

# Chapter 3

## The DEVS Formalism

**Rhys Goldstein**

*Autodesk Research, Canada*

**Gabriel A. Wainer**

*Carleton University, Canada*

**Azam Khan**

*Autodesk Research, Canada*

### ABSTRACT

*The DEVS formalism is a set of conventions introduced in 1976 for the specification of discrete event simulation models. This chapter explains the core concepts of DEVS by applying the formalism to a single ongoing example. First, the example is introduced as a set of informal requirements from which a formal specification is to be developed. Readers are then presented with alternative sets of modeling conventions which, lacking the DEVS formalism's approach to representing state, prove inadequate for the example. The chapter exploits the DEVS formalism's support for modular model design, as the system in the example is specified first in parts and later as a combination of those parts. The concept of legitimacy is demonstrated on various model specifications, and the relationship between DEVS and both object-oriented programming and parallel computing is discussed.*

### INTRODUCTION

The DEVS (Discrete Event System Specification) formalism is a set of conventions for specifying discrete event simulation models. It was introduced in 1976 with the publication of Bernard Zeigler's *Theory of Modeling and Simulation* (Zeigler, 1976).

DOI: 10.4018/978-1-4666-4369-7.ch003

## The DEVS Formalism

While the latest edition of that book (Zeigler et al., 2000) provides a comprehensive overview of DEVS theory, here we focus on the application of the core concepts. The chapter is organized around a particular example: the simulation of an automatic lighting system in an office environment. We develop this example from a set of informal requirements to a complete formal specification.

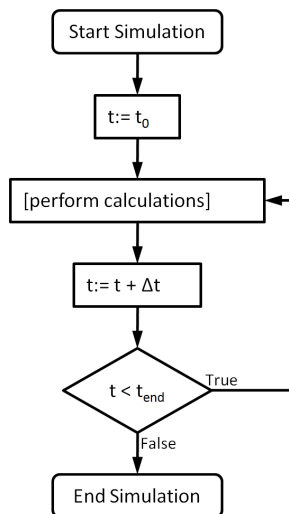
Before we begin, let us clarify the difference between a discrete time simulation and a discrete event simulation. Numerous simulations are implemented with a time variable  $t$  that starts at some initial value  $t_0$ , and increases by a fixed time step  $\Delta t$  between calculations. The flowchart in Figure 1 outlines the procedure.

This type of simulation is a discrete time simulation, as  $t$  is effectively a discrete variable. The approach is simple and familiar, but limited in that the duration between any pair of inputs, outputs, or state transitions must be a multiple of  $\Delta t$ .

DEVS can be applied to discrete time simulation, but it is best suited to the discrete event approach for which it was invented. In a discrete event simulation, time is continuous. Any pair of events can be separated by any length of time, and there is generally no need for a global  $\Delta t$ . Later in the chapter we will present a procedure like that in Figure 1, but suitable for all discrete event simulations.

The adoption of a discrete event approach impacts the model development process. For example, suppose one designs separate models for different parts of a larger system. Ideally, modeling the overall system would be a simple matter of combining these submodels. With discrete time simulation, one would have to choose a single  $\Delta t$  appropriate for every submodel, or invent some scheme by which only certain submodels experience events at any given iteration. With DEVS, two

Figure 1. Discrete time simulation procedure



39 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: [www.igi-global.com/chapter/devs-formalism/77798](http://www.igi-global.com/chapter/devs-formalism/77798)

## Related Content

---

### Modeling for System's Understanding

Marisa A. Sánchez (2014). *Formal Languages for Computer Simulation: Transdisciplinary Models and Applications* (pp. 38-61).

[www.irma-international.org/chapter/modeling-system-understanding/77797](http://www.irma-international.org/chapter/modeling-system-understanding/77797)

### Reducing Simulation Runtime in Wireless Sensor Networks: A Simulation Framework to Reduce WSN Simulation Runtime by Using Multiple Simultaneous Instances

Pedro Pinto, António Alberto Pinto and Manuel Ricardo (2016). *Handbook of Research on Computational Simulation and Modeling in Engineering* (pp. 726-741).

[www.irma-international.org/chapter/reducing-simulation-runtime-in-wireless-sensor-networks/137461](http://www.irma-international.org/chapter/reducing-simulation-runtime-in-wireless-sensor-networks/137461)

### Virtual Modelling of Prehistoric Sites and Artefacts by Automatic Point-Cloud Surveys

Mercedes Farjas, Francisco J. García-Lázaro, Julio Zancajo, Teresa Mostaza and Nieves Quesada (2011). *Virtual Technologies for Business and Industrial Applications: Innovative and Synergistic Approaches* (pp. 201-217).

[www.irma-international.org/chapter/virtual-modelling-prehistoric-sites-artefacts/43412](http://www.irma-international.org/chapter/virtual-modelling-prehistoric-sites-artefacts/43412)

### Sigma Tuning of Gaussian Kernels Detection of Ischemia from Magnetocardiograms

Long Han, Mark J. Embrechts, Boleslaw K. Szymanski, Karsten Sternickel and Alexander Ross (2011). *Computational Modeling and Simulation of Intellect: Current State and Future Perspectives* (pp. 206-223).

[www.irma-international.org/chapter/sigma-tuning-gaussian-kernels-detection/53307](http://www.irma-international.org/chapter/sigma-tuning-gaussian-kernels-detection/53307)

### AI and Digital Twins: A Breakthrough Approach to Mental Health Diagnosis and Therapy

Sanket Dan, Jayeeta Ghosh, Bikramjit Sarkar and Jayshree Bhattacharya (2025). *AI-Powered Digital Twins for Predictive Healthcare: Creating Virtual Replicas of Humans* (pp. 201-232).

[www.irma-international.org/chapter/ai-and-digital-twins/373670](http://www.irma-international.org/chapter/ai-and-digital-twins/373670)