Discrete-Event Systems Specification (DEVS) Study Group

Background

Since the early '70s, the Modeling and Simulation community tried to formulate approaches to modeling as system specification formalisms. In many cases, model mathematical representation of systems existed before their computerized incarnations. For instance, differential equation systems, having continuous states and continuous time, were formulated as the class of DESS (Differential Equation System Specifications). Also, systems that operated on a discrete time base such as automata were formulated as the class of DTSS (Discrete Time System Specifications). However, the Discrete Event System Specifications (DEVS) were largely prisoners of their simulation language implementations or algorithmic code expressions.

Indeed, there was a prevalent belief that discrete event “world views” constituted new mutant forms of simulation, unrelated to the traditional mainstream paradigms. Fortunately, that situation changed as the benefits of abstractions in control and design became clear. Witness the variety of discrete event dynamic system formalisms that have emerged . While each one – examples are Petri Nets, Min-Max algebra, and GSMP (generalized semi-markov processes) – has its application area, none were developed deliberately as subclasses of the systems theory formalism.

The first efforts to include DEVS formalism into an organized system-theory based framework were devoted to build a theory to help bring some coherence and unity to the field of modeling and simulation. Although nearly a quarter of a century later has seen many advances in the field, we believe that the need for a widely accepted framework and theoretical foundation is even more necessary today. Modeling and simulation lore is still fragmented across the disciplines making it difficult to share in the advances, reuse other discipline’s ideas, and work collaboratively in multidisciplinary teams. As a consequence of the growing specialization of knowledge there is even more fragmentation in the field now then ever. The need for “knowledge workers” who can synthesize disciplinary fragments into cohesive wholes is increasingly recognized. Modeling and simulation – as a generic, non-discipline specific, set of activities – can provide a framework of concepts and tools for such knowledge work.

There has been much significant progress in the area. Model building and simulation execution have been made easier and faster by riding piggyback on the technology advances in software and hardware. However, 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 received a lot less attention. These issues moved to the front and center under the impetus of the High Level Architecture (HLA) standard mandated by the United States Department of Defense for all its contractors and agencies.

The High Level Architecture (HLA) standard focused on interoperabillity of existing geographically dispersed simulation assets. However, non-DoD applications of distributed modeling and simulation, such as in distributed business enterprises and e-commerce, are becoming increasingly important as complexity increases and lead-times diminish. Developers of such applications may find the constraints placed by HLA, however critical to DoD simulations, overly burdensome for their situations. Moreover, some of the difficulties faced in creating distributed modeling and simulation environments are not addressed by the HLA prescriptions and run-time infrastructure. CORBA is a widely accepted middleware to support distributed software systems and offers complementary services to those of HLA. Other support middleware with application in different areas include MPI or PVM, as other standard syncrhonization mechanisms provided by Operating Systems.

These issues can be avoided when the DEVS (Discrete Event System Specification) formalism is used as the specification paradigm. DEVS is an increasingly accepted framework for understanding and supporting the activities of modeling and simulation. DEVS is a sound formal modeling and simulation (M&S) framework based on generic dynamic systems concepts. It is a mathematical formalism with well defined concepts of coupling of components, hierarchical, modular construction, support for discrete event approximation of continuous systems and an object-oriented substrate supporting repository reuse.

Perhaps the most basic concept is that of mathematical systems theory. DEVS theory provides a fundamental, rigorous mathematical formalism for representing dynamical systems. There are two main, and orthogonal, aspects to the theory:

levels of system specification – these are the levels at which we can describe how systems behave and the mechanisms that make them work the way they do.

systems specification formalisms – these are the types of modeling styles, such continuous or discrete, that modelers can use to build system models.

Although the theory is quite intuitive, it does present an abstract way of thinking about the world with independence of the simulation mechanisms, underlying hardware and middleware.

In May 1997, a research project entitled "DEVS Formalism as a Framework for HLA Predictive Contract Methodology" awarded by DARPA demonstrated how an HLA-compliant DEVS environment can significantly improve the performance of large-scale distributed modeling and simulation exercises. HLA effort to establish a common technical framework facilitating the interoperability of all types of models and simulations among themselves and with C4I systems. DEVS has a theoretical foundation which makes it in principle independent of various programming languages and hardware platforms.

In addition to HLA implementations, others are running under CORBA, MPI, standard Unix sockets, and others. There is a wide variety of groups working on extensions to the DEVS formalism, with several groups developing different modeling tools based on these extensions. Research and development groups are found throughout the world, including the US, Canada, Korea, Japan, UK, Portugal, France, Austria, Germany, Argentina and Mexico. Although, each development group adheres to the basic DEVS definitions, implementation issues arise due to differences in underlying platforms and these differences offer barriers to the reuse and sharing of DEVS models.

During SCSC '2000 in Vancouver, a group of DEVS researchers decided to form a study group to pursue the development of a standard for DEVS-based tools and extensions, to be presented to SISO for future recommendation as a IEEE standard.

Issues to Investigate

The DEVS formalism continues to expand in scope, depth and wide spread use. For example, DEVS is a prime candidate to provide the simulation model representation standard for Simulation Based Acquisition. As indicated, implementations of DEVS agree on the underlying mathematical formalism but are not necessarily interoperable due to different computer representations

The objective is to develop standards for a computer processable representation of DEVS that supports common understanding, sharing and interoperability of DEVS implementations. Computer processable forms include all forms of simulation and real-time execution as well as various forms of syntactic and semantic analysis.

Three main tasks will be undertaken:

- Different approaches will be examined including compiling, translation, object orientation (standardization of the supporting classes) and combinations of these methods.

- The relation to other applicable standards such as HLA (http://hla.dmso.mil), CORBA (http://www.omg.org/corba) and XML (http://www.w3.org/XML/) will be considered.

- Simulation interoperability of DEVS with non-DEVS simulation models will be examined.

Standardization of basic primitive and compound DEVS modeling constructs (syntactic and semantic) in support of higher-level extensions such as agent, cellular, and dynamic simulation models.

Among the issues that may enter into consideration of the three main tasks are:

- Standardization of DEVS modeling constructs and semantics w.r.t. those offered by Modelica (www.modelica.org).

- Standardization of DEVS model types in relation to AI-based models (performing and simulated agents).

- Standardization of DEVS models applied to Real-Time systems (RT-DEVS).

- Standardization of model persistence and databases.

- Semantic-based representation of family of models in support of Simulation Based Acquisition.

- Standardization of DEVS models libraries

- Support for web-based model definition and execution.

- Identifying, developing, and cataloging application model frameworks (similar to the concept of design patterns in SE) based on major domains of interest.

Initial schedule of Activities

The following are the preliminary key dates and activities to be followed by the SG. New participants will be invited, leading to possible meetings not included here:

Mar 2001- SAC Approves Study/Drafting Group
Mar 2001- SG to meet (via teleconference) to set-up first meeting during ASTC'01.
Apr 2001- SG pre-meeting in ASTC'01, Seattle, Washington
Jul 2001 - Initial SG Kick Off Meeting at SCSC
Oct 2001 – Meeting at the DEVS Workshop, ESS, Marseilles, France.
Dec 2001 – Meeting at WinterSim
Mar 2001 - Mar 2002 - SG to meet (via teleconference) on re-occurring basis.

RESOURCES

Coordination: Gabriel A. Wainer. gwainer@sce.carleton.ca

Mailing list:
Post message: DEVSTD@yahoogroups.com
Subscribe: DEVSTD-subscribe@yahoogroups.com
Unsubscribe: DEVSTD-unsubscribe@yahoogroups.com
List owner: DEVSTD-owner@yahoogroups.com

Old posted messages: http://groups.yahoo.com/group/DEVSTD

Group URL: http://www.sce.carleton.ca/faculty/wainer/devsgroup.htm

Other resources: http://www.acims.arizona.edu/
http://www.sce.carleton.ca/faculty/wainer/celldevs/