

# Chapter 5

## Multiformalism Modeling Compositionality in SIMTHESys

**Enrico Barbierato**  
*Università di Torino, Italy*

### ABSTRACT

*Multiformalism has emerged as a sound technique to define a complex system as the composition of a set of sub-components, each one modeled according to the best-suited formalism. Existing literature offers a wide choice of frameworks and tools that exploit model composition following different approaches. This chapter provides an insight into the composition approach used by SIMTHESys (a framework for the development of modeling languages and the solution of multiformalism models) in order to compose easily and consistently primitives belonging to different (custom) modeling languages. A case study is presented to illustrate the effectiveness of the proposed composition formalism.*

### INTRODUCTION

The modeling of real world systems is intrinsically complex.

The issue arises from the fact that several factors must be taken in account: for example, a system can be distributed over different components or can be characterized by an intricate dynamics. As some of the aspects of the system can be ignored, the researcher can invest more time and efforts to study the most relevant aspects. This classification suggests the possibility of building an abstraction of the system or a model that is defined according to a formalism. The advantages provided by these approach are several, being the most notable the capability of measuring the system dependability (the ability of a system to provide a specific ser-

vice) and its performability (how well a system performs, including the occurrence of faults).

One of the most powerful ways to understand how the parts of a system interact is modeling. Modeling power and Decision power of a formalism play an important role in this field. The former corresponds to the capability of describing efficiently a real system. The latter is inversely proportional to the complexity of computation and the set of computing resources requested to define properties and to execute the model (i.e. to implement a control function defined in terms of the formalism). Sub-formalisms originated by structural restrictions reduce modeling power and tend to increase the model size. On the other hand, they increase the decision power.

A further level of analysis considers qualitative and quantitative modeling. In the first case, the models are characterized by a variable with finite value domains and rely on an abstract logic description of the system behavior. In the second case, real valued variables are used to describe physical entities of a system through differential and algebraic equations.

The scale of the systems currently in charge to support human activities and the concurrency of different non-functional specifications require that modeling techniques would offer abstraction mechanisms that allow modelers to face such complexity while exploiting mathematically founded methods. Such abstraction mechanisms can be constituted of formal modeling languages and their solution algorithms, that can be based either on simulation or analytical methods. A broad and well spread example set is given by stochastic modeling techniques such as Petri nets, fault trees, stochastic automata, process algebras, queuing networks and many others that have been given a stochastic variant or interpretation.

During the last decades the intricacy of modern systems has grown significantly, including operating systems, networks, hardware and applications. Not only the choice of a specific formalism to build a model is difficult, but experience has shown that there is no a unique formalism able to model a system the best. Even in a specific domain the specifications used to build a system may vary. In a similar way, there is no a unique solution method.

The need to investigate disciplines such as engineering and biology characterized by complexity and heterogeneity has seen the rapid development of different formalisms and solution techniques supporting research in these fields. Informally, these approaches can be classified within three categories: i) sub-models composition, ii) multi-formalism integration and iii) formal models generation from high-level specifications. Multiple tools integration in a common environment plays an important role. A common approach in the

Software Performance Engineering area consists of implementing model-to-model transformations from UML description to formal models oriented to performance and dependability. However, integrating different tools raises different issues.

Firstly, it is necessary to understand how to report the results from a tool in a common way and secondly how to build the interoperability between the tools, considering the output values provided by one tool as the input for another tool.

Model composition has emerged as a sound technique showing how a complex model can be described as the composition of a set of sub-models. Initially, modelers consider several abstractions of the system under study, then they decompose it into two or more sub-components in order to analyze and understand more easily the single parts. Finally, the sub-components are put together following a bottom-up fashion in order to generate the initial system.

Literature offers different approaches to model composition describing different techniques and operators required. The most notable proposals include i) combinatorial methods, ii) qualitative model checking and iii) state-based stochastic methods.

The advantage of multiformalism results from the capability to assign the modeling of each sub-component to an expert in the specific subject, who is free to model that part using the most suitable formalism. Multiformalism has proven to be successful in different areas such as biology, fault-tolerant computing and disaster recovery being the most notable. As a result, this interdisciplinary aspect has created interesting links between different communities of modelers.

A variety of different software tools have been implemented to date. Early attempts included SHARPE (Trivedi 2002), a modeling framework capable of studying Markov models, queueing networks expressed in product form and generalized stochastic Petri Nets (GSPN). SMART (Ciardo G., Jones R. L. et al. 2006) is a software package used to design complex discrete-state systems; it

21 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/multiformalism-modeling-compositionality-in-simthesys/91942](http://www.igi-global.com/chapter/multiformalism-modeling-compositionality-in-simthesys/91942)

## Related Content

---

### Providing Automated Holistic Process and Knowledge Assistance during Software Modernization

Gregor Grambow, Roy Oberhauser and Manfred Reichert (2014). *Uncovering Essential Software Artifacts through Business Process Archeology* (pp. 20-63).

[www.irma-international.org/chapter/providing-automated-holistic-process-and-knowledge-assistance-during-software-modernization/96613](http://www.irma-international.org/chapter/providing-automated-holistic-process-and-knowledge-assistance-during-software-modernization/96613)

### The SOF Programming Paradigm: A Sequence of Pure Functions

Antoine Bossard (2022). *International Journal of Software Innovation* (pp. 1-14).

[www.irma-international.org/article/the-sof-programming-paradigm/309965](http://www.irma-international.org/article/the-sof-programming-paradigm/309965)

### TESTAR: Tool Support for Test Automation at the User Interface Level

Tanja E.J. Vos, Peter M. Kruse, Nelly Condori-Fernández, Sebastian Bauersfeld and Joachim Wegener (2015). *International Journal of Information System Modeling and Design* (pp. 46-83).

[www.irma-international.org/article/testar/126956](http://www.irma-international.org/article/testar/126956)

### Integrating Software Engineering and Costing Aspects within Project Management Tools

Roy Gelbard, Jeffrey Kantor and Liran Edelist (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 1358-1374).

[www.irma-international.org/chapter/integrating-software-engineering-costing-aspects/29450](http://www.irma-international.org/chapter/integrating-software-engineering-costing-aspects/29450)

### Using Description Logics for the Provision of Context-Driven Content Adaptation Services

Stephen J.H. Yang, Jia Zhang, Jeff J.S. Huang and Jeffrey J.P. Tsai (2012). *Theoretical and Analytical Service-Focused Systems Design and Development* (pp. 176-209).

[www.irma-international.org/chapter/using-description-logics-provision-context/66799](http://www.irma-international.org/chapter/using-description-logics-provision-context/66799)