSign -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_OrigHiPrioSign">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_OrigIntactCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_OrigIntactCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>

```

```

<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd:integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_OrigLoPrioSign -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_OrigLoPrioSign">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd:integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_OrigStrmCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_OrigStrmCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd:integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_OrigSubscrTrafficCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_OrigSubscrTrafficCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd:integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_Registration -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_Registration">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd:integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_TermBgrdCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_TermBgrdCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd:integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_TermCauseUnknown -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_TermCauseUnknown">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd:integer"/>
</owl:DatatypeProperty>

```



```

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_TermConvCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_TermConvCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_TermHiPrioSign -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_TermHiPrioSign">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_TermIntactCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_TermIntactCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_TermLoPrioSign -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_TermLoPrioSign">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_TermStrmCall -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_TermStrmCall">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#RRC_SuccConnEstab_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#RRC_SuccConnEstab_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Standalone_CS_SRB_Drop_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Standalone_CS_SRB_Drop_Rate">

```

```

<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Standalone_CS_SRB_Drop_Rate_name -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Standalone_CS_SRB_Drop_Rate_name">

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Standalone_CS_SRB_Drop_Rate_value -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Standalone_CS_SRB_Drop_Rate_value">

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Standalone_PS_SRB_Drop_Rate -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Standalone_PS_SRB_Drop_Rate">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#System_CPULoad_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#System_CPULoad_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#System_MemoryUsage_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#System_MemoryUsage_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#System_SoftwareRestart_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#System_SoftwareRestart_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_RxCsDropPacket_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_RxCsDropPacket_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>

```

```

</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_RxCsOctet_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_RxCsOctet_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_RxCsPacket_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_RxCsPacket_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_RxOamOctet_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_RxOamOctet_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_RxOamPacket_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_RxOamPacket_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_RxPsDropPacket_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_RxPsDropPacket_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_RxPsOctet_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_RxPsOctet_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_RxPsPacket_Sum -->

```

```

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_RxPsPacket_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_RxSigOctet_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_RxSigOctet_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_RxSigPacket_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_RxSigPacket_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_RxSyncOctet_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_RxSyncOctet_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_RxSyncPacket_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_RxSyncPacket_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_TxCsOctet_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_TxCsOctet_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_TxCsPacket_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_TxCsPacket_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>

```

```

<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_TxOamOctet_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_TxOamOctet_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_TxOamPacket_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_TxOamPacket_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_TxPsOctet_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_TxPsOctet_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_TxPsPacket_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_TxPsPacket_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_TxSigOctet_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_TxSigOctet_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_TxSigPacket_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_TxSigPacket_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_TxSyncOctet_Sum -->

```

```

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_TxSyncOctet_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Traffic_TxSyncPacket_Sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Traffic_TxSyncPacket_Sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#UL_E-DCH_Data_Volume_on_RLC_SDU -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#UL_E-
DCH_Data_Volume_on_RLC_SDU">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_AttActiveCallReDirect2G -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_AttActiveCallReDirect2G">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_AttActiveCallReDirectUMTS -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_AttActiveCallReDirectUMTS">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_Att_FastAuth_EAPAKA -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_Att_FastAuth_EAPAKA">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_Att_FullAuth_EAPAKA -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_Att_FullAuth_EAPAKA">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>

```

```

<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_Fail_FastAuth_EAPAKA -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_Fail_FastAuth_EAPAKA">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_Fail_FullAuth_EAPAKA -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_Fail_FullAuth_EAPAKA">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_HARQRetransmissionsThree_EDCH_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_HARQRetransmissionsThree_EDCH_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_HARQRetransmissionsTwo_EDCH_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_HARQRetransmissionsTwo_EDCH_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_HARQRetransmissionsZero_EDCH_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_HARQRetransmissionsZero_EDCH_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_HHO_ACRPSCallDrop -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_HHO_ACRPSCallDrop">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

```

```

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_HHO_AttBsrToBsrCSHOSameCluster -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_HHO_AttBsrToBsrCSHOSameCluster">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_HHO_AttBsrUmtsInterFreq -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_HHO_AttBsrUmtsInterFreq">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_HHO_AttBsrUmtsIntraFreq -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_HHO_AttBsrUmtsIntraFreq">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_HHO_PSCallDrop -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_HHO_PSCallDrop">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_HHO_SuccBsrToBsrCSHOSameCluster -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_HHO_SuccBsrToBsrCSHOSameCluster">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_HHO_SuccBsrUmtsInterFreq -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_HHO_SuccBsrUmtsInterFreq">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_HHO_SuccBsrUmtsIntraFreq -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_HHO_SuccBsrUmtsIntraFreq">

```



```

<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_MaxAvailDLBandwidth -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_MaxAvailDLBandwidth">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_MaxAvailULBandwidth -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_MaxAvailULBandwidth">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_NumAttActCallTransCAC -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_NumAttActCallTransCAC">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_NumCallRejectTransCAC -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_NumCallRejectTransCAC">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_NumDLTransCACFail -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_NumDLTransCACFail">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_NumSuccActCallTransCAC -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_NumSuccActCallTransCAC">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

```

```

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_NumULTransCACFail -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_NumULTransCACFail">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_NumUserBits_EDCH_PS -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_NumUserBits_EDCH_PS">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#VS_RAB_Rel_DropPS_EDCH -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#VS_RAB_Rel_DropPS_EDCH">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Vs_HARQFail_EDCH_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Vs_HARQFail_EDCH_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Vs_HARQRetransmissionsGTthree_EDCH_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Vs_HARQRetransmissionsGTthree_EDCH_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Vs_HARQRetransmissionsOne_EDCH_sum -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Vs_HARQRetransmissionsOne_EDCH_sum">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Vs_SuccActiveCallReDirect2G -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-

```

```

lucent.com/femtoinstances#Vs_SuccActiveCallReDirect2G">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#Vs_SuccActiveCallReDirectUMTS -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-
lucent.com/femtoinstances#Vs_SuccActiveCallReDirectUMTS">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#NPM"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#datatypeProperty_3 -->

<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#datatypeProperty_3"/>

<!-- http://femto.testbed.alcatel-lucent.com/femtoinstances#femtoid -->

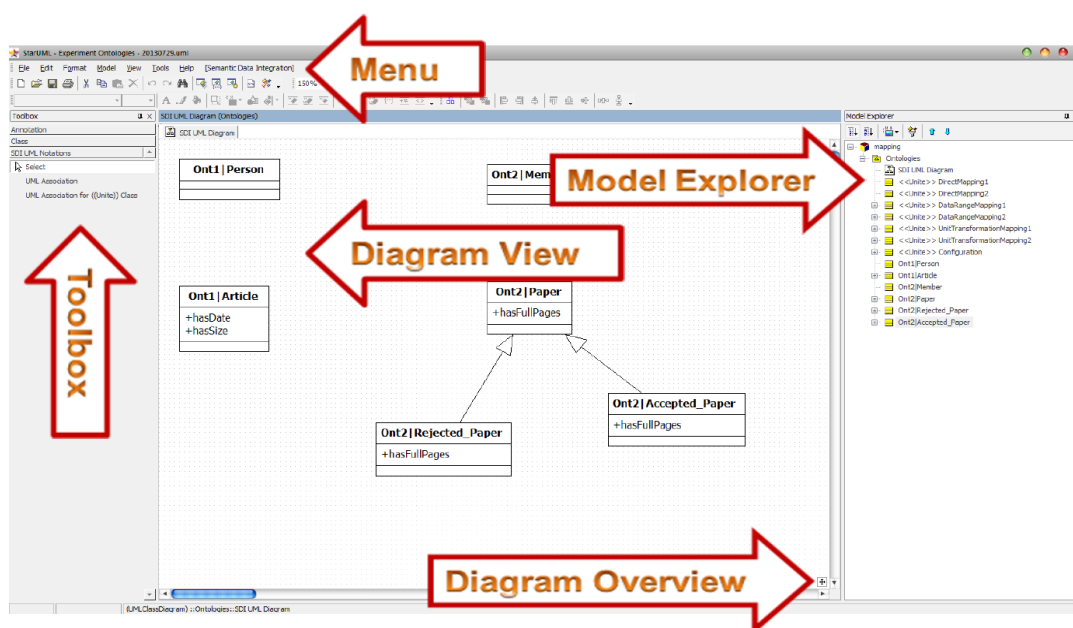
<owl:DatatypeProperty rdf:about="http://femto.testbed.alcatel-lucent.com/femtoinstances#femtoid">
<rdfs:domain rdf:resource="http://femto.testbed.alcatel-lucent.com/femtoinstances#Femto"/>
</owl:DatatypeProperty>

</rdf:RDF>

```

APPENDIX B SDI tool user manual in GUI mode with examples

Reference Manual for SDI Tool



The tool is composed by four separate parts:

- ✓ Diagram View, Diagram Overview: to show UML Diagram
- ✓ Toolbox: to connect <<Unite>> UML class with UML classes
- ✓ Model Explorer: to drag and drop <<Unite>> UML classes
- ✓ Menu: to generate mapping outputs

The contents of this manual:

1. Example of how to create DirectMapping	2
2. Example of how to create DataRangeMapping	5
3. Example of how to create UnitTransformationMapping	8
4. Example of how to generate Mapping Outputs	11

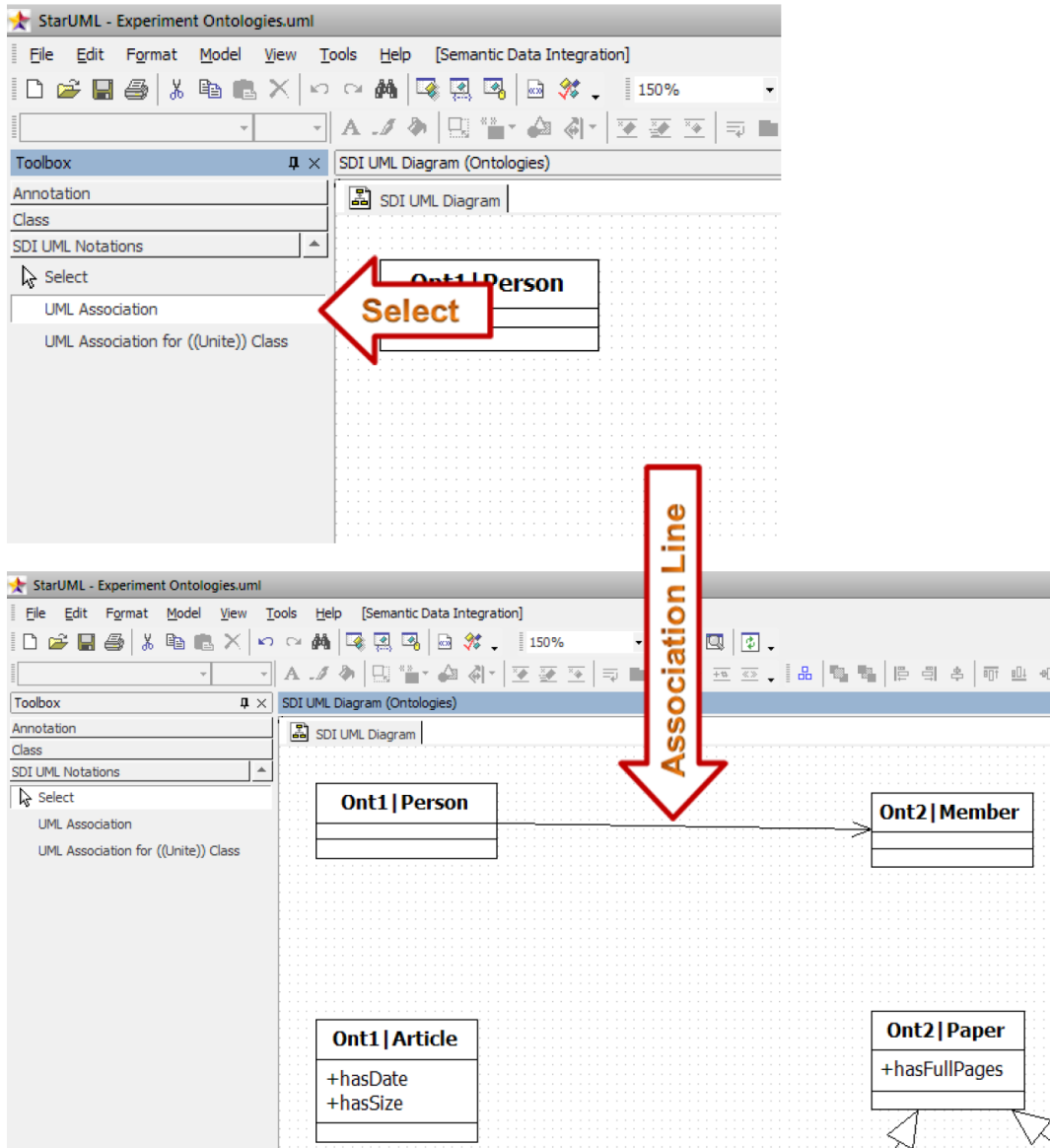
Appendix B.1 Example of how to create direct mapping

1. Example of how to create DirectMapping

This mapping type represents the expression for one ontology class (A) corresponding to another ontology class (B) with the integration property corresponding to “Unite the instances of class (A) to class (B)”.

1. Create UML Association between source and target UML Classes

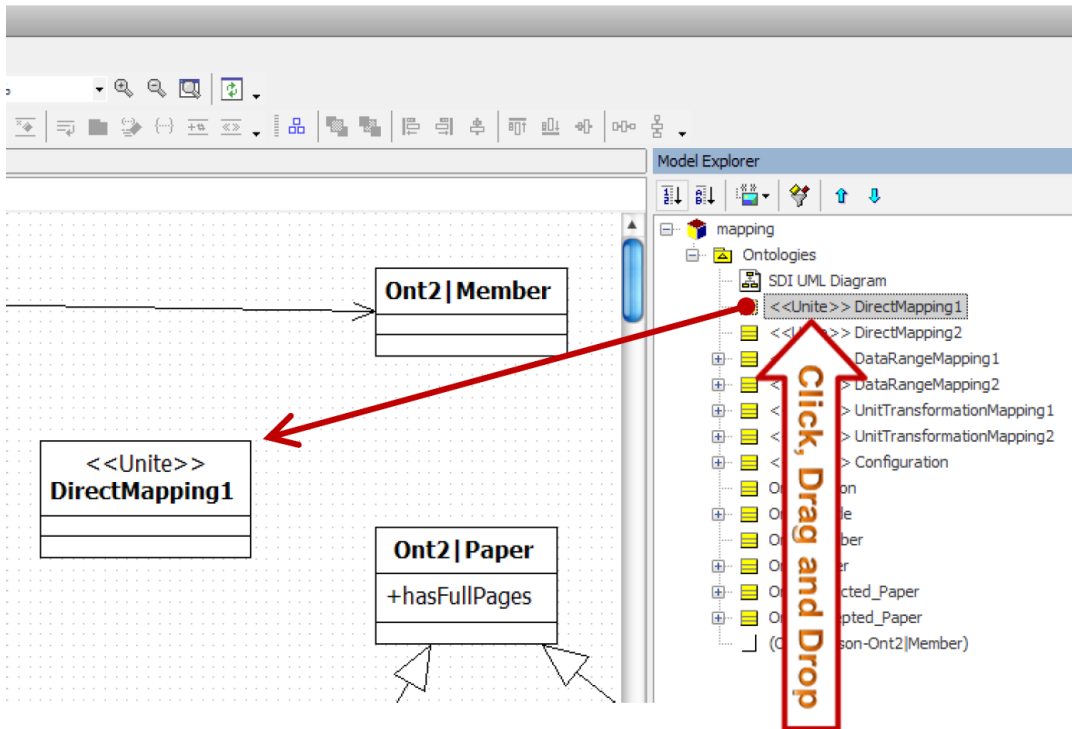
- Select UML Association in Toolbox.
- Click source UML class in Diagram View and drag to target UML class.



Now, you can see Directed UML Association Line between Classes.

2. Create <<Unite>> stereotyped UML Class

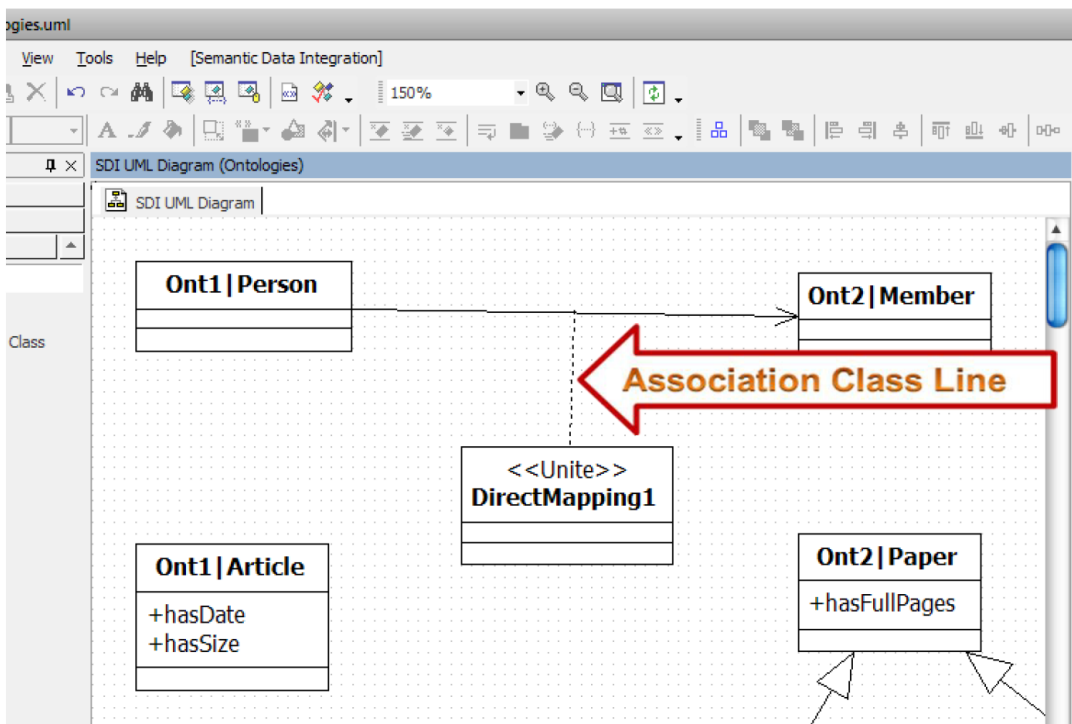
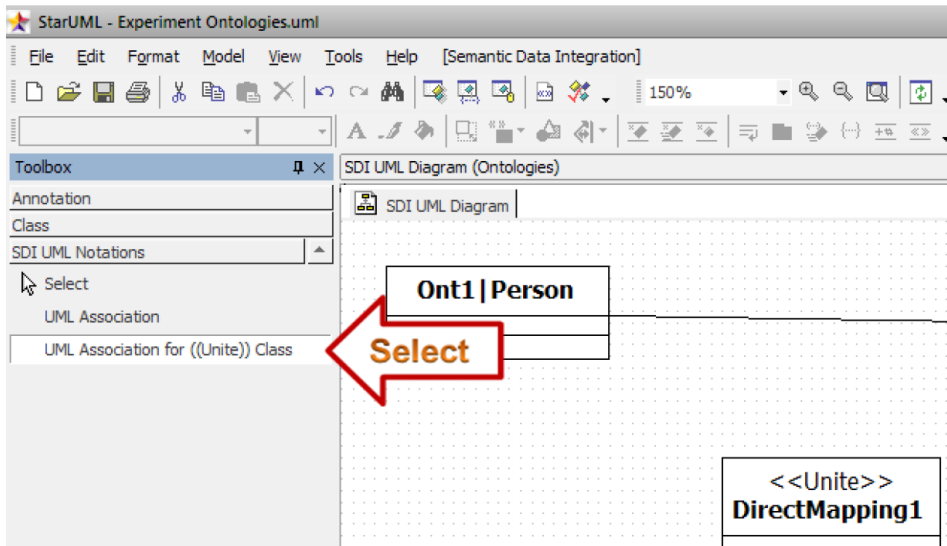
- Select <<Unite>> DirectMapping1 or DirectMapping2 in **Model Explorer**.
- Drag and Drop <<Unite>> DirectMapping1 or 2 in Model Explorer into Diagram View.



Now, you can see <<Unite>> DirectMapping1 UML Class.

3. Create <<Unite>> stereotyped UML Association Class

- Select UML Association for ((Unite)) Class in Toolbox.
- Click **UML Association Line** in Diagram View and drag to <<Unite>> stereotyped UML class.



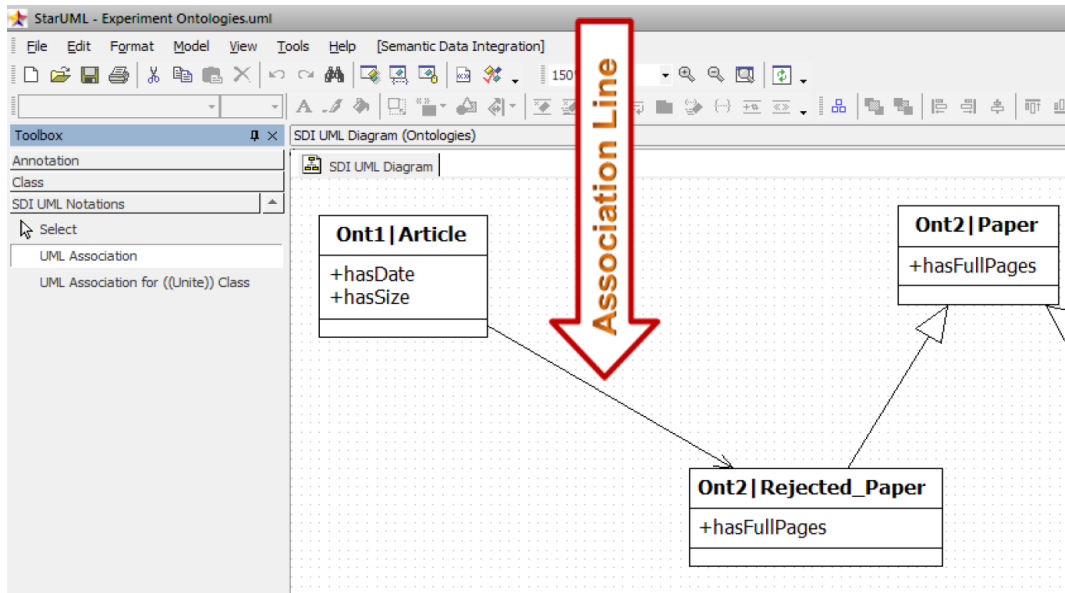
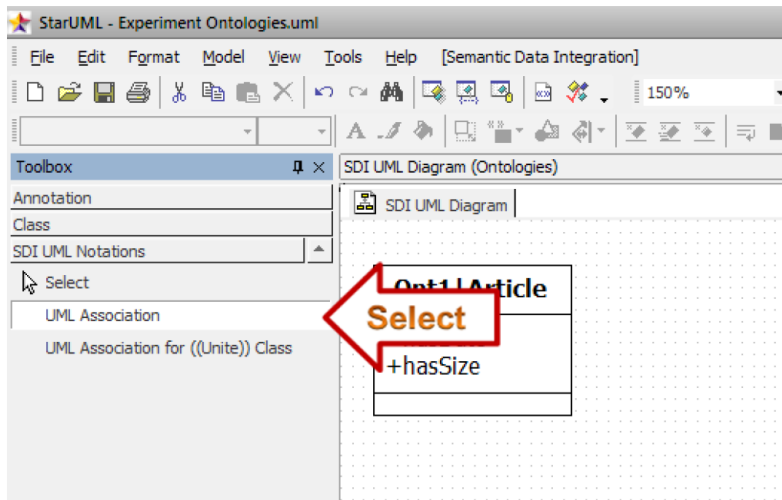
Now, you can see UML Association Class Line between UML Association and <<Unite>> stereotyped UML Class.

Appendix B.2 Example of how to create data range mapping

2. Example of how to create DataRangeMapping

This mapping type represents the expression for an ontology class (A), that is constrained by the data range of the datatype properties, corresponding to another ontology class (B) with the integration property corresponding to “Unite the instances of class (A) to class (B) if the instance of class (A) satisfies the data range constraint”.

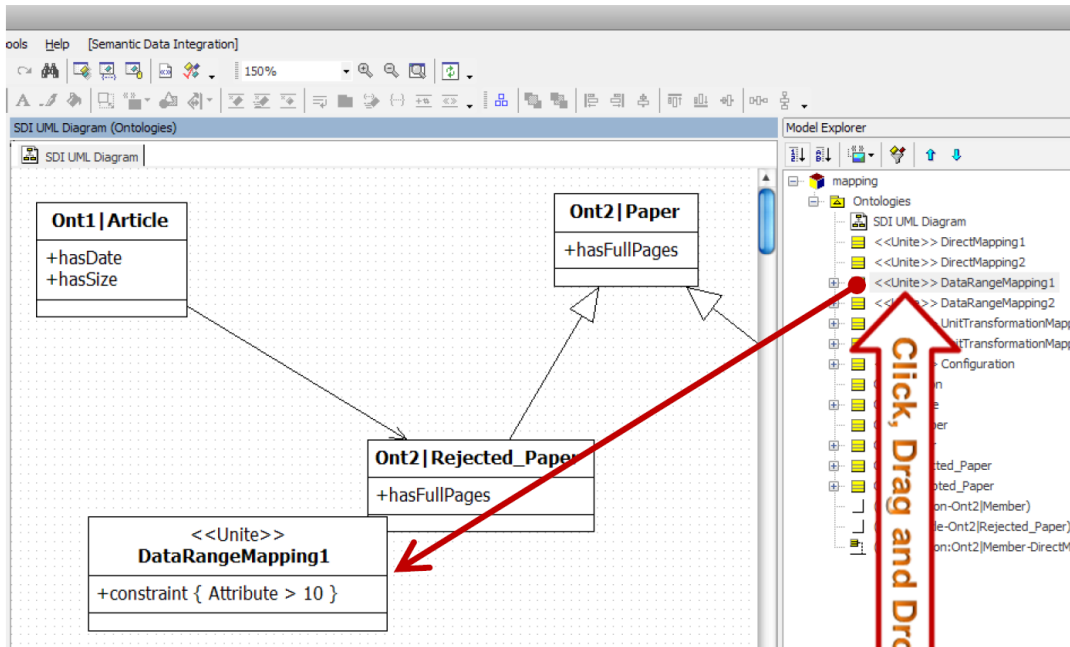
1. **Create UML Association between source and target UML Classes**
 - Select UML Association in Toolbox.
 - Click source UML class in Diagram View and drag to target UML class.



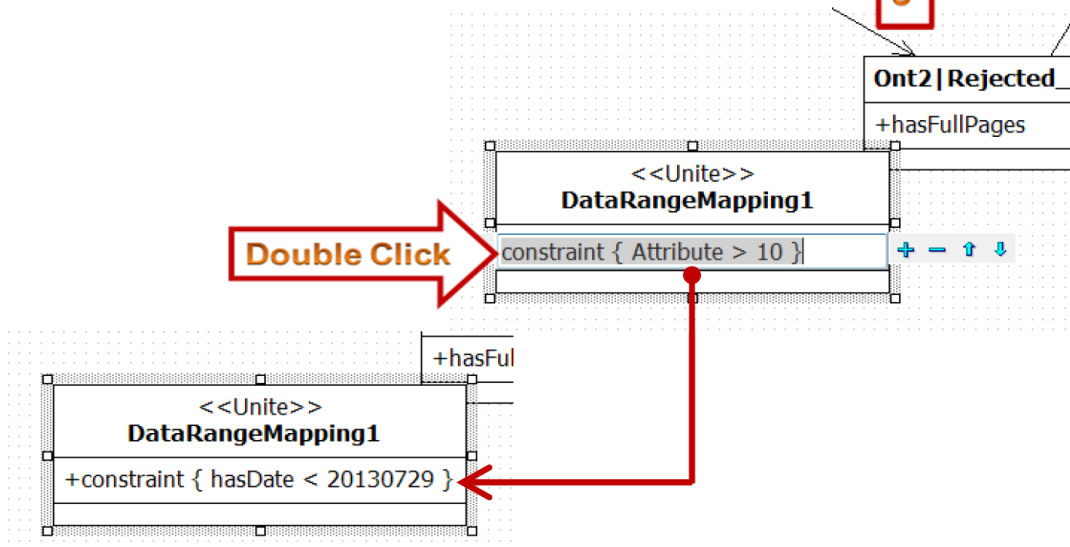
Now, you can see Directed UML Association Line between Classes.

2. Create <<Unite>> stereotyped UML Class

- Select <<Unite>> DataRangeMapping1 or DataRangeMapping2 in **Model Explorer**.
- Drag and Drop <<Unite>> DataRangeMapping1 or 2 in Model Explorer into Diagram View.
- **Double click on the attribute** to change the attribute in <<Unite>> DataRangeMapping1 Class.
- Modify default UML constraint which is in between “{“ and “}” according to your design of the mapping.



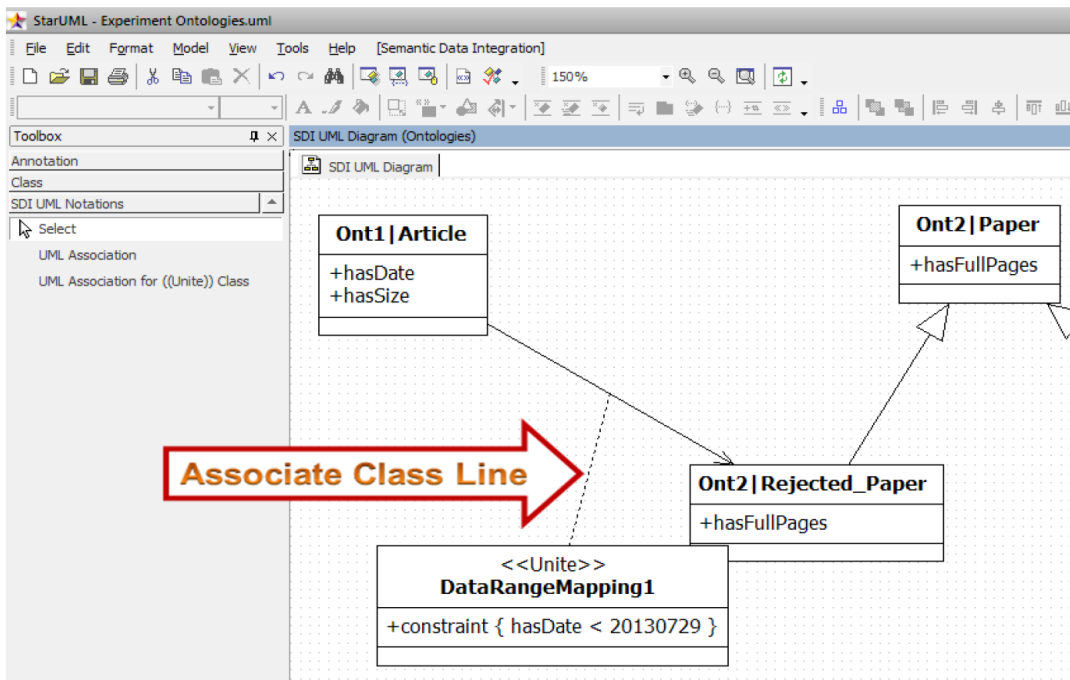
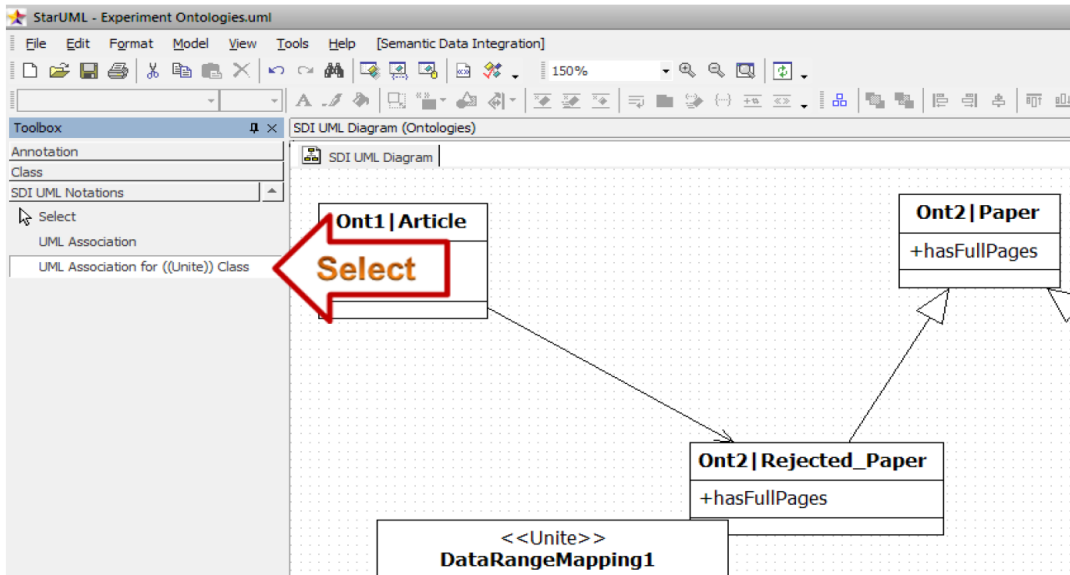
Click, Drag and Drop



Now, you can see <<Unite>> DataRangeMapping1 UML Class with your intended constraint.

3. Create <<Unite>> stereotyped UML Association Class

- Select UML Association for ((Unite)) Class in Toolbox.
- Click **UML Association Line** in Diagram View and drag to <<Unite>> stereotyped UML class.



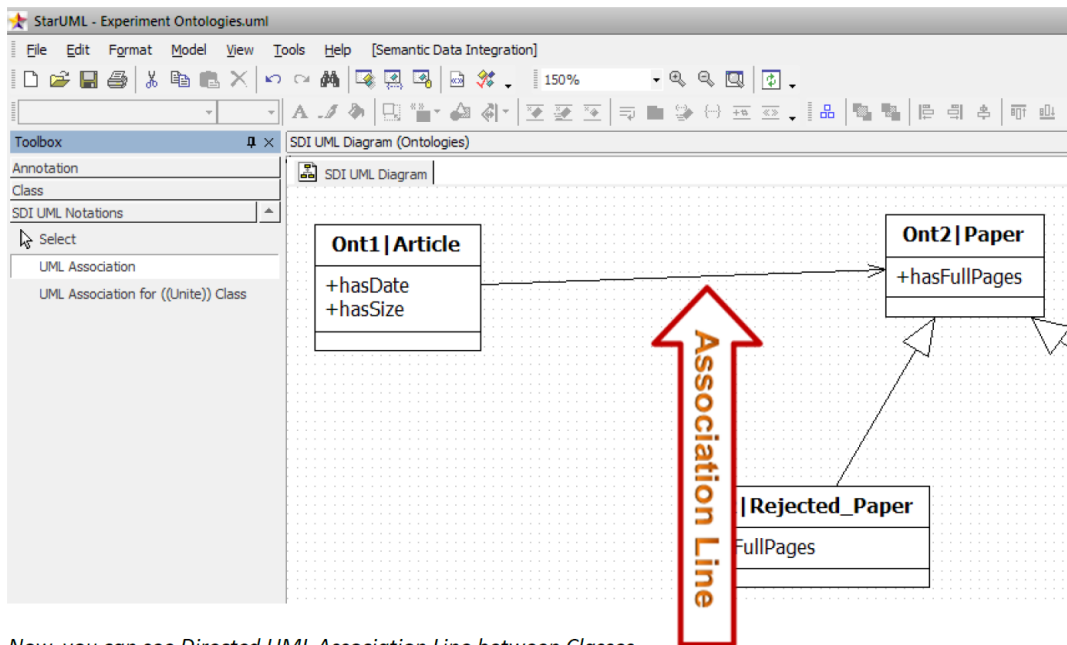
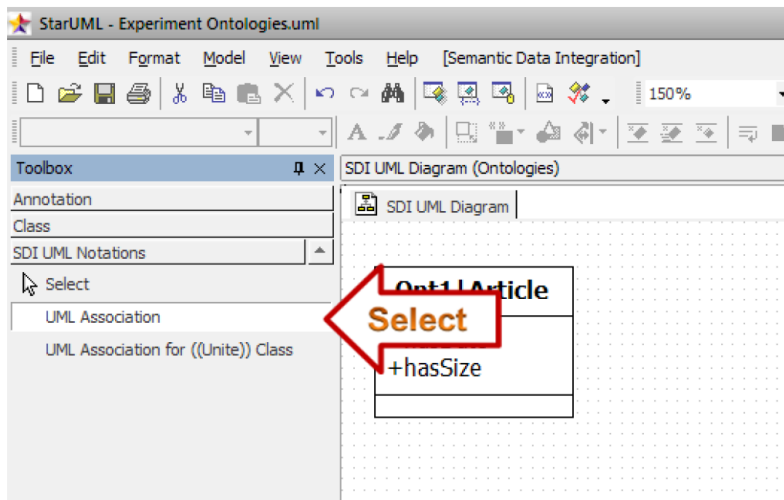
Now, you can see UML Association Class Line between UML Association and <<Unite>> stereotyped UML Class.

Appendix B.3 Example of how to create unit transformation mapping

3. Example of how to create UnitTransformationMapping

This mapping type represents the expression for an ontology class (A), that is constrained by the arithmetic transformation of the data value of the datatype properties, corresponding to another ontology class (B) with the integration property corresponding to "Unite the instances of class (A) to class (B) with the unit transformation according to the constraint".

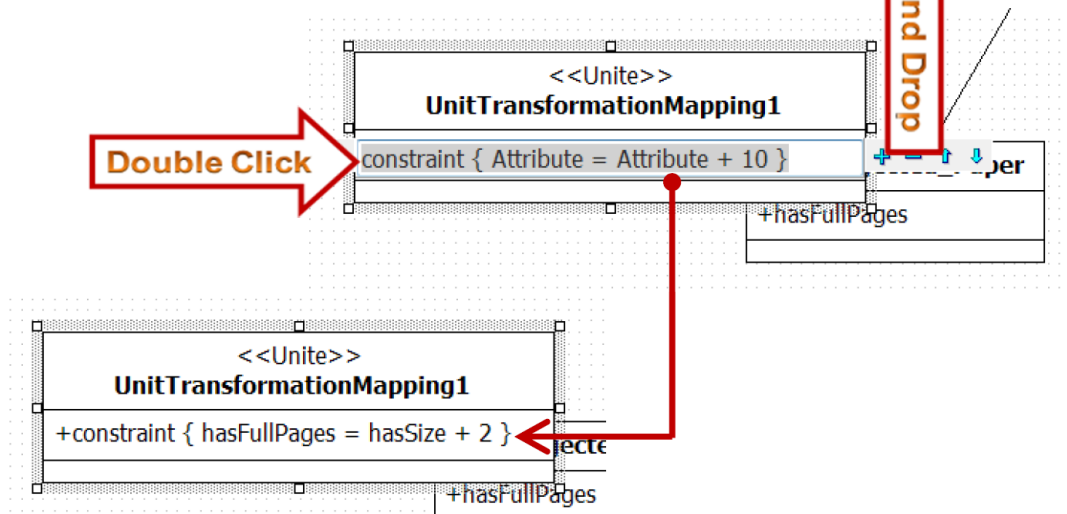
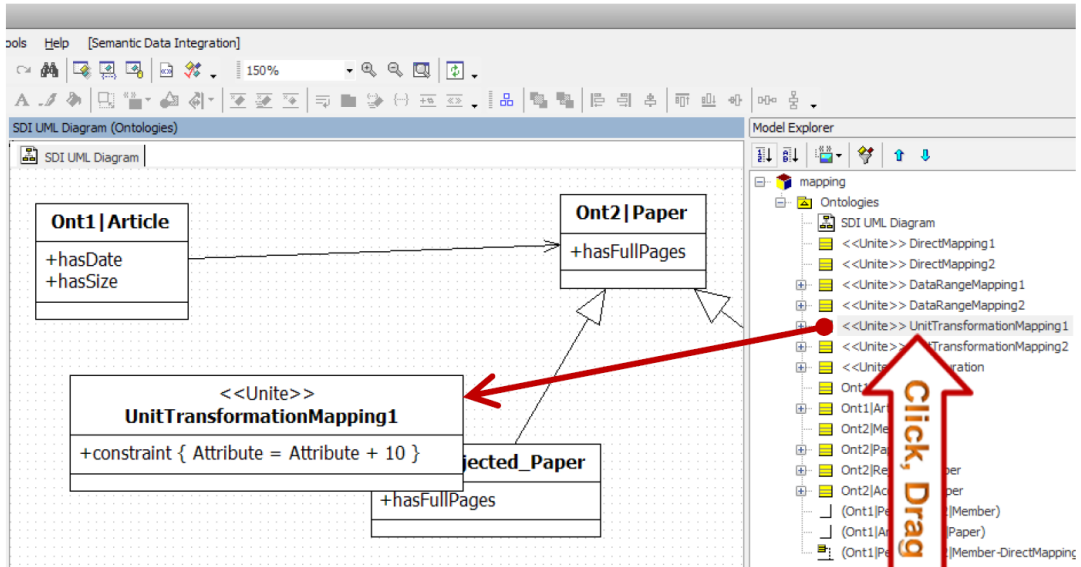
1. **Create UML Association between source and target UML Classes**
 - Select UML Association in Toolbox.
 - Click source UML class in Diagram View and drag to target UML class.



Now, you can see Directed UML Association Line between Classes.

2. Create <<Unite>> stereotyped UML Class

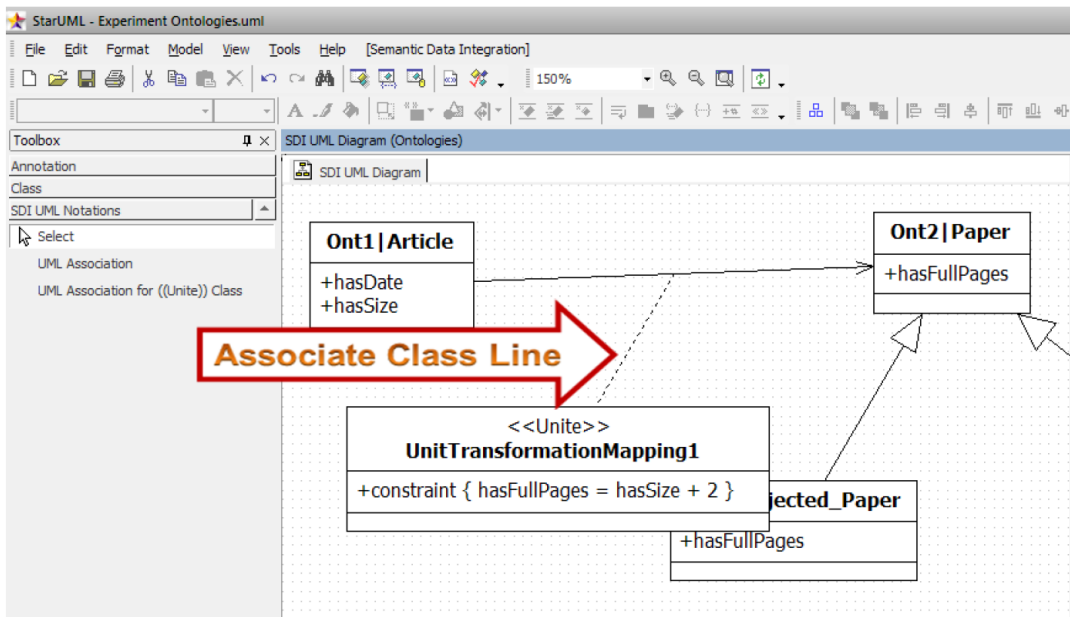
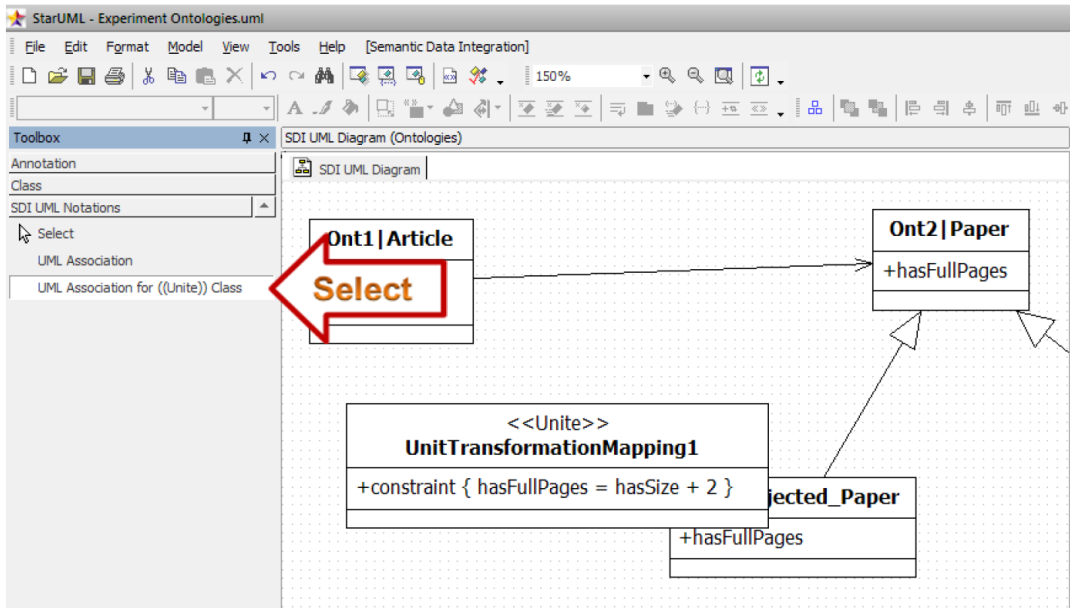
- Select <<Unite>> UnitTransformationMapping1 or 2 in **Model Explorer**.
- Drag and Drop <<Unite>> UnitTransformationMapping1 or 2 in Model Explorer into Diagram View.
- **Double click on the attribute** to change the attribute in <<Unite>> UnitTransformationMapping1 Class.
- Modify default UML constraint which is in between “{” and “}” according to your design of the mapping.



Now, you can see <<Unite>> UnitTransformationMapping1 UML Class with your intended constraint.

3. Create <<Unite>> stereotyped UML Association Class

- Select UML Association for ((Unite)) Class in Toolbox.
- Click **UML Association Line** in Diagram View and drag to <<Unite>> stereotyped UML class.

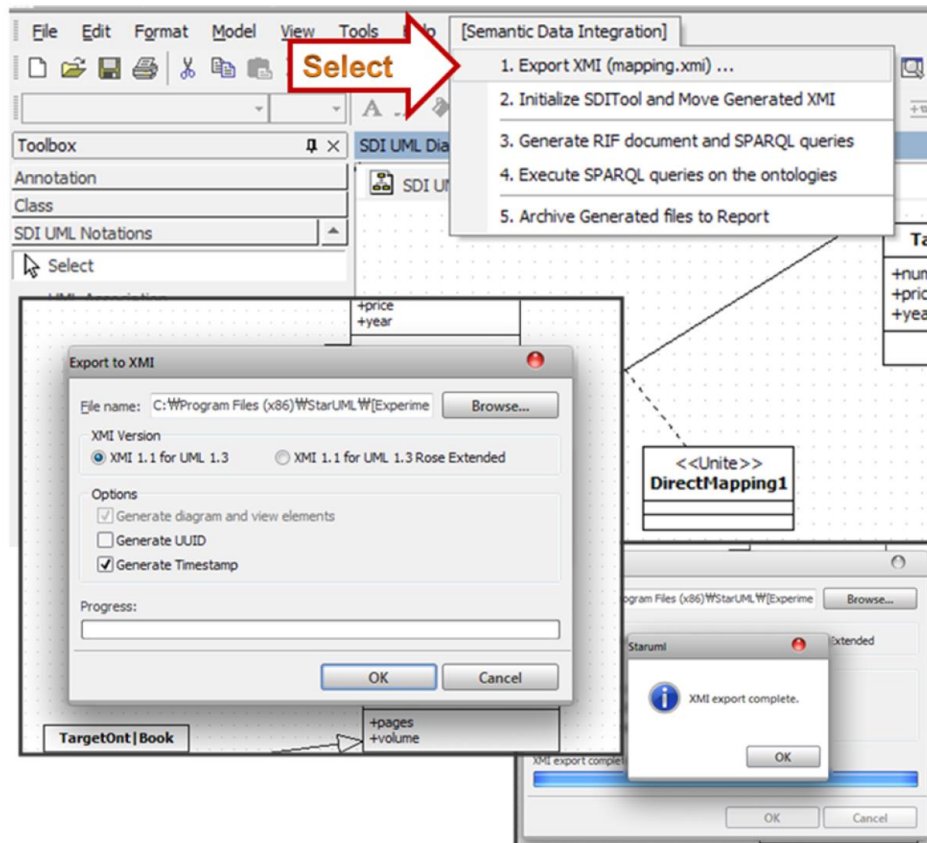


Now, you can see UML Association Class Line between UML Association and <<Unite>> stereotyped UML Class.

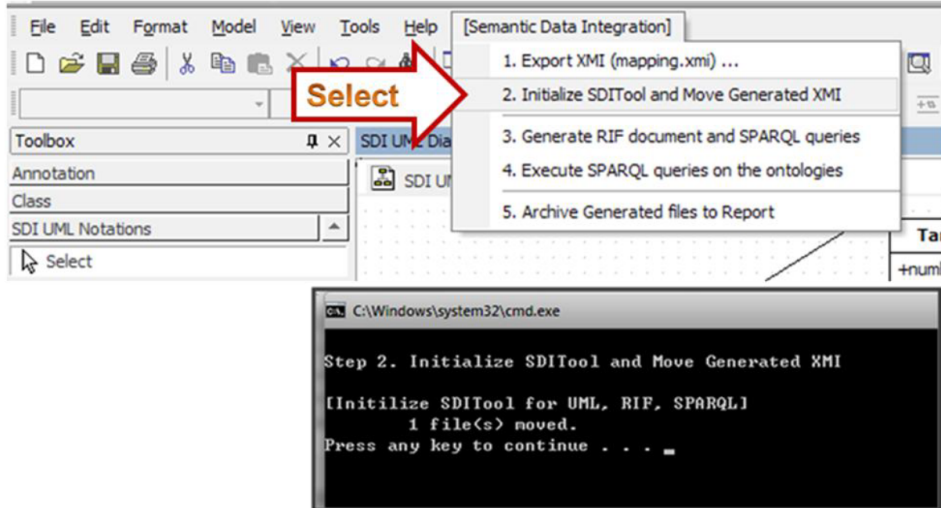
Appendix B.4 Example of how to generate mapping outputs

4. Example of how to generate Mapping Outputs

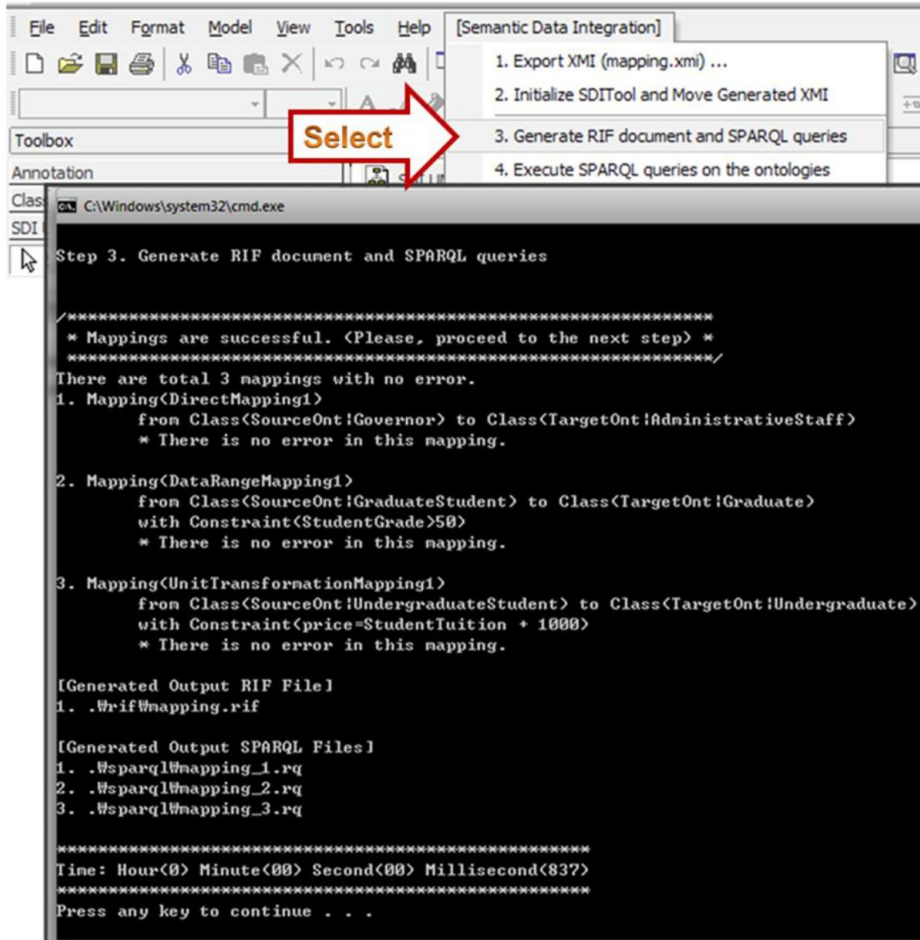
- **Export XMI**
 - Select **[Semantic Data Integration]** in the Menu.
 - Click **"1. Export XMI (mapping.xmi) ..."** in the Menu.
 - Click OK when Export to XMI popup window appears.
 - Click OK when XMI export complete window appears.



- **Initialize SDITool and Move Generated XMI**
 - Select [Semantic Data Integration] in the Menu.
 - Click **"2. Initialize SDITool and Move Generated XMI"** in the Menu.
 - CMD window appears with 1 file(s) moved information.
 - Click any key to make the window disappear.

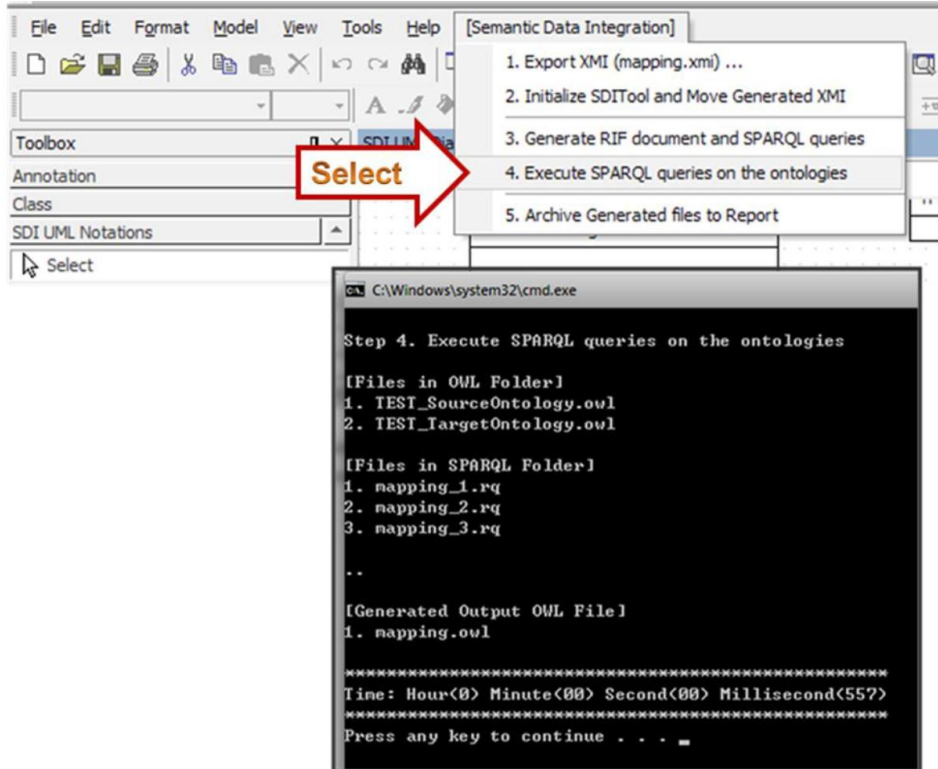


- **Generate RIF document and SPARQL queries**
 - Select [Semantic Data Integration] in the Menu.
 - Click “3. Generate RIF Document and SPARQL queries” in the Menu.
 - CMD window appears with the mappings information.
 - Click any key to make the window disappear.



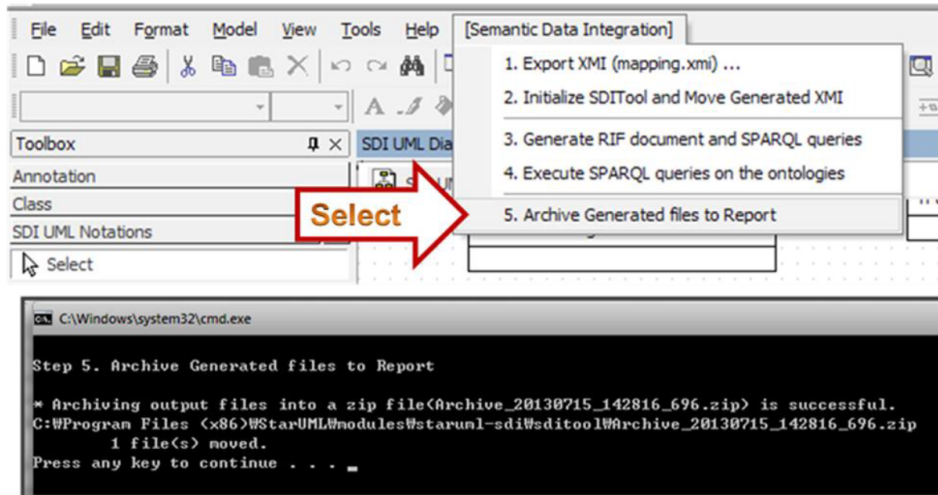
*If * Mappings are not successful, please check the mappings according to the error and re-generate mapping outputs from the Export XMI.*

- **Execute SPARQL queries on the ontologies**
 - Select [Semantic Data Integration] in the Menu.
 - Click **“4. Execute SPARQL queries on the ontologies”** in the Menu.
 - CMD window appears with the output information.
 - Click any key to make the window disappear.



- **Archive Generated Files**

- Select [Semantic Data Integration] in the Menu.
- Click **"5. Archive Generated files to Report"** in the Menu.
- CMD window appears with the archived file information.
- Click any key to make the window disappear.



- ✓ **Thank you.**
The mapping experiment on SDI Tool is finished. (Please answer the questionnaire.)

APPENDIX C SDI tool java archive file usage in command line mode

Appendix C.1 Usage of “sditool.jar” in command line mode

1. “java -jar sditool.jar -I”

- Initialize the tool by creating/deleting the folders for UML, RIF, SPARQL.

2. “java -jar sditool.jar” (Without any parameter)

- Generate RIF document and SPARQL queries from UML file.

3. “java -jar sditool.jar -S”

- Compute SPARQL queries on merged OWL file and generate result OWL file.

4. “java -jar sditool.jar -A”

- Archive the output files into a zip file with the unique name.

5. “-L” or “-D” parameter can be used in any of above execution.

“-L” parameter creates log files

“-D” parameter shows log only in console.

e.g. “java -jar sditool.jar -S -D”

- This will compute SPARQL queries and the process will be shown in console.

Appendix C.2 Composition of the sditool.jar file

root folder - There is an executable “sditool.jar” file of the tool and a sample outputs archived file.

lib folder - This folder has library (jar) files used in the tool. There are 10 external libraries: apache-jena-core-2.10.0.jar, apache-jena-arq-2.10.0.jar, apache-jena-iri-0.9.5.jar, commons-codec-1.6.jar, log4j-1.2.16.jar, slf4j-api-1.6.4.jar, slf4j-log4j12-1.6.4.jar, xercesImpl-2.10.0.jar, xml-apis-1.4.01.jar, commons-compress-1.5.jar.

bin folder - This folder has compiled class files.

src folder - This folder has source code files.

uml folder - This folder has “mapping.xml” file that is in serialized XMI format of UML representation.

owl folder - This folder has source and target ontology files.

[Model] folder - This folder has UML diagrams of the data structure.

APPENDIX D Mapping design worksheets

Appendix D.1 Mapping design worksheets for direct mapping type

Mapping Design Worksheet for Direct Mapping Type

Participant ID: _____

Instruction:

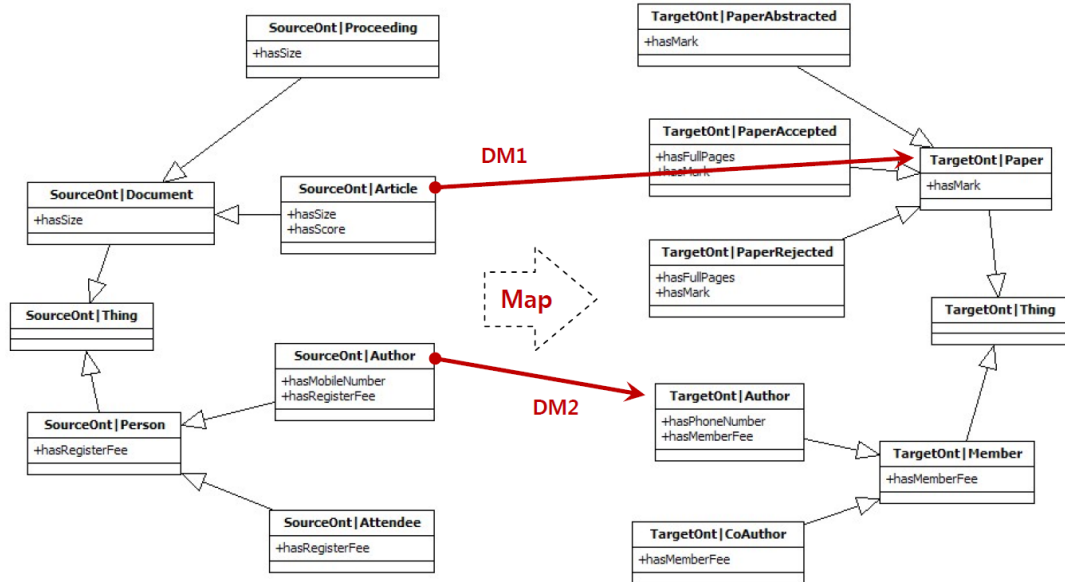
- 1) Please review "Example Mapping Design for Direct Mapping Type".
- 2) Please describe briefly two direct mappings on UML representation(3).
- 3) Please describe detail intention of the mapping in section (4).

Sections:

(1) Mapping Design Worksheet for Direct Mapping Type Instruction	(2) Example Mapping Design for Direct Mapping Type
(3) UML Representation of the conference ontologies	(4) Description of Mapping Intention for Direct Mapping - ID 1 & 2

(2) Example Mapping Design for Direct Mapping Type

This mapping type represents the expression for one ontology class (A) corresponding to another ontology class (B) with the integration property corresponding to “Unite the instances of class (A) to class (B)”.



Direct Mapping 1

Please explain your intention of the mapping.

Instances in [**SourceOnt|Article**] class need to be mapped to [**TargetOnt|Paper**] class.

because **Article can be subclass of Paper.**

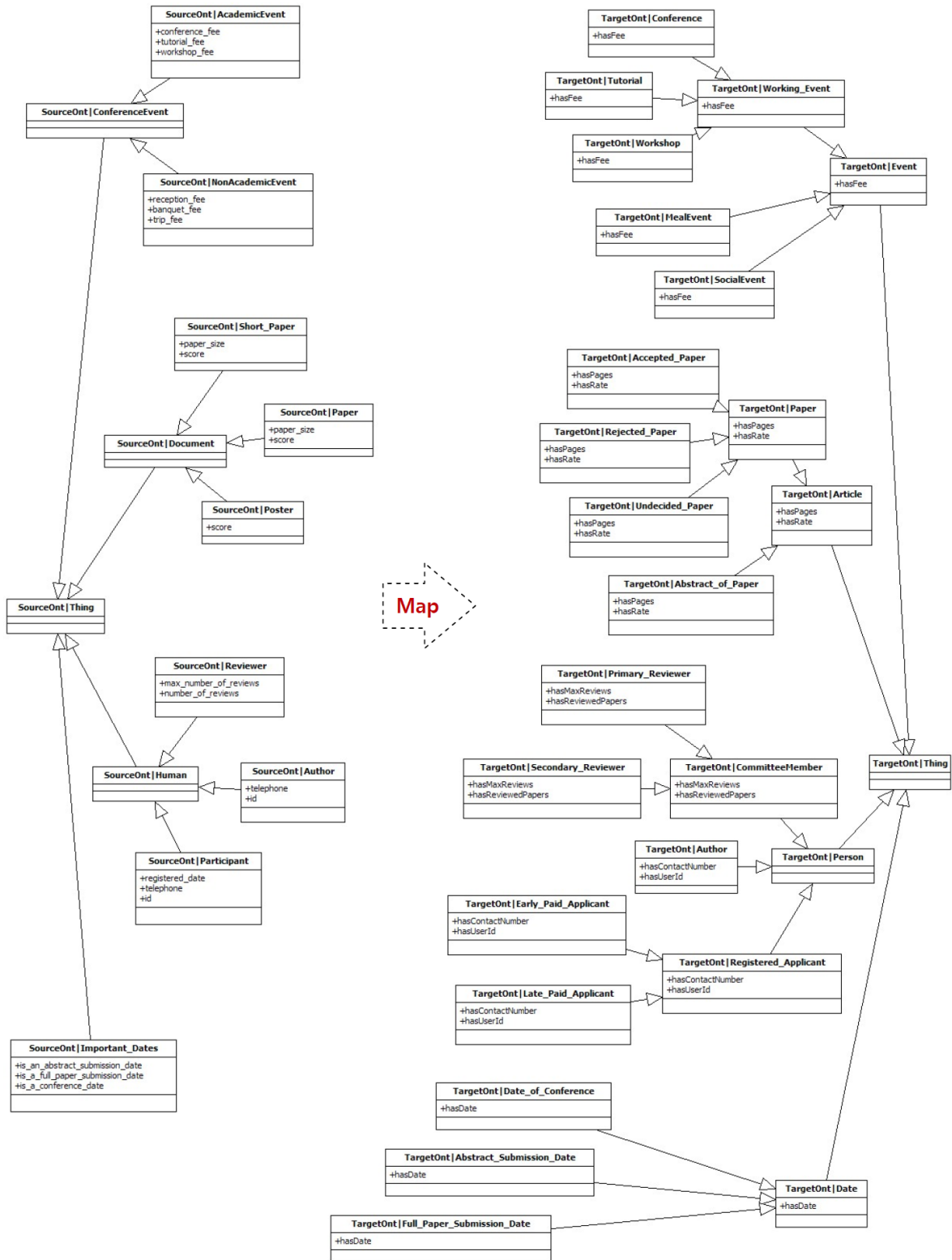
Direct Mapping 2

Please explain your intention of the mapping.

Instances in [**SourceOnt|Author**] class need to be mapped to [**TargetOnt|Author**] class.

because **Both Author can be considered same.**

(3) UML Representation of the conference ontologies



(4) Description of Mapping Intention for Direct Mapping - ID 1 & 2

For the conference ontologies:

You can make any assumption on the elements (please describe) and assume all the attributes are numeric types. The followings are format examples in detail for each attribute:

1. **conference_fee, tutorial_fee, workshop_fee**: #.## (Upto 2 digits below point, e.g. 123.45, Currency is not yet decided.)
2. **reception_fee, banquet_fee, trip_fee, hasFee**: #.## (Upto 2 digits below point, e.g. 1234.56, Currency is not yet decided.)
3. **paper_size, hasPages**: ## (Decimal Range from 1 to 40, e.g. 20)
4. **score**: ## (Decimal Range from 0 to 10, e.g. 5)
5. **hasRate**: ### (Decimal Range from 0 to 100, e.g. 50)
6. **max_number_of_reviews, hasMaxReviews**: ## (Decimal Range from 0 to 20, e.g. 10)
7. **number_of_reviews, hasReviewedPapers**: ## (Decimal Range from 0 to 20, e.g. 10)
8. **telephone, hasContactNumber**: ##### (10 Decimal Digits, e.g. 1234567890)
9. **id, hasUserId**: ##### (8 Decimal Digits, e.g. 12345678)
10. **is_an_abstract_submission_date, is_a_full_paper_submission_date**: YYYYDDMM (e.g. 20130731)
11. **is_a_conference_date, registered_date, hasDate**: YYYYDDMM (e.g. 20130801)

Direct Mapping 1

Please explain your intention of the mapping.

Instances in [] class need to be integrated into [] class.

Because

Direct Mapping 2

Please explain your intention of the mapping.

Instances in [] class need to be integrated into [] class.

Because

Participant ID: _____

Mapping Design Worksheet for Data Range Mapping Type

Instruction:

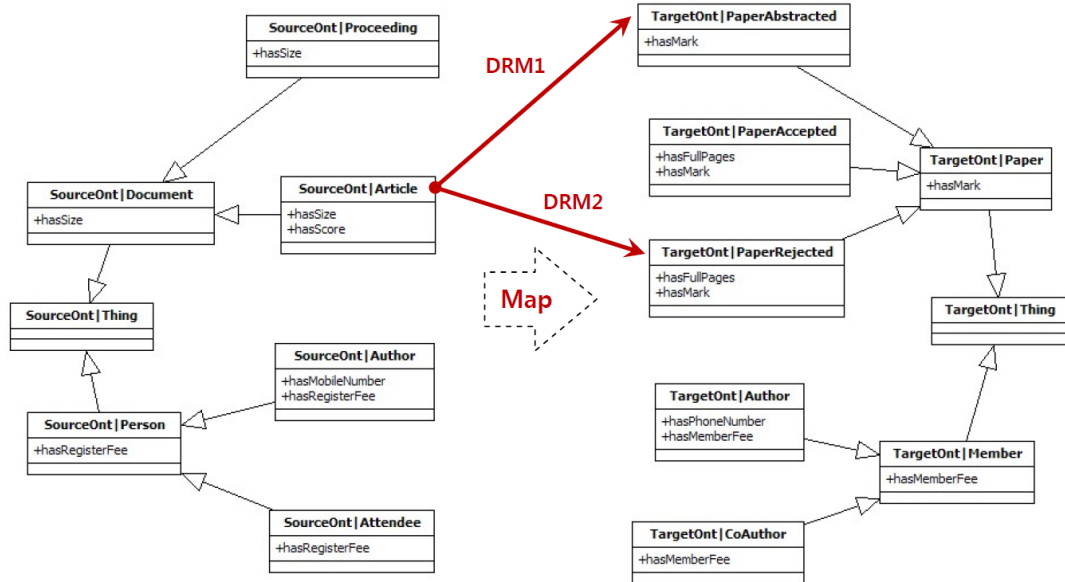
- 1) Please review "Example Mapping Design for Data Range Mapping Type".
- 2) Please describe briefly two data range mappings on UML representation(3).
- 3) Please describe detail intention of the mapping in section (4).

Sections:

<p>(1)</p> <p>Mapping Design Worksheet for Data Range Mapping Type Instruction</p>	<p>(2)</p> <p>Example Mapping Design for Data Range Mapping Type</p>
<p>(3)</p> <p>UML Representation of the conference ontologies</p>	<p>(4)</p> <p>Description of Mapping Intention for Data Range Mapping - ID 1 & 2</p>

(2) Example Mapping Design for Data Range Mapping Type

This mapping type represents the expression for an ontology class (A), that is constrained by the data range of the datatype properties, corresponding to another ontology class (B) with the integration property corresponding to “Unite the instances of class (A) to class (B) if the instance of class (A) satisfies the data range constraint”.



Data Range Mapping 1

Please explain your intention of the mapping.

Instances in [**SourceOnt|Article**] class need to be integrated into [**TargetOnt|PaperAbstracted**] class

with the constraint { **hasSize** <= 2 }

because **Articles, that have less than or equal to 2 pages, are an abstracted paper.**

Data Range Mapping 2

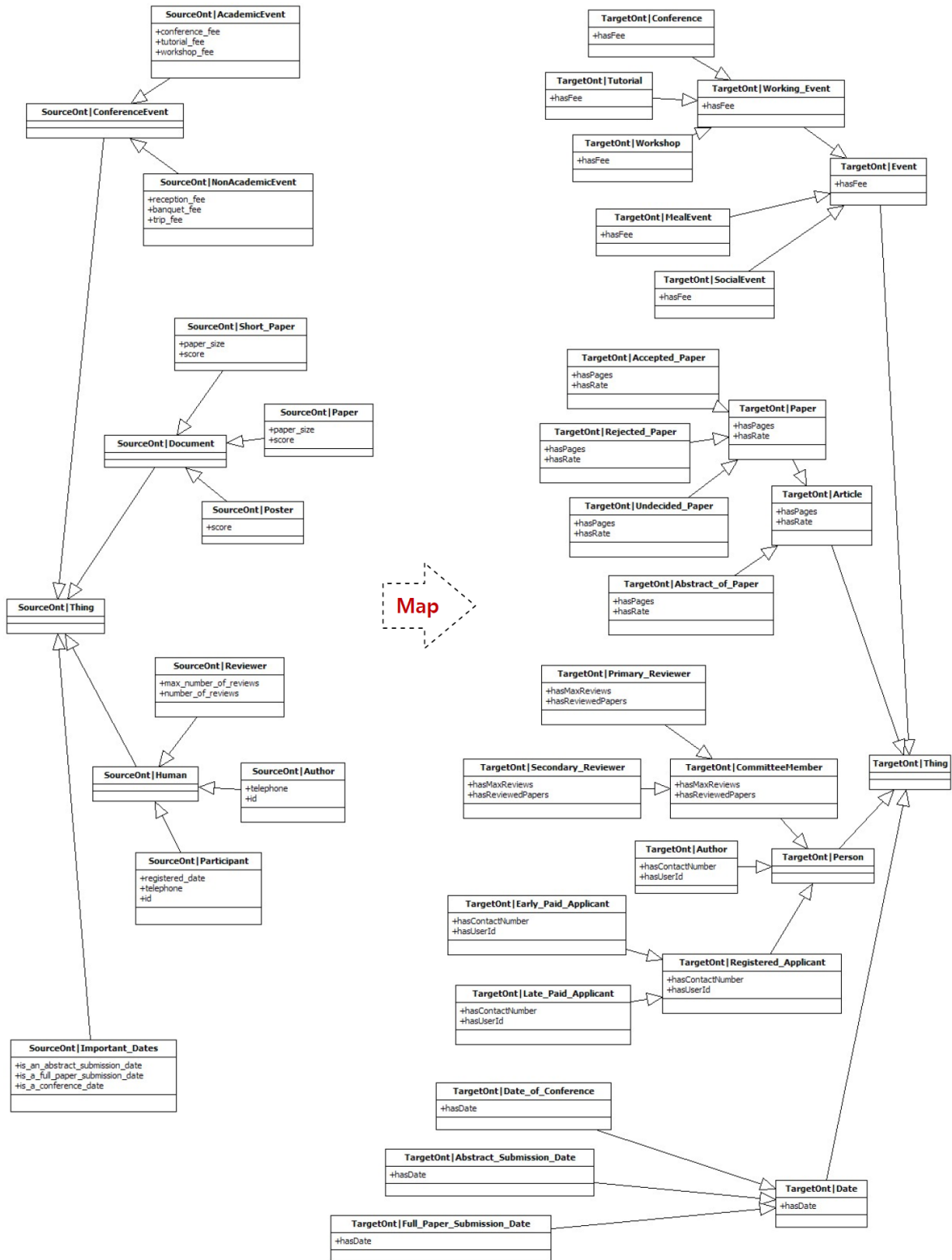
Please explain your intention of the mapping.

Instances in [**SourceOnt|Article**] class need to be integrated into [**TargetOnt|PaperRejected**] class

with the constraint { **hasScore** < 5 }

because **Articles, that have less than 5 score, are a rejected paper.**

(3) UML Representation of the conference ontologies



(4) Description of Mapping Intention for Data Range Mapping - ID 1 & 2

For the conference ontologies:

You can make any assumption on the elements (please describe) and assume all the attributes are numeric types. The followings are format examples in detail for each attribute:

1. **conference_fee, tutorial_fee, workshop_fee**: #.## (Upto 2 digits below point, e.g. 123.45, Currency is not yet decided.)
2. **reception_fee, banquet_fee, trip_fee, hasFee**: #.## (Upto 2 digits below point, e.g. 1234.56, Currency is not yet decided.)
3. **paper_size, hasPages**: ## (Decimal Range from 1 to 40, e.g. 20)
4. **score**: ## (Decimal Range from 0 to 10, e.g. 5)
5. **hasRate**: ### (Decimal Range from 0 to 100, e.g. 50)
6. **max_number_of_reviews, hasMaxReviews**: ## (Decimal Range from 0 to 20, e.g. 10)
7. **number_of_reviews, hasReviewedPapers**: ## (Decimal Range from 0 to 20, e.g. 10)
8. **telephone, hasContactNumber**: ##### (10 Decimal Digits, e.g. 1234567890)
9. **id, hasUserId**: ##### (8 Decimal Digits, e.g. 12345678)
10. **is_an_abstract_submission_date, is_a_full_paper_submission_date**: YYYYDDMM (e.g. 20130731)
11. **is_a_conference_date, registered_date, hasDate**: YYYYDDMM (e.g. 20130801)

Data Range Mapping 1

Please explain your intention of the mapping.

Instances in [] class need to be integrated into [] class.

with the constraint { }

because

Data Range Mapping 2

Please explain your intention of the mapping.

Instances in [] class need to be integrated into [] class.

with the constraint { }

because

Appendix D.3 Mapping design worksheets for unit transformation mapping type

Mapping Design Worksheet for Unit Transformation Mapping Type

Participant ID: _____

Instruction:

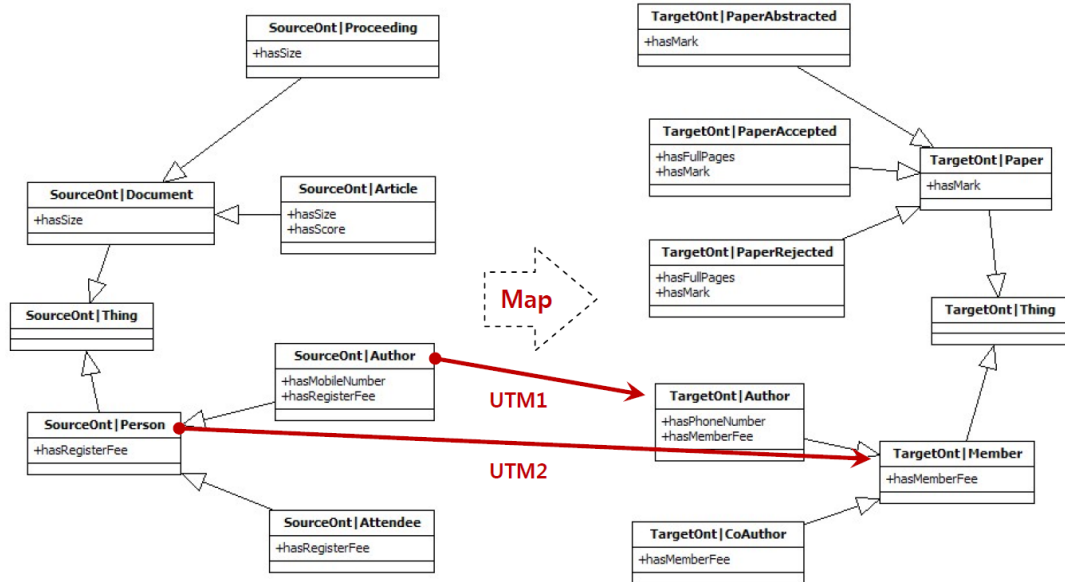
- 1) Please review "Example Mapping Design for Unit Transformation Mapping Type".
- 2) Please describe briefly two unit transformation mappings on UML representation(3).
- 3) Please describe detail intention of the mapping in section (4).

Sections:

<p>(1)</p> <p>Mapping Design Worksheet for Unit Transformation Mapping Type Instruction</p>	<p>(2)</p> <p>Example Mapping Design for Unit Transformation Mapping Type</p>
<p>(3)</p> <p>UML Representation of the conference ontologies</p>	<p>(4)</p> <p>Description of Mapping Intention for Unit Transformation Mapping - ID 1 & 2</p>

(2) Example Mapping Design for Unit Transformation Mapping Type

This mapping type represents the expression for an ontology class (A), that is constrained by the arithmetic transformation of the data value of the datatype properties, corresponding to another ontology class (B) with the integration property corresponding to “Unite the instances of class (A) to class (B) with the unit transformation according the constraint”.



Unit Transformation Mapping 1

Please explain your intention of the mapping.

Instances in [**SourceOnt|Person**] class need to be integrated into [**TargetOnt|Member**] class

with the constraint { **hasMemberFee = hasRegisterFee * 1.12** }

because **There is currency difference between RegisterFee and MemberFee.**

Unit Transformation Mapping 2

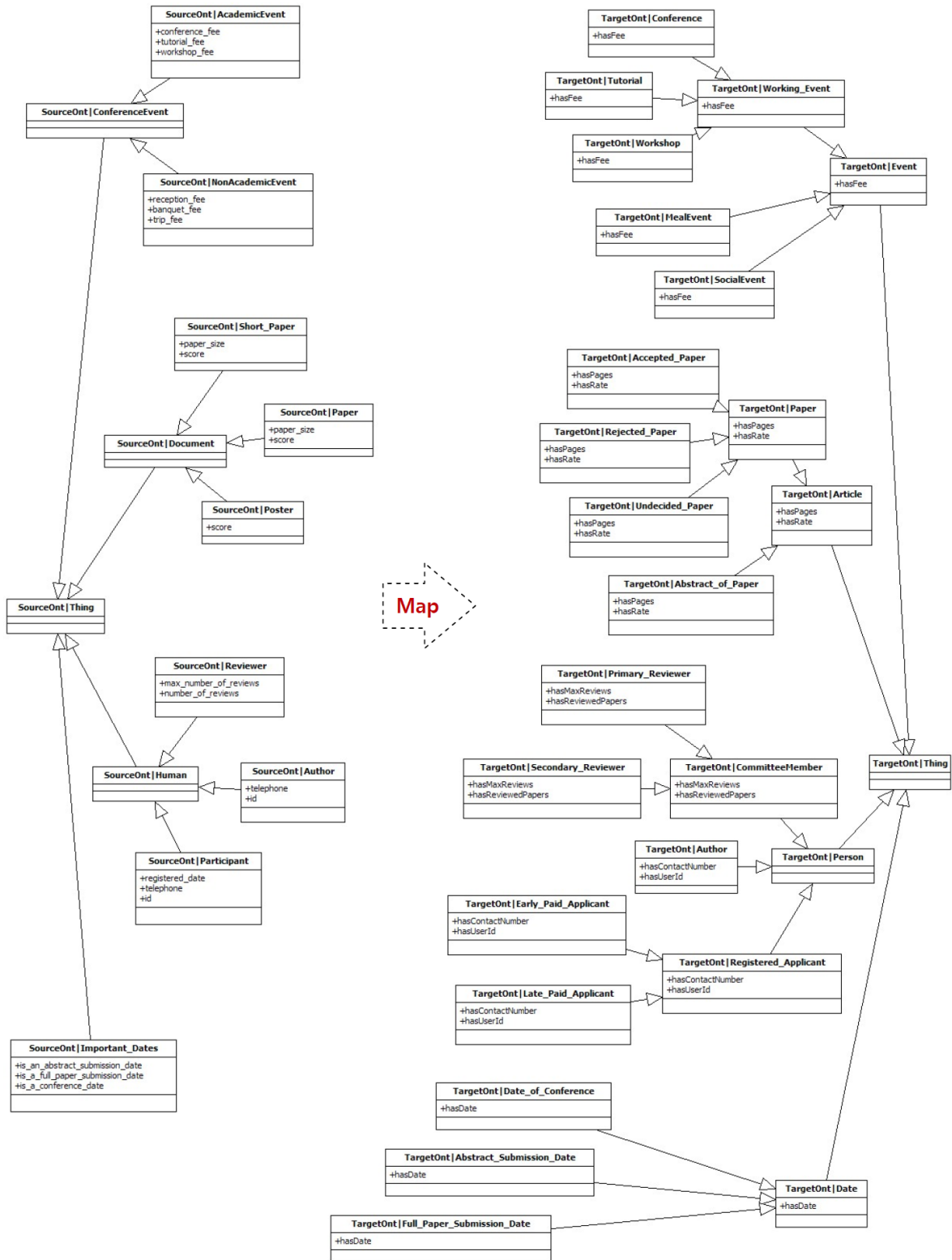
Please explain your intention of the mapping.

Instances in [**SourceOnt|Author**] class need to be integrated into [**TargetOnt|Author**] class

with the constraint { **hasPhoneNumber = hasMobileNumber** }

because **PhoneNumber value needs to be same as the value of MobileNumber.**

(3) UML Representation of the conference ontologies



(4) Description of Mapping Intention for Unit Transformation Mapping - ID 1 & 2

For the conference ontologies:

You can make any assumption on the elements (please describe) and assume all the attributes are numeric types. The followings are format examples in detail for each attribute:

1. **conference_fee, tutorial_fee, workshop_fee**: #.## (Upto 2 digits below point, e.g. 123.45, Currency is not yet decided.)
2. **reception_fee, banquet_fee, trip_fee, hasFee**: #.## (Upto 2 digits below point, e.g. 1234.56, Currency is not yet decided.)
3. **paper_size, hasPages**: ## (Decimal Range from 1 to 40, e.g. 20)
4. **score**: ## (Decimal Range from 0 to 10, e.g. 5)
5. **hasRate**: ### (Decimal Range from 0 to 100, e.g. 50)
6. **max_number_of_reviews, hasMaxReviews**: ## (Decimal Range from 0 to 20, e.g. 10)
7. **number_of_reviews, hasReviewedPapers**: ## (Decimal Range from 0 to 20, e.g. 10)
8. **telephone, hasContactNumber**: ##### (10 Decimal Digits, e.g. 1234567890)
9. **id, hasUserId**: ##### (8 Decimal Digits, e.g. 12345678)
10. **is_an_abstract_submission_date, is_a_full_paper_submission_date**: YYYYDDMM (e.g. 20130731)
11. **is_a_conference_date, registered_date, hasDate**: YYYYDDMM (e.g. 20130801)

Unit Transformation Mapping 1

Please explain your intention of the mapping.

Instances in [] class need to be integrated into [] class.

with the constraint { }

because

Unit Transformation Mapping 2

Please explain your intention of the mapping.

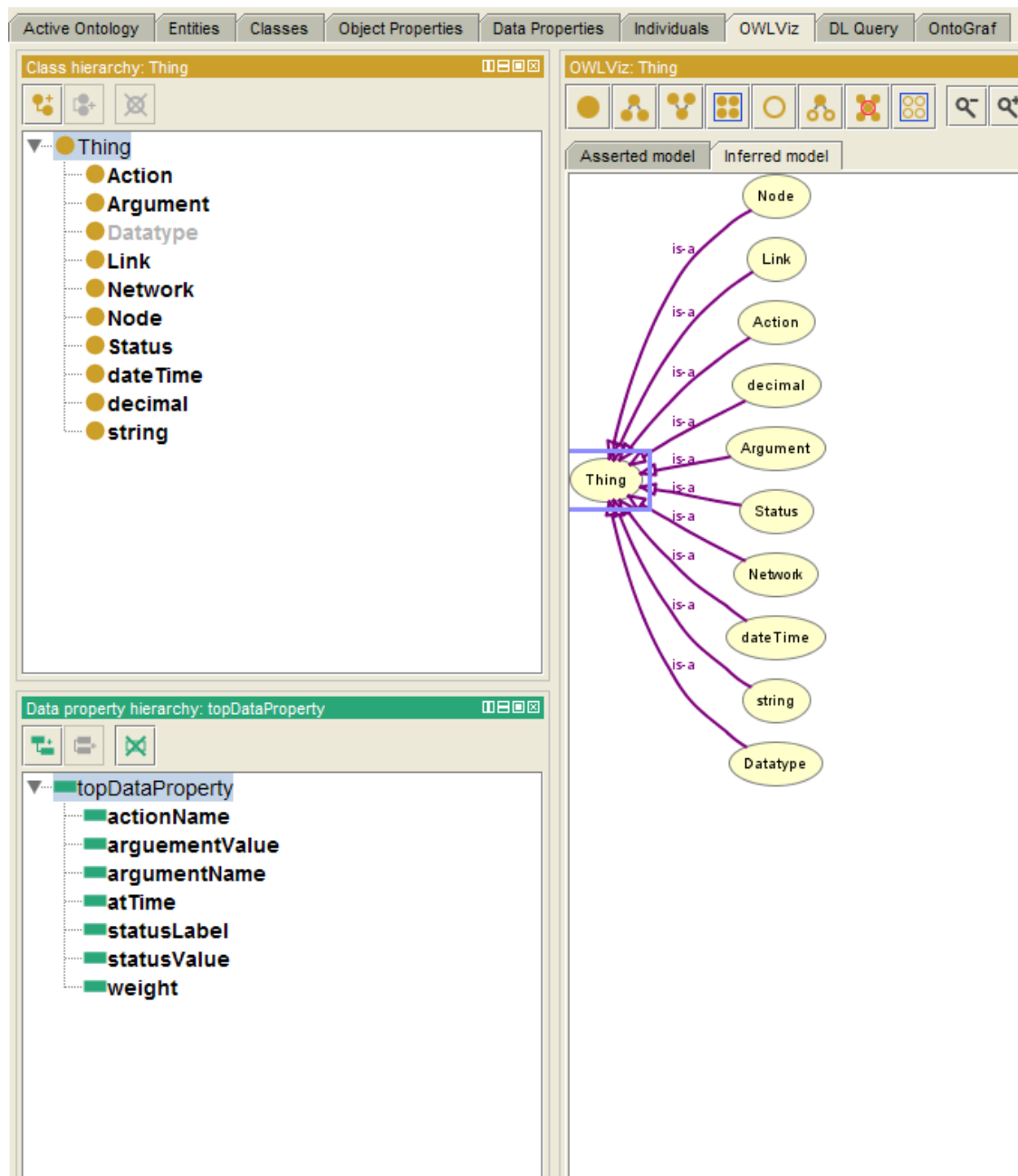
Instances in [] class need to be integrated into [] class.

with the constraint { }

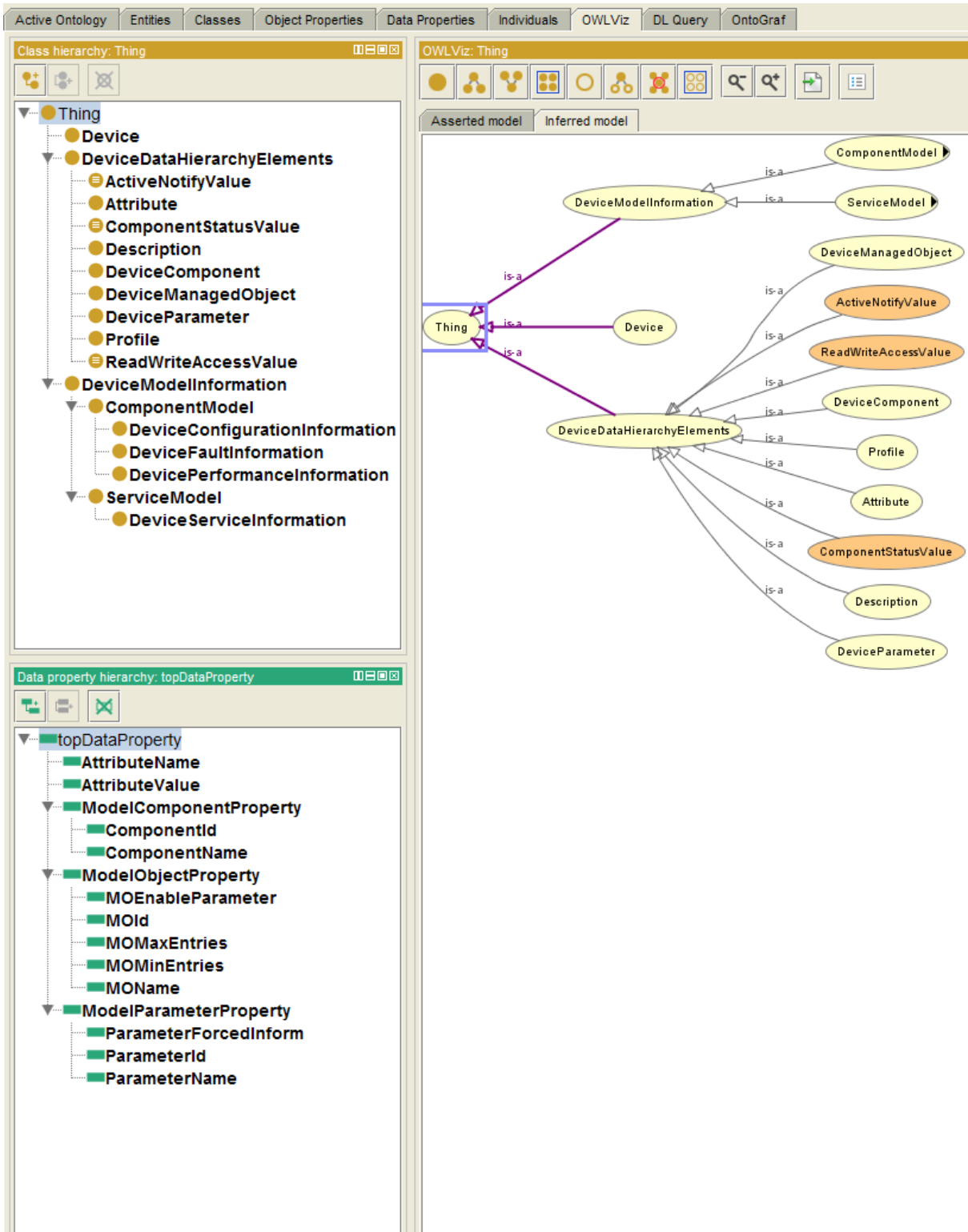
because

APPENDIX E OWL view of initial experiment ontologies in Protégé tool

Appendix E.1 NC ontology



Appendix E.2 TR069 ontology

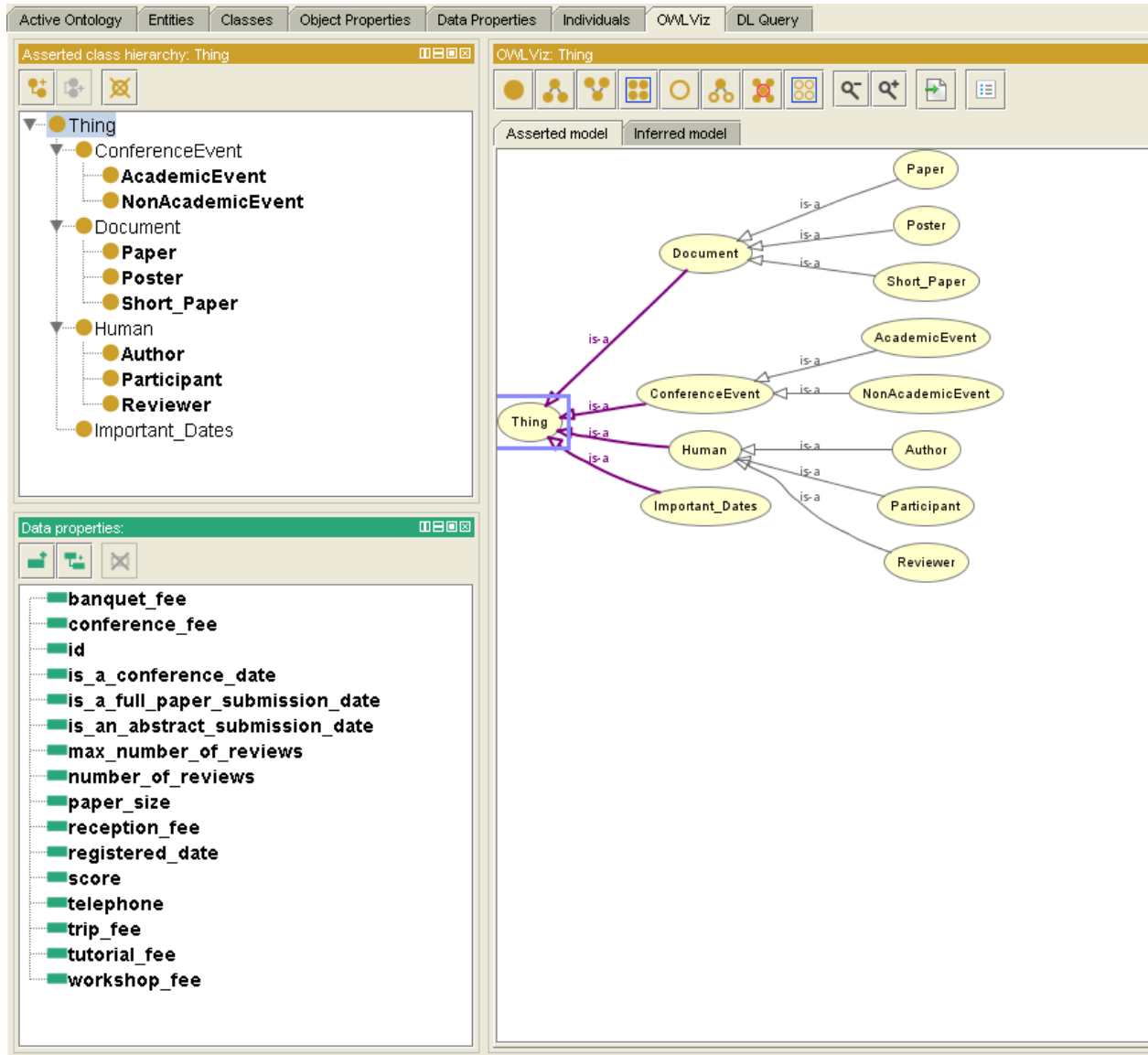


Appendix E.3 Femto ontology

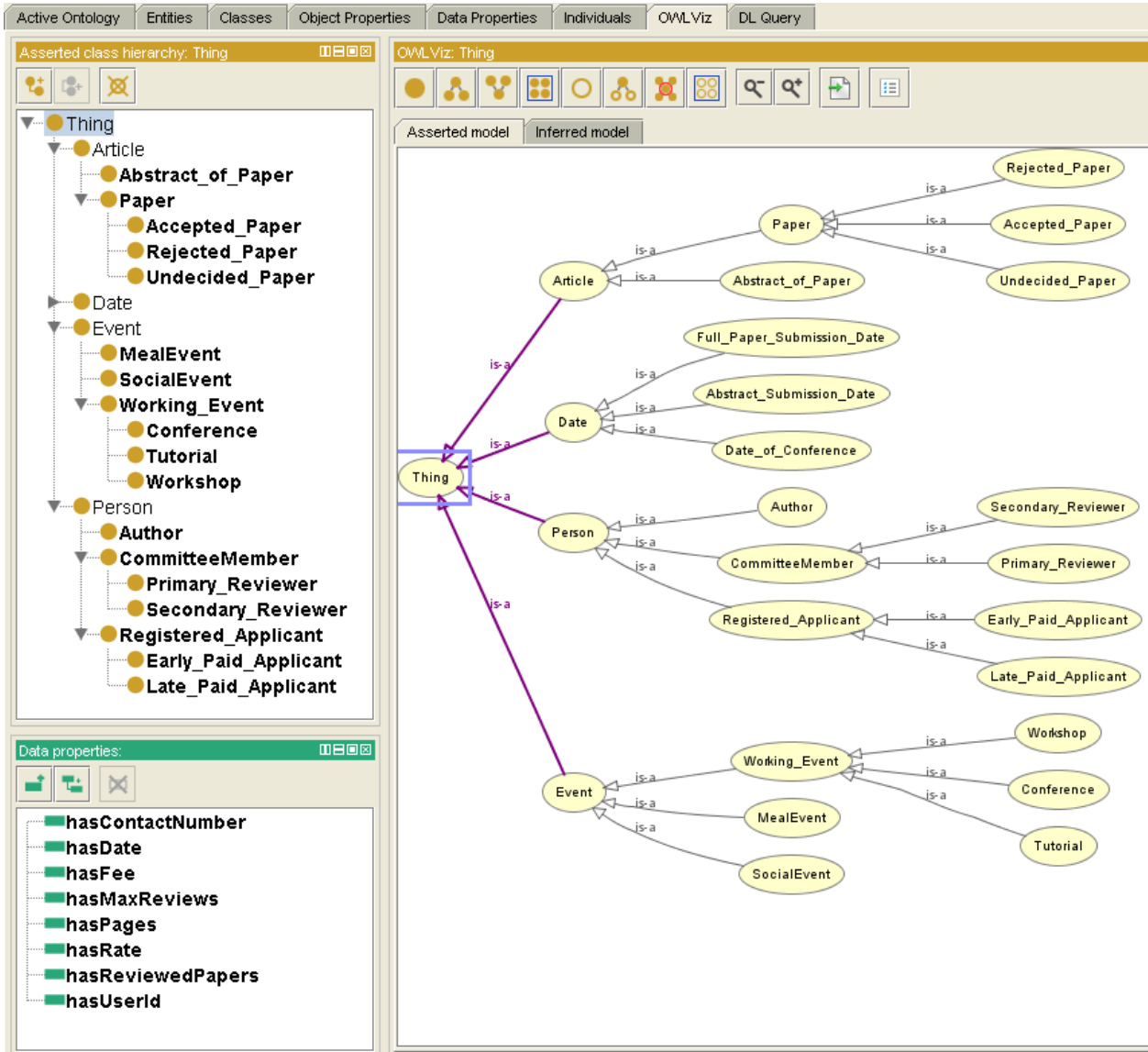


APPENDIX F OWL view of experiment ontologies in Protégé tool

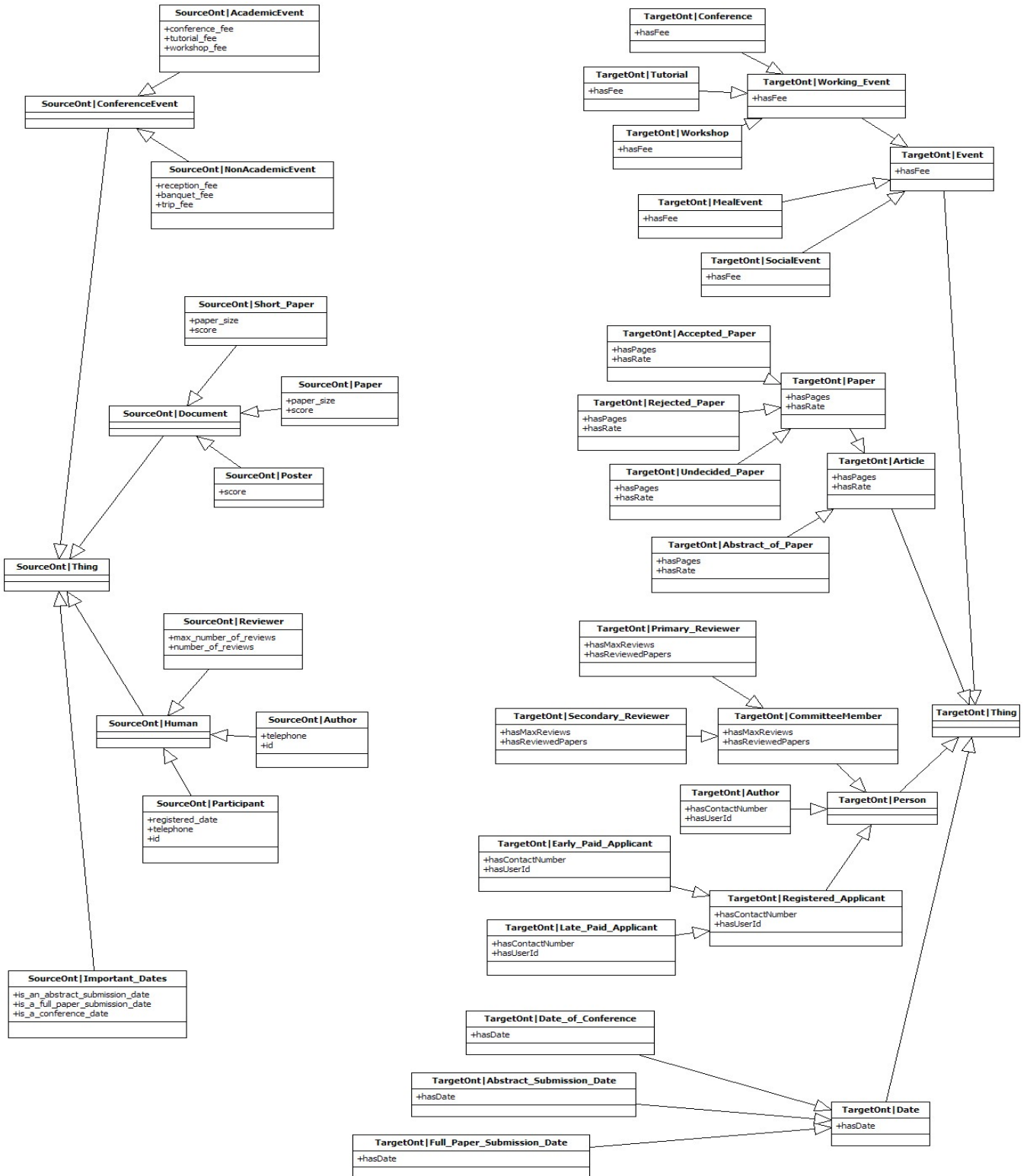
Appendix F.1 Source conference ontology



Appendix F.2 Target conference ontology



APPENDIX G Potential mapping candidates in experiment ontologies



There are 46 potential mapping candidates in the experiment ontologies.

Note: There can be more mapping candidates depending on the assumption made by experiment participants.

Appendix G.1 Potential mapping candidates in direct mapping type

<Source Ontology> - <Target Ontology>

1. ConferenceEvent - Event
2. AcademicEvent - Working_Event
3. NonAcademicEvent - SocialEvent
4. Document - Article
5. Short_Paper - Paper
6. Paper - Paper
7. Human - Person
8. Author - Author
9. Reviewer - CommitteeMember
10. Participant - Registered_Applicant
11. Important_Dates - Date

Appendix G.2 Potential mapping candidates in data range mapping type

<Source Ontology> - <Target Ontology> { <Constraint> }

1. AcademicEvent - Conference { conference_fee > 0 }
2. AcademicEvent - Tutorial { tutorial_fee > 0 }
3. AcademicEvent - Workshop { workshop_fee > 0 }
4. NonAcademicEvent - MealEvent { reception_fee > 0 || banquet_fee > 0 }
5. NonAcademicEvent - SocialEvent { trip_fee > 0 }
6. Paper - Abstract_of_Paper { paper_size <= 4 }
7. Paper - Accepted_Paper { score > 5 }
8. Paper - Rejected_Paper { score < 5 }
9. Paper - Undecided_Paper { score == 5 }
10. Short_Paper - Abstract_of_Paper { paper_size < 2 }
11. Short_Paper - Accepted_Paper { score > 5 }
12. Short_Paper - Rejected_Paper { score < 5 }
13. Short_Paper - Undecided_Paper { score == 5 }
14. Reviewer - Primary_Reviewer { number_of_reviews >= 10 }
15. Reviewer - Secondary_Reviewer { number_of_reviews < 10 }
16. Reviewer - Primary_Reviewer { max_number_of_reviews >= 15 }
17. Reviewer - Secondary_Reviewer { max_number_of_reviews < 15 }
18. Participant - Early_Paid_Applicant { registered_date < 20130701 }
19. Participant - Late_Paid_Applicant { registered_date > 20130831 }

Appendix G.3 Potential mapping candidates in unit transformation mapping type

<Source Ontology> - <Target Ontology> { <Constraint> }

1. AcademicEvent - Working_Event { hasFee = (conference_fee + tutorial_fee + workshop_fee) * 1.2 }
2. NonAcademicEvent - MealEvent { hasFee = (reception_fee + banquet_fee) * 1.2 }
3. NonAcademicEvent - SocialEvent { hasFee = trip_fee * 1.2 }
4. Short_Paper - Paper { hasRate = score * 10 }
5. Paper - Paper { hasRate = score * 10 }
6. Short_Paper - Paper { hasPages = paper_size + 2 }
7. Paper - Paper { hasPages = paper_size + 2 }
8. Reviewer - CommitteeMember { hasMaxReviews = max_number_of_reviews + 5 }
9. Reviewer - CommitteeMember { hasReviewedPapers = number_of_reviews }
10. Author - Author { hasContactNumber = telephone }
11. Author - Author { hasUserId = id * 1000 + telephone }
12. Participant - Registered_Applicant { hasContactNumber = telephone }
13. Participant - Registered_Applicant { hasUserId = id * 1000 + telephone }
14. Important_Dates - Date_of_Conference { hasDate = is_a_conference_date }
15. Important_Dates - Abstract_Submission_Date { hasDate = is_an_abstract_submission_date }
16. Important_Dates - Full_Paper_Submission_Date { hasDate = is_a_full_paper_submission_date }

APPENDIX H Brief mapping guide for core mapping types

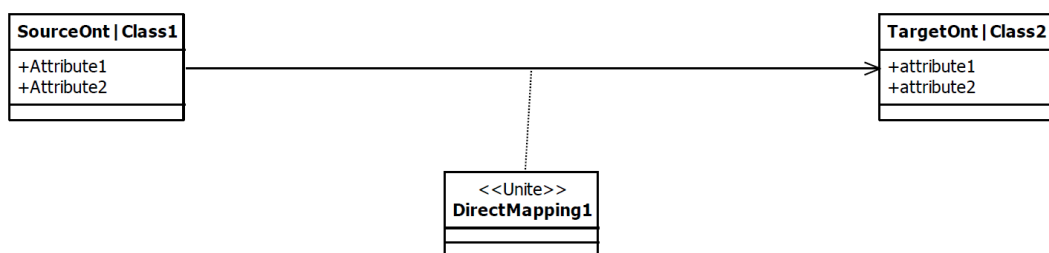
Appendix H.1 Brief mapping guide for direct mapping type

Reference Brief Mapping Guide

1. Direct Mapping Type

- This mapping type represents the expression for one ontology class (A) corresponding to another ontology class (B) with the integration property corresponding to “Unite the instances of class (A) to class (B)”.

- **Example of UML Representation**



- **Above UML Representation can be interpreted as:**
 - The instances of “SourceOnt|Class1” can be mapped to “TargetOnt|Class2”.

Note!

Direct Mapping does not require any constraint.

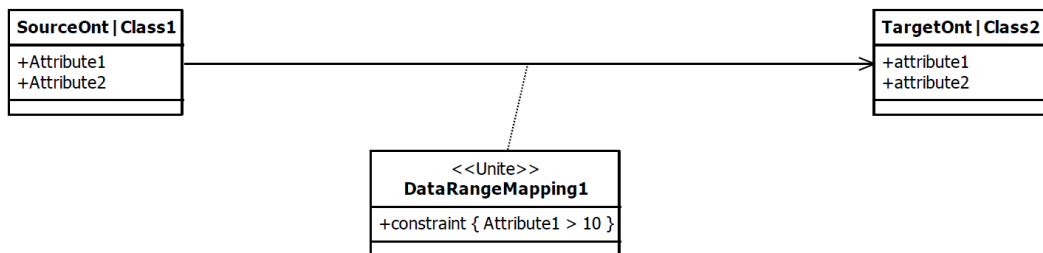
Reference

Brief Mapping Guide

2. Data Range Mapping Type

- This mapping type represents the expression for an ontology class (A), that is constrained by the data range of the datatype properties, corresponding to another ontology class (B) with the integration property corresponding to “Unite the instances of class (A) to class (B) if the instance of class (A) satisfies the data range constraint”.

- **Example of UML Representation**



- **Above UML Representation can be interpreted as:**

- The instances of “SourceOnt|Class1” can be mapped to “TargetOnt|Class2”, if the attribute value of “Attribute1” of the instance is bigger than 10.

Note!

Data Range Mapping requires a constraint expressed with comparison operation.

- eg1. `Attribute1 > 10`
- eg2. `Attribute1 <= 100`
- eg3. `Attribute1 > 10 && Attribute1 < 90`
- eg4. `Attribute1 > 50 || Attribute2 < 50`

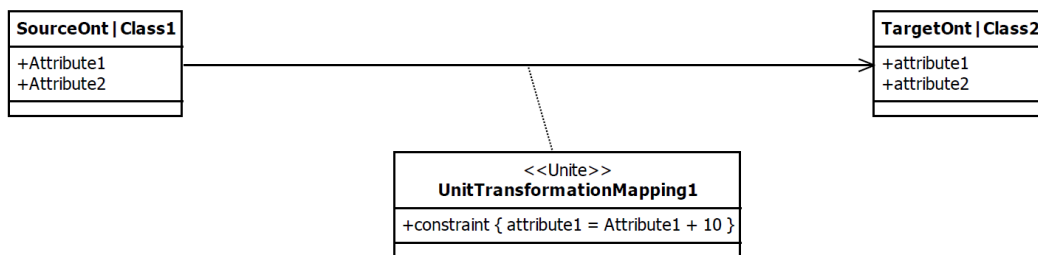
Reference

Brief Mapping Guide

3. Unit Transformation Mapping Type

- This mapping type represents the expression for an ontology class (A), that is constrained by the arithmetic transformation of the data value of the datatype properties, corresponding to another ontology class (B) with the integration property corresponding to "Unite the instances of class (A) to class (B) with the unit transformation according the constraint".

- **Example of UML Representation**



- **Above UML Representation can be interpreted as:**

- The instances of "SourceOnt|Class1" can be mapped to "TargetOnt|Class2" with the constraint that the value of "attribute1" in "TargetOnt|Class2" will have the value of "Attribute1" in "SourceOnt|Class1" plus 10.

Note!

Unit Transformation Mapping requires a constraint expressed with arithmetic operation.

- eg1. $attribute1 = Attribute1 + 10$
- eg2. $attribute1 = Attribute1 / 100$
- eg3. $attribute1 = (Attribute1 + Attribute2) / 2$
- eg4. $attribute1 = Attribute1$

APPENDIX I Questionnaire sheets based on the System

Usability Scale

Participant ID: _____

Questionnaire

Instruction: For each of the following statements, mark one box that best describes your reactions to the UML representation and the tool you have just used to design the semantic mappings and to auto-generate semantic mapping documents. Each question is optional. Feel free to omit a response to any question; however the researcher would be grateful if all questions are responded to.

Part 1 / 2:

SUS (System Usability Scale) and Research Specific Questions for the user satisfaction of using UML representation of ontologies to design the semantic mappings.

	Strongly Disagree				Strongly Agree
1. I think that I would like to design the semantic mapping using the UML representation of ontologies frequently.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. I found the semantic mapping on the UML representation of ontologies is unnecessarily complex.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. I thought the semantic mapping on the UML representation of ontologies was easy to design.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. I think that I would need the support of an expert person to be able to design the semantic mapping on the UML representation of ontologies.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. I found UML notations related to the semantic data and mappings in UML class diagram were well integrated.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. I thought there was too much inconsistency to represent semantic data and mappings using UML notations.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. I would imagine that most people would learn to design the semantic mapping using the UML representation of ontologies very quickly.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. I found the UML representation of ontologies very cumbersome to design the semantic mapping.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9. I felt very confident to design the semantic mapping using the UML representation of ontologies.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. I needed to learn a lot of things before I could get going to design the semantic mapping with the UML representation of ontologies.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11. I found Data Range Mapping and Unit Transformation Mapping useful in expressing more precise mapping.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

12. I think UML representation of semantic data and mapping is easy to understand for system engineers.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
---	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

Part 2 / 2:

SUS (System Usability Scale) and Research Specific Questions for the user satisfaction of using the SDI Tool to auto-generate semantic mapping documents (and an integrated ontology).

	Strongly Disagree				Strongly Agree
1. I think that I would like to use the SDI tool frequently.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. I found the SDI tool is unnecessarily complex.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. I thought the SDI tool was easy to use.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. I think that I would need the support of a technical person to be able to use the SDI tool.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. I found the various functions in the SDI tool were well integrated.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. I thought there was too much inconsistency in the SDI tool.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. I would imagine that most people would learn to use the SDI tool very quickly.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. I found the SDI tool very cumbersome to use.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9. I felt very confident using the SDI tool.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. I needed to learn a lot of things before I could get going with the SDI tool.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11. I think the SDI tool requires ontology expert knowledge to conduct semantic data integration.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The experiment is completed. Thank you!

APPENDIX J Experiment overall task sheet

Participant ID: _____

Date: ____/____/____

Task Sheet

1. Brief Explanation about SDI Tool

- 1) Watch overview video

2. Describe Mappings on Mapping Design Worksheets

- 1) You need to describe in total six different mappings:
 - Two mappings of direct mapping type using the mapping design worksheet for direct mapping type
 - Two mappings of data range mapping type using the mapping design worksheet for data range mapping type
 - Two mappings of unit transformation mapping type using the mapping design worksheet for unit transformation mapping type

Total three mapping design worksheets will be given, and please describe two mappings for each mapping design worksheet.

Please indicate start and end time of the mappings' design. (Hour:Minute Format)

Start time:

End time:

3. Watch the video of SDI tool use examples

4. Conduct Semantic Data Integration using SDI Tool

- 1) You need to conduct integration task with the tool according to the description on the mapping design worksheets. You can refer to Brief Mapping Guide and Manual for SDI Tool that describes examples of how to use the tool.

Please indicate start and end time of the integration task on the tool. (Hour:Minute Format)

Start time:

End time:

5. Answer Questionnaire

- 1) Please now fill in the questionnaire.

Thank you!

APPENDIX K Scanned paper result from the experiment

Appendix K.1 Experiment result from participant ID02

02

TRINITY COLLEGE DUBLIN

INFORMED CONSENT FORM

LEAD RESEARCHER: Seung-Hwa Chung

BACKGROUND OF RESEARCH:

Semantic Data Integration is normally conducted by ontology experts, and most existing semantic mapping tools support only one-to-one correspondence mapping and typically require a high level of technical expertise which limits the users of the tools to knowledge engineers or users who have received substantial training.

The experiment will evaluate the usability of a SDI Tool, designed to offer usable semantic data integration for non-ontology expert system engineers. It uses UML class diagrams as a semantic data and mapping representation for the users and then auto-generates semantic mapping documents (and an integrated ontology).

PROCEDURES OF THIS STUDY:

In the experiment, participants will conduct a semantic integration task on the given UML representation of ontologies using the developed SDI Tool. The selected ontology domain, conferences, is general and understandable by most people with a system engineering background. This experiment tries to avoid a domain specific ontology that requires deep understanding of the specific knowledge, so that system engineers in any domain can conduct the experiment and be able to describe the intentions behind their mappings.

The participant will be asked to complete several tasks in the course of the experiment: designing total six different mappings on the mapping design worksheets (two mappings in direct mapping type, two mappings in data range mapping type and two mappings in unit transformation mapping type); conducting semantic data integration using SDI tool according to the designed mappings; and answering a questionnaire. However the participation is voluntary and individual tasks or the experiment as a whole may be abandoned.

In the case of the questionnaire, all participants can omit the entire questionnaire or individual questions if they wish. At the end of the study the lead researcher may contact the participant via email for feedback on the experiment outcomes or vice versa.

There are no particular risks to participating in this experiment beyond those inherent using any computer application.

PUBLICATION:

All the research results will be used within the following venues, and will be presented without any personal identifier of participants. Individual results will be aggregated anonymously and research reported on aggregate results:

1. The researcher's thesis
2. Conference papers

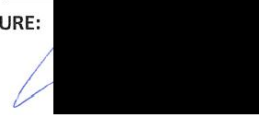
DECLARATION:

- I am 18 years or older and am competent to provide consent.
- I have read, or had read to me, a document providing information about this research and this consent form. I have had the opportunity to ask questions and all my questions have been answered to my satisfaction and understand the description of the research that is being provided to me.
- I agree that my data is used for scientific purposes and I have no objection that my data is published in scientific publications in a way that does not reveal my identity.
- I understand that if I make illicit activities known, these will be reported to appropriate authorities.
- I understand that I may stop electronic recordings at any time, and that I may at any time, even subsequent to my participation have such recordings destroyed (except in situations such as above).
- I understand that, subject to the constraints above, no recordings will be replayed in any public forum or made available to any audience other than the current researchers/research team.
- I freely and voluntarily agree to be part of this research study, though without prejudice to my legal and ethical rights.
- I understand that I may refuse to answer any question and that I may withdraw at any time without penalty.
- I understand that my participation is fully anonymous and that no personal details about me will be recorded.
- I understand that if I or anyone in my family has a history of epilepsy then I am proceeding at my own risk.
- I have received a copy of this agreement.

PARTICIPANT'S NAME:



PARTICIPANT'S SIGNATURE:



Date:

Statement of investigator's responsibility:

I have explained the nature and purpose of this research study, the procedures to be undertaken and any risks that may be involved. I have offered to answer any questions and fully answered such questions. I believe that the participant understands my explanation and has freely given informed consent.

RESEARCHERS CONTACT DETAILS:

Name: Seung-Hwa Chung
Email: chung@tcd.ie

INVESTIGATOR'S SIGNATURE:

Date:

Participant ID: 02

Date: 2013, 9, 13

Task Sheet

1. Brief Explanation about SDI Tool

- 1) Watch overview video

2. Describe Mappings on Mapping Design Worksheets

- 1) You need to describe in total six different mappings:

- Two mappings of direct mapping type using the mapping design worksheet for direct mapping type
- Two mappings of data range mapping type using the mapping design worksheet for data range mapping type
- Two mappings of unit transformation mapping type using the mapping design worksheet for unit transformation mapping type

Total three mapping design worksheets will be given, and please describe two mappings for each mapping design worksheet.

Please indicate start and end time of the mappings' design. (Hour:Minute Format)

Start time: 9:55 am End time: 10:15 am

20 m

3. Watch the video of SDI tool use examples

4. Conduct Semantic Data Integration using SDI Tool

- 1) You need to conduct integration task with the tool according to the description on the mapping design worksheets. You can refer to Brief Mapping Guide and Manual for SDI Tool that describes examples of how to use the tool.

Please indicate start and end time of the integration task on the tool. (Hour:Minute Format)

Start time: 10:20 am End time: 10:28 am

8 m

5. Answer Questionnaire

- 1) Please now fill in the questionnaire.

Thank you!

Mapping Design Worksheet for Direct Mapping Type

Instruction:

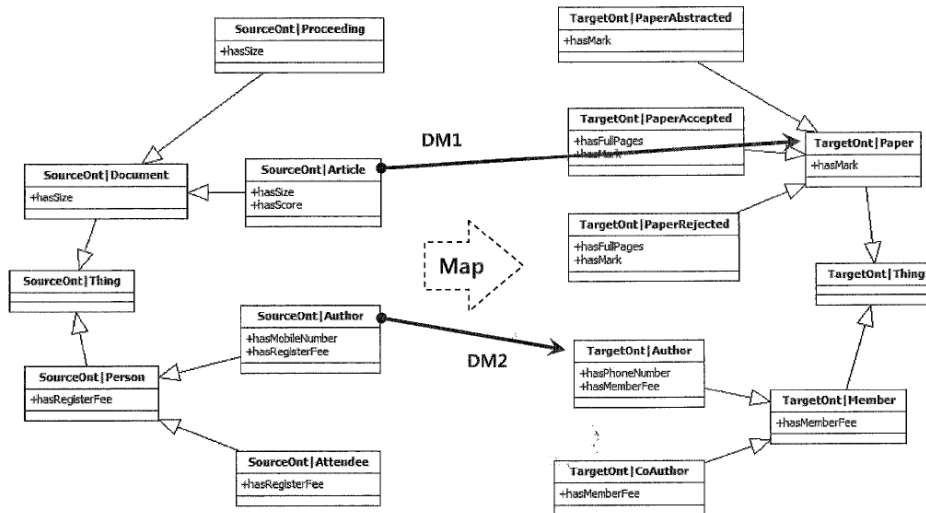
- 1) Please review "Example Mapping Design for Direct Mapping Type".
- 2) Please describe briefly two direct mappings on UML representation(3).
- 3) Please describe detail intention of the mapping in section (4).

Sections:

<p>(1)</p> <p>Mapping Design Worksheet for Direct Mapping Type Instruction</p>	<p>(2)</p> <p>Example Mapping Design for Direct Mapping Type</p>
<p>(3)</p> <p>UML Representation of the conference ontologies</p>	<p>(4)</p> <p>Description of Mapping Intention for Direct Mapping - ID 1 & 2</p>

(2) Example Mapping Design for Direct Mapping Type

This mapping type represents the expression for one ontology class (A) corresponding to another ontology class (B) with the integration property corresponding to "Unite the instances of class (A) to class (B)".



Direct Mapping 1

Please explain your intention of the mapping.

Instances in [**SourceOnt|Article**] class need to be mapped to [**TargetOnt|Paper**] class.

because **Article can be subclass of Paper.**

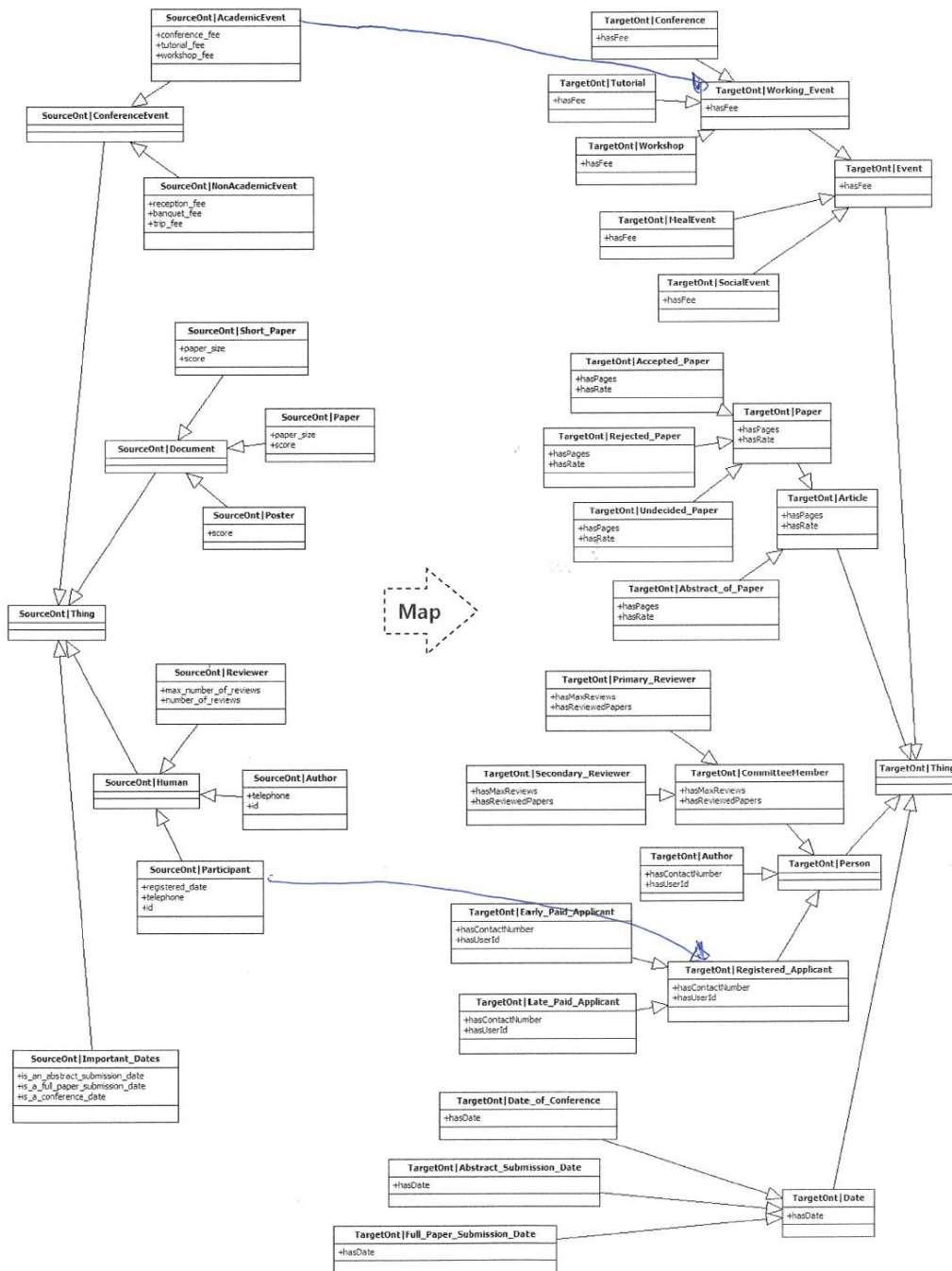
Direct Mapping 2

Please explain your intention of the mapping.

Instances in [**SourceOnt|Author**] class need to be mapped to [**TargetOnt|Author**] class.

because **Both Author can be considered same.**

(3) UML Representation of the conference ontologies



(4) Description of Mapping Intention for Direct Mapping - ID 1 & 2

For the conference ontologies:

You can make any assumption on the elements (please describe) and assume all the attributes are numeric types. The followings are format examples in detail for each attribute:

1. **conference_fee, tutorial_fee, workshop_fee**: ### (Upto 2 digits below point, e.g. 123.45, Currency is not yet decided.)
2. **reception_fee, banquet_fee, trip_fee, hasFee**: ### (Upto 2 digits below point, e.g. 1234.56, Currency is not yet decided.)
3. **paper_size, hasPages**: ## (Decimal Range from 1 to 40, e.g. 20)
4. **score**: ## (Decimal Range from 0 to 10, e.g. 5)
5. **hasRate**: ### (Decimal Range from 0 to 100, e.g. 50)
6. **max_number_of_reviews, hasMaxReviews**: ## (Decimal Range from 0 to 20, e.g. 10)
7. **number_of_reviews, hasReviewedPapers**: ## (Decimal Range from 0 to 20, e.g. 10)
8. **telephone, hasContactNumber**: ##### (10 Decimal Digits, e.g. 1234567890)
9. **id, hasUserId**: ##### (8 Decimal Digits, e.g. 12345678)
10. **is_an_abstract_submission_date, is_a_full_paper_submission_date**: YYYYDDMM (e.g. 20130731)
11. **is_a_conference_date, registered_date, hasDate**: YYYYDDMM (e.g. 20130801)

Direct Mapping 1

Please explain your intention of the mapping.

Instances in [*Source Ont | Academic Event*] class need to be integrated into [*Target Ont | Working Event*] class.

Because

Both can be considered for same.

Direct Mapping 2

Please explain your intention of the mapping.

Instances in [*Source Ont | Participant*] class need to be integrated into [*Target Ont | Registered Applicant*] class.

Because

Both are the same.

Mapping Design Worksheet for Data Range Mapping Type

Instruction:

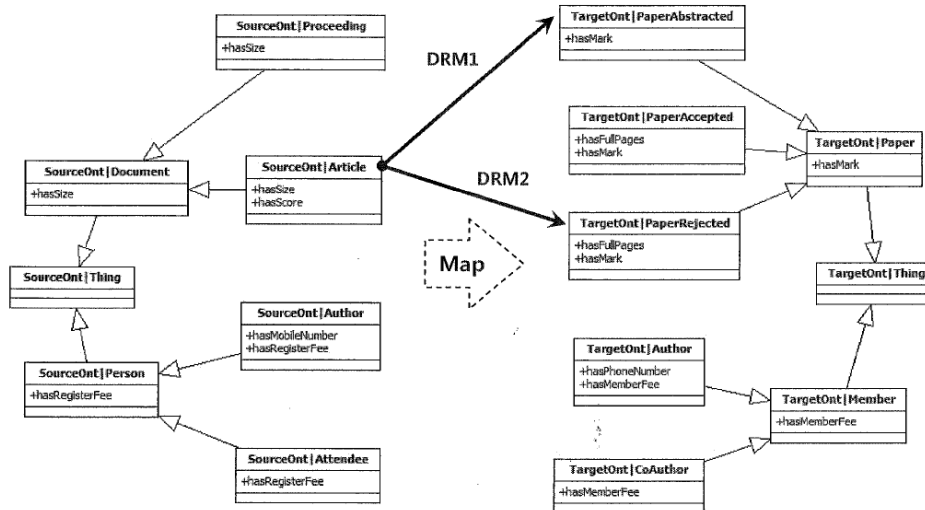
- 1) Please review "Example Mapping Design for Data Range Mapping Type".
- 2) Please describe briefly two data range mappings on UML representation(3).
- 3) Please describe detail intention of the mapping in section (4).

Sections:

<p>(1)</p> <p>Mapping Design Worksheet for Data Range Mapping Type Instruction</p>	<p>(2)</p> <p>Example Mapping Design for Data Range Mapping Type</p>
<p>(3)</p> <p>UML Representation of the conference ontologies</p>	<p>(4)</p> <p>Description of Mapping Intention for Data Range Mapping - ID 1 & 2</p>

(2) Example Mapping Design for Data Range Mapping Type

This mapping type represents the expression for an ontology class (A), that is constrained by the data range of the datatype properties, corresponding to another ontology class (B) with the integration property corresponding to "Unite the instances of class (A) to class (B) if the instance of class (A) satisfies the data range constraint".



Data Range Mapping 1

Please explain your intention of the mapping.

Instances in [SourceOnt|Article] class need to be integrated into [TargetOnt|PaperAbstracted] class with the constraint { **hasSize <= 2** }

because **Articles, that have less than or equal to 2 pages, are an abstracted paper.**

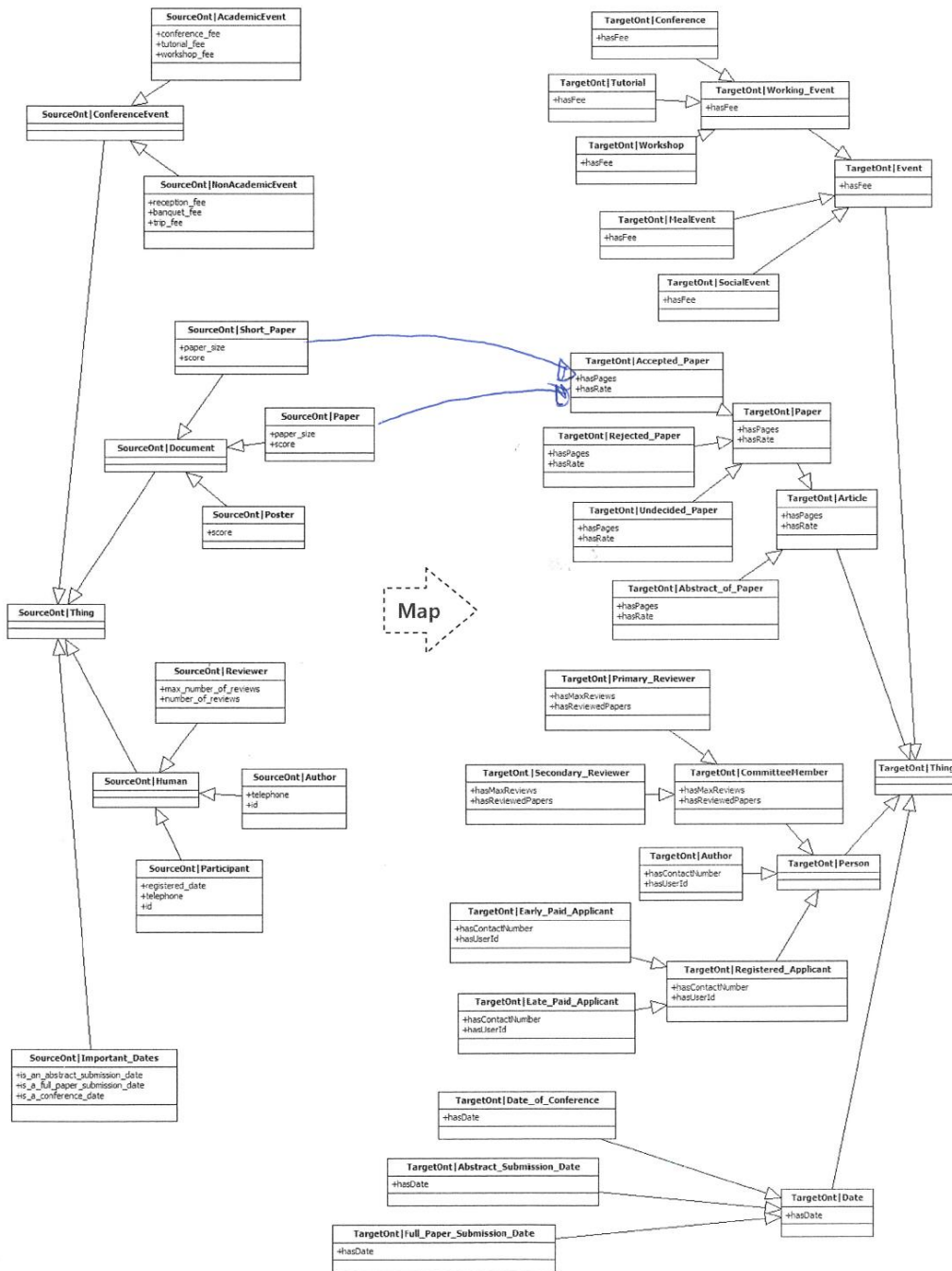
Data Range Mapping 2

Please explain your intention of the mapping.

Instances in [SourceOnt|Article] class need to be integrated into [TargetOnt|PaperRejected] class with the constraint { **hasScore < 5** }

because **Articles, that have less than 5 score, are a rejected paper.**

(3) UML Representation of the conference ontologies



(4) Description of Mapping Intention for Data Range Mapping - ID 1 & 2

For the conference ontologies:

You can make any assumption on the elements (please describe) and assume all the attributes are numeric types. The followings are format examples in detail for each attribute:

1. **conference_fee, tutorial_fee, workshop_fee:** ### (Upto 2 digits below point, e.g. 123.45, Currency is not yet decided.)
2. **reception_fee, banquet_fee, trip_fee, hasFee:** ### (Upto 2 digits below point, e.g. 1234.56, Currency is not yet decided.)
3. **paper_size, hasPages:** ## (Decimal Range from 1 to 40, e.g. 20)
4. **score:** ## (Decimal Range from 0 to 10, e.g. 5)
5. **hasRate:** ### (Decimal Range from 0 to 100, e.g. 50)
6. **max_number_of_reviews, hasMaxReviews:** ## (Decimal Range from 0 to 20, e.g. 10)
7. **number_of_reviews, hasReviewedPapers:** ## (Decimal Range from 0 to 20, e.g. 10)
8. **telephone, hasContactNumber:** ##### (10 Decimal Digits, e.g. 1234567890)
9. **id, hasUserId:** ##### (8 Decimal Digits, e.g. 12345678)
10. **is_an_abstract_submission_date, is_a_full_paper_submission_date:** YYYYDDMM (e.g. 20130731)
11. **is_a_conference_date, registered_date, hasDate:** YYYYDDMM (e.g. 20130801)

Data Range Mapping 1

Please explain your intention of the mapping.

Instances in [*Source* | *Skat Paper*] class need to be integrated into [*Target* | *Accepted Paper*] class.

with the constraint { *paper-size* ≤ 2 }

because *Skat* papers are published if they have less than 3 pages.

Data Range Mapping 2

Please explain your intention of the mapping.

Instances in [*Source* | *Paper*] class need to be integrated into [*Target* | *Accepted Paper*] class.

with the constraint { *paper-size* > 6 }

because normal length paper must be at least 6 pages.

Mapping Design Worksheet for Unit Transformation Mapping Type

Instruction:

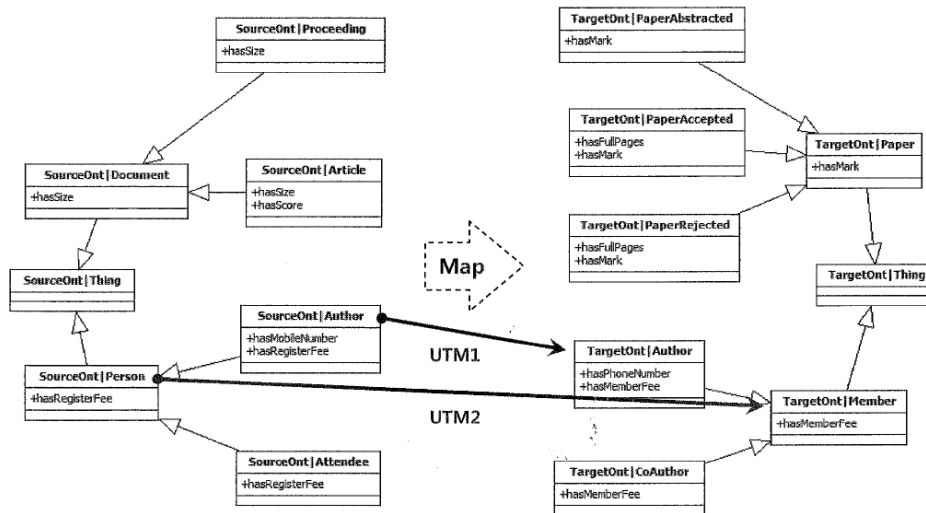
- 1) Please review "Example Mapping Design for Unit Transformation Mapping Type".
- 2) Please describe briefly two unit transformation mappings on UML representation(3).
- 3) Please describe detail intention of the mapping in section (4).

Sections:

(1) Mapping Design Worksheet for Unit Transformation Mapping Type Instruction	(2) Example Mapping Design for Unit Transformation Mapping Type
(3) UML Representation of the conference ontologies	(4) Description of Mapping Intention for Unit Transformation Mapping - ID 1 & 2

(2) Example Mapping Design for Unit Transformation Mapping Type

This mapping type represents the expression for an ontology class (A), that is constrained by the arithmetic transformation of the data value of the datatype properties, corresponding to another ontology class (B) with the integration property corresponding to "Unite the instances of class (A) to class (B) with the unit transformation according the constraint".



Unit Transformation Mapping 1

Please explain your intention of the mapping.

Instances in [**SourceOnt|Person**] class need to be integrated into [**TargetOnt|Member**] class with the constraint { **hasMemberFee = hasRegisterFee * 1.12** }

because **There is currency difference between RegisterFee and MemberFee.**

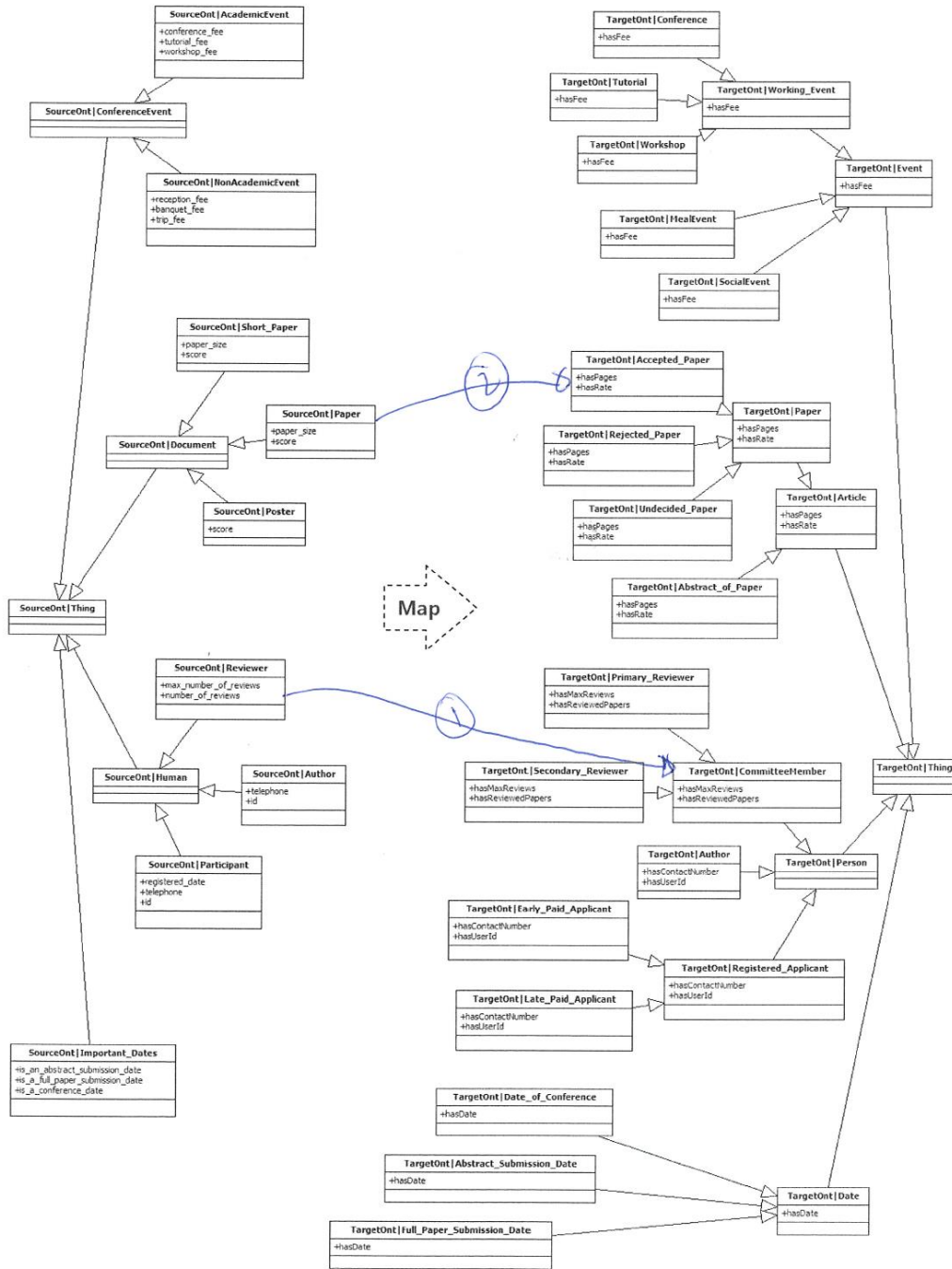
Unit Transformation Mapping 2

Please explain your intention of the mapping.

Instances in [**SourceOnt|Author**] class need to be integrated into [**TargetOnt|Author**] class with the constraint { **hasPhoneNumber = hasMobileNumber** }

because **PhoneNumber value needs to be same as the value of MobileNumber.**

(3) UML Representation of the conference ontologies



(4) Description of Mapping Intention for Unit Transformation Mapping - ID 1 & 2

For the conference ontologies:

You can make any assumption on the elements (please describe) and assume all the attributes are numeric types.

The followings are format examples in detail for each attribute:

1. **conference_fee, tutorial_fee, workshop_fee:** ### (Upto 2 digits below point, e.g. 123.45, Currency is not yet decided.)
2. **reception_fee, banquet_fee, trip_fee, hasFee:** ### (Upto 2 digits below point, e.g. 1234.56, Currency is not yet decided.)
3. **paper_size, hasPages:** ## (Decimal Range from 1 to 40, e.g. 20)
4. **score:** ## (Decimal Range from 0 to 10, e.g. 5)
5. **hasRate:** ### (Decimal Range from 0 to 100, e.g. 50)
6. **max_number_of_reviews, hasMaxReviews:** ## (Decimal Range from 0 to 20, e.g. 10)
7. **number_of_reviews, hasReviewedPapers:** ## (Decimal Range from 0 to 20, e.g. 10)
8. **telephone, hasContactNumber:** ##### (10 Decimal Digits, e.g. 1234567890)
9. **id, hasUserId:** ##### (8 Decimal Digits, e.g. 12345678)
10. **is_an_abstract_submission_date, is_a_full_paper_submission_date:** YYYYDDMM (e.g. 20130731)
11. **is_a_conference_date, registered_date, hasDate:** YYYYDDMM (e.g. 20130801)

Unit Transformation Mapping 1

Please explain your intention of the mapping.

Instances in [*Source* | *reviewer*] class need to be integrated into [*Target* | *Committee member*] class.

with the constraint { $\text{number_of_reviews} = \text{hasReviewedPapers}$ }

because *the number of reviews should be the same*

Unit Transformation Mapping 2

Please explain your intention of the mapping.

Instances in [*Source* | *Paper*] class need to be integrated into [*Target* | *Accepted paper*] class.

with the constraint { $\text{hasRate} = \text{score} \div 100$ }

because *scores should be tracked as percentages.*

Participant ID: _____

02

Questionnaire

Instruction: For each of the following statements, mark one box that best describes your reactions to the UML representation and the tool you have just used to design the semantic mappings and to auto-generate semantic mapping documents. Each question is optional. Feel free to omit a response to any question; however the researcher would be grateful if all questions are responded to.

Part 1 / 2:

SUS (System Usability Scale) and Research Specific Questions for the user satisfaction of using UML representation of ontologies to design the semantic mappings.

	Strongly Disagree			Strongly Agree			
1. I think that I would like to design the semantic mapping using the UML representation of ontologies frequently.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
2. I found the semantic mapping on the UML representation of ontologies is unnecessarily complex.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4
3. I thought the semantic mapping on the UML representation of ontologies was easy to design.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
4. I think that I would need the support of an expert person to be able to design the semantic mapping on the UML representation of ontologies.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4
5. I found UML notations related to the semantic data and mappings in UML class diagram were well integrated.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
6. I thought there was too much inconsistency to represent semantic data and mappings using UML notations.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4
7. I would imagine that most people would learn to design the semantic mapping using the UML representation of ontologies very quickly.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
8. I found the UML representation of ontologies very cumbersome to design the semantic mapping.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
9. I felt very confident to design the semantic mapping using the UML representation of ontologies.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
10. I needed to learn a lot of things before I could get going to design the semantic mapping with the UML representation of ontologies.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4
11. I found Data Range Mapping and Unit Transformation Mapping useful in expressing more precise mapping.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

70, 5, 77.5, 1

1

12. I think UML representation of semantic data and mapping is easy to understand for system engineers.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
---	--------------------------	--------------------------	--------------------------	--------------------------	-------------------------------------

Part 2 / 2:

SUS (System Usability Scale) and Research Specific Questions for the user satisfaction of using the SDI Tool to auto-generate semantic mapping documents (and an integrated ontology).

	Strongly Disagree			Strongly Agree			
1. I think that I would like to use the SDI tool frequently.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
2. I found the SDI tool is unnecessarily complex.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
3. I thought the SDI tool was easy to use.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
4. I think that I would need the support of a technical person to be able to use the SDI tool.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
5. I found the various functions in the SDI tool were well integrated.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2
6. I thought there was too much inconsistency in the SDI tool.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
7. I would imagine that most people would learn to use the SDI tool very quickly.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	4
8. I found the SDI tool very cumbersome to use.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
9. I felt very confident using the SDI tool.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	4
10. I needed to learn a lot of things before I could get going with the SDI tool.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
11. I think the SDI tool requires ontology expert knowledge to conduct semantic data integration.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

The experiment is completed. Thank you!

Appendix K.2 Experiment result from participant ID13

13

TRINITY COLLEGE DUBLIN

INFORMED CONSENT FORM

LEAD RESEARCHER: Seung-Hwa Chung

BACKGROUND OF RESEARCH:

Semantic Data Integration is normally conducted by ontology experts, and most existing semantic mapping tools support only one-to-one correspondence mapping and typically require a high level of technical expertise which limits the users of the tools to knowledge engineers or users who have received substantial training.

The experiment will evaluate the usability of a SDI Tool, designed to offer usable semantic data integration for non-ontology expert system engineers. It uses UML class diagrams as a semantic data and mapping representation for the users and then auto-generates semantic mapping documents (and an integrated ontology).

PROCEDURES OF THIS STUDY:

In the experiment, participants will conduct a semantic integration task on the given UML representation of ontologies using the developed SDI Tool. The selected ontology domain, conferences, is general and understandable by most people with a system engineering background. This experiment tries to avoid a domain specific ontology that requires deep understanding of the specific knowledge, so that system engineers in any domain can conduct the experiment and be able to describe the intentions behind their mappings.

The participant will be asked to complete several tasks in the course of the experiment: designing total six different mappings on the mapping design worksheets (two mappings in direct mapping type, two mappings in data range mapping type and two mappings in unit transformation mapping type); conducting semantic data integration using SDI tool according to the designed mappings; and answering a questionnaire. However the participation is voluntary and individual tasks or the experiment as a whole may be abandoned.

In the case of the questionnaire, all participants can omit the entire questionnaire or individual questions if they wish. At the end of the study the lead researcher may contact the participant via email for feedback on the experiment outcomes or vice versa.

There are no particular risks to participating in this experiment beyond those inherent using any computer application.

PUBLICATION:

All the research results will be used within the following venues, and will be presented without any personal identifier of participants. Individual results will be aggregated anonymously and research reported on aggregate results:

1. The researcher's thesis
2. Conference papers

* Mistake on the tool
② (Conference Event → Event)
but (Conference Event → Workshop Event)

DECLARATION:

- I am 18 years or older and am competent to provide consent.
- I have read, or had read to me, a document providing information about this research and this consent form. I have had the opportunity to ask questions and all my questions have been answered to my satisfaction and understand the description of the research that is being provided to me.
- I agree that my data is used for scientific purposes and I have no objection that my data is published in scientific publications in a way that does not reveal my identity.
- I understand that if I make illicit activities known, these will be reported to appropriate authorities.
- I understand that I may stop electronic recordings at any time, and that I may at any time, even subsequent to my participation have such recordings destroyed (except in situations such as above).
- I understand that, subject to the constraints above, no recordings will be replayed in any public forum or made available to any audience other than the current researchers/research team.
- I freely and voluntarily agree to be part of this research study, though without prejudice to my legal and ethical rights.
- I understand that I may refuse to answer any question and that I may withdraw at any time without penalty.
- I understand that my participation is fully anonymous and that no personal details about me will be recorded.
- I understand that if I or anyone in my family has a history of epilepsy then I am proceeding at my own risk.
- I have received a copy of this agreement.

PARTICIPANT'S NAME:



PARTICIPANT'S SIGNATURE:



Date:

1/10/2013

Statement of investigator's responsibility:

I have explained the nature and purpose of this research study, the procedures to be undertaken and any risks that may be involved. I have offered to answer any questions and fully answered such questions. I believe that the participant understands my explanation and has freely given informed consent.

RESEARCHERS CONTACT DETAILS:

Name: Seung-Hwa Chung
Email: chung@tcd.ie

INVESTIGATOR'S SIGNATURE:

Date:

Participant ID: 13

Date: 2013, 10, 1

Task Sheet

1. Brief Explanation about SDI Tool

- 1) Watch overview video

2. Describe Mappings on Mapping Design Worksheets

- 1) You need to describe in total six different mappings:
 - Two mappings of direct mapping type using the mapping design worksheet for direct mapping type
 - Two mappings of data range mapping type using the mapping design worksheet for data range mapping type
 - Two mappings of unit transformation mapping type using the mapping design worksheet for unit transformation mapping type

Total three mapping design worksheets will be given, and please describe two mappings for each mapping design worksheet.

Please indicate start and end time of the mappings' design. (Hour:Minute Format)

Start time: <u>2:42 pm</u>	End time: <u>3:9 pm</u>
----------------------------	-------------------------

3. Watch the video of SDI tool use examples

27m

4. Conduct Semantic Data Integration using SDI Tool

- 1) You need to conduct integration task with the tool according to the description on the mapping design worksheets. You can refer to Brief Mapping Guide and Manual for SDI Tool that describes examples of how to use the tool.

Please indicate start and end time of the integration task on the tool. (Hour:Minute Format)

Start time: <u>3:15 pm</u>	End time: <u>3:24 pm</u>
----------------------------	--------------------------

5. Answer Questionnaire

- 1) Please now fill in the questionnaire.

9m

Thank you!

Mapping Design Worksheet for Direct Mapping Type

Participant ID: _____

13

Instruction:

- 1) Please review "Example Mapping Design for Direct Mapping Type".
- 2) Please describe briefly two direct mappings on UML representation(3).
- 3) Please describe detail intention of the mapping in section (4).

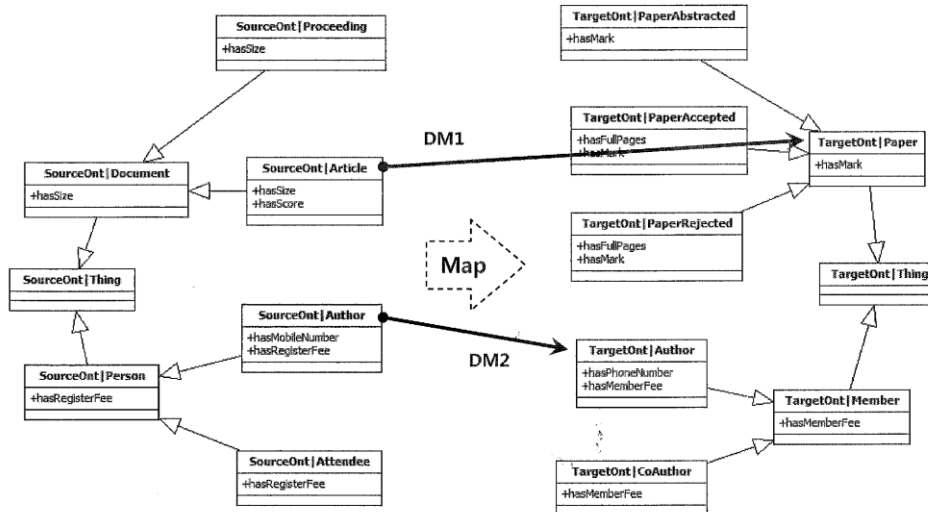
Sections:

(1) Mapping Design Worksheet for Direct Mapping Type Instruction	(2) Example Mapping Design for Direct Mapping Type
(3) UML Representation of the conference ontologies	(4) Description of Mapping Intention for Direct Mapping - ID 1 & 2

1

(2) Example Mapping Design for Direct Mapping Type

This mapping type represents the expression for one ontology class (A) corresponding to another ontology class (B) with the integration property corresponding to "Unite the instances of class (A) to class (B)".



Direct Mapping 1

Please explain your intention of the mapping.

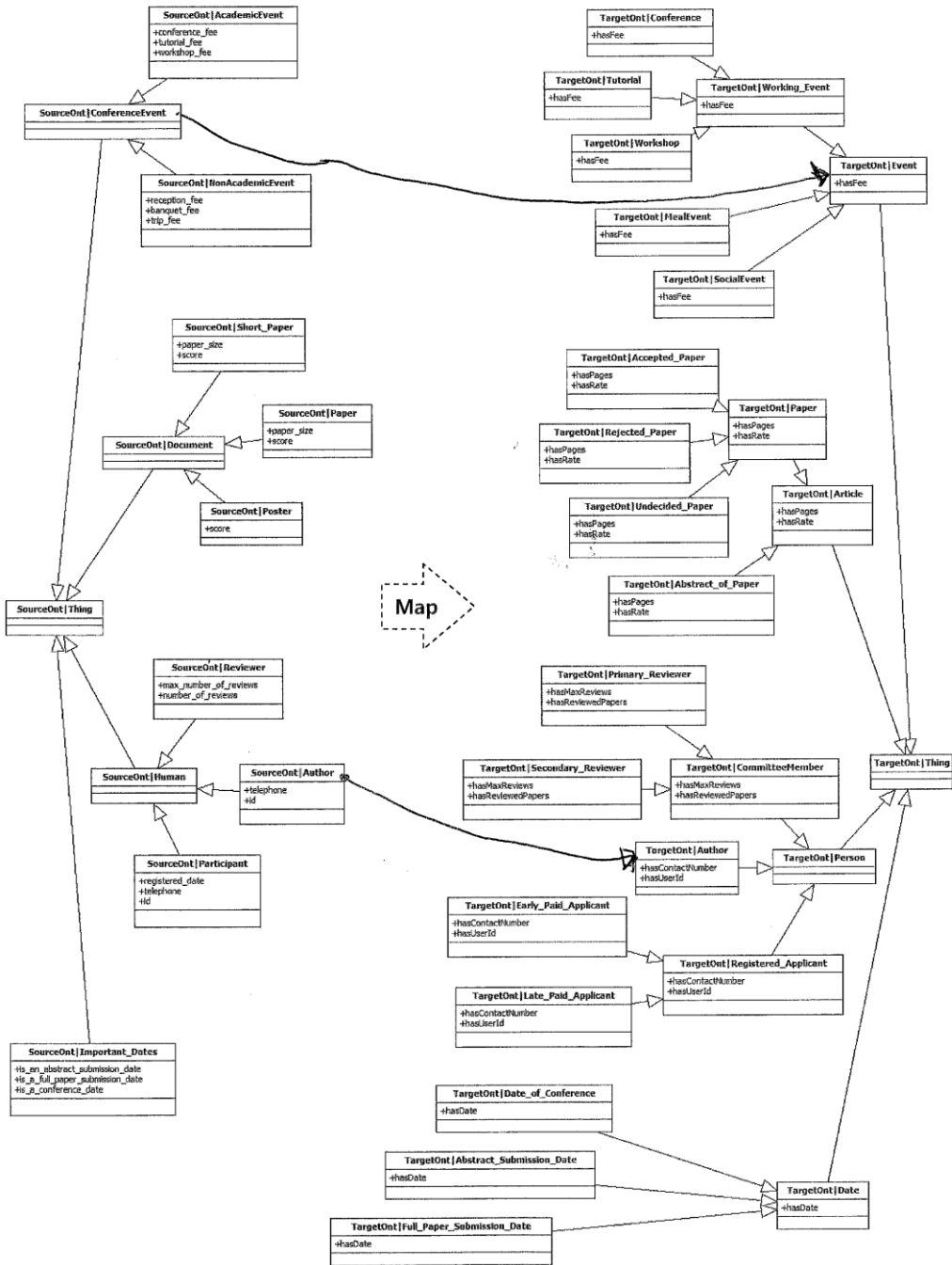
Instances in [**SourceOnt|Article**] class need to be mapped to [**TargetOnt|Paper**] class.
because **Article can be subclass of Paper.**

Direct Mapping 2

Please explain your intention of the mapping.

Instances in [**SourceOnt|Author**] class need to be mapped to [**TargetOnt|Author**] class.
because **Both Author can be considered same.**

(3) UML Representation of the conference ontologies



(4) Description of Mapping Intention for Direct Mapping - ID 1 & 2

For the conference ontologies:

You can make any assumption on the elements (please describe) and assume all the attributes are numeric types. The followings are format examples in detail for each attribute:

1. `conference_fee`, `tutorial_fee`, `workshop_fee`: ### (Upto 2 digits below point, e.g. 123.45, Currency is not yet decided.)
2. `reception_fee`, `banquet_fee`, `trip_fee`, `hasFee`: ### (Upto 2 digits below point, e.g. 1234.56, Currency is not yet decided.)
3. `paper_size`, `hasPages`: ## (Decimal Range from 1 to 40, e.g. 20)
4. `score`: ## (Decimal Range from 0 to 10, e.g. 5)
5. `hasRate`: ### (Decimal Range from 0 to 100, e.g. 50)
6. `max_number_of_reviews`, `hasMaxReviews`: ## (Decimal Range from 0 to 20, e.g. 10)
7. `number_of_reviews`, `hasReviewedPapers`: ## (Decimal Range from 0 to 20, e.g. 10)
8. `telephone`, `hasContactNumber`: ##### (10 Decimal Digits, e.g. 1234567890)
9. `id`, `hasUserId`: ##### (8 Decimal Digits, e.g. 12345678)
10. `is_an_abstract_submission_date`, `is_a_full_paper_submission_date`: YYYYDDMM (e.g. 20130731)
11. `is_a_conference_date`, `registered_date`, `hasDate`: YYYYDDMM (e.g. 20130801)

Direct Mapping 1

Please explain your intention of the mapping.

Instances in [SO: `ConferenceEvent`] class need to be integrated into [TO: `Event`] class.

Because

They have the same level of abstraction and their subclasses are comparable.

Direct Mapping 2

Please explain your intention of the mapping.

Instances in [SO: `Author`] class need to be integrated into [TO: `Author`] class.

Because

Similarity of attributes. Neither class has subclasses.

Mapping Design Worksheet for Data Range Mapping Type

Instruction:

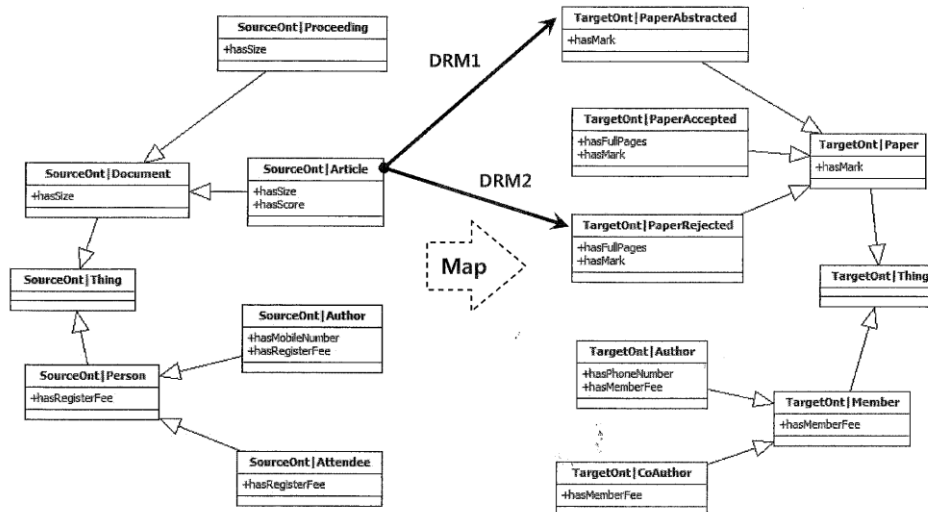
- 1) Please review "Example Mapping Design for Data Range Mapping Type".
- 2) Please describe briefly two data range mappings on UML representation(3).
- 3) Please describe detail intention of the mapping in section (4).

Sections:

<p>(1)</p> <p>Mapping Design Worksheet for Data Range Mapping Type Instruction</p>	<p>(2)</p> <p>Example Mapping Design for Data Range Mapping Type</p>
<p>(3)</p> <p>UML Representation of the conference ontologies</p>	<p>(4)</p> <p>Description of Mapping Intention for Data Range Mapping - ID 1 & 2</p>

(2) Example Mapping Design for Data Range Mapping Type

This mapping type represents the expression for an ontology class (A), that is constrained by the data range of the datatype properties, corresponding to another ontology class (B) with the integration property corresponding to "Unite the instances of class (A) to class (B) if the instance of class (A) satisfies the data range constraint".



Data Range Mapping 1

Please explain your intention of the mapping.

Instances in [**SourceOnt|Article**] class need to be integrated into [**TargetOnt|PaperAbstracted**] class with the constraint { **hasSize <= 2** }

because **Articles, that have less than or equal to 2 pages, are an abstracted paper.**

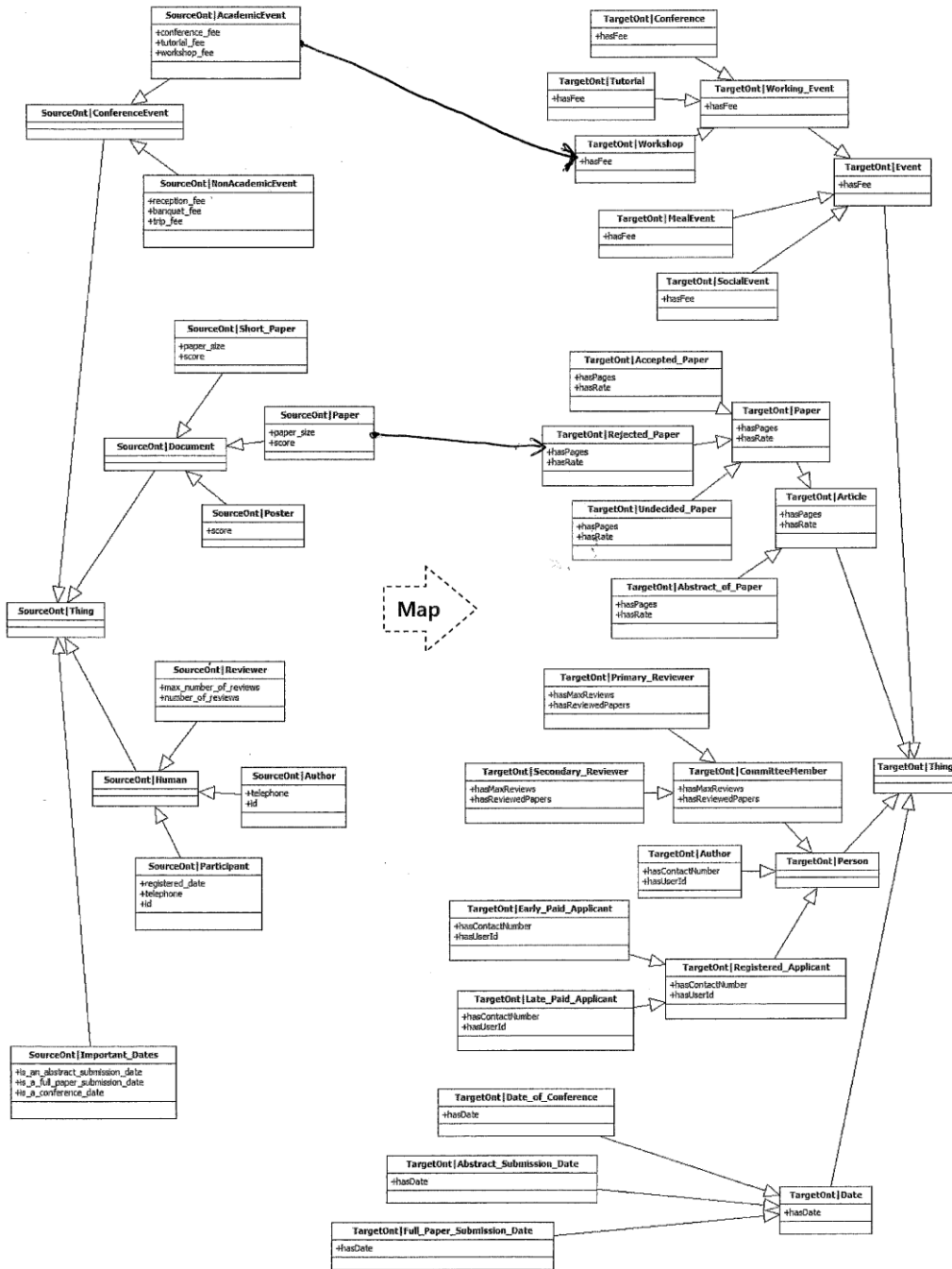
Data Range Mapping 2

Please explain your intention of the mapping.

Instances in [**SourceOnt|Article**] class need to be integrated into [**TargetOnt|PaperRejected**] class with the constraint { **hasScore < 5** }

because **Articles, that have less than 5 score, are a rejected paper.**

(3) UML Representation of the conference ontologies



(4) Description of Mapping Intention for Data Range Mapping - ID 1 & 2

For the conference ontologies:

You can make any assumption on the elements (please describe) and assume all the attributes are numeric types.

The followings are format examples in detail for each attribute:

1. **conference_fee, tutorial_fee, workshop_fee:** #.## (Upto 2 digits below point, e.g. 123.45, Currency is not yet decided.)
2. **reception_fee, banquet_fee, trip_fee, hasFee:** #.## (Upto 2 digits below point, e.g. 1234.56, Currency is not yet decided.)
3. **paper_size, hasPages:** ## (Decimal Range from 1 to 40, e.g. 20)
4. **score:** ## (Decimal Range from 0 to 10, e.g. 5)
5. **hasRate:** ### (Decimal Range from 0 to 100, e.g. 50)
6. **max_number_of_reviews, hasMaxReviews:** ## (Decimal Range from 0 to 20, e.g. 10)
7. **number_of_reviews, hasReviewedPapers:** ## (Decimal Range from 0 to 20, e.g. 10)
8. **telephone, hasContactNumber:** ##### (10 Decimal Digits, e.g. 1234567890)
9. **id, hasUserId:** ##### (8 Decimal Digits, e.g. 12345678)
10. **is_an_abstract_submission_date, is_a_full_paper_submission_date:** YYYYDDMM (e.g. 20130731)
11. **is_a_conference_date, registered_date, hasDate:** YYYYDDMM (e.g. 20130801)

Data Range Mapping 1

Please explain your intention of the mapping.

Instances in [SO: Paper] class need to be integrated into [TO: Rejected_Paper] class.

with the constraint { score < 6 }

because

Papers are going to be rejected if score < 6

Data Range Mapping 2

Please explain your intention of the mapping.

Instances in [SO: AcademicEvent] class need to be integrated into [TO: Workshop] class.

with the constraint { workshop_fee > 0 }

because

When a workshop fee is paid these classes are equivalent
(except for the case of a free workshop)

Mapping Design Worksheet for Unit Transformation Mapping Type

Instruction:

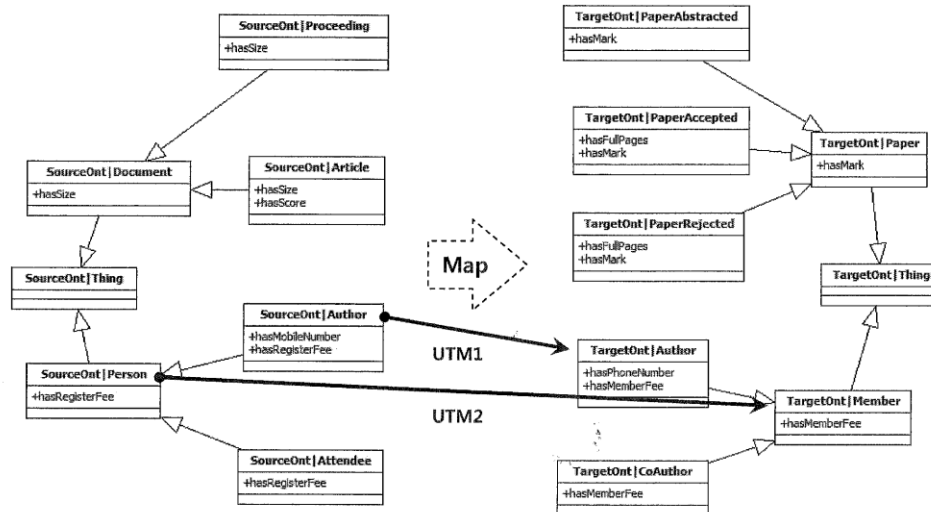
- 1) Please review "Example Mapping Design for Unit Transformation Mapping Type".
- 2) Please describe briefly two unit transformation mappings on UML representation(3).
- 3) Please describe detail intention of the mapping in section (4).

Sections:

<p>(1)</p> <p>Mapping Design Worksheet for Unit Transformation Mapping Type Instruction</p>	<p>(2)</p> <p>Example Mapping Design for Unit Transformation Mapping Type</p>
<p>(3)</p> <p>UML Representation of the conference ontologies</p>	<p>(4)</p> <p>Description of Mapping Intention for Unit Transformation Mapping - ID 1 & 2</p>

(2) Example Mapping Design for Unit Transformation Mapping Type

This mapping type represents the expression for an ontology class (A), that is constrained by the arithmetic transformation of the data value of the datatype properties, corresponding to another ontology class (B) with the integration property corresponding to "Unite the instances of class (A) to class (B) with the unit transformation according the constraint".



Unit Transformation Mapping 1

Please explain your intention of the mapping.

Instances in [**SourceOnt|Person**] class need to be integrated into [**TargetOnt|Member**] class with the constraint { **hasMemberFee = hasRegisterFee * 1.12** }

because **There is currency difference between RegisterFee and MemberFee.**

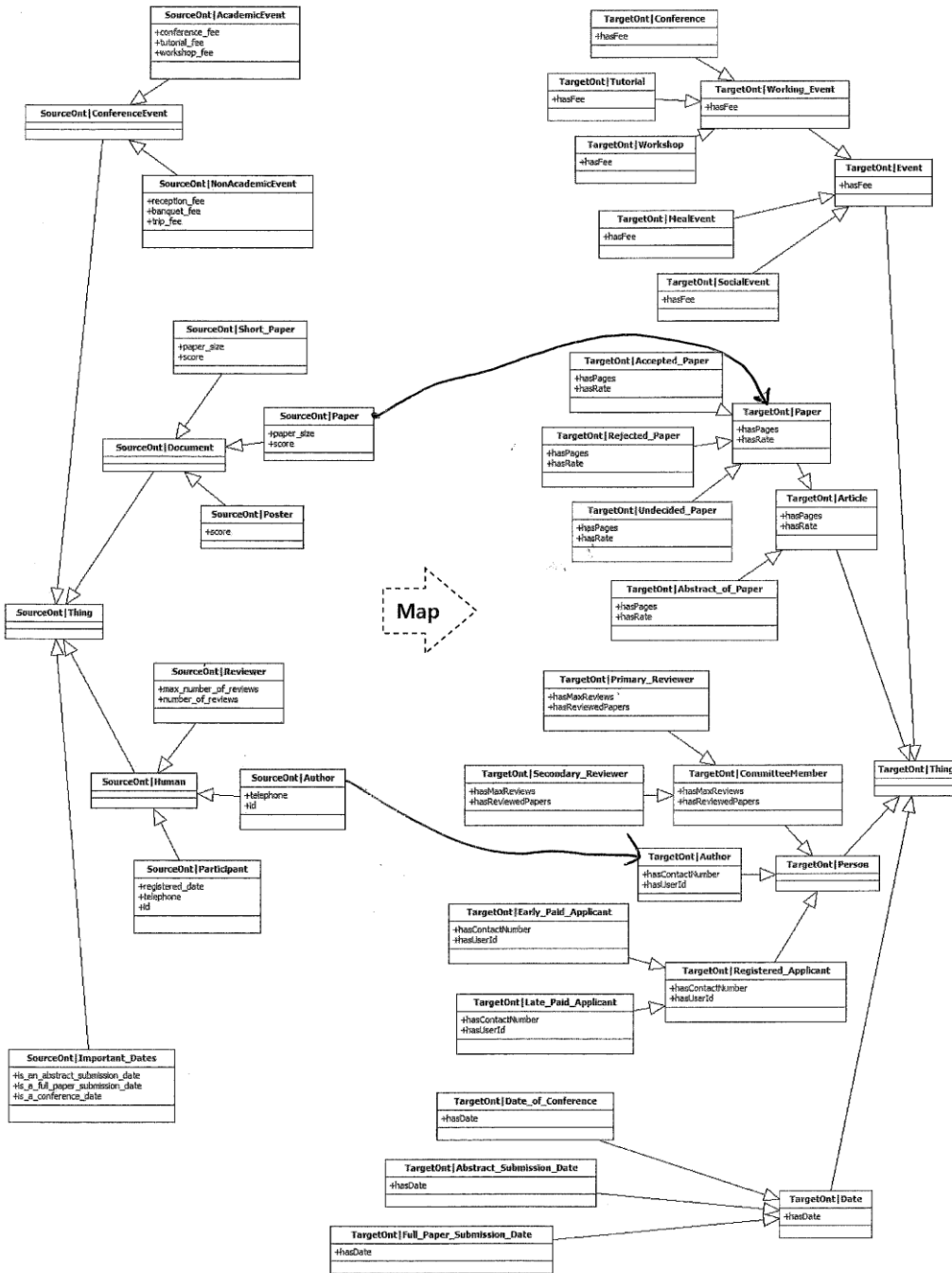
Unit Transformation Mapping 2

Please explain your intention of the mapping.

Instances in [**SourceOnt|Author**] class need to be integrated into [**TargetOnt|Author**] class with the constraint { **hasPhoneNumber = hasMobileNumber** }

because **PhoneNumber value needs to be same as the value of MobileNumber.**

(3) UML Representation of the conference ontologies



(4) Description of Mapping Intention for Unit Transformation Mapping - ID 1 & 2

For the conference ontologies:

You can make any assumption on the elements (please describe) and assume all the attributes are numeric types.

The followings are format examples in detail for each attribute:

1. `conference_fee`, `tutorial_fee`, `workshop_fee`: `###` (Upto 2 digits below point, e.g. 123.45, Currency is not yet decided.)
2. `reception_fee`, `banquet_fee`, `trip_fee`, `hasFee`: `###` (Upto 2 digits below point, e.g. 1234.56, Currency is not yet decided.)
3. `paper_size`, `hasPages`: `##` (Decimal Range from 1 to 40, e.g. 20)
4. `score`: `##` (Decimal Range from 0 to 10, e.g. 5)
5. `hasRate`: `###` (Decimal Range from 0 to 100, e.g. 50)
6. `max_number_of_reviews`, `hasMaxReviews`: `##` (Decimal Range from 0 to 20, e.g. 10)
7. `number_of_reviews`, `hasReviewedPapers`: `##` (Decimal Range from 0 to 20, e.g. 10)
8. `telephone`, `hasContactNumber`: `#####` (10 Decimal Digits, e.g. 1234567890)
9. `id`, `hasUserId`: `#####` (8 Decimal Digits, e.g. 12345678)
10. `is_an_abstract_submission_date`, `is_a_full_paper_submission_date`: `YYYYDDMM` (e.g. 20130731)
11. `is_a_conference_date`, `registered_date`, `hasDate`: `YYYYDDMM` (e.g. 20130801)

Unit Transformation Mapping 1

Please explain your intention of the mapping.

Instances in [SO: *Paper*] class need to be integrated into [TO: *Paper*] class.

with the constraint { *hasRate* = *score* * 10 }

because

hasRate has a numeric range 10 times greater than *score*

Unit Transformation Mapping 2

Please explain your intention of the mapping.

Instances in [SO: *Author*] class need to be integrated into [TO: *Author*] class.

with the constraint { *hasContactNumber* = *telephone* }

because

They are equivalent

Participant ID:

13

Questionnaire

Instruction: For each of the following statements, mark one box that best describes your reactions to the UML representation and the tool you have just used to design the semantic mappings and to auto-generate semantic mapping documents. Each question is optional. Feel free to omit a response to any question; however the researcher would be grateful if all questions are responded to.

Part 1 / 2:

SUS (System Usability Scale) and Research Specific Questions for the user satisfaction of using UML representation of ontologies to design the semantic mappings.

	Strongly Disagree					Strongly Agree	
1. I think that I would like to design the semantic mapping using the UML representation of ontologies frequently.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		3
2. I found the semantic mapping on the UML representation of ontologies is unnecessarily complex.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		3
3. I thought the semantic mapping on the UML representation of ontologies was easy to design.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		2
4. I think that I would need the support of an expert person to be able to design the semantic mapping on the UML representation of ontologies.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		3
5. I found UML notations related to the semantic data and mappings in UML class diagram were well integrated.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
6. I thought there was too much inconsistency to represent semantic data and mappings using UML notations.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		4
7. I would imagine that most people would learn to design the semantic mapping using the UML representation of ontologies very quickly.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		3
8. I found the UML representation of ontologies very cumbersome to design the semantic mapping.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		3
9. I felt very confident to design the semantic mapping using the UML representation of ontologies.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		2
10. I needed to learn a lot of things before I could get going to design the semantic mapping with the UML representation of ontologies.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		2
11. I found Data Range Mapping and Unit Transformation Mapping useful in expressing more precise mapping.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

62.5 , 4 , 52.5 , 2

1

12. I think UML representation of semantic data and mapping is easy to understand for system engineers.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
---	--------------------------	--------------------------	--------------------------	-------------------------------------	--------------------------

Part 2 / 2:

SUS (System Usability Scale) and Research Specific Questions for the user satisfaction of using the SDI Tool to auto-generate semantic mapping documents (and an integrated ontology).

	Strongly Disagree			Strongly Agree			
1. I think that I would like to use the SDI tool frequently.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2
2. I found the SDI tool is unnecessarily complex.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
3. I thought the SDI tool was easy to use.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2
4. I think that I would need the support of a technical person to be able to use the SDI tool.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2
5. I found the various functions in the SDI tool were well integrated.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
6. I thought there was too much inconsistency in the SDI tool.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
7. I would imagine that most people would learn to use the SDI tool very quickly.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
8. I found the SDI tool very cumbersome to use.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
9. I felt very confident using the SDI tool.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2
10. I needed to learn a lot of things before I could get going with the SDI tool.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2
11. I think the SDI tool requires ontology expert knowledge to conduct semantic data integration.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

The experiment is completed. Thank you!

✓ If you have any comments

The interface for the tool presents several UI challenges. Firstly there are lots of opportunities to make mistakes, more constraint in the actions the user can perform is necessary. Allowing users to enter free text is liable to errors that could be avoided by using drop downs. Secondly there are too many options in the right-hand column when I'm only using 6 of the many available. Thirdly there is an issue with association arrows overlap and it becomes hard to distinguish associations.

Thank you!

APPENDIX L Accompanying DVD: Table of Contents

- Thesis.pdf: This file is a digital copy of the thesis.

- Experiment 2012 for Confirmation Report Folder: This folder contains the initial experiment on the tool that implements the ODM and OCL combined abstract mapping syntax.
 - Experiment Preparation and Code Subfolder: This folder contains the experiment ontologies, tool source code, demo samples and documents.
 - Experiment Results from Participants Subfolder: This folder contains the experiment results from the participants.

- Experiment 2013 for PhD Thesis Folder: This folder contains the system engineers experiment on the SDI tool.
 - Experiment Ethical Applications Subfolder: This folder contains the approved ethical applications for the experiment.
 - Experiment Materials and Code Subfolder: This folder contains the experiment ontologies, tool source code and hand-out materials for the experiment.
 - Experiment Results from Participants Subfolder: This folder contains the experiment input/output file results from the participants and a sample of a scanned full paper record documentation from participant ID02 and ID13.