Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 320 (2016) 21–34

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

An Illustrative Use Case of the DIVERSITY Platform based on UML Interaction Scenarios

Mathilde Arnaud[1] Boutheina Bannour[2] Arnault Lapitre[3]

CEA, LIST, Laboratory of Model Driven Engineering for Embedded Systems, Point Courrier 174, 91191 Gif-sur-Yvette, France

Abstract

DIVERSITY is a multi-purpose customizable platform based on symbolic execution. DIVERSITY has been designed for the purpose of managing the diversity of different semantics, but also the diversity of possible analyses based on symbolic execution. In this paper, we show how the input language of DIVERSITY can be used to encode the semantics of UML scenarios which include timing constraints expressed with the VSL language (standardized in the UML profile for embedded systems MARTE). We apply symbolic execution on practical scenarios of a system-on-chip example[3] in order to select test behaviors using an advanced exploration strategy implemented in DIVERSITY.

Keywords: Symbolic execution and tools, Modeling languages semantics, UML Scenario-based Interactions, VSL/MARTE timing constraints, Test selection strategy and coverage.

1 Introduction

Symbolic execution was first defined for programs [15]. The underlying concept consists in executing programs, not for concrete numerical values but for symbolic parameters, and computing logical constraints on those parameters at each step of the execution. Symbolic execution allows computing semantics of programs or models and representing them efficiently in an abstract manner. Model-based testing (MBT) is one of the most popular applications of symbolic execution[11,10,14,2]. Symbolic execution has been used to select some parts of the resulting symbolic representation of models, which may be infinite due to the presence of unbounded loops for example, according to some coverage objective. Test data are then generated from those chosen parts using constraint solving techniques. The increased efficiency

[1] Email: mathilde.arnaud@cea.fr

[2] Email: boutheina.bannour@cea.fr

[3] Email: arnault.lapitre@cea.fr

[3] Work partially supported by the European project OpenES www.openes-project.org

http://dx.doi.org/10.1016/j.entcs.2016.01.003

1571-0661/

of solvers in recent years [9,8,4] has helped symbolic execution to be adopted more widely for this purpose. Many symbolic execution based tools for formal treatments have been developed for diverse usages, for example the ones used in (Model-Based) Testing cited in the following survey [1]. Compared to these tools, the objective of the DIVERSITY platform is to offer an extensible platform to take into consideration various formal analysis possibilities. For this, DIVERSITY provides a common symbolic execution platform:

- generic enough to take into account semantics of a wide range of models;
- extensible to allow customizing of the basic symbolic treatments to implement specific formal functionalities (e.g. MBT algorithms, exploration strategies, etc.).

DIVERSITY is on its way to becoming an Eclipse open-source project [6]. In this paper, we give a brief introduction to DIVERSITY, and in particular we provide an example of its use. To illustrate extensibility, we show how an adaptation of the exploration strategy Hit-or-Jump [5],a heuristic whose aim is to achieve targeted test coverage, can be easily integrated into the customizable symbolic execution process. To illustrate the generality of DIVERSITY, we show how it provides interesting support of the semantics of the UML sequence diagrams [13]. Sequence diagrams display the UML graphical language used to describe the interaction behavior of system components. First, we have identified a subset of the input language of DIVERSITY to encode this interaction language. In fact, DIVERSITY provides a pivot language called *xLIA*(executable Language for Interaction and Architecture) which is a generic language with a variety of primitives which allow encoding a diversity of classical semantics. In particular, xLIA supports classical automata syntax involving symbolic data and communication actions. For MBT purposes, we have in previous work [3] provided a formal treatment of the semantics of UML sequence diagrams which involve timing constraints, specified using the Value Specification Language (VSL, standardized in the UML profile for MARTE [12]), by translating them into a kind of transition-labeled symbolic automata. In this paper, we extend this work by showing how these automata can be implemented in xLIA in an efficient way using asynchronous communication mechanisms and facilities to encode MARTE timing constraints. TIOSTS can be easily encoded as a subset of xLIA with a simple mechanism for communication. It appeared while implementing the translation mechanism described in [3] that it is not an efficient representation for symbolic execution in terms of performance, especially the message representation resulted in unnecessary computations. Thus it is useful to choose a different way of translating messages that alleviates this effect. We want to put particular emphasis on describing the translation mechanism and the use of DIVERSITY for coverage analysis, as an illustration of the more generic abilities of the tool.

Overview. Section 2 presents the transition-labeled automata in xLIA which are used to encode sequence diagrams and their associated symbolic semantics. Section 3 presents the symbolic execution process in DIVERSITY and how it is coupled with the Hit-or-Jump exploration strategy. Section 4 gives an example of the specification of a timed interaction behavior of a system-on-chip using sequence diagrams.