

# Chapter 5

## DEVS–Based Simulation Interoperability

**Thomas Wutzler**

*Max Planck Institute for Biogeochemistry, Germany*

**Hessam Sarjoughian**

*Arizona Center for Integrative Modeling and Simulation, USA*

### ABSTRACT

*This chapter introduces the usage of DEVS for the purpose of implementing interoperability across heterogeneous simulation models. It shows that the DEVS framework provides a simple, yet effective conceptual basis for handling simulation interoperability. It discusses the various useful properties of the DEVS framework, describes the Shared Abstract Model (SAM) approach for interoperating simulation models, and compares it to other approaches. The DEVS approach enables formal model specification with component models implemented in multiple programming languages. The simplicity of the integration of component models designed in the DEVS, DTSS, and DESS simulation formalisms and implemented in the programming languages Java and C++ is demonstrated by a basic educational example and by a real world forest carbon accounting model. The authors hope, that readers will appreciate the combination of generalness and simplicity and that readers will consider using the DEVS approach for simulation interoperability in their own projects.*

### INTRODUCTION

Interoperability among simulators continues to be of key interest within the simulation community. A chief reason for this interest is the existence of heterogeneous legacy simulations which are developed using a variety of programming practices and software engineering approaches. For many studied

problems there exist already simulations models. Much work has been dedicated to develop the models, estimate parameters, verify the simulations and validate the model assumptions by comparing model results to observations. Hence, it is desirable to reuse such existing models for additional tasks and to integrate parts of several existing models into a common framework. However, this integration is hampered by the fact that the existing models have been developed in different programming

DOI: 10.4018/978-1-60566-774-4.ch005

languages, with different software and different software engineering approaches. There exist already several approaches to implement a joint execution and some of them are summarized in the background section. In this book chapter we suggest an additional approach to realize interoperability among disparate simulators that utilizes the Discrete Event System Specification (DEVS). We argue that this approach yields several benefits. One of the most important benefits is the reduced complexity for adapting heterogeneous legacy models.

The interoperability issue will be exemplified with a typical problem which we encountered in our work. The task was to construct a forest ecosystem carbon tracking model. With such a model the exchange of carbon between the forest ecosystem and the atmosphere can be studied. One can then implement and test several scenarios of forest management and simulate how these influence the exchange of carbon dioxide with the atmosphere to explore how these management strategies influence atmosphere carbon concentrations and global warming. There already existed a model for the stand growth, i.e. the growth of trees within a plot, which was based on extensive measurement of tree properties within the study region (Appendix A). It accounted for competition among different trees and incorporated knowledge of many species. Hence, the source code was quite extensive. There already existed also a model for the soil carbon balance that have been used and validated by many research groups (Appendix B). Only the management strategies had to be implemented as a new component model.

However, the stand growth model was specified as discrete-time model (DTSS) and implemented in JAVA and the soil carbon balance model was specified as continuous (differential equations) model (DESS) and implemented in C++ (see Figure 1). So how could we couple the heterogeneous models and perform a joint and integrated simulation?

The purpose of this chapter is to discuss the benefits of DEVS concept compared to several commonly applied alternatives of tackling the problem of simulation interoperability. To this end we introduce the Shared Abstract Model (SAM) interoperability approach (Wutzler & Sarjoughian, 2007). We will exemplify its application first by a simple educational example model. This simple example provides the basis to compare the SAM approach to other DEVS-based approaches. Finally we present a setup using SAM that we used to solve the problem posed in Figure 1. Next, we start with a discussion of several commonly used alternatives of tackling the problem of simulation interoperability.

## **BACKGROUND**

There exist several alternative approaches to construct a model from the component models such as the one described in the Introduction section (refer to Figure 1). One solution to the interoperability task is to develop all models using a single programming language. We could have reimplemented the soil model in JAVA as a time-stepped model. The reimplementation alternative, however, impedes independent development of the component models by different research groups. This disadvantage combined with weak support for simulation model verification and validation results makes reimplementation unattractive and generally impractical. In particular, there is a growing need for alternative simulation interoperability approaches in the domain of environmental modeling and simulation (Papajorgji et al., 2004; Filippi & Bisgambiglia, 2004; Roxburgh & Davies, 2006).

A second alternative is to use the High Level Architecture (HLA) simulation middleware (IEEE, 2000). It supports interoperation among simulated and physical software/systems. The HLA standard is widely used and considered more powerful in terms of supporting different kinds

15 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-based-simulation-interoperability/38258](http://www.igi-global.com/chapter/devs-based-simulation-interoperability/38258)

## Related Content

---

### Grid-Enabling Applications with JGRIM

Cristian Mateos, Alejandro Zunino and Marcelo Campo (2009). *International Journal of Grid and High Performance Computing* (pp. 52-72).

[www.irma-international.org/article/grid-enabling-applications-jgrim/3970](http://www.irma-international.org/article/grid-enabling-applications-jgrim/3970)

### Introduction

Valentin Cristea, Ciprian Dobre, Corina Stratan and Florin Pop (2010). *Large-Scale Distributed Computing and Applications: Models and Trends* (pp. 1-22).

[www.irma-international.org/chapter/introduction/43100](http://www.irma-international.org/chapter/introduction/43100)

### Granular Computing Based Data Mining in the Views of Rough Set and Fuzzy Set

Guoyin Wang, Jun Hu, Qinghua Zhang, Xianquan Liu and Jiaqing Zhou (2010). *Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation* (pp. 148-178).

[www.irma-international.org/chapter/granular-computing-based-data-mining/44703](http://www.irma-international.org/chapter/granular-computing-based-data-mining/44703)

### Proactive Auto-Scaling Algorithm (PASA) for Cloud Application

Mohammad Sadegh Aslanpour and Seyed Ebrahim Dashti (2017). *International Journal of Grid and High Performance Computing* (pp. 1-16).

[www.irma-international.org/article/proactive-auto-scaling-algorithm-pasa-for-cloud-application/185770](http://www.irma-international.org/article/proactive-auto-scaling-algorithm-pasa-for-cloud-application/185770)

### A Comprehensive Survey on Sentiment Analysis in Twitter Data

Hema Krishnan, M. Sudheep Elayidom and Santhanakrishnan T. (2022). *International Journal of Distributed Systems and Technologies* (pp. 1-22).

[www.irma-international.org/article/a-comprehensive-survey-on-sentiment-analysis-in-twitter-data/300352](http://www.irma-international.org/article/a-comprehensive-survey-on-sentiment-analysis-in-twitter-data/300352)