Formal Modeling and Analysis of Object Oriented Systems using Triple Graph Grammars

Wafa Chama, MISC Laboratory, University Constantine 2-Abdelhamid Mehri, Constantine, Algeria

Allaoua Chaoui, MISC Laboratory, University Constantine 2-Abdelhamid Mehri, Constantine, Algeria

Seidali Rehab, MISC Laboratory, NTIC Faculty, University Constantine 2-Abdelhamid Mehri, Constantine, Algeria

ABSTRACT

This paper proposes a Model Driven Engineering automatic translation approach based on the integration of rewriting logic formal specification and UML semi-formal models. This integration is a contribution in formalizing UML models since it lacks for formal semantics. It aims at providing UML with the capabilities of rewriting logic and its Maude language to control and detect incoherencies in their diagrams. Rewriting logic Maude language allows simulation and verification of system’s properties using its LTL model-checker. This automatic translation approach is based on meta-modeling and graph transformation since UML diagrams are graphs. More precisely, the authors have proposed five meta-models and three triple graph grammars to perform the translation process. The authors have used Eclipse Generative Modeling tools: Eclipse Modeling Framework (EMF) for meta-modeling, Graphical Modeling Framework (GMF) for generating visual modeling tools and TGG Interpreter for proposing triple graph grammars. The approach is illustrated through an example.

Keywords: EMF, GMF, Graph Transformation, LTL Model Checker, Maude Specifications, TGG Interpreter, UML Models

1. INTRODUCTION

With the revolution of computer science, the complexity of software systems has been grown. Several efforts have been done to improve the modeling techniques. UML (Unified Modeling Language) (Seidl et al., 2015) has become the graphical standard for high quality design, which recognized by the Object Management Group. UML offers a collection of both structural and behavioral diagrams which are suitable for defining the different views of a software system.
object oriented system is modeled using a family of UML diagrams, where each one defines a
dynamic feature (Seidl et al., 2015). Class diagram is the main building block expressing
the static view in a system with classes and relationships between them. Statechart diagram
specifies the dynamic structure of a system by describing the lifecycle of a single object during
its lifetime in response to events. Communication diagram is an interaction diagram (subset of
behavior diagrams) which can model the interactions and relationships between a set of objects
by sending and receiving messages (Seidl et al., 2015). These three types of UML diagrams are
enough to represent complementary aspects of a system.

In spite of its advantages, UML suffers from a lack of formal semantics to control and detect
incoherencies either between a single model diagrams or between elements in the same diagram.
This weakness can generate ambiguous models and also creates several problems.

The combined use of graphic methods and rigorous techniques based on mathematical theo-
ories is required in software development projects. Formal methods offer high level of rigorous
and analyzable descriptions allowing simulation and verification process. Maude is the powerful
rewriting logic formal specification for concurrent systems (Clavel et al., 2015). It offers many
possibilities of verification and validation through its own LTL model checker (Eker et al., 2003).

In this paper, our aim is bridging the gap between UML graphic semi-formal models and
Maude formal textual specifications. The idea is to satisfy UML’s needs by the use of formal
specifications and model checking (see Figure 1). Our approach facilitates the early detection
of errors such as deadlock and livelock in the whole model, or incoherencies between an UML
model diagrams, or between elements in the same diagram.

Our approach provides a formal framework that allows an automatic translation of UML
models into its semantically equivalent Maude specifications. To achieve our goal, we have
used Triple Graph Grammars (TGG) suite (Kindler & Wagner, 2007), interpreter (TGG, 2014)
and a set of Eclipse tools. Firstly, we have implemented meta-models for UML class, statechart,
communication diagrams, correspondences and Maude language using EMF: Eclipse Modeling
Framework (Steinberg et al., 2009). Then, using GMF: Graphical Modeling Framework (Gouyette,
2010), we have produced three visual tools for supporting our UML diagrams. Finally, we have
proposed fourteen TGG transformation rules to perform the mapping. Once the semantically
equivalent Maude code is generated, the LTL Maude model checker can be used.

Figure 1. Our proposed approach
Network Analyzer Development Comparison with Independent Data and Benchmark Products: Network Traffic Utilization
www.igi-global.com/article/network-analyzer-development-comparison-independent/44966?camid=4v1a

VANET Authentication Using Trust Distribution in a Clustered Environment
www.igi-global.com/chapter/vanet-authentication-using-trust-distribution/61751?camid=4v1a