Computers and Mathematics with Applications 64 (2012) 3828–3839

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

The SIMTHESys multiformalism modeling framework

M. Iacono [a,*], E. Barbierato [b], M. Gribaudo [c]

[a] Seconda Università di Napoli, Dip. di Studi Europei e Mediterranei, Caserta, Italy
[b] Università di Torino, Dip. di Informatica, Torino, Italy
[c] Politecnico di Milano, Dip. di Elettronica e Informazione, Milano, Italy

ARTICLE INFO

Keywords:
Modeling languages
Metamodeling
Performance evaluation

ABSTRACT

The usage of models is a fundamental activity in designing and verifying a system. Mastering different modeling techniques and scaling their application to complex systems is not an easy task and requires both advanced skills and proper tools. One of the means that allow modelers to leverage the power of proper modeling techniques (e.g. stochastic techniques) is the application of abstractions by using high level formal modeling languages. This paper presents SIMTHESys, a framework for the development of formal modeling languages and the solution of multiformalism models by automatically generated solvers based on different solving engines.

1. Introduction

Modeling is one of the most powerful tools developed to master the complexity of reality. 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. A broad and well spread example set is given by stochastic modeling techniques.

Literature offers many such modeling formalisms, more or less abstract and fit to model systems from a given point of view: examples are Petri nets, fault trees, stochastic automata, process algebras, queuing networks and many others that have been given a stochastic variant or interpretation. Nevertheless, all these useful and powerful abstractions are built over common low-level tools, such as Markov chains. Their existence enables modelers to cope with more difficult problems keeping the same well-known advantages.

SIMTHESys (Structured Infrastructure for Multiformalism modeling and Testing of Heterogeneous formalisms and Extensions for SYStems) is a framework for the definition of new formalisms and the generation of related solvers, that allow the combination of more formalisms in the same models. Formalisms and models definition is supported by a family of languages based on XML. SIMTHESys is mainly a conceptual framework that assists the process of defining new formalisms. As the definition of new formalisms would not be useful without the availability of related solvers, SIMTHESys is complemented with a solver generation tool (namely SIMTHESys*ER*), capable of automatically building simple multiformalism solvers based on generic elementary solvers (solving engines, in SIMTHESys terminology). SIMTHESys supports the design and the experimentation of modeling abstractions that can be used to ease the modeling process by encapsulating different solution methods and techniques into high level, customizable modeling languages.

* Corresponding author.
E-mail address: mauro.iacono@unina2.it (M. Iacono).

0898-1221/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2012.03.009

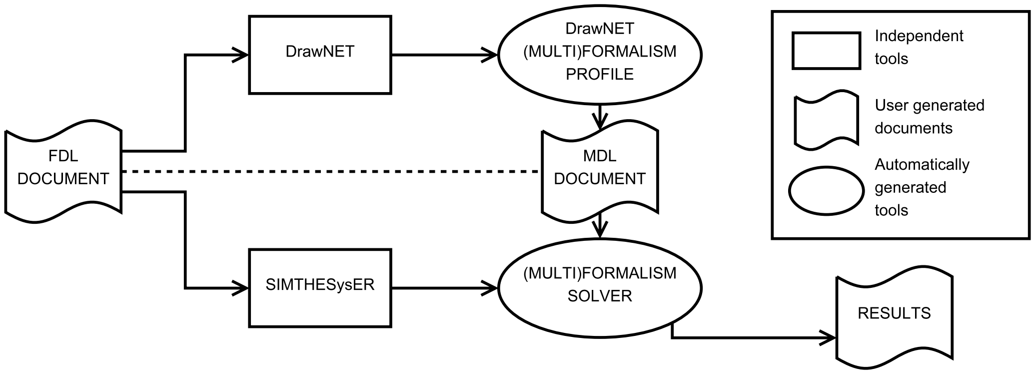

Fig. 1. The SIMTHESys workflow.

Stochastic formalisms are supported by the framework in the current phase of the project, though any kind of formalisms can be implemented in line of principle.

The goal of this paper is to introduce SIMTHESys, its main ideas and its contribution to the field of multiformalism modeling: a formal description of the framework is thus out of the scope of this paper. Currently, being the focus of our research the exploration of a new multiformalism modeling paradigm, optimization of solvers (indeed an important topic) is out of the scope of our research and will be faced in the future.

In this paper the SIMTHESys modeling framework is presented: Section 2 explains how formalisms and models can be designed, and with which limitations and possibilities; Section 3 examines the architecture that allows solvers generation and the model solution process; Section 4 shows some examples of use of SIMTHESys; finally Section 5 presents conclusions and future work. For this purpose, some general information about the framework and other similar approaches is needed.

1.1. Framework description

A sound description of the framework should deal with both model description and analysis aspects. The SIMTHESys framework consists of the modeling stack, the solving stack and SIMTHESys*ER*. The two goals of the framework (producing solvers for new formalisms and describing models to be solved) are accomplished by two complementary processes, that are both represented in the SIMTHESys workflow in Fig. 1. In the figure, tools and documents that are involved in the two processes are depicted with different symbols. The solver generation process, described in Section 3.3, is described by the lower part of the figure. The figure also includes DrawNET [1], an external general-purpose GUI generator that is adopted as graphical interface for the framework and is used to assist the user in writing MDL model documents by a graphical editor. DrawNET can generate a custom GUI by associating graphical primitives to the information obtained from the analysis of a FDL formalism document. Besides SIMTHESys*ER* and DrawNET, the ovals represent the automatically generated tools obtained by their execution, that form the elements of the modeling process. The dotted line stresses the fact that the MDL document must conform to the FDL document in the process.

1.2. Comparison with other frameworks

SIMTHESys is based on the application of metamodeling techniques to the description and the analysis of performance models (where performance should be interpreted in the broadest sense). Metamodeling is used to found formalisms extensibility and multiformalism capabilities on common features, supplied by the SIMTHESys metametamodel. The metametamodel is the foundation on which the metamodels are built. Metamodels form the layer of SIMTHESys on which the main focus is, because they are the formalisms with which models can be created. To better focus the perspective in which metamodeling is used in SIMTHESys, a light comparison with eCore [2] is useful. eCore is the metamodeling stack on which the Eclipse Modeling Framework is founded. eCore is used to allow the description of software entities independently from the platform on which they will be implemented: an eCore model is thus equivalent to an object-oriented application with its business logic, and it is used to generate plain source code in the desired language for the desired platform when needed. The focus in eCore is thus on models, and the eCore metamodel, designed to describe a generic object oriented language for software development, is generally used as it is. In the case of SIMTHESys, the purpose is to develop different metamodels, each of which is a formalism and shares the common metametamodel, so that models written according to different metamodels can interact though showing different characteristics, logic and internal organization.

To the best of our knowledge, SIMTHESys [3–6] is the only framework designed to support rapid formalism development (and automatic solver synthesis). Nevertheless, other extensible multiformalism modeling frameworks have provided the main contributions to the field. A different metamodeling approach has been adopted by AToM^3 [7], that exploits