Framework for the Interoperability of Software Engineering Metamodels

Muhammad Atif Qureshi

A thesis submitted for the degree of Doctor of Philosophy
University of Technology, Sydney

July 2014
Declaration

This is to certify that this thesis comprises my original work toward the Doctor of Philosophy degree and due acknowledgement has been made in the text of all other materials used. The work has not been submitted previously, in whole or part, to qualify for any other academic award. Any editorial work, paid or unpaid, carried out by a third party is acknowledged.

Muhammad Atif Qureshi

(July 2014)
Dedicated to my mother. At her knees my education commenced and to her I
owe all that I am or hope to be.
Abstract

A model, represented as a concrete artefact, is an abstraction of reality according to a certain conceptualization. A model can support communication and analysis about relevant aspects of the underlying domain. A model must be expressed in some language and such languages are defined using metamodels. Many metamodels have been proposed and used in the software engineering literature. Some define modelling languages that are general in nature but the literature of modelling is dominated by domain-specific modelling languages or metamodels. Most of these metamodels have been developed independently from each other and any shared concepts are only accidental. Widespread adoption of these metamodels is hindered by differences between metamodels’ concepts. Using more than one modelling language during software development requires some sort of interoperability between the metamodels of those modelling languages. This interoperability is also required to allow mappings between models developed using different modelling languages.

These metamodels are not static in nature and are continuously evolving. This evolution has increased their size and complexity over time. This complexity increases when more than one metamodel is used.
during software development. Interoperability of a pair of metamodels can reduce their joint size and complexity (elaborated in detail in Chapter 7). The need for interoperability between metamodels is also raised by many research communities.

In this thesis, we have developed a framework that can be used for metamodel interoperability. The framework compares metamodel elements based on their syntax, semantics and structure. The semantics of metamodel elements are further investigated for linguistic and ontological semantics.

Since terms such as interoperability, bridging, merging and mapping have all been used, often loosely, with reference to metamodel compatibility, we will define these terms under the generic term harmonization.

Metamodels share some similarities with other domains, e.g. ontologies and schemas. In this thesis, we have also explored the techniques available in these domains that might be useful for metamodel interoperability. We have applied our framework to different metamodels and have shown how metamodels can be used in an interoperable fashion.

The results achieved are analysed and we have shown how interoperability of metamodels can reduce their size and their joint complexity, hence making them easier to understand and use.
Acknowledgement

Firstly, my greatest regards to the Almighty Allah for bestowing upon me the blessings, throughout my life, that make me able to achieve this milestone.

There are no words I could possibly write to articulate my gratitude to my mother, whose never ending prayers followed me everywhere and gave me the courage to face the complexities of life and complete this project successfully.

I must express my gratitude to my wife for her continued support and encouragement. Completing this work would have been all the more difficult without her support. She was undoubtedly the best partner I could have had during the ups and downs of my journey. Also, special thanks to my daughters, who always give me a new hope and strength.

I am immeasurably proud of my brothers and my sisters. I am grateful to them, for all their love, care and everlasting support in these many years of physical distance.
I would like to thank my supervisor, Prof. Brian Henderson-Sellers, for the patient guidance, encouragement and advice he has provided throughout my time as his student. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly. I would also like to thank Dr Bruce Howarth for his help regarding proof reading and improving the presentation of this thesis.

This thesis is dedicated to all these people. I could not have completed my research without the support of all these wonderful people!
Publications

Publications arising from this thesis.


# Contents

1 Introduction 1

1.1 Background .............................................. 2
1.2 Motivation ............................................. 4
1.3 Objectives ............................................. 6
1.4 Scope .................................................. 7
1.5 The Approach ......................................... 8

2 Model-Driven Engineering 12

2.1 Models .................................................. 14
2.1.1 Descriptive and Specification Models ................. 14
2.2 Modelling Languages ................................... 18
2.3 Metamodels ............................................ 21
2.4 Model-Driven Architecture (MDA) ....................... 26
2.4.1 Linguistic and Ontological Metamodelling ........... 30
2.5 Conclusion ............................................. 33

3 Metamodels and Ontologies 35

3.1 Ontologies in Software Engineering ...................... 37
3.1.1 Ontologies and Metamodels ......................... 41
3.2 Conclusion ............................................. 47
4 Similarity Matching Techniques 48

4.1 Harmonization of Terms 49

4.1.1 Matching 50

4.1.2 Interoperability 53

4.1.3 Merging and Integration 54

4.1.4 Alignment 56

4.1.5 Bridging 56

4.1.6 Mapping 57

4.2 Similarity Techniques in Models and Metamodels 59

4.3 Similarity Techniques in Ontologies 62

4.4 Similarity Techniques in Schemas 64

4.5 Similarity Techniques in Business Process Models 66

4.6 Conclusion 68

5 The Interoperability Framework 70

5.1 Selection of Metamodels 73

5.2 Linguistic Analysis 74

5.2.1 Syntactic Analysis 74

5.2.2 Semantic Analysis 75

5.2.3 Structural Analysis 77

5.3 Ontological Analysis 81

5.3.1 Linguistically Same Concepts 81

5.3.2 Linguistically Similar Concepts 82

5.3.3 Linguistically Different Concepts 82

5.3.4 Holonyms and Meronyms 83

5.3.5 Hypernyms and Hyponyms 84

5.4 Interoperability 86

5.5 Conclusion 86
CONTENTS

6 Application of the Framework 88

6.1 Example 1: BPMN and OSM ................. 89
   6.1.1 BPMN ........................................ 89
   6.1.2 OSM ........................................... 90
   6.1.3 Syntactic Similarity of BPMN and OSM .... 91
   6.1.4 Semantic Similarity of BPMN and OSM .... 93
   6.1.5 Structural Similarity between BPMN and OSM ... 93
   6.1.6 Ontological Analysis ......................... 95
   6.1.7 Interoperability of BPMN and OSM .......... 102

6.2 Example 2: MAS Metamodels ................. 105
   6.2.1 ADELFE ...................................... 106
   6.2.2 Gaia .......................................... 107
   6.2.3 PASSI ......................................... 108
   6.2.4 Syntactic Similarity of ADELFE, Gaia and PASSI . . . 109
   6.2.5 Semantic Similarity of Gaia, ADELFE and PASSI . . 111
   6.2.6 Ontological Analysis ......................... 113
   6.2.7 Interoperability of Gaia, ADELFE and PASSI .... 118

6.3 Example 3: MaSE and Prometheus .......... 120
   6.3.1 MaSE .......................................... 120
   6.3.2 Prometheus ................................. 120
   6.3.3 Syntactic Similarity of MaSE and Prometheus .. 121
   6.3.4 Semantic Similarity of MaSE and Prometheus ... 123
   6.3.5 Ontological Analysis ......................... 124

6.4 Example 4: Ecore and UML ................... 128
   6.4.1 Ecore and UML ........................... 128
   6.4.2 Syntactic Similarity of ECore and UML .... 129
   6.4.3 Semantic Similarity of ECore and UML ....... 131
CONTENTS

6.4.4 Structural Similarity of ECore and UML . . . . . . . . . . 132
6.4.5 Ontological Analysis . . . . . . . . . . . . . . . . . . . . . 132
6.5 Example 5: SPEM and BPMN . . . . . . . . . . . . . . . . . . 137
   6.5.1 SPEM and BPMN . . . . . . . . . . . . . . . . . . . . . . 137
   6.5.2 The Scenario . . . . . . . . . . . . . . . . . . . . . . . . . 141
6.6 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 144

7 Quality Assessment 145
   7.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . 145
   7.2 Quality Assessment . . . . . . . . . . . . . . . . . . . . . . . . 145
   7.3 Size and Complexity Measures . . . . . . . . . . . . . . . . . . 149
   7.4 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . 152

8 Conclusions 154
   8.1 Lessons Learned . . . . . . . . . . . . . . . . . . . . . . . . . . 155
      8.1.1 Syntactic Matching . . . . . . . . . . . . . . . . . . . . . 155
      8.1.2 Semantics Matching . . . . . . . . . . . . . . . . . . . . . 156
      8.1.3 Comparing Structures . . . . . . . . . . . . . . . . . . . . 158
      8.1.4 Automation . . . . . . . . . . . . . . . . . . . . . . . . . . 160
   8.2 Future Research Plans . . . . . . . . . . . . . . . . . . . . . . . 162
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Homomorphism between a model and the SUS (González-Pérez and Henderson-Sellers, 2009)</td>
<td>15</td>
</tr>
<tr>
<td>2.2</td>
<td>Type and token models (Kühne, 2005)</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>Relationship between a model and the SUS</td>
<td>18</td>
</tr>
<tr>
<td>2.4</td>
<td>Relationship between a language, model and model-unit-kind</td>
<td>19</td>
</tr>
<tr>
<td>2.5</td>
<td>The concept of notation (Gonzalez-Perez and Henderson-Sellers, 2007)</td>
<td>20</td>
</tr>
<tr>
<td>2.6</td>
<td>Misleading interpretation of a metamodel</td>
<td>22</td>
</tr>
<tr>
<td>2.7</td>
<td>Relationships among model, modelling language and metamodel</td>
<td>22</td>
</tr>
<tr>
<td>2.8</td>
<td>Bézivin and Gerbe (2001)’s interpretation of metamodel</td>
<td>23</td>
</tr>
<tr>
<td>2.9</td>
<td>Relationship between model, metamodel and modelling language, adapted from (Gonzalez-Perez and Henderson-Sellers, 2007) and (Kühne 2006b)</td>
<td>25</td>
</tr>
<tr>
<td>2.10</td>
<td>OMG four-layer modelling infrastructure, (Seidewitz, 2003)</td>
<td>26</td>
</tr>
<tr>
<td>2.11</td>
<td>instanceOf relationship between different levels of models using strict metamodelling</td>
<td>27</td>
</tr>
<tr>
<td>2.12</td>
<td>Value of an attribute at instantiation</td>
<td>28</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

2.13 Multiple instanceOf relationships between class and object . . . . 28
2.14 Linguistic and ontological relationships, adapted from González-
Pérez and Henderson-Sellers[2006] . . . . . . . . . . . . . . . . . . 31
2.15 Re-arrangements of linguistic and ontological relationships, adapted
from González-Pérez and Henderson-Sellers[2006] . . . . . . . . . 31
2.16 Example of powertype pattern (Henderson-Sellers and González-
Pérez[2005]) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

3.1 Roles of ontologies in software engineering (Happel and Seedorf,
2006) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
3.2 Three-level ontology architecture suggested by the Ontology Defi-
nition Metamodel of the Object Management Group (after Figure
12 in Henderson-Sellers[2011]) . . . . . . . . . . . . . . . . . . . . 41
3.3 Three domains defined by the category of users. For models cre-
ated in any one of these domains, a modelling language (ML)
is used. That ML is often the same for the three layers and
can be defined in one of a number of ways (after Figure 6.1 in
(Henderson-Sellers[2012]) . . . . . . . . . . . . . . . . . . . . . . . 42
3.4 Relationship between metamodels and domain ontologies . . . 44

4.1 Harmonization and matching . . . . . . . . . . . . . . . . . . . 50
4.2 The MDA architecture from CIM (computationally independent
model) to PIM (platform independent model) to PSM (platform
specific model) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
4.3 Metamodel transformation . . . . . . . . . . . . . . . . . . . . . 53
4.4 Metamodel interoperability . . . . . . . . . . . . . . . . . . . . . 54
4.5 Merging of concepts from BPMN and OSM . . . . . . . . . . . . 55
4.6 Bridging of concepts in BPMN and OSM . . . . . . . . . . . . . 57
<table>
<thead>
<tr>
<th><strong>LIST OF FIGURES</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4.7  Mapping of concepts in BPMN and OSM</td>
<td>58</td>
</tr>
<tr>
<td>4.8  Ontology merging method (Stumme and Maedche, 2001)</td>
<td>62</td>
</tr>
<tr>
<td>4.9  Classification of schema matching techniques (Rahm and Bernstein, 2001)</td>
<td>65</td>
</tr>
<tr>
<td>5.1  Metamodel interoperability framework</td>
<td>72</td>
</tr>
<tr>
<td>5.2  Structural neighbours of a class C (de Sousa Jr et al., 2009)</td>
<td>78</td>
</tr>
<tr>
<td>5.3  Activity and ActivityResource as holonym/meronym in BPMN</td>
<td>83</td>
</tr>
<tr>
<td>5.4  Event as hypernym in BPMN (OMG, 2011a)</td>
<td>84</td>
</tr>
<tr>
<td>6.1  Participant in OSM</td>
<td>96</td>
</tr>
<tr>
<td>6.2  Participant in BPMN</td>
<td>96</td>
</tr>
<tr>
<td>6.3  Property in BPMN</td>
<td>97</td>
</tr>
<tr>
<td>6.4  Property in OSM</td>
<td>97</td>
</tr>
<tr>
<td>6.5  Performer in BPMN</td>
<td>98</td>
</tr>
<tr>
<td>6.6  Relationship in BPMN</td>
<td>100</td>
</tr>
<tr>
<td>6.7  Interaction specification in BPMN</td>
<td>101</td>
</tr>
<tr>
<td>6.8  Interoperability of BPMN and OSM (dashed lines show mappings between elements of the two metamodels)</td>
<td>104</td>
</tr>
<tr>
<td>6.9  MAS metamodel adopted in ADELFE (Bernon et al., 2003)</td>
<td>106</td>
</tr>
<tr>
<td>6.10 MAS metamodel adopted in Gaia (Bernon et al., 2003)</td>
<td>108</td>
</tr>
<tr>
<td>6.11 MAS metamodel adopted in PASSI (Bernon et al., 2003)</td>
<td>109</td>
</tr>
<tr>
<td>6.12 Mapping/merging of concepts to our proposed merged metamodel</td>
<td>119</td>
</tr>
<tr>
<td>6.13 MAS metamodel adopted in OMaSE</td>
<td>121</td>
</tr>
<tr>
<td>6.14 MAS metamodel adopted in Prometheus</td>
<td>122</td>
</tr>
<tr>
<td>6.15 Merged metamodel for MaSE and Prometheus</td>
<td>126</td>
</tr>
<tr>
<td>6.16 Mapping from MaSE to merged metamodel</td>
<td>127</td>
</tr>
</tbody>
</table>
6.17 Mapping from Prometheus to merged metamodel .......... 127
6.18 Ecore metamodel ........................................ 129
6.19 UML class diagram metamodel ................................. 130
6.20 Merged metamodel for Ecore and UML2CD .............. 136
6.21 SPEM metamodel (partial) .................................. 139
6.22 BPMN metamodel (partial) ................................ 140
6.23 An example scenario using the merged metamodel .......... 143

7.1 Improvements in precision ....................................... 148

8.1 Number of concepts having structural similarity of some degree before using Equation 5.3 and after ................ 161
List of Tables

3.1 Similarities and differences between metamodels and ontologies . . . 45
3.2 Definitions of the terms used in table 3.1 . . . . . . . . . . . . . . . 46
5.1 Edit distance values between Process and Activity . . . . . . . . 76
5.2 SSM values between Process and Activity . . . . . . . . . . . . . 77
6.1 Syntactic similarity between BPMN and OSM . . . . . . . . . . . 92
6.2 Semantic similarity between BPMN and OSM . . . . . . . . . . . 94
6.3 Structural similarity (structSim) between BPMN and OSM . . . . 94
6.4 Candidate concepts of BPMN and OSM for interoperability . . . 95
6.5 Matched concepts of BPMN and OSM after ontological analysis . . 102
6.6 Syntactic similarity between Gaia and ADELFE . . . . . . . . . . 109
6.7 Syntactic similarity between Gaia and PASSI . . . . . . . . . . . . 110
6.8 Syntactic similarity between ADELFE and PASSI . . . . . . . . 110
6.9 Syntactically same concepts among Gaia, ADELFE and PASSI . . 110
6.10 Semantic similarity between Gaia and ADELFE . . . . . . . . . . 111
6.11 Semantic similarity between Gaia and PASSI . . . . . . . . . . . 112
6.12 Semantic similarity between ADELFE and PASSI . . . . . . . . . 112
6.13 Candidate concepts for merging among Gaia, ADELFE and PASSI . 112
6.14 SSM between MaSE and Prometheus . . . . . . . . . . . . . . . . . 122
**LIST OF TABLES**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.15</td>
<td>Synonyms for concepts in MaSE</td>
<td>123</td>
</tr>
<tr>
<td>6.16</td>
<td>Candidate concepts of MaSE and Prometheus for merging</td>
<td>123</td>
</tr>
<tr>
<td>6.17</td>
<td>Syntactic similarity between Ecore and UML2CD</td>
<td>131</td>
</tr>
<tr>
<td>6.18</td>
<td>Semantic similarity between Ecore and UML2CD</td>
<td>131</td>
</tr>
<tr>
<td>6.19</td>
<td>Structural similarity between Ecore and UML2CD</td>
<td>133</td>
</tr>
<tr>
<td>6.20</td>
<td>Candidate concepts between Ecore and UML2CD for interoperability</td>
<td>134</td>
</tr>
<tr>
<td>6.21</td>
<td>Similarities between BPMN and SPEM</td>
<td>141</td>
</tr>
<tr>
<td>7.1</td>
<td>Precision and recall of matching using old techniques</td>
<td>147</td>
</tr>
<tr>
<td>7.2</td>
<td>Precision and recall of matching using the interoperability framework</td>
<td>148</td>
</tr>
<tr>
<td>7.3</td>
<td>Size and complexity improvement</td>
<td>152</td>
</tr>
<tr>
<td>8.1</td>
<td>Effect of Equation 8.1 on structural similarity</td>
<td>161</td>
</tr>
<tr>
<td>8.2</td>
<td>BPMN Concepts List</td>
<td>186</td>
</tr>
<tr>
<td>8.2</td>
<td>BPMN Concepts List</td>
<td>187</td>
</tr>
<tr>
<td>8.2</td>
<td>BPMN Concepts List</td>
<td>188</td>
</tr>
<tr>
<td>8.2</td>
<td>BPMN Concepts List</td>
<td>189</td>
</tr>
<tr>
<td>8.2</td>
<td>BPMN Concepts List</td>
<td>190</td>
</tr>
<tr>
<td>8.3</td>
<td>ECore Concepts List</td>
<td>190</td>
</tr>
<tr>
<td>8.4</td>
<td>UML Class Diagram Concepts List</td>
<td>190</td>
</tr>
<tr>
<td>8.4</td>
<td>UML Class Diagram Concepts List</td>
<td>191</td>
</tr>
<tr>
<td>8.5</td>
<td>OSM Concepts List</td>
<td>191</td>
</tr>
<tr>
<td>8.6</td>
<td>Concepts List of Gaia, ADELFE and PASSI</td>
<td>192</td>
</tr>
<tr>
<td>8.7</td>
<td>OMaSE Concepts List</td>
<td>193</td>
</tr>
<tr>
<td>8.8</td>
<td>Prometheus Concepts List</td>
<td>193</td>
</tr>
<tr>
<td>8.9</td>
<td>BPMN Synonyms List</td>
<td>194</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>8.9</td>
<td>BPMN Synonyms List</td>
<td>195</td>
</tr>
<tr>
<td>8.10</td>
<td>OSM Synonyms List</td>
<td>195</td>
</tr>
<tr>
<td>8.11</td>
<td>Semantic Similarity between BPMN and OSM</td>
<td>196</td>
</tr>
<tr>
<td>8.11</td>
<td>Semantic Similarity between BPMN and OSM</td>
<td>197</td>
</tr>
<tr>
<td>8.11</td>
<td>Semantic Similarity between BPMN and OSM</td>
<td>198</td>
</tr>
<tr>
<td>8.11</td>
<td>Semantic Similarity between BPMN and OSM</td>
<td>199</td>
</tr>
<tr>
<td>8.11</td>
<td>Semantic Similarity between BPMN and OSM</td>
<td>200</td>
</tr>
<tr>
<td>8.11</td>
<td>Semantic Similarity between BPMN and OSM</td>
<td>201</td>
</tr>
<tr>
<td>8.11</td>
<td>Semantic Similarity between BPMN and OSM</td>
<td>202</td>
</tr>
</tbody>
</table>