Reo2MC: a Tool Chain for Performance Analysis of Coordination Models

Farhad Arbab, Sun Meng, Young-Joo Moon, Marta Kwiatkowska, Hongyang Qu

Reo2MC: a Tool Chain for Performance Analysis of Coordination Models

Farhad Arbab[1] and Sun Meng[1] and Young-Joo Moon[1] and Marta Kwiatkowska[2] and Hongyang Qu[2]

[1] CWI, Amsterdam, The Netherlands
[2] Oxford University Computing Laboratory, UK

Abstract. In this paper, we present Reo2MC, a tool chain for the performance evaluation of coordination models. Given a coordination model represented by a stochastic Reo connector, Reo2MC is able to automatically generate the Quantitative Intentional Automaton (QIA) as its operational semantics, and the corresponding Continuous-Time Markov Chain (CTMC), which allows us to apply existing CTMC tools, e.g., PRISM, for performance analysis of Reo connectors. In support of understanding connector behavior and performance properties, the tool also provides the graphical representation of the QIA and Markov Chains.

1 Introduction

The growing complexity and importance of coordination models in software applications necessarily lead to a higher relevance of performance issues for coordinators during development of systems. In this context, the performance of such models plays an important role in the quality of the final software system. Unfortunately, the lack of tools that support the performance analysis of coordination models makes it difficult to automatically verify the performance properties of coordination models.

Reo [2] is a channel-based exogenous coordination model wherein complex coordinators, called connectors, are compositionally built out of simpler ones, called *channels*. Stochastic Reo [3] extends Reo by annotating with stochastic properties, such as processing delays on channels and arrival rates of data/requests from outside; the arrival rates are influenced by the environment while the processing delays are influenced by the primitive behavior of connectors. With these stochastic properties, Stochastic Reo specifies not only the behaviour of connectors, but also the behaviour of the environment. The operational semantics of Stochastic Reo is provided by QIA which are an extention of Constraint Automata (CA) [4]. The states of a QIA show the configurations of a system and their pending I/O requests, relevant to the corresponding processing and data arrivals, respectively. This paper introduces Reo2MC, a tool for automatically generating CTMCs out of Stochastic Reo connectors. Reo2MC has been implemented using QIA as an intermediate bridge for translating Reo circuits to homogeneous CTMCs (see [3]).

Reo2MC is a plug-in for the Eclipse Coordination Tools (ECT) [1]. It is fully automated, unlike theorem proving methods that may require user assistance. The tool is able to automatically derive the QIA semantics of Reo models and their corresponding CTMCs. The generated CTMCs can be represented in different textual file formats,

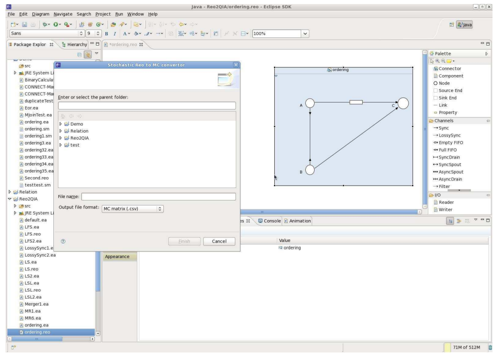

Fig. 1. A Snapshot of Reo2MC

to be used as input files for other tools, such as PRISM and Maple, to analyze the performance of the connector. The integration with ECT makes it possible to specify a connector by the graphical editor inside ECT or synthesize it from other specifications like UML or BPMN automatically, and then generate its corresponding QIA and CTMC to check its behavioral properties and analyze its performance. Figure 1 depicts a snapshot of the Reo2MC tool in ECT.

2 Tool Usage

Reo2MC works as an Eclipse application through a graphical user interface (see Figure 1). The execution flow is depicted as in Figure 2: the user provides as an input a Reo circuit, which is obtained either by using the graphical editor in ECT, or by automatic synthesis from other specifications, such as UML sequence diagrams or BPMN models, the tools for which are also provided in ECT, and a textual description of the stochastic constraints on the connector and its environment. Once all this input has been provided, the model can be automatically translated into QIA and CTMC, both represented in XML. The GMF framework in Eclipse is used to generate the graphical representation of QIA and CTMCs, and the XML files of the generated CTMCs can be parsed into several other file formats, which are used as input to PRISM, MATLAB and Maple to analyze the performance of connectors.

3 Experiment and discussion

In order to demonstrate the tool chain, we use PRISM [5] to construct an experiment using a small example: Mary bought a new mobile phone and wants to transfer her contracts stored in the old phone to the new phone. To keep the model simple, we assume there are three contacts that need to be transferred. Figure 3 shows the Reo circuit connecting two phones. The new phone “CM2” sends three requests sequentially, each of which is used to get a contact, through the output port “out”. The new phone

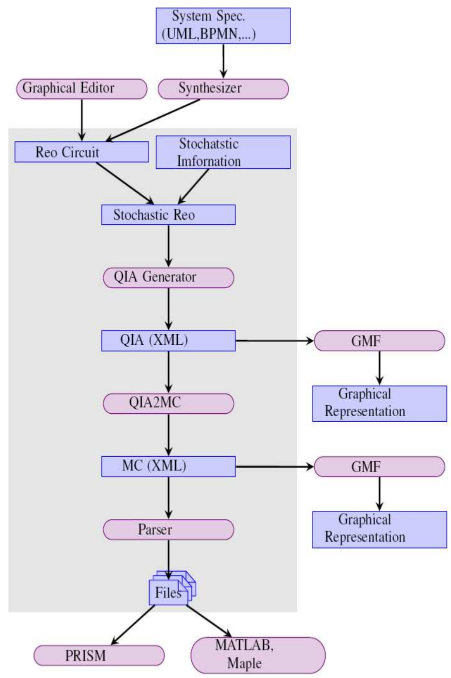

Fig. 2. Architecture of Reo2MC

"CM1" receives the requests from the input port "in" and sends through its "out" port the contacts, which are received by "CM1" in the "in" port. The Reo circuit in the system also guarantees the ordering among the data flow: each request is followed by a reply before next request is issued. Figure 4 shows the trend of the probability of completing the contact transmission within different intervals, which is specified as the following PRISM CSL formula

$$P = ?[true\ U[0, T]\ TransComplete].$$

The generated CTMC model has 2128 states and 7174 transitions.

PRISM is a popular probabilistic model checker, which can handle discrete and continuous time Markov chain models and is able to not only verify them against probabilistic temporal logic formulae, but also supports reward-based properties. Both states and transitions in a system can be associated with rewards, and both instantaneous and cumulative rewards properties can be checked. Our future work will aim to add rewards to QIA so that we can handle non-functional properties required by QoS analysis.

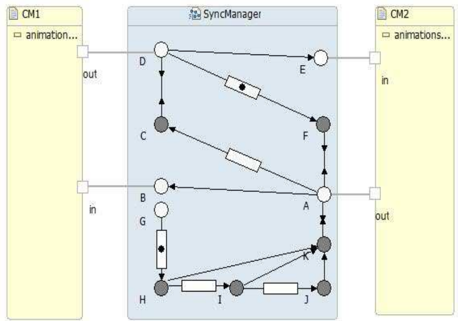

Fig. 3. The Mary Scenario.

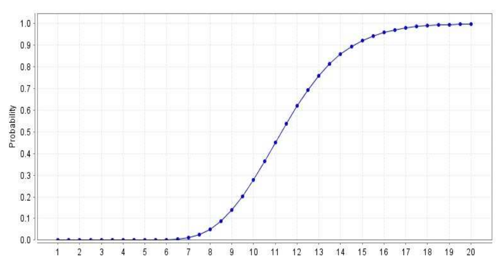

Fig. 4. The experimental results.

Acknowledgements

This work is jointly supported by a grant from the GLANCE funding program of the Dutch National Organisation for Scientific Research (NWO), through project CooPer (600.643.000.05N12), and by the European Commission Framework Program 7 project CONNECT (IST Project Number 231167).

References

1. Eclipse Coordination Tools. http://reo.project.cwi.nl/.
2. F. Arbab. Reo: a channel-based coordination model for component composition. *Mathematical Structures in Computer Science*, 14(3):329–366, 2004.
3. F. Arbab, T. Chothia, R. van der Mei, S. Meng, Y.-J. Moon, and C. Verhoef. From Cooridination to Stochastic Models of QoS. In *Proceedings of COORDINATION'09*, pages 268–287, LNCS 5521, Springer, 2009.
4. C. Baier, M. Sirjani, F. Arbab, and J. J. M. M. Rutten. Modeling component connectors in Reo by constraint automata. *Science of Computer Programming*, 61(2):75–113, 2006.
5. M. Kwiatkowska. Quantitative verification: Models, techniques and tools. In *Proceedings of ESEC/FSE 2007*, pages 449–458. ACM Press, September 2007.