Chapter 2
Combining Heterogeneity, Compositionality, and Automatic Generation in Formal Modelling

Stefano Marrone
Seconda Università di Napoli, Italy

Nicola Mazzocca
Università di Napoli "Federico II", Italy

Roberto Nardone
Università di Napoli "Federico II", Italy

Valeria Vittorini
Università di Napoli "Federico II", Italy

ABSTRACT

Critical computer-based systems have an increasing complexity due to the number of components, to their heterogeneity, and to the relationships among them. Such systems must meet strict non-functional requirements and should be able to cope with competitive market needs. The adoption of formal methods is often advocated in order to provide formal proof, but their application does not scale with the growing size of systems. The aim of this chapter is to introduce a modelling and analysis methodology that allows the combination of three proven research trends in formal modelling of large systems: formal model generation (by means of model-driven techniques), multiformalism, and compositional approaches. In this chapter there is also a discussion about enabling techniques. The proposed approach has been applied to the performability modelling and evaluation of flexible manufacturing systems.

DOI: 10.4018/978-1-4666-4659-9.ch002

FORMAL MODELLING OF CRITICAL SYSTEMS

Computer-based systems are now present in our daily life and affect many aspects providing essential support to several human activities, from transportation to industrial automation systems. They generally have large distributed architectures where the complexity is due to heterogeneity of system elements which are not only hardware and software components, but also procedures and people. Their criticality implies the necessity to meet several requirements, often dictated by international standards, whose fulfillment must be demonstrated in order to achieve necessary certifications. The application of formal methods in industry (highly recommended if not mandatory) is slowed down by the complexity of the model and heterogeneity (which reflects the system complexity), and by the need to have highly skilled personnel involved in model development. In other words, there is a request for modelling methodologies and supporting tools which can hide the complexity of the modelling process, without losing expressive power and solving efficiency.

The scientific literature addresses three main research directions in order to overcome the described problems:

- **Divide-and-Conquer:** Focusing on methods and techniques for developing models through submodel composition.
- **Multi-Formalism and Multi-Paradigm:** Exploring the possibility of combining different formalisms and modelling paradigms in defining the overall model.
- **Automatic Generation:** Generating formal models suitable for the analysis from high-level specifications.

In the first approach a model consists of several submodels tied together by appropriate rules and composition operators: in order to provide appropriate methods for solving submodels and aggregating results, composition techniques and operators definition are necessary. In (Nicol, 2004) a comprehensive review of the state of the art in such techniques is presented. These methods include techniques to limit the size of the state space (*largeness avoidance*) and techniques to manage the size of the models (*largeness tolerance*).

The second approach deals with the integration between submodels expressed by different formalisms and also based on different modelling paradigms. The first steps in this direction have been made by SMART (Ciardo, 2001) and SHARPE (Trivedi, 2002): the first integrates Stochastic Petri Nets, Markov chains in continuous and discrete time, while the latter integrates Fault Trees (FT), Generalized Stochastic Petri Nets (GSPN), some types of queuing networks and Markov Processes. Then, some approaches and tools based on explicit methodologies for the development of multiformal models have been proposed: two different frameworks described in literature are Möbius (Deavours, 2002) and OsMoSys (Vittorini, 2004). An interesting example of multi-paradigm modelling is realized by AToM^3 (de Lara, 2002), which implements an approach based on meta-modelling and graph transformation techniques. Other recent frameworks focuses the attention on the practical use of these techniques (Iacono, 2012).

The third approach deals with the generation of formal models from high-level specifications. As surveyed in (Bernardi, 2011b), the scientific community has also explored such way and several approaches presented in the literature are: in (Pai, 2002) and in (D'Ambrogio, 2002) the generation of Fault Tree and Dynamic Fault Tree (DFT) models from a set of UML diagrams is described; in (Bondavalli, 2011), UML diagrams are used to generate Timed Petri Nets-based models; generation of Stochastic Reward Nets from Statechart and Activity Diagrams are presented respectively in (Huszerl, 2002) and (Tadano, 2011). These approaches often rely on high-level languages oriented to Non-Functional Properties (NFPs)