Available online at www.sciencedirect.com
SCIENCE DIRECT®

Electronic Notes in Theoretical Computer Science

ELSEVIER Electronic Notes in Theoretical Computer Science 130 (2005) 263–280
www.elsevier.com/locate/entcs

Coherently Explaining UML Statechart and Collaboration Diagrams by Graph Transformations

Paul Ziemann [1,2]

Department of Computer Science
University of Bremen
Bremen, Germany

Karsten Hölscher[3]

Department of Computer Science
University of Bremen
Bremen, Germany

Martin Gogolla[4]

Department of Computer Science
University of Bremen
Bremen, Germany

Abstract

In this paper we continue our work on the formalization and validation of UML models by means of graph transformation systems. We here concentrate on statechart and collaboration diagrams albeit our approach covers use case, class, object, and sequence diagrams as well. The statechart and collaboration diagrams describe the operations of the underlying class diagram and include OCL expressions as guards and parts of message expressions. We illustrate in detail the generation of graph transformation rules for the statechart and collaboration diagrams.

Keywords: UML, graph transformation, integrated formal semantics, OCL

1571-0661/$ – see front matter
doi:10.1016/j.entcs.2005.03.014

1 Introduction

The Unified Modeling Language (UML) has recently become a widely accepted standard for the visualization, specification, construction, and documentation of object-oriented software systems. It is well established and used in industry as well as in research. The UML is a graphical language that comprises a number of different diagram types for different purposes. The syntax of these diagram types is defined in the UML metamodel [9]. But the semantics of the language constructs is only given in natural language. As the UML is supposed to support a software engineer in constructing precise models, a formal foundation for UML is needed. The graphical notation is enhanced by the Object Constraint Language (OCL), which permits to formulate constraints that cannot be expressed by the diagrams in a textual way. OCL is formally defined in [11]. Currently, the 2.0 version of UML is about to be finalized [10] but the language definition will still be informal.

By translating a given UML model into a graph transformation system we provide an integrated formal semantics for a large part of UML. Integrated means that a model in our approach may comprise use case, class, object, statechart and interaction (collaboration and sequence) diagrams. We stick to UML 1.5 but UML 2.0 likewise includes the UML concepts covered by us, albeit some details and the naming have changed in some cases. In particular, collaboration diagrams are called communication diagrams in UML 2.0.

The graph transformation system consists of graph transformation rules and a working graph, which represents a snapshot of the current state (hence called system state) of the modeled system. The system state changes during a run of the system, i.e. graph transformation rules are applied rewriting parts of the working graph. Using our approach modelers can validate a system model by performing system runs and comparing their expectations with the results of these runs.

In this paper we present the fundamental concept of system states and the translation of a given UML model into a graph transformation system focusing on the rules evolving from statechart and collaboration diagrams. An important aspect for us is that our approach integrates the OCL. OCL expressions that appear in the model are used in rules as well. When applying

[1] Research partially supported by the EC Research Training Network SegraVis (Syntactic and Semantic Integration of Visual Modeling Techniques) and by the German Research Foundation (DFG) as part of the Collaborative Research Centre 637 *Autonomous Cooperating Logistic Processes — A Paradigm Shift and its Limitations* and the project UML-AID.
[2] Email: ziemann@tzi.de
[3] Email: hoelscher@tzi.de
[4] Email: gogolla@tzi