A Model Transformation Approach for Specifying Real-Time Systems and Its Verification Using RT-Maude

Messaoud Bendiaf, Department of Computer Science University of Biskra, Biskra, Algeria
Mustapha Bourahla, Department of Computer Science University of M’sila, M’sila, Algeria
Malika Boudia, Department of Computer Science University of Biskra, Biskra, Algeria
Seidali Rehab, MISC Laboratory, University of Constantine 2 - Abdelhamid Mehri, Constantine, Algeria

ABSTRACT

Real-time systems must be properly validated and verified before their manufacturing and deployment in order to increase their reliability and reduce their maintenance cost. Models have been used for a long time to build complex systems, in virtually every engineering field. This is because they provide invaluable help in making important design decisions before the system is implemented. In this paper, the authors propose an approach based on model transformation to apply formal verification techniques to demonstrate the correctness of system designs. At the first step, they describe real-time systems by state chart (machine) diagrams, as source models to generate RT-Maude models (target models). The second step is to use the result models to verify the real-time systems against specified LTL properties using Maude LTL Model-Checker. This approach is illustrated through an example.

KEYWORDS

Formal Specification and Verification, Model Checking, Model Transformation, Model-to-Text Transformation, Real-Time Systems, Rewriting Logic, RT-Maude, Triple Graph Grammars

1. INTRODUCTION

With the advance of new technologies such as multi-core technology, real-time systems are becoming more complex and the supporting algorithms and protocols are becoming more sophisticated. Therefore, there is a clear need for different kinds of automated formal analysis that can be used to analyze systems behaviors before the arduous task of proving the correctness of a complex system is attempted.

Such real-time systems are very critical and complex. The formal specification in early phases reduces the rate of error prone in later phase. The specified user requirements can be easily validated and verified before detailed design and development.

Temporal Logic is used to specify functional or nonfunctional requirements. Temporal logic is the collection of rules to specify these requirements. There are different types of temporal logic like Linear Temporal Logic (LTL), Computation Tree Logic (CTL), Extended Computational Tree Logic, etc. (Bouyer, 2009). The model checking tools are used to specify the user requirements in a formal way and the functionality of the model is verified against these requirements.

When designing real-time systems, it is often difficult to guarantee that the specification of a system actually fulfills the needs, i.e., whether it satisfies the design requirements. Especially for

DOI: 10.4018/IJITWE.2017100102
critical applications, for example in real-time domains, there is a need to prove that the designed system has certain properties under certain conditions.

Model transformation plays a central role in Model-Driven Engineering (MDE) and supporting bidirectionality is a current challenge with important applications. Triple Graph Grammars (TGGs) are a formally founded, bidirectional model transformation language shown by numerous case studies to be promising and useful in practice. TGGs have been researched for more than 15 years and multiple TGG tools are under active development.

A Real-Time Maude specification provides a precise high-level mathematical model of the system that can be directly simulated in Real-Time Maude, and that can be subjected to search and model checking analysis to explore all possible behaviors from a given initial state to systematically search for “corner case” bugs.

This paper is organized as follows. Section 2 outlines some related works. In section 3, we recall some basic notions about Real-Time Maude and Model Transformation Based on TGGs. We describe in detail the proposed approach in section 4. In section 5, we illustrate our approach through an example. Finally, section 6 concludes the paper and gives some perspectives of this work.

2. RELATED WORK

In (Bourahla, 2009), author has presented an approach for the analysis of real-time embedded systems. The analysis is based on the verification of the schedulability of tasks in addition to the verification of requirements on the overall system. In (Sammi, 2010), authors have presented a survey about the formal specification languages which are used to specify Real-time systems, then they have compared formal languages which are used to specify real-time systems such as VDM++, RTSJ, ASTRAL. A comparison criterion was established to select the best suitable specification languages for Real-Time systems.

In (Boucherit, 2011), authors have proposed a new formal approach based on rewriting logic, in which they attempted to bridge the gap between agent based system analysis and its specification. In addition, their approach included a well-known and effective verification technique, model checking, and allowed independent of the used formalism to verify an important number of properties deemed relevant on critical system based on agent paradigm.

In (Basit, 2014a), a methodology has been proposed for the modeling and verification of embedded systems in parallel and distributed environments. Authors have demonstrated the suitability of the framework by applying it on the case study of embedded security system. The case study was modeled using SysML’s state machine diagrams and then semantics were described to translate these state machine diagrams to DVE (DiVinE) based model. The translated model is verified against specified LTL properties using DiVinE.

The following work of (Basit, 2014a), authors have proposed a methodology to use both types of model checkers (sequential and parallel) for verification of real-time system that are graphically modeled using SysML. The case study was modeled using state machine diagram of SysML and verified against specified timed properties using UPPAAL while the untimed properties are verified using DiVinE (Basit, 2014b).

In (Kristin, 2010), author has provided a survey of a perspective on the formal verification technique of linear temporal logic (LTL) symbolic model checking, from its history and evolution leading up to the state-of-the-art. In addition, the author has delivered a complete end-to-end analysis embracing a user’s perspective by applying each step to a real-life aerospace example. In (Jussi, 2014),
Employing Graph Network Analysis for Web Service Composition
www.igi-global.com/article/employing-graph-network-analysis-web/2635?camid=4v1a

Web-Enabled Technologies Assessment and Management: Critical Issues
www.igi-global.com/chapter/web-enabled-technologies-assessment-management/26106?camid=4v1a