Alberto Falcone

Distribution, Reuse and Interoperability of simulation models in heterogeneous distributed computing environments

Ph.D. Thesis

September 30, 2017

Abstract

Systems Engineering is an interdisciplinary field of engineering and engineering management that is gaining a central role in a variety of industrial and scientific domains ranging from e-science to space exploration, due to the increasing complexity of system requirements and thus of the related engineering problems. Systems Engineering represents one of the most important and effective methods for designing and studying both *Large-Scale System* and *System of Systems (SoS)* over their life cycles.

A *Large-Scale System* is defined as:

"A group of subsystems that are interconnected and organized so to form a whole system with clearly defined boundaries. Each subsystem is self-contained and independent from the other ones but it cannot work individually."

Whereas a *SoS* is defined as:

"A complex purposeful whole that is composed of complex, independent, self-organizing, component parts whose high levels of interoperability enable them to be recomposed into different configurations and even different systems of systems; is characterized by poorly-defined issues that significantly affect its behavior and make it difficult to understand; has ambiguous boundaries with critical contextual influences involving a mix of technical/non-technical factors; and exhibits emergent nonlinear properties. The complexity of a system of systems is a function of the number and diversity of its components and their linkages. System of systems linkages range from loosely to closely connected, but all systems of systems exhibit non-deterministic evolution and behavior and are cybernetically self-organizing."

In both the cases (*Large-Scale System* and *SoSs*), each component contributes to the functioning of the entire system but, in general, the behavior of the whole system cannot be straightforwardly derived from the behavior of its components.

Systems engineering utilizes *systems thinking* principles to organize this body of knowledge and uses tools that include modeling and simulation (M&S), requirements analysis and scheduling to manage the complexity of such systems. By using M&S methods, tools and techniques, it is possible to reproduce the structure and behavior of systems over the time so as to observe and analyze them. The use of M&S techniques offer many advantages, such as the possibility to study the behavior of a system without physically building it, and the evaluation and comparison of different design choices, policies, and operating procedures through experiments in a controlled (virtual) environment. Despite the above sketched advantages, M&S has important challenges many of those related to the significant efforts required for producing a full-fledged simulation model and analyzing simulation results. Moreover, it is often hard to *reuse* already available simulation models; indeed, there is a lack of mechanisms to make *interoperable* simulation models built on different simulation platforms and a scarce support to enable their *execution on distributed infrastructures*. To overcome these challenges, many research efforts are focusing on the definition of methods, models and techniques to support the reuse and interoperability of simulation models and their execution on distributed computing environment. Two of the most popular efforts going in these directions are the Functional Mock-up Interface (FMI) and IEEE 1516-2010 - High Level Architecture (HLA) standards. However, each of the two mentioned proposals addresses part of the above issues and great benefits could derive from their combined exploitation.

In this context, the research presented in this Ph.D. thesis has been focused on the definition of models, methods and techniques to address, in an integrated way, the issues of *reuse*, *distribution*, and *interoperability* among heterogeneous simulation models. The research activity has been conducted in cooperation with the Software, Robotics, and Simulation Division (ER) - Simulation and Graphics Branch (ER7) of the NASA's Lyndon B. Johnson Space Center (JSC) in Houston (Texas, USA) where I spent nine months of my Ph.D. program.

Starting from the research objectives above described, the main contributions resulting from the research activity presented in this Ph.D. thesis concern the definition of:

1. A software framework, which is called *HLA Development Kit*, that aims at facilitating the design and develop of distributed simulators compliant with the IEEE 1516.2010 - High Level Architecture (HLA) standard.
2. A Model-Driven method, which is called *MONADS*, that makes easier for Systems Engineers to design a Complex System and simulate it on a distributed simulation environment, without asking them to explicitly deal with the intricacies and difficulties of currently available standards and technologies.
3. Two methods, *HLA for FMI* and *FMI for HLA*, to address in an integrated way the issues of *reuse*, *distribution* and *interoperability* among

heterogeneous simulation components through the integration of the functionalities offered by the HLA and FMI standards.

Concerning the first contribution, the *HLA Development Kit software Framework (DKF)* is a general-purpose, domain-independent framework, fully implemented in the Java language and released under the open source policy Lesser GNU Public License (LGPL), which facilitates the development of HLA Federates. The DKF allows developers to focus on the specific aspects of their own Federates rather than dealing with the common HLA aspects such as the management of the simulation time; the connection/disconnection to/from the HLA RTI; the publish, subscribe and updating of *HLA ObjectClass* and *HLA InteractionClass* elements. The DKF has been designed and developed in the context of the research activities carried out within the SMASH-Lab (System Modeling And Simulation Hub - Laboratory) of the University of Calabria (Italy) working in cooperation with the Software, Robotics, and Simulation Division (ER) of the NASA's Lyndon B. Johnson Space Center (JSC) in Houston (Texas, USA). It has been successfully experimented in the SEE project since the 2015 edition. In the last edition, the Universities of Calabria (Italy), Bordeaux (France), Brunel (London, UK) and the Faculdade de Engenharia de Sorocaba, FACENS (Brazil) developed their SEE-Federates by using the Kit.

Concerning the second contribution, the result is a Model-Driven method called MONADS (MOdel-driveN Architecture for Distributed Simulation). The MONADS method aims at facilitating the distributed simulation of complex systems, specified by using UML/SysML, according to the Model-Driven Systems Engineering (MDSE) paradigm. Moreover, the HLA simulation code, generated starting from SysML models by a chain of *model-to-model* and *model-to-text* transformations, is based on the *HLA Development Kit software Framework (DKF)*. This research activity is carried out working in cooperation with the Laboratory of Software Engineering, Department of Enterprise Engineering of the University of Tor Vergata (Rome).

In the end, regarding the third contribution. Although the HLA and FMI standards start from different objectives and are based on different techniques, they have several common features that can be jointly exploited so as to create a full-fledged solution to enable reuse, interoperability and distributed execution of simulation models. In this context, two solutions on how to fruitfully combine the HLA and FMI standards have been defined: (i) *HLA for FMI*, in which a FMU is enriched with HLA features and services; and, (ii) *FMI for HLA*, in which simulation modules that are available as FMUs are reused in a HLA simulation environment without modifying them.

The results achieved lead to an improvement in the Systems Engineering research domain. Specifically, the presented solutions provide a new and flexible way on how to design and manage complex systems in distributed computing environments.