

## Chapter 9

# A Symbolic Approach to the Analysis of Multi-Formalism Markov Reward Models

**Kai Lampka**

*Uppsala University, Sweden*

**Markus Siegle**

*Bundeswehr University Munich, Germany*

### ABSTRACT

*When modelling large systems, modularity is an important concept, as it aids modellers to master the complexity of their model. Moreover, employing different modelling formalisms within the same modelling project has the potential to ease the description of various parts or aspects of the overall system. In the area of performability modelling, formalisms such as stochastic reward nets, stochastic process algebras, stochastic automata, or stochastic UML state charts are often used, and several of these may be employed within one modelling project. This chapter presents an approach for efficiently constructing a symbolic representation in the form of a zero-suppressed Binary Decision Diagram (BDD), which represents the Markov Reward Model underlying a multi-formalism high-level model. In this approach, the interaction between the submodels may be established either by the sharing of state variables or by the synchronisation of common activities. It is shown that the Decision Diagram data structure and the associated algorithms enable highly efficient state space generation and different forms of analysis of the underlying Markov Reward Model (e.g. calculation of reward measures or asserting non-functional system properties by means of model checking techniques).*

DOI: 10.4018/978-1-4666-4659-9.ch009

## 1 INTRODUCTION

*Motivation:* Due to the complexity of today's systems, performance and dependability models should be built in a structured, i.e. Modular and hierarchical fashion. Employing different modelling formalisms within the same overall model can greatly assist the modeller in describing different aspects of the system in a clear and concise way. (generalised) stochastic petri nets, stochastic activity networks, stochastic automata, stochastic process algebras, stochastic extensions of uml state charts and other modelling formalisms may thus be employed in an overall multi-formalism model. There are two basic forms of interaction between the submodels of an overall model, both well understood in the theory of concurrent processes: one of them is the sharing of state variables, which is a very general concept supported by literally every modelling formalism mentioned above. In this approach, a subset of the state variables of a submodel is shared with one or more other submodels, so these state variables can be considered as global variables. The other form of interaction is synchronisation of common activities, which means that a designated subset of events may only take place jointly between two or more submodels. The consequence is that submodels have to wait for each other to perform these synchronised activities and are blocked as long as the partners are not ready to proceed. For the analysis, the high-level model description needs to be transformed into its low-level counterpart, of which this chapter assumes that it can be formalised as a Markov Reward Model (MRM), widespread in the performance and dependability literature. MRMs are continuous-time Markov chains augmented by reward (or cost) functions which enable the description and computation of a wide range of interesting performance and dependability measures. Examples for such measures are the expected accumulated reward gained during the mission of a spacecraft or the mean energy consumption per unit time of a production system.

A well-known drawback of state-based analysis is the problem of state space explosion, which means that the number of reachable states may grow exponentially in the number of concurrent activities of the high-level model. Among the techniques devised for coping with this problem, symbolic, i.e. decision diagram based approaches have shown to be particularly effective.

(Reduced Ordered Binary) Decision Diagrams (DD) are very useful for efficiently constructing and compactly representing the MRM underlying a high-level model description. Among the tools which successfully employ DD-based techniques are stochastic model checkers like PRISM (PRISM web page), CASPA (Kuntz et al. 2004) and SMART (SMART web page). These tools are able to analyse models with hundreds of millions of states. However, when it comes to tools which support multiple modelling languages, e.g. the Moebius performance analysis framework (Moebius web page), it is important that the DD-based analysis of MRMs is independent of the concrete model description method. This is not only because different entities of a system model might be described in a different method, but future extensions of the set of modelling formalisms should not require a re-implementation of the analysis engine. Independence of modelling formalism and DD-based analysis can be achieved by carrying out standard state space traversal and step-wise augmentation of the DD encoding the MRM. It is important, however, to note that these steps cannot practically be performed in a state-by-state manner, since this would lead to an unacceptable runtime overhead. Instead, operations which process sets of states and sets of transitions within one step are needed. Such operations can be provided by a DD environment, but they need to be used with great care and insight into the structure of the high-level model. Otherwise, negative effects<sup>1</sup> may compromise the efficiency of the approach.

*Contribution:* In order to address these problems, this chapter introduces a scheme for efficiently constructing a DD-based representation

24 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:  
[www.igi-global.com/chapter/a-symbolic-approach-to-the-analysis-of-multi-formalism-markov-reward-models/91947](http://www.igi-global.com/chapter/a-symbolic-approach-to-the-analysis-of-multi-formalism-markov-reward-models/91947)

## Related Content

---

### Countering Cross-Site Scripting in Web-Based Applications

Loye Lynn Ray (2018). *Application Development and Design: Concepts, Methodologies, Tools, and Applications* (pp. 370-383).

[www.irma-international.org/chapter/countering-cross-site-scripting-in-web-based-applications/188214](http://www.irma-international.org/chapter/countering-cross-site-scripting-in-web-based-applications/188214)

### Identifying Non-Performing Students in Higher Educational Institutions Using Data Mining Techniques

Deepti Aggarwal, Sonu Mittal and Vikram Bali (2021). *International Journal of Information System Modeling and Design* (pp. 94-110).

[www.irma-international.org/article/identifying-non-performing-students-in-higher-educational-institutions-using-data-mining-techniques/273228](http://www.irma-international.org/article/identifying-non-performing-students-in-higher-educational-institutions-using-data-mining-techniques/273228)

### Applying Software Engineering Design Principles to Agile Architecture

Chung-Yeung Pang (2022). *Research Anthology on Agile Software, Software Development, and Testing* (pp. 330-355).

[www.irma-international.org/chapter/applying-software-engineering-design-principles-to-agile-architecture/294471](http://www.irma-international.org/chapter/applying-software-engineering-design-principles-to-agile-architecture/294471)

### Evaluation and Ranking of E-Government Websites Using Weighted-Combinative Distance-Based Assessment Approach

Aakash Gupta and Mohit Bansal (2022). *International Journal of Software Innovation* (pp. 1-15).

[www.irma-international.org/article/evaluation-and-ranking-of-e-government-websites-using-weighted-combinative-distance-based-assessment-approach/309729](http://www.irma-international.org/article/evaluation-and-ranking-of-e-government-websites-using-weighted-combinative-distance-based-assessment-approach/309729)

### A Survey on Different Approaches to Automating the Design Phase in the Software Development Life Cycle

Sahana Prabhu Shankar, Harshit Agrawal and Naresh E. (2022). *Research Anthology on Agile Software, Software Development, and Testing* (pp. 542-564).

[www.irma-international.org/chapter/a-survey-on-different-approaches-to-automating-the-design-phase-in-the-software-development-life-cycle/294482](http://www.irma-international.org/chapter/a-survey-on-different-approaches-to-automating-the-design-phase-in-the-software-development-life-cycle/294482)