Chapter 1
An Introduction to Multiformalism Modeling

Marco Gribaudo
Politecnico di Milano, Italy

Mauro Iacono
Seconda Università degli Studi di Napoli, Italy

ABSTRACT

The fundamental need for models in every field of design stems from the absolute and essential necessity for complexity domination. The prediction and verification competence is a must-have ability to allow an efficient and effective design cycle, as much as the complexity of the requirements, in specification, extension or criticalness of the design object, grows. Researchers and practitioners have developed different modeling languages to suit the needs of special classes of problems, such as general formalism like Fault Trees for dependability and availability models, several Petri Nets variants for performance and correctness evaluation, or dedicated modeling languages that map model elements onto domain entities. Richness and variety of formalisms, each of which tailored on a specific problem, empowers the modeling process, but also results in the explosion of many different submodels when coping with a complex and articulated system. These models are logically correlated and are used to take local decisions about single aspects of the overall problem. In practice they might not be aligned to each other and the mutual influences connecting them could be lost, compromising the general results. Multiformalism is an approach to modeling that aims to enable modelers to exploit together different formalisms for different aspects of the same system, while keeping the coherence and the mutual influences of the different parts of the overall model. Different concepts and interpretation of multiformalism modeling (and related problems, such multiparadigm and multisolution) have been experimented and are documented in literature. In this chapter, the main ideas and results in the field are introduced, together with an analysis and comparison.

DOI: 10.4018/978-1-4666-4659-9.ch001

INTRODUCTION

As rich hardware becomes more and more a commodity and the availability of reliable high speed network technologies increases, the expectations in terms of complexity of functions and integration of components about systems and applications grow, posing significant design challenges. Systems and applications consequently grow in scale, number of components and articulation of internal relationships. More constraints on performance, reliability, safety, security are expected to be satisfied during the design process. The result is that the process of specification of artifacts becomes a design into the design. Keeping the coherence between specifications expressed at different degrees of abstractions, related to different problems (but mutually influential on each other), requires a significant effort and consequent resources.

Moreover, the number of different logical and physical layers of artifacts possibly requires the collaboration of experts, with specialized backgrounds and skills, each used to a proper modeling logic, tools and methodology, and generally focused on different areas of the project.

Each of the problems faced during the design process led, with time, to the definition of a different approach to modeling, capturing and abstracting all the elements and the relations that influence or describe the system under that specific perspective. Such an abstraction process led to the birth and the definition (and generally the formalization) of proper modeling languages, which are a mean to produce abstraction artifacts (in graphical or textual form) by which problems can be described, communicated, documented, eventually analyzed and evaluated. The possibility of obtaining a manual or, even better, an automatic evaluation process producing qualitative or quantitative results in the field of interest is a useful feature. In general it is considered as the main goal in the definition of a modeling language supporting and speeding up the design cycle, in the early decision phases or as a verification tool to be applied to design choices, as an alternative or an integration to more classical test- or prototype-based approaches. In the rest of this chapter, the focus will be on the subset of all possible modeling languages aiming to offer a tool producing and/or transforming automatically evaluable models (let this informally defined subset be dubbed for convenience in the rest of this chapter *solvable modeling languages* or, alternatively, in coherence with the terminology that will be introduced in the following, *solvable formalisms*).

Solvable modeling languages imply the existence of proper *solving processes*, i.e. sequences of operations or transformations or other manipulations producing the final goal of the evaluation. They are implemented by *solvers*, software tools that perform such processes by steps or integrally. Solving processes for different solvable modeling languages (e. g. performance and availability oriented languages) are generally different, and based on different principles; as a consequence, it is unlikely that a single approach to solution could encompass all possible needs. One of the main challenge with multiformalism modeling is the invention of a possible solving approach to handle this problem. For example, when the dynamic aspects of a system are involved in the evaluation and probabilistic description of its elementary behaviors are available, a possibility is given by general-purpose event-based simulators. However, this approach presents some drawbacks: i) it does not preserve the richness of description proper of specific modeling languages, and could produce approximate results, ii) it requires long setups of the simulation scenario due to the complexity of the modeling power that multiformalism modeling offers, iii) it requires long simulations for systems with many interactions and many different aspects of it modeled together. iv) it requires the modelers to focus on the design and implementation of the simulation process rather than on the modeling process itself, and v) it can force modelers to acquire a different background with respect to their own specialization. Resorting