
# SG

# "DEVS STANDARDIZATION STUDY GROUP"

# INTERIM FINAL REPORT

Submitted to:

The SISO Standards Activities Committee (SAC)

Date: _____

SISO-REF-nnn-2005

Submitted By:
Gabriel A. Wainer

# Table of Contents

## 1.0 INTRODUCTION / OVERVIEW:

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, systems that operated on a discrete time base, etc. However, the Discrete Event System Specifications (DEVS) were largely prisoners of their simulation language implementations or algorithmic code expressions.

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 is even more necessary today. 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 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. The 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. The HLA is not addressing how to solve the problem on creating models to be executed in the simulation environment.

Instead, DEVS has a theoretical foundation which makes it in principle independent of various programming languages and hardware platforms. There is a wide variety of groups working on extensions to the DEVS formalism, with several modeling tools based on these extensions. The goal of the study group will be to find a core of the DEVS formalism that is suitable for standardization of activities at the level of modeling. It will bridge the gap between existing simulation frameworks and modeling activities using a standard notation.

## 2.0 PDG/SG OFFICERS:

| | |
|---|---|
| **Chair** | ***Gabriel Wainer***<br>**Carleton University. Ottawa, ON. Canada.** |
| **Vice Chair** | ***Bernard Zeigler***<br>**ACIMS. University of Arizona. Tucson, AZ.** |
| **Secretaries** | ***Hessam Sarjoughian***<br>**Arizona State University. Tempe, AZ.**<br>***James Nutaro***<br>**ACIMS. University of Arizona. Tucson, AZ.**<br>***Trevor Pearce***<br>**Carleton University. Ottawa, ON. Canada**<br>***Fabrice Bernardi***<br>**University of Corsica**<br>***Hwang Moon***<br>**Wayne State University** |

## 3.0 PDG/SG MEMBER LIST:

<deleted; login into the SG to access to this information >

## 4.0 TASK DESCRIPTIONS FROM THE PN / TOR:

The objective is to study the possibility of developing 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.

The following three main tasks will be undertaken:

1. Perform an analysis study for the potential establishment of a core for a DEVS standard. The actual standards development work for the 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 would then be recommended by the DEVS study group for a potential development task for a DEVS SISO Product Development Group (PDG).

2. Analyze the relationship of DEVS to standards such as HLA, CORBA, XML Modelica, etc..

3. Analyze and describe the potential creation of a proposed standardized language for defining DEVS models, and standardization of DEVS models libraries. The proposed language could then be refined and standardized by a SISO Product Development Group (PDG).

## 5.0 PRODUCT DESCRIPTIONS FROM THE PN / TOR:

- Interim Progress Reports during SIW conferences.
- Analysis report summarizing the results from the three tasks described above.
- Final reports from findings of any study group sub-groups or committees.
- Recommendation to the SISO on any potential DEVS standards development followed by a completed Product Nomination Form describing the potential DEVS SISO standards products.

## 6.0 SIGNIFICANT RESULTS AND/OR ACHIEVEMENTS

During the period of the Study Group, we identified three main areas of interest by the team members:

- Team 1: Building DEVS language+libraries: it was decided that it is needed to provide a new set of constructions that could help in spreading the basics of DEVS (mostly for newcomers to the field, students, and industrial/government participants new to this field).

- Team 2: implementation of services to interoperate at least two existing DEVS tools: a team was interested in carrying out experiments on interoperating two DEVS tools at the level of the model, following the ideas presented in the TOR.

- Team 3: discussion of the contents of a DEVS kernel. This is the main core task of the SG, which includes discussions on what should be included on a kernel for the DEVS standard notation.

### 6.1. Discussion on advances of Team 1:

- 3 different members in the SG created versions of graphical-based tools with educational purposes
- Some of these tools have started to being used for teaching DEVS concepts (Carleton University, McGill, Univ. Marseille)
- At present, the experience is being extended to other members

- 4 members in this team are preparing a joint paper, to be used as a case study/set of requirements. The goal is to compare different DEVS tools features. Each team will define the model using different DEVS tools (PyDEVS, DEVSJava, CD++).

6.2. Discussion on Advances of Team 2

Initially there were some discussions on how to make ADEVS and CD++ toolkits interoperate. Starting in May 2005, a new student is working on interoperation studies of the CD++ and DEVSJava toolkits
Work period: May 2006-April 2006

6.3. Discussion on Advances of Team 3

Generic discussion started during SIW meetings. These results would be the basis for a Product Nomination. This is the Main core of a Product Development Group. The SG is not yet ready to face this task in full detail
The results of Team 1 would provide a basis for future discussion
The results of Team 2 would identify a common interface to become a source for the kernel
Dr. Nutaro proposed some basic ideas, presented in Section 7. The relationship with the DEVS-Bus proposal should be discussed.

## 7.0 SUMMARY OF TECHNICAL FINDINGS

- Standardized APIs have helped to create widely reused software "components" (Standardized thread APIs; OpenGL). This promotes use of advanced technology (e.g., sophisticated computer graphics software) as components in large software systems

- Simulation engines are represented by a single object, and the simulation engine object interface usually has methods for
  - Obtaining outputs
  - Injecting inputs
  - Obtaining the time of next event
  - Computing the next state

In general, these methods are applied to the 'top' of a hierarchical model.

Thus: what to standardize?

The proposal is to standardize the simulation engine API, using an approach similar to other API standards (pthreads, OpenGL, MPI, OpenMP, etc.)

What are the benefits of this approach?
- To permit software system designers to include simulation components in their high level designs

- To select a simulator implementation can be done later, in the implementation phase
- This selection can match the modeling problem, and not be driven by integration requirements

In this way, external interfaces (e.g., for DIS or HLA) can be readily adapted to an existing simulation. Consequently, new capabilities for legacy simulation systems can be constructed using formal techniques and supporting tools. Incremental adoption of modern M&S technology into pre-existing simulation systems….especially important in military applications where 10+ yr. old simulation software is not uncommon

In terms of DEVS modeling languages, different proposals are based on state-based notation, states are represented by bubbles including an identifier and a state lifetime. When the lifetime is consumed, an internal transition function is executed, as in the following figure:

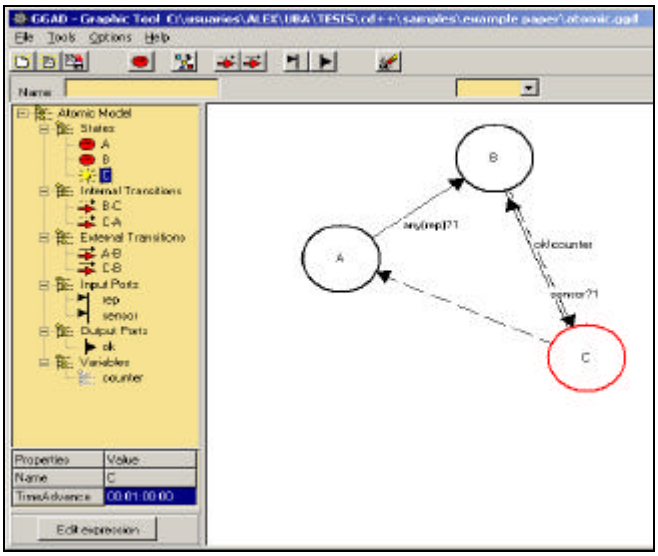

Figure 1: An atomic model defined as a DEVS graph.

A graphical notation like this one can be used as a basis to generate a standard modeling notation. We need to define how to translate such notation into an interoperable definition. A proposal considers:

[modelname] defines the atomic or coupled model name, which will be used subsequently. Model states are declared as: state: state1 state2 ...
States are associated to a time advance value. This attribute are initialized with the name of the object and the list of valid attributes for that object, as follows:
state1 : time-expression
One of the states must be declared as the initial state of the model: initial: statename
Then, I/O ports are declared either as follows:
in : inport1 inport2 ...
out : outport1 outport2 ...

The internal transitions use the following syntax:

int:source destination [outport!value]* ( { (action;)* } )?
Here, the source and destination represent the initial and final states associated with the execution of the transition function. As the output function should also execute before the internal transition, an output value can be associated with the internal transition. Finally, if the user wants to execute a complex function to generate an output, one or more actions can be defined. External transitions are defined using the following expression:
ext : source destination EXPRESSION ( { (action;)* } )?
In this case, the expression should hold, and then the model will change from state source to state destination, while also executing one or more actions.

Coupled models are also better defined using a graphical notation as the following:

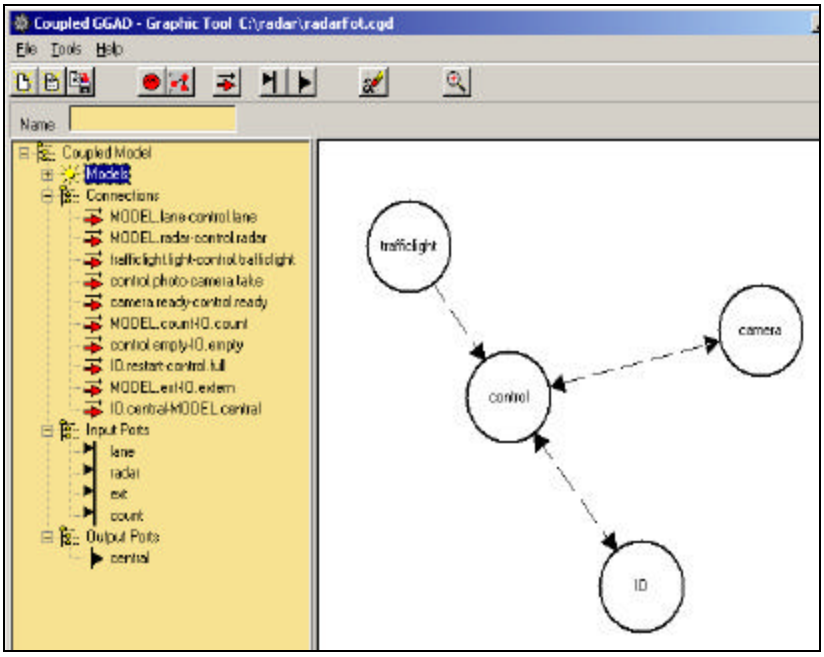


Figure 2: Structure of a photographic radar coupled model.

## 8.0 RECOMMENDATIONS FOR FUTURE SISO ACTIONS - THE WAY AHEAD

- Organization of Educational activities
  - Creation of a DEVS tutorial on-line
  - Presentation of a Tutorial at Spring SIW
  - Several tutorials spread during the year, in SISO and other conferences SCSC/SPECTS/Wintersim
  - SIW meetings will always start with a 1-hour tutorial for newcomers

- Team members started seeking funding for a DEVS workshop organization. Will invite partners from Industry, Government and Academia. Will be held in the country providing the funding (France? Canada – SISO Canada participation?)

- Other members encouraged to follow this model (and organize a workshop within SISO SIW if US funds are obtained)

- Different ongoing activities
  - Seeking funding opportunities for the year 2005-06
  - Good basis for the creation of a Product Nomination, and organization of a PDG
  - Need one more year to finish the current developments

## 9.0 REFERENCES

Zeigler, B. P., Fulton, D., Nutaro, J., Hammonds, P., "M&S Enalbed Testing of Distributed Systems: Beyond Interoperability to Combat Effectiveness Assessment":, 9th Annual Modeling and Simulation Workshop, Dec. 8-11, 2003, ITEA White Sands Chapter.

Zeigler, B.P., Fulton, D., Hammonds, P., Nutoro., J., "Framework for M&S-Based System Development and Testing in Net-centric Environment,", to appear in ITEA Journal, November 2005.

Zeigler, B.P., S.B. Hall, and H.S. Sarjoughian, "Exploiting HLA and DEVS to Promote Interoperability and Reuse in Lockheed's Corporate Environment," Simulation J., Vol. 73, No. 4, pp. 288-295, November 1999.

Zeigler, B.P., H.S. Sarjoughian, "Support for Hierarchical Modular Component-based Model Construction in DEVS/HLA", Simulation Interoperability Workshop, March, Orlando, FL., 1999.

Lake, T., B.P. Zeigler, H.S. Sarjoughian, J. Nutaro, "DEVS Simulation and HLA Lookahead," Simulator Interoperability Workshop (SIW), 00S-SIW-160.

James Nutaro. A Middleware Based Standard for DEVS Simulator Interoperability . 04F-SIW-114

Gabriel Wainer, Trevor Pearce, Amir Saghir . HLA to simulate computer systems at the hardware platform level. 04F-SIW-116

Peter MacSween, Gabriel Wainer. On the Construction of Complex Models Using Reusable Components. 04S-SIW-124

Rami Madhoun, Gabriel Wainer. Modeling Space-Shaped Defense Applications with Cell-DEVS. 05S-SIW-050