THIS IS A MINI-FOREWORD INTRODUCING THE NEXT 3 CHAPTERS.

CHAPTER 15

DEVS Standardization: foundations and trends

Gabriel A. Wainer, Khaldoon Al-Zoubi, Olivier Dalle, David R.C. Hill, Saurabh Mittal, José L. Risco Martín, Hessam Sarjoughian, Luc Touraille, Mamadou K. Traoré, Bernard P. Zeigler.

Since the early 70s, the Modeling and Simulation (M&S) community has been attempting to formulate approaches to modeling as system specification formalisms. In many cases, different models (i.e., mathematical representations) of the source systems existed before their computerized incarnations. Examples of this include, for instance, differential equations (which have continuous states and continuous time), models of systems operating on a discrete time base, etc. As discussed in other chapters of this book, the DEVS formalism was defined to bring coherence to the field of discrete-event M&S by using an organized system-theoretical framework. In spite of this coherence and the unity that this theory introduces into the field of M&S, today, almost forty years later, the Discrete Event System Specifications are still held hostage by their simulation language implementations and algorithmic code expressions.

We believe that the need for a widely accepted framework is now more necessary than ever because of the fragmentation in the field that has resulted from the growing specialization of knowledge. The need for "knowledge workers" who can synthesize disciplinary fragments into cohesive wholes is increasingly recognized. M&S—as a generic, non-discipline specific, set of activities—can provide a framework of concepts and tools for such knowledge work.

The following chapters present different approaches to solve these problems. The presented approaches each provide a solid foundation for standardization and by constrasting their specific strengths and less developed qualities, the significant progress in the area is shown. These approaches build on new methods and the underlying technologies have made model building and simulation execution easier and faster by riding piggyback on the technology advances in software and hardware. More recently, fundamental issues such as model credibility (e.g., validation, verification, and model family consistency) and interoperation (e.g., repositories, reuse of components, and resolution matching) have moved to the front and center under the impetus of the High Level Architecture (HLA) standard. The HLA focused on interoperability of existing geographically dispersed M&S assets, and it was initially built focusing on defense applications (which still are the main application focus of the standard). These days, however, other users are interested in distributed simulation applications as well, including those in business enterprises and e-commerce (which are becoming important as complexity increases and lead-times diminish). The HLA does not address this kind of applications easily. In addition, it does not focus on how to solve the problem of creating models to be executed in the simulation environment.

Instead, the DEVS theoretical foundation makes it (in principle) independent of the various programming languages and hardware platforms. Clear examples of this have been seen in previous chapters, where we could see the wide variety of groups working on extensions to the DEVS formalism (each of them focusing on real-time modeling, cellular model definition, dynamic structure applications, etc.) and several modeling tools based on these extensions. Nevertheless, there still is a need to standardize these notations and simulation tools. Within a

Study Group of the Simulation Interoperability Standards Organization (SISO), a standard has been under development to support interoperability of DEVS models implemented in different platforms as well as with legacy simulations. The goal of the study group is to find a core of the DEVS formalism that is suitable for standardization of activities at the level of modeling, thereby bridging the gap between existing simulation frameworks and modeling activities using a standard notation. In this part, we will discuss fundamentals of various designs on how to achieve these goals through the standardization of DEVS models and their simulators.

One of the main issues to be addressed is a direct consequence of separating the model from the simulator: this results in multiple ways in which the same model can be simulated—all adhering to the abstract simulator specification. For instance, there are *virtual-time* simulators (where the simulator can skip from one event time to the next without traversing the intervening time interval) and *real-time* simulators (where time is interpreted as wall clock readings, so the simulator must wait for the interval to its next scheduled event to expire before handling the event). In addition to the different combinations of model type/simulation software, a standard allows for the use of different forms of distribution of model components (e.g., single processor vs. multi-processor, and within the latter, conservative vs. optimistic time advance for virtual-time as well as centralized vs. non-centralized time control in real-time execution). The standard can also be independent of different implementation platforms, such as Windows vs. UNIX; different programming languages, such as Java vs. C++; and different networking and middleware frameworks such as .Net vs. Apache. As we can see, such a standard can have multiple simulation scenarios. For example, a model may be simulated in virtual-time and in real-time both in distributed and non-distributed fashion.

The remaining three chapters of this book discuss in detail the various recent standardization approaches in DEVS simulation tools. In most cases, Web Services technology is highly leveraged, which has proven useful in achieving model and simulation interoperability. Developments in the World Wide Web have prompted many efforts in the distributed simulation field for modeling, executing simulation, and creating model libraries that can be assembled and executed over HTTP. By means of XML and web services technology these efforts have entered into a new phase with exponentially increasing utility as new standards are becoming available.

ACKNOWLEDGEMENTS

This section of the book has been built based on the effort of numerous other researchers (a non-comprehensive list can be found in the DEVS Standardization Study Group webpage, http://cell-devs.sce.carleton.ca/devsgroup/). In particular, Prof. Tag-Gon Kim (KAIST, Korea) has sponsored this group, with the goal of demonstrating the feasibility of a DEVS Simulator standard. Numerous other members of the Study Group have participated on the standardization efforts, including Fernando Barros, Claudia Frydman, Norbert Giambiasi, Jan Himmelspach, Xiaolin Hu, Moon H. Hwang, Jim Nutaro, Lin Uhrmacher, and Hans Vangheluwe. Pieter J. Mosterman contributed with the numerous editions and organization of these chapters.