Representing Fibonacci function through cellular automata using Specification and Description Language

Pau Fonseca i Casas
Universitat Politècnica de Catalunya
Jordi Girona 1-3
08034, Barcelona, Catalunya, SPAIN
(+34)934017732
pau@fib.upc.edu

Màxim Colls
Universitat Politècnica de Catalunya
Jordi Girona 1-3
08034, Barcelona, Catalunya, SPAIN
(+34)934017732
maxim.colls@fib.upc.edu

Josep Casanovas
Universitat Politècnica de Catalunya
Jordi Girona 1-3
08034, Barcelona, Catalunya, SPAIN
(+34)934017732
josepk@fib.upc.edu

Abstract

In this poster we show how to use Specification and Description Language (SDL) to represent cellular automata models. To achieve that we use a generalization of the common cellular automata, named m:n-CA^k. Also we add some extension to SDL language to simplify the representation of these automata. Thanks to SDL and m:n-CA^k the behavior of the cellular automata model can be defined in a graphical way allowing the complete and unambiguous description of the simulation model that uses it. SDL is a modern object oriented language that allows the definition of distributed systems. It has focused on the modeling of reactive, state/event driven systems, and has been standardized by the International Telecommunications Union (ITU) in the Z.100.

Keywords: SDL, Simulation, Formalisms, cellular automata.

1. INTRODUCTION

The construction of a simulation model sometimes lacks in the formalization process needed to understand the model before any implementation. This model behavior understanding helps in the implementation process and in the communication between the different personnel involved in the model construction. Also can be considered a product itself [1]. The proposed methodology (and infrastructure) allows the definition (and implementation) of a simulation model that uses cellular automata [2] following the Specification and Description Language.

2. M:N-CAK CELLULAR AUTOMATA

The main objective of m:n-CA^k is to simplify the use of geographical and environmental information in a simulation model. As a result, m:n-CA^k cellular automata is defined. Mainly is a generalization that simplifies the use of raster and vectorial data in a simulation model, and also extends the definition of common cellular automaton over the Topology mathematical concept. m:n-CA^k was first defined on [3], in the next lines we describe the more important aspects of this cellular automaton generalization.

First said that m:n-CA^k means multi n-dimensional cellular automaton. Mainly is a generalization of cellular automata defined as follows:

Definition 1. m:n-AC^k

A multi n dimensional cellular automaton is a cellular automaton generalization composed by m layers with n dimensions each one.

The representation is:

$$m.:n - AC^k$$

Where

m: is the automaton number of layers.

n: is the different layers dimension.

k: is the number of main layers (1 by default).

With this notation, a usual two dimensional cellular automaton is represented by a 1:2-AC. A transition in a m:2-AC cellular automata is defined as in a 2-dimensional cellular automata, but main layer cell state is a combination of data contained in the m-1 secondary layers at the same position.

3. SDL EXTENSIONS

In order to simplify the representation (avoiding the representation of all the cellular automaton cells) we decide to add a new kind of agent to SDL [4] language, the **mnca**. This agent has the same behavior as the agent block, with a particularity, **mnca** agent is defining the entire cellular automaton, but only is needed to represent one cell. This implies that is needed to use a declarations section that defines the structure of the cellular automata (mainly the dimension of the cellular automata, and the size of each one of these dimensions). Since **mnca** block represents all the cells of the cellular automaton, is needed to define how send a signal from one cell to other cell of the same **mnca**.

To do this we are using an extension of the language that allows completing the definition of a signal (Figure 1).

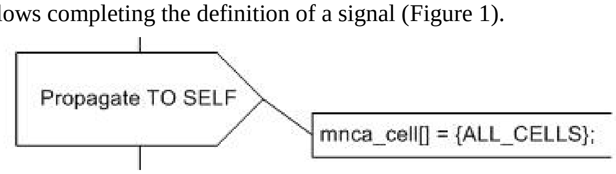

Figure 1: Sending a signal from on cell to other in the same mnca block.

First, note that the signal is send to the same element (TO SELF), as is defined in the standard. In order to distinguish between the different cells represented in the **mnca** SDL agent we are using the extension mnca_cell[]={cells} that defines the cells of the **mnca** block that receives the signal.

We can define an array of cells (the cells that can receive the signal). Also we can use ALL_CELLS, to send the signal to all the cells of the **mnca** agent.

As an example we represent cellular automata that calculate the well known Fibonacci function. Looking this structure the m:n-CA^k cellular automata used here is 1:2-CA over N.

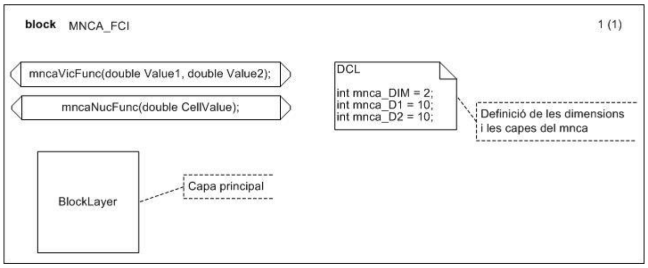

Figure 2. m:n-CA^k cell representation for Fibonacci model. Here we can see the relation between the different layers.

From this complete representation of the cellular automata structure we can go further to define the behavior (**Figure 3**).

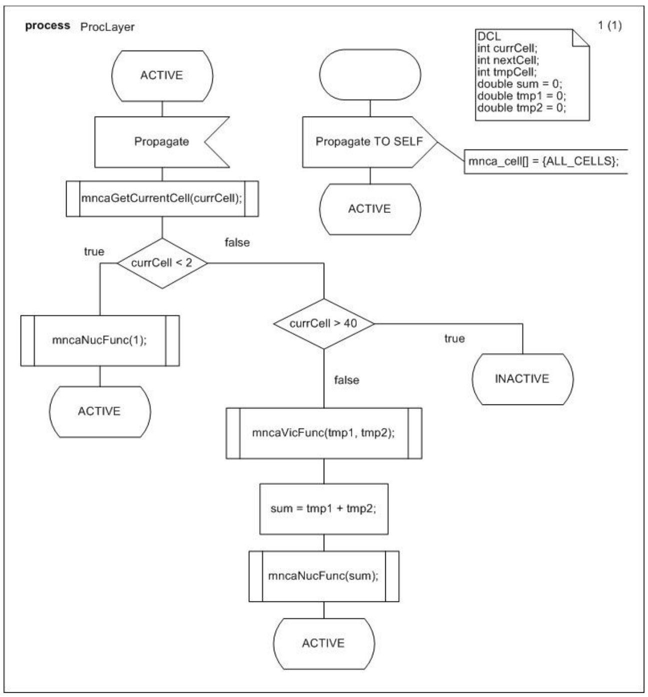

Figure 3. representation of the behavior of the cell.

The model is represented using Microsoft Visio®. From this representation an XML is obtained (thanks to a plug-in developed by our team). This XML is loaded in SDLPS [5] who performs the simulation. We are using XML file instead of the SDL-PR representation because XML offers different advantages to manage the data. Also allows the incorporation of blocs and metadata inside the XML model representation. This metadata is useful to add information about the graphical representation that we want that each one of the different SDL agents have.

The initial conditions of the cellular automata can be defined using GIS data that follows IDRISI32 format. The output is a new file that again follows IDRISI32 format. As an example you can obtain the next output for our model.

0\n1\n1\n2\n3\n5\n8\n13\n21\n34\n55\n89\n (…)

This information (following the IDRISI format) is represented in two files. (i) Layer.doc, the file containing the description of the raster file. (ii) Layer.img, the file containing the data of the raster file.

4. CONCLUSIONS AND FUTURE WORK

This poster proposes a solution to represent the behavior of cellular automata graphically using the Specification and Description Language.

To do this we propose two solutions to two existing problems. First, we had shown how to deal with time in SDL. Second how to represent cellular automata using the language. To do this first an extension of the common cellular automata is used. This extension allows the definition of several functions that helps in the description of its behavior. From these extensions we had shown how a Fibonacci model can be represented using SDL. The graphical representation of the cellular automata behavior helps in the understanding of its behavior. In this kind of models, where usually are formed by multidisciplinary individuals, these graphical tool can be very valuable.

The future work is focused in the implementation of all this extensions in a way that allows a distributed execution of the automata with a dynamic assignation of the existing resources. Also we are developing some different models using this approximation to represent environmental phenomena, based on models developed using other approximations like slap avalanches [6] or wildfires [7].

References

[1] Brade, D. (2000). Enhancing modeling and simulation accreditation by structuring verification and validation results. In J. A. Joines, R. R. Barton, K. Kang, & P. A. Fishwick (Ed.), *Winter Simulation Conference.*

[2] Emmeche, C. (1998). *Vida Simulada en el ordenador*. Barcelona, Catalunya: Gedisa.

[3] Fonseca i Casas, P., & Casanovas, J. (2005). Simplifying GIS data use inside discrete event simulation model through m_n-AC cellular automaton. *Proceedings ESS 2005.*

[4] Telecommunication standardization sector of ITU. (1999). *Specification and Description Language (SDL).* Retrieved April 2008, from Series Z: Languages and general software aspects for telecommunication systems.: http://www.itu.int/ITU-T/studygroups/com17/languages/index.html

[5] Fonseca i Casas, P. (2008). SDL distributed simulator. *Winter Simulation Conference 2008.* Miami: INFORMS.

[6] Fonseca i Casas, P., & Rodríguez Fontoba, S. (2007). Using GIS data in a m:n-ACk cellular automaton to perform an avalanche simulation. *Geographical Information Science Research UK Conference 2007.* National University of Ireland Maynooth.

[7] Fonseca i Casas, P., Casanvas, J., & Montero, J. (2004). GIS and simulation system integration in a virtual reality environment. *Proceedings GISRUK 2004*, (pp. 403-408).