FORMAL DEVELOPMENT OF OPEN DISTRIBUTED SYSTEMS: INTEGRATION OF UML AND PVS

Doctoral Dissertation

by

Demissie Bediye Aredo

Submitted to the Faculty of Mathematics and Natural Sciences, at the University of Oslo in partial fulfilment of the requirements for the degree Dr. Scient. in Computer Science

August 2004
To my sister

Dirribee Bediye Aredo
Abstract

In this thesis, a research work conducted on formalization of the Unified Modeling Language (UML) notations is reported. Formal semantic definitions for UML modeling constructs are provided by systematically transforming them into suitable and well-defined entities in the specification language of the Prototype Verification System (PVS). As UML is an industry standard modeling language consisting of several aspects of object-oriented modeling techniques, it is not feasible to cover all semantic aspects of the UML notations. Static structural models (class diagrams), and dynamic behavioral models (sequence and statecharts diagrams) are the main focus of the thesis.

A strategy for deriving semantic models directly from UML graphical models, and a framework for integrating the UML modeling techniques with formal analysis techniques of the PVS environment is proposed. Transformation of UML graphical models into PVS specifications results in semantic models that are amenable to rigorous analysis, thereby overcoming limitations inherent in the semi-formal UML notations. This paves a way for developing formal techniques that support rigorous development of distributed systems through transformation and enhancement of OO modeling techniques.

Integrating semi-formal graphical modeling techniques with a mathematically based development method(s) results in a development framework that supports rigorous model analysis, while useful features of the graphical modeling techniques are preserved. Automation of the derivation of formal specifications from graphical UML models based on the proposed semantics is vital as model analysis usually involves manipulation of large volume of information. In this regard, we have developed a prototype of a CASE tool that integrates the general-purpose PVS tool set with a UML CASE tool. The tool supports formal development of distributed systems from requirement capture to code generation and allows developers to deal with the graphical models they have developed while the rigorous analysis is performed at the back-end.

This work contributes to the ongoing effort to provide formal semantics for the UML notations, with the aim of clarifying and disambiguating the language as well as supporting development of semantically-based CASE tools. Moreover, it allows exploitation of the synergy between formal methods (FM) and semi-formal modeling languages, which in turn improves the use of FMs in industrial settings.
Acknowledgements

This work was financially supported by a grant from the Research Council of Norway under the research program for distributed IT-systems. Additional funding was provided by the Department of Informatics, University of Oslo, Norway. The work was carried out at the Department of Informatics, University of Oslo, and the Institute for Energy Technology (IFE), Halden, Norway, from February 1998 – March 2001.

I would like to thank my supervisors Prof. Olaf Owe, and Dr. Wenhui Zhang for their follow-ups, encouragements, and invaluable comments without which this work would not have come to completion.

I am indebted to my earlier supervisor Prof. Ketil Stølen who guided me through the early months of ‘chaos’ and confusion. Colleagues who worked on the ADAPT-FT project in general, and Drs. Issa Traoré, Isabelle Ryl, and Einar Johnsen in particular deserve special thanks for their support.

I always remember the informal and friendly atmosphere I enjoyed with the personnel and academic staff at the Department of Informatics, University of Oslo. I am grateful to all staff members at the Department of Informatics, in particular Mr. Narve Trædal for his courage in dealing with the administrative component of the thesis work, most of the formal procedures were unnoticeable.

I had the pleasure of staying at IFE, in Halden, during my PhD candidacy. The people at IFE are all wonderful, and their support made the completion of this work possible. I am also grateful to the Research Council of Norway for the financial support – a crucial component for the successful completion of this thesis.

I am also thankful to the Department of Computer Science, at the University of Kent at Canterbury (UKC), for allowing me to use the facilities in their Computing Laboratory. Dr. Stuart Kent and Prof. Keith Mander deserve special thanks for expressing their interest in my work, and above all for making my stay at UKC so comfortable.

Finally, my most sincere thanks go to my family for their patience, and support in any way possible throughout the years. They had suffered my absence.

August 2004, Oslo, Norway
Demissie B. Aredo
# Table of Contents

Abstract i

Acknowledgements iii

Table of Contents v

Executive Summary vii

1 Introduction 1
   1.1 Background .................................................. 1
   1.2 The Problem Statement ..................................... 3
   1.3 Formal Methods ............................................. 4
   1.4 Involved Notations and Formalisms ......................... 6
      1.4.1 The Prototype Verification System ................. 7
      1.4.2 The Unified Modeling Language ................... 8
   1.5 Formal Semantic Definitions ............................. 9

2 Formalization of UML Notations 13
   2.1 Motivation .................................................. 13
   2.2 Formalization Approaches .................................. 15
   2.3 State-of-the-Art ............................................ 16
   2.4 Formalization Issues ....................................... 19
      2.4.1 Composition of UML Models ..................... 19
      2.4.2 Checking Consistency of UML models ........ 20
      2.4.3 Refinement ....................................... 21
      2.4.4 Formal Reasoning ................................. 22

3 Summary of Contributions 23
   3.1 Formal Development of Distributed Systems ............. 24
   3.2 Semantics of Structural UML Models .................... 26
   3.3 Semantics of UML Sequence Diagrams .................... 27
   3.4 Semantics of UML Statecharts in PVS .................. 28
   3.5 Tracking Inconsistencies in Integrated Platforms ..... 29
   3.6 Enhancing Structured Reviews with Model-Based Verification ...... 30
3.7 Summary of Major Achievements ........................................... 31
  3.7.1 Semantic Definitions for UML Notations .......................... 31
  3.7.2 A Framework for Formal Development ODSs ....................... 32
  3.7.3 CASE Tool Support ...................................................... 34

4 Conclusions and Future Work ............................................. 37
  4.1 Conclusions .................................................................. 37
  4.2 Future Work ............................................................... 38

A Formal Development of Open Distributed Systems: Towards an Integrated Framework 47

B Towards formalization of Structural UML Models in PVS 61

C An Integrated Framework for Formal Development of Open Distributed Systems 77

D A Framework for Semantics of UML Sequence Diagrams in PVS 95

E Semantics of UML Statecharts in PVS .................................. 119

F Tracking Inconsistencies in an Integrated Platform .................. 135

G Enhancing Structured Review with Model-based Verification 157

H Formal System Development Using Method Integration: a Case Study 193
Executive Summary

The Unified Modeling Language (UML) [79, 91, 11] is an important industry standard (standardized by the Object Management Group (OMG)) for modeling software systems that has rapidly become popular among the software communities. The popularity of UML can largely be attributed to its graphical and intuitively understandable visual notations, and its capabilities to support encapsulation, data abstraction, extensibility, and reusability. It is indisputable that the UML reflects some of the best modeling experiences and incorporates notations that have proven useful in practice. Using UML for effective formal analysis in industrial setting could, however, be problematic due to the lack of precise semantic definitions for its graphical notations. The lack of firm semantic foundations for UML modeling constructs can lead to a number of problems: understanding of the models can be more apparent than real; developers may waste considerable time resolving disputes over usage and interpretation of notations; and model analysis and communication could be difficult [42, 100]. Defining precise semantics of a modeling language is a prerequisite for developing semantically based CASE tools, and for model communication.

The primary objective of this thesis is to investigate semantics of UML description techniques, to make them amenable to rigorous model analysis by transforming them into semantic models. The specification language of the Prototype Verification System (PVS) [81, 82, 97] is used as an underlying semantic domain. A general framework for transforming graphical UML models into formal descriptions in the PVS specification language is also proposed. This paves a way for formal development of systems through a systematic transformation of UML models. The framework is used to transform UML modeling constructs, namely, static structural modeling constructs such as class diagrams, and dynamic behavioral modeling constructs such as sequence diagrams, and statecharts into semantic models in the PVS specification language.

Transforming UML models into corresponding semantic models in the PVS specification language enables rigorous model analysis using the formal techniques of PVS and its tools such as type-checker, theorem-prover, and model-checker. Analysis of the resulting semantic models of reasonably large systems may involve processing of large size of software artifacts, which calls for a mechanized support - a criteria for whole-scale application of formal analysis techniques. In this regard, we have developed a platform
that integrates a UML CASE tool and the PVS tool set. The platform supports formal development of distributed systems from requirement capture to code production and allows system designers to analyze the graphical models they have developed, while the formal stuff is processed at the back-end.

This work is part of a long-term vision to explore how formal methods can be used to underpin practical tools for analyzing UML models. It contributes to the ongoing effort to meet the needs of software industry - improved quality and reliability, and lower production cost - by providing mathematical basis for the UML modeling techniques with the aim of clarifying the semantics of the language as well as supporting the development of semantically-based CASE tools.

**Organization of the Thesis**

The thesis is organized into several chapters. In Chapter 1, the problem to be addressed is introduced. Moreover, relevant aspects of formal methods and semantics, and modeling notations and methods involved in this work, namely the UML and the PVS are briefly introduced. In Chapter 2, some of the central concepts of formalization of OO modeling techniques are discussed. A literature survey of formalization of OO modeling languages with emphasis put on the formal semantics for UML notations is presented. In Chapter 3, a brief summary of the publications constituting the thesis and the main achievements are presented, whereas full texts of the publications are included as appendices. Finally, in Chapter 4, concluding remarks and future research issues are presented.

**List of Contributions**

The thesis consists of a number of stand-alone publications each of which is addressing a specific research issue. A roman-numbered list of the publications is given below. In later sections, we refer to the publications by their respective numbers in the list. The publications are listed in the order they have been summarized in chapter 3 to obtain a logical flow. The versions of the publications included in the sequel may differ from the published ones due to minor editorial fixes, reformatting necessary to give the thesis a uniform layout, and in some cases discussions of new issues.


The publications coauthored with Prof. Stølen were published while he was my principal supervisor. The cooperation with Dr. Traoré started when he held a one year post-doc position associated with the ADAPT-FT project, which also included my own doctoral fellowship.

**Other Related Publications**

My contributions to the following publications are results of the work done in the context of the thesis project, but not included in the thesis. Cooperation with the coauthors started at the time they were working on the ADAPT-FT project¹.

¹http://www.ifi.uio.no/adapt/
