DEVS: Past, Present, Future

Bernard Zeigler
Prof. Emeritus, Univ. Arizona
Chief Scientist, RTSync

Journees DEVS Francophone
Cargese, Corsica
April 11, 2016
Outline...

- As DEVS matures new generations are curious about its origins
  - How it was conceived within the intellectual climate of its early years,
  - How it related to simulation language development and discrete event dynamic systems.
- However, looking backward is preparation for going forward
- From a historical perspective, we
  - Present a time line of DEVS milestones
  - Review DEVS briefly
  - Examine a role for DEVS in M&S methodology
  - Survey the state of DEVS today
  - Discuss DEVS research and development
  - Hints of Future Development
Origins of DEVS

Background:
Discrete Event Systems Specification (DEVS) family of M&S formalisms

• DEVS formalizes what a model is, what it must contain, and what it doesn’t contain (experimentation and simulation control parameters are not contained in the model)

• DEVS is **universal** and **unique** for discrete event system models: any system that accepts events as inputs over time and generates events as outputs over time is equivalent to a DEVS: its behavior and structure can be described by such a DEVS.

• DEVS-compliant simulators execute DEVS models **correctly**, repeatably, and efficiently. Closure under coupling guarantees correctness in hierarchical composition of components.

• DEVS models can be simulated on **multiple different execution platforms**, including those on desktops (for development) and those on high-performance platforms, such as multi-core processors

• **Correctness without performance hit**: The Parallel DEVS Simulation Protocol provides close to the best possible performance except possibly where activity is very low or coupling is very small.
Some Highlights in DEVS Development

1976
Classic DEVS Defined

1987
Hierarchical, Modular DEVS

1990
SES and Model-Base

1991
DEV&DESS

1996
Parallel DEVS

1997
Dynamic Structure DEVS

2001
GDEVS

2002
Quantized DEVS

2011
Finite Deterministic DEVS

2014
Modelica&DEVS

DEVS Universality
**Associated Articles**


Chow, Parallel DEVS: A Parallel, Hierarchical Modular Modeling Formalism and its Distributed Simulator, Transactions of the SCS, Vol 13, #2, pp. 55-68, 1996


Quantized-state systems: a DEVS Approach for continuous system simulation,E Kofman, S Junco - Transactions of The SCS,Volume 18, # 1, pp. 2-8 2001


Moon Hwang, Taxonomy of DEVS subclasses for standardization, TMS-DEVS Pages 152-159 , Moon Ho Hwang

Universality: Lattice of DEVS-Representable Formalisms (Vangeleuwe)

Category theory?
DEVS Place within M&S Methodology

Analyzability vs. Indispensability

- Complex Models: Not Analyzable. Simulation needed.
- Trend: With high performance technology.

Low Analyzability vs. High Indispensability

Low Indispensability vs. High Analyzability
DEVS Place within M&S Methodology

Analyzability vs. Simulation Indispensability

- Simplistic Models: Analyzable. Simulation not needed
- Complex Models: Not Analyzable. Simulation needed

M&S Methodology Domain

DEVS focus here

High Analyzability

Low Analyzability

Low Simulation Indispensability

High Simulation Indispensability
DEVS Place within M&S Methodology

Analyzability

High

Low

DEVS focus here

Develop Abstractions

Structural Properties flow

High

Low

Construct Valid Simplifications

Behavioral Parameter Value Constraints flow

Low

High

Simulation Indispensability

M&S Methodology Domain

• Computational complexity
• Understandability
• Descriptive entropy
• Preservation/predictive ability

Parameter Value Constraints flow
Preservation/Predictive Ability ("predictivity") of models

- **Preservation**: Does the lumped model preserve a given property of the base model?
- **Predictivity**: Does a given property of the lumped model imply that the property holds for the base model?
- **Example**: Recurrent (cyclic) vs Absorbing (acyclic) behavior
DEVS makes it easy to cross deterministic / stochastic lines

- **Deterministic DEVS (FDDEVS)**
- **Stochastic DEVS (FPDEVS, Markov)**

**Markov model**
- Probability assigned to transition
- Probability $P = 2/3$
- Extends to Lumping of Markov Models

**Deterministic model**
- Count the number of transition from First block to Second
- Morphism
- Probability $P = 2/3$
Preservation/Predictive Ability ("predictivity") of Markov models from analysis of their underlying Directed Graphs (DG)

* Theorem If $C$ is a directed cycle, then $G \text{ hom} \to C$ iff $G$ contains only cycles of length divisible by the length of $C$.

Pavol Hell, Huishan Zhou, Xuding Zhu Homomorphisms to oriented cycles. 2003
Approximate Morpisms: What’s the probability of finding a reasonably good aggregation when sampling at random?
### DEVS Formalism Provides Frameworks in New Areas

<table>
<thead>
<tr>
<th>Application Area</th>
<th>Novel Feature</th>
<th>Unique Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVS Framework for Simulating Continuous Time Production Flows In Food Industry</td>
<td>New framework for carrying out simulations of continuous-time stochastic</td>
<td>Keep track of parameters related to the process and the flowing material (temperature, concentration of pollutant) is also considered.</td>
</tr>
<tr>
<td></td>
<td>processes</td>
<td></td>
</tr>
<tr>
<td>Development of DEVS Models for Building Energy Design</td>
<td>Allow different professions involved in the building design process to work independently to create an integrated model.</td>
<td>Results indicate that the DEVS formalism is a promising way to improve poor interoperability between models of different domains involved in building performance simulations.</td>
</tr>
<tr>
<td>Quantum Key Distribution (QKD) system with its components using DEVS</td>
<td>DEVS assures the developed component models are composable and exhibit temporal behavior independent of the simulation environment.</td>
<td>Enable users to assemble and simulate any collection of compatible components to represent complete QKD system architectures.</td>
</tr>
<tr>
<td>DEVs Framework for transportation evacuation integrating event scheduling into an agent-based method.</td>
<td>This framework has a unique hybrid simulation space that includes a flexible-structured network and eliminates time-step scheduling used in classic agent-based models.</td>
<td>Hybrid space overcomes the cellular space limitation and provides flexibilities in simulating evacuation scenarios. Model is significantly more efficient than a Repast model.</td>
</tr>
</tbody>
</table>
Hints of Future Development

• Basic Systems Foundations – Iterative Specification, Time base refinement
• Coupling Formalism Specification
• Dynamic Structure Framework
• Universal Specification Language for DEVS – CML-DEVS: sets, first order logic
• Extensions of Abstract Simulator Concept
• Category Systems Organization
  • Subclasses of DEVS – (Semi-)Markov Models – metrics
  • Approximate morphism – quasi-lumpability– metrics
• DEVS Pathways web-based execution
• DEVS as a basis M&S as a service – cloud, web, virtual containers - VLE in France, NATO interest
• Continued Development of new DEVS Tools: http://www.sce.carleton.ca/faculty/wainer/standard/tools.htm
• More books, videos, demos. Etc.
Videos: www.ms4systems.com and YouTube

Computer Simulation Pioneer: NCSU Simulation Archive

Formalizing Porter's Integrated Practice Unit with System-of-Systems Modeling and Simulation

Extra-Clinical Care Coordination: Pathways Community HUB Model

The Role of Modeling and Simulation in Coordination of Health Care

Modeling and Simulation for Engineering of Self-Improving Service Systems of Systems: Barriers and Prospects

Introduction to MS4 Me and Markov modeling