The Meaning of UML Models

Greg O’Keefe

A thesis submitted for the degree of
Doctor of Philosophy at
The Australian National University

September 2008
revised February 2010
This thesis is my own original work.

Gregory C. O'Keefe
Acknowledgements

Thanks to my supervisor Raj Goré for pushing me to do my own thing, and to clearly justify its research value. He had faith in me even when I did not. Thanks also to Shayne Flint, Clive Boughton, Lynette Johns-Boast and Stephen Mellor for many helpful exchanges on engineering and model-driven software development. I talked modelling with innumerable people at the ECMDA-FA and MoDELS conferences in 2006, and I want to thank them all even though they are too many to name. Thanks to Jean-Marc Jézéquel and his team for welcoming me to their lab between these conferences. The ANU has provided me with financial support and an exciting intellectual environment to work in. For this, I am very grateful.
Abstract

The Unified Modelling Language (UML) is intended to express complex ideas in an intuitive and easily understood way. It is important because it is widely used in software engineering and other disciplines. Although an official definition document exists, there is much debate over the precise meaning of UML models.

In response, the academic community have put forward many different proposals for formalising UML, but it is not at all obvious how to decide between them. Indeed, given that UML practitioners are inclined to reject formalisms as non-intuitive, it is not even obvious that the definition should be “formal” at all. Rather than searching for yet another formalisation of UML, our main aim is to determine what would constitute a good definition of UML.

The first chapter sets the UML definition problem in a broad context, relating it to work in logic and the philosophy of science. More specific conclusions about the nature of model driven development are reached in the beginning of Chapter 2. We then develop criteria for a definition of UML. Applying these criteria to the existing definition, we find that it is lacking in clarity. We then set out to test the precision of the definition. The test is to take an apparently inconsistent model, and determine whether it really is inconsistent according to the definition.

Many people have proposed that UML models are graphs, but few have justified this choice using the official definition of UML. We begin Chapter 3 by arguing from the official definition that UML models are graphs and that instantiation is a graph homomorphism into an interpretation functor. The official definition of UML defines the semantics against its abstract syntax, which is in turn defined by a UML model. Chapters 3 and 4 prepare for our test by resolving this apparent circularity. The result is a semantics for the metamodel fragment of the language.

In Chapter 5, we find, contrary to popular belief, that the official definition does provide sufficient semantics to classify the example model as inconsistent. Moreover, the sustained study of the semantics in Chapters 3 to 5 confirms our initial argument that the semantic domain is graphs. The Actions are the building blocks of UML’s prescriptive dynamics. We see that they can be naturally defined as graph transformation rules. Sequence diagrams are the main example of descriptive dynamics, but we find that their official semantics are broken. The “recorded history” approach should be replaced, we suggest, by a graph-oriented dynamic logic.

Chapter 6 presents our early work on dynamic logic for UML sequence diagrams and further explores the proposed semantic repairs. In Chapter 7, guided by the criteria developed in Chapter 2, we critically survey the UML formalisation literature and conclude that an existing body of graph transformation based work known as “dynamic metamodelling” is very close to what is required.

The final chapter draws together our conclusions. It proposes a category theoretic construction to merge models of the syntax and semantic domain, yielding a type graph for the graph transformation system which defines the dynamic semantics of the language. Finally, it outlines the further work required to realise a satisfactory definition of UML.
Intended Audience

The dissertation is deliberately written for workers in model driven development who do not necessarily have a strong mathematical background. Where mathematics is used, the ideas are mostly explained in plain English as well. Some of the mathematical ideas are not fully developed, but merely indicate what existing research may be relevant.
Contents

1 Logic and Modelling 1
  1.1 Logic .................................................. 2
  1.2 Conceptual Modelling .................................. 3
  1.3 Models and Statements .................................. 4
  1.4 The UML Model ......................................... 8

2 The Purpose of UML and Criteria for a Definition 11
  2.1 What is a Model? ....................................... 12
  2.2 Working with Ideas ..................................... 19
  2.3 Diagrams and Languages ............................... 20
  2.4 Semantics ............................................... 24
  2.5 Metamodelling and Reflection .......................... 27
  2.6 Semantic Variation Points .............................. 29
  2.7 The UML Definition Evaluated .......................... 30
  2.8 Conclusion ............................................. 32

3 Semantic Preliminaries 34
  3.1 The Semantic Domain .................................... 35
  3.2 Object Diagrams and System States ....................... 38
  3.3 Class Diagrams and Models ................................ 41
  3.4 The Metamodel .......................................... 50
  3.5 Reflection ............................................... 54
  3.6 Reconciliation of the Two Views ......................... 59
  3.7 Conclusion ............................................. 64

4 Breaking the Symmetry of Association Ends 66
  4.1 Naive Symmetry ......................................... 67
  4.2 Derived Property Values ................................ 73
  4.3 Conclusion ............................................. 77

5 Testing the Definition 79
  5.1 An Apparently Inconsistent Model ....................... 80
  5.2 The Definition of UML and our Class Model ............. 81
  5.3 UML Dynamics .......................................... 87
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4</td>
<td>Emergent Behavior</td>
<td>92</td>
</tr>
<tr>
<td>5.5</td>
<td>Causation and Behavior</td>
<td>98</td>
</tr>
<tr>
<td>5.6</td>
<td>Graph Transformation and Action</td>
<td>105</td>
</tr>
<tr>
<td>5.7</td>
<td>Consistency</td>
<td>109</td>
</tr>
<tr>
<td>6</td>
<td>Dynamic Logic for UML Consistency</td>
<td>115</td>
</tr>
<tr>
<td>6.1</td>
<td>Dynamic Logic</td>
<td>116</td>
</tr>
<tr>
<td>6.2</td>
<td>The Model as Dynamic Logic Formulae</td>
<td>119</td>
</tr>
<tr>
<td>6.3</td>
<td>Weaving and Consistency</td>
<td>124</td>
</tr>
<tr>
<td>6.4</td>
<td>Interactions as Dynamic Formulae</td>
<td>128</td>
</tr>
<tr>
<td>7</td>
<td>The UML Formalisation Literature</td>
<td>130</td>
</tr>
<tr>
<td>7.1</td>
<td>Relational and Entity-Relationship Models</td>
<td>131</td>
</tr>
<tr>
<td>7.2</td>
<td>Graph Transformation</td>
<td>136</td>
</tr>
<tr>
<td>7.3</td>
<td>Logics</td>
<td>148</td>
</tr>
<tr>
<td>7.4</td>
<td>Other Techniques</td>
<td>154</td>
</tr>
<tr>
<td>7.5</td>
<td>Comparison and Conclusion</td>
<td>163</td>
</tr>
<tr>
<td>8</td>
<td>How UML Should be Defined</td>
<td>165</td>
</tr>
<tr>
<td>8.1</td>
<td>What a UML Model Means</td>
<td>166</td>
</tr>
<tr>
<td>8.2</td>
<td>Towards a Well Defined and Understood UML</td>
<td>171</td>
</tr>
<tr>
<td>8.3</td>
<td>Findings</td>
<td>175</td>
</tr>
<tr>
<td>8.4</td>
<td>Conclusion</td>
<td>178</td>
</tr>
</tbody>
</table>