N-dimensional Cell-DEVS Models

GABRIEL A. WAINER  
gabriel.wainer@sce.carleton.ca  
Department of Systems and Computer Engineering, Carleton University, 4456 Mackenzie Building, 1125 Colonel By Drive, Ottawa, ON K1S 5B6, Canada

NORBERT GIAMBIASSI  
norbert.giambiasi@iuspim.u-3mrs.fr  
DIAM-IUSPIM, Université d’Aix-Marseille III, Av. Escadrille Normandie Niemen, 13397 Marseilles, Cedex 20, France

Abstract. This article presents an extension to the timed binary Cell-DEVS paradigm. The goal is to allow the modeling of n-dimensional generic cell spaces, including transport or inertial delays for each cell. The automatic definition of cell spaces is achieved, simplifying the construction of new models. The model definition is independent of the simulation mechanism, easing the verification of the structural models. It was shown that the Cell-DEVS models can be integrated in a DEVS hierarchy, improving the definition and description of complex systems. This approach allows improvements in the execution times and precision for the cell spaces simulations due to the use of a continuous time base.

Keywords: DEVS models, modeling paradigms, cellular automata, discrete event simulation, cell-DEVS models

1. Introduction

Complex physical systems have been studied for centuries using different approaches. In most cases, partial differential equations have been the tool of choice. Nevertheless, the appearance of digital computers allowed scientists and engineers to attack the problem from two different point of views. Even in these days, difference equations are implemented in digital computers to solve this class of problems. Unfortunately, the complexity of certain problems is such that no solutions can be found. In these cases, the use of computer simulation enables to solve more complex problems, providing solutions to specific problems.

Many of these systems can be represented as cell spaces. The cellular automata formalism (Wolfram, 1986; Toffoli and Margolus, 1987) has been widely used to describe complex systems with these characteristics. Cellular automata are discrete-time discrete models described as cells organized as n-dimensional infinite lattices. These automata evolve by executing a global transition function that updates the state of every cell in the space. Each cell in the automaton has a discrete value that is changed by a local computing function. The behavior of this function depends on the results of a function that executes locally in each cell. This function uses the present value for the cell and a finite set of neighbor cells to compute the new state (see Figure 1).

Conceptually, these local functions are computed synchronously and in parallel, using the state values of the present cell and its neighbors. The use of discrete time poses
constraints in the precision and execution performance of these complex models. To achieve the desired accuracy, smaller time slots must be used, increasing processing time. To avoid these problems, asynchronous solutions can be used.

In general several cells do not need to be updated in every step, wasting computation time. These problems can be solved using a continuous time base, providing instantaneous events that can occur asynchronously at unpredictable times. This approach was considered in Zeigler (1976; 1984), where discrete event cellular models were presented. Discrete event cellular models were applied in real world applications in later works (Moon et al., 1996; Zeigler et al., 1998). These works presented the use of DEVS (Zeigler, 1976; Zeigler et al., 2000) as the modeling technique to be applied to improve the performance in cellular models.

DEVS is used to specify formally discrete events systems using a modular description. The quantitative complexity of the problems is attacked by using a hierarchical approach. A model is seen as composed by behavioral (atomic) submodels that can be combined into structural (coupled) models. As the formalism is closed under coupling, coupled models can be seen as new base models that can integrated hierarchically. This strategy allows the reuse of tested models, improving the safety of the simulations and allow reducing the development times. DEVS provides the advantages of being a formal approach. Formal specification mechanisms are useful to improve the security and development costs of a simulation. A formal conceptual model can be validated, improving the error detection process and reducing testing time. DEVS models are closed under coupling, therefore a coupled model is equivalent to an atomic one, improving reuse. DEVS supplies facilities to translate the formal specifications into executable models. In this way, the behavior of a conceptual model can be validated against the real system, and the response of the executable model can be verified against the conceptual specification.

DEVS, as a discrete event paradigm, uses a continuous time base, which allows accurate timing representation. Precision of the conceptual models can be improved, and CPU time requirements reduced. Higher timing precision can be obtained without using small discrete time segments (that would increase the number of simulation cycles).

Recalling the definitions, a DEVS atomic model can be formally described as:

\[ M = (X, S, \delta_s, \delta_e, \lambda, D) \]

![Figure 1. Sketch of a cellular automaton.](image)