Simulation Modelling Practice and Theory 000 (2017) 1–10

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

An approach to agent-based modeling with Modelica

Victorino Sanz [a,*], Federico Bergero [b], Alfonso Urquia [a]

[a] Departamento de Informática y Automática, UNED, Juan del Rosal, 16, Madrid, 28040, Spain
[b] French-Argentine International Center for Information and Systems Sciences (CIFASIS-CONICET), 27 de Febrero 210 bis, S2000EZP, Rosario, Argentina

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Agent-based model
Modelica
Hybrid systems

ABSTRACT

Modelica is a free, general-purpose object-oriented equation-based modeling language. It is mainly designed to describe systems using the physical modeling approach. Our proposal to describe Agent-Based Models (ABMs) in Modelica is discussed in this manuscript. The contribution of the presented work is twofold: firstly, to analyze the conceptual requirements to describe ABMs in Modelica; and secondly, to develop a prototype implementation following the previous analysis. Agents are described using a message passing communication mechanism previously proposed by the authors. Additional extensions to this mechanism are proposed in order to describe agent interactions. The environment, where the agents live, is described as a two-dimensional cellular automaton. A new Modelica library, named ABMLib, developed to support this functionality, is presented. A prototype implementation of the message passing mechanism and ABMLib models has been performed to demonstrate the functionality of the library as a proof-of-concept for this proposal. The library is freely available at www.euclides.dia.uned.es/vsanz.

1. Introduction

Agent-Based Models (ABMs) are discrete-event models composed of a variable number of “living” objects, named agents, that behave following a pre-defined set of rules (i.e., agent behavior), and interact among them and with their environment (i.e., the physical space where the agents “live”) [1]. The individual behavior of each agent is defined using simple rules, or algorithms, but the simulation of the whole model may lead to complex and emergent behaviors. In this manuscript, the description of ABMs using the Modelica language is discussed.

Modelica supports the description of mathematical models following the physical modeling paradigm [2]. Modelica models are described as a combination of acausal equations, algorithms and events, using the hybrid DAE formalism (cf. [3] for a detailed description of the formalism). The causality of the model is automatically computed by means of symbolic manipulations of the equations before generating the executable code [4].

The description of ABMs in Modelica could be used to perform a qualitative description of models, or parts of models, in contrast with the quantitative approach given by equation-based modeling [5]. ABMs can be used to represent heterogeneous objects in the model (i.e., agents of the same type are used to represent different individuals in the model with different characteristics or even different behavior) while equations are used to represent homogeneous quantities. Adaptive

* Corresponding author.
E-mail addresses: vsanz@dia.uned.es (V. Sanz), bergero@cifasis-conicet.gov.ar (F. Bergero), aurquia@dia.uned.es (A. Urquia).

https://doi.org/10.1016/j.simpat.2017.12.012
1569-190X/

or learning behaviors can also be described in terms of ABMs. In this way, the combination of ABMs with other Modelica models enhances the functionality of the language and the description of more complex hybrid systems.

Other authors have made efforts to combine equation-based models and ABMs. The LEADSTO language combines dynamic systems of equations with ABMs [6]. Another approach has been to combine ABMs with System Dynamics, using Anylogic, to describe health-care systems [7]. Humans are described using agents, while System Dynamics is used to describe disease dynamics. A similar approach is used to simulate antibiotic resistance in hospital wards [8]. Intra-host dynamics (i.e., bacterial-level processes inside individuals) are described using differential equations, while inter-host dynamics (i.e., relations between humans) are described using ABMs. Also, Dymola and JADE have been combined using a co-simulation approach to describe control for office spaces [9].

The proposal presented in this manuscript is to describe agents as individual messages flowing between components of a flowchart diagram, which is analogous to a coupled DEVS model [10]. Agent behavior is described by the components of the flowchart diagram independently of the environment, but agents can interact with it. The environment is represented as a two-dimensional cellular automaton, using CellularAutomataLib2 [11]. Some extensions to the message passing communication mechanism, previously proposed by the authors [12], are presented to describe agent's interactions. All this functionality is included in a new Modelica library, named ABMLib, designed and developed by the authors. ABMLib models can be also combined with other Modelica models. ABMLib approach is similar to the description of systems using process calculus, where the processes communicate using messages [13]. However, in this proposal agent actions are not synchronized by means of communication rendezvous, but at discrete points in time [14]. ABMLib approach is also similar to actor-oriented models, but in this case messages represent the agents themselves and not the data flowing between actors.

A prototype implementation of the message passing mechanism and the new library has been developed and tested, and an application example using a Lotka–Volterra model is presented in this manuscript. The presented model combines ABMs with continuous-time equations in order to illustrate the benefits of supporting ABMs in Modelica.

The structure of the manuscript is as follows. The requirements to describe ABMs in Modelica are discussed in Section 2. The message passing communication mechanism is briefly described in Section 3, together with the proposed extensions required for ABM development. The ABMLib library is described in Section 4, and the combination of ABMLib models with other Modelica models is described in Section 5. The Lotka–Volterra model described using ABMLib and other Modelica functionality is presented in Section 6. Finally, some conclusions and future work ideas are given in Section 7.

2. Requirements for describing ABMs in Modelica

Modelica and ABM are conceptually different. ABMs are composed of agents, environments and interactions [1]. Modelica models are mainly described by means of equations, while the behavior of agents is described using rules. Agents can be created or removed during the simulation run, while the number of variables and equations in Modelica has to remain constant. ABMs are usually dependent on the spatial coordinates, with the agents moving and interacting around the environment, while the only independent variable in Modelica is the time. The functionality of the Modelica language that can be used to describe these characteristics is discussed below.

- Agents are described by means of their state and behavior. Modelica simple data types and complex data structures can be used to represent the state of the agents. Usually, the behavior of agents is described as a set of rules or actions. Modelica algorithm sections can be used to describe the behavior of the agents.
  However, Modelica does not support changes in the number of variables and/or equations during the simulation. These changes occur during the creation or removal of agents from the model. An additional mechanism needs to be used to represent the variation of agents during the simulation.
  The proposed approach is to graphically represent the behavior of agents as a flowchart diagram. Agents are represented as messages that are sent from one component to another in the diagram. The components represent the individual actions performed by agents. The diagram includes components to create agents and to remove them from the simulation. The components of the diagram communicate using the message passing communication mechanism previously proposed by the authors [12], which is capable of dealing with a variable number of messages during the simulation run.
- The environment represents the physical space where the agents behave and interact, and can be defined in multiple ways depending on the necessities of the model. Some authors consider the environment as a set of passive agents, since they can also have state and behavior [15]. Also, the environment could only represent feasible agent interactions (e.g., links in a social network).
  As a first approach, our proposal is to represent the environment as a cellular automaton. This is a two-dimensional square lattice, where the states of the cells are represented using some variables and all the cells share the same behavior defined using a transition function. The state of the cells is periodically updated using the transition function. The CellularAutomataLib2 library, developed by the authors, supports the description and efficient simulation of cellular automata in Modelica [11].
- Agents can interact with other agents or with the environment. Modelica provides functionality to access models and variables in the hierarchy of models and libraries of models. This functionality includes the dot-notation, that allows to access models in other libraries or packages, or the `inner`/`outer` variable modifiers that can be used to access variables and models defined in another part of the hierarchy. Additionally, CellularAutomataLib2 includes interface models