

TOWARDS A REPRESENTATION OF ENVIRONMENTAL MODELS USING SPECIFICATION AND DESCRIPTION LANGUAGE

From the Fibonacci model to a wildfire model

Pau Fonseca i Casas, Màxim Colls, Josep Casanovas

Universitat Politècnica de Catalunya, Jordi Girona 1,3, Barcelona, Catalunya, Spain
pau@fib.upc.edu, maxim.colls@fib.upc.edu, josepk@fib.upc.edu

Keywords: SDL, Simulation, Formalisms, cellular automata.

Abstract: In this paper we explore how we can use Specification and Description Language (SDL) to represent environmental models. Since the main concern in this kind of models is the representation of the geographical information data, we analyze how we can represent this information in the SDL diagrams. We base our approach using two examples, a representation of the Fibonacci function using a cellular automaton, and the representation of a wildfire model. To achieve this we propose the use of a language extension to Specification and Description Language. This allows the simplification of the representation of cellular automata. Thanks to this we can define the behavior of environmental models in a graphical way allowing its complete and unambiguous description. SDL is a modern object oriented formalism 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.

1. INTRODUCTION

From the different phases of a simulation model construction, the formalization phase, sometimes is missed. This phase is needed in order to understand the model before any implementation. The specification 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 (Brade, 2000).

Different formalisms exist in order to represent a simulation model, like Petri Nets or DEVS among others. Some tools have been built in order to allow help the model implementation from the specification (De Lara, et al., 2002), (Praehofer, et al., 1993) and some allows the distribute execution of the models like CD++ (Wainer, et al., 2003).

We select Specification and Description Language in order to represent our simulation, an ISO language than allow representing the models graphically. SDL (Doldi, 2003), (Telecommunication standardization sector of ITU, 1999) is the acronym of Specification and Description Language.

SDL is an object-oriented, formal language defined by the International Telecommunication Union – Telecommunication Standardization Sector (ITU-T) (formerly *Comité Consultatif International Télégraphique et Téléphonique* [CCITT]) as Recommendation Z.100 (Telecommunication standardization sector of ITU, 1999). The language is designed to specify complex, event-driven, real-time, interactive applications involving many concurrent activities using discrete signals to enable communication.

SDL is a powerful and modern language widely used in different areas, not only in simulation area. It has been standardized by the International Telecommunications Union (ITU) in the Z.100, and can be used easily in combination with UML.

The definition of the model is based on different components:

- **Structure:** system, blocks, processes and processes hierarchy.
- **Behavior:** defined through the different processes.
- **Data:** based on Abstract Data Types (ADT).
- **Communication:** signals, with the parameters and channels that the signals use to travel.

- **Inheritances:** to describe the relationships between, and specialization of, the model elements.

The language has 4 levels (i) System, (ii) Blocks, (iii) Processes and (iv) Procedures (Figure 1).

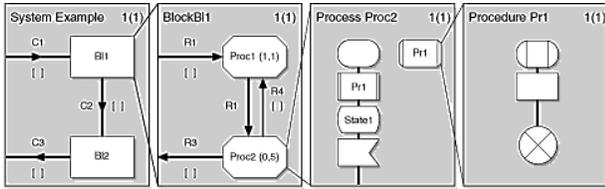


Figure 1: SDL levels (IEC)

To know more about SDL the recommendation Z.100 (Telecommunication standardization sector of ITU, 1999) can be consulted, also a lot of information can be reviewed in the www.sdl-forum.org website.

In our domain the signals are equivalent to the events that rule the modification of the states of the different elements, for that in this paper the use of event or signal is equivalent.

1.1 Cellular automata

Cellular automata must be defined in SDL because simplify the interaction of simulation models and GIS data (Benenson, et al., 2004). The cellular automaton we are using is an extension of the common cellular automaton named $m:n-CA^k$. Its definition can be found on (Fonseca i Casas, et al., 2005). Cellular automata are discrete dynamical systems whose behavior is completely specified in terms of a local relation (Emmeche, 1998).

One-dimensional cellular automata are based in a row of "cells" and a set of "rules". A two-dimensional cellular automata uses rectangular grids of cells. Each one of the different cells can be in one of different "states" (the number of possible states depends on the automata). Thinking states as numbers, in a two-state automata, each cell can be only in 1 or 2 state. Cells represent automata space; time advances in discrete steps following "the rules", the laws of "automata universe", usually expressed in a small look-up table. At each step every cell computes its new state in function of its closer neighbors. Thus, system's laws are local and uniform.

2. IMPLEMENTATION

To implement our models we can use different existing tools that understand SDL language, like Cinderella (CINDERELLA SOFTWARE, 2007) or Telelogic (IBM, 2009) of IBM. We develop our tool in order to improve the existing solutions adding some new capabilities:

- Allow to work with the delaying signals (as we can see below).
- Allow to work with cellular automata.
- Allow to work with intelligent agents.
- Allow a distributed simulation of the models.

For these reasons we decide to implement our tool named SDLPS (Fonseca i Casas, 2008).

2.1 Proposed SDL extensions

In SDLPS all the signals can carry the parameter defined in the structure represented in the Figure 2.

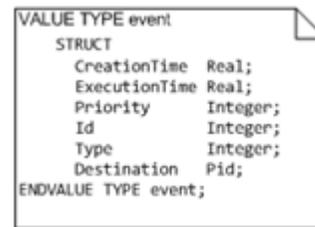


Figure 2: Structure related to the SDLPS signals.

ExecutionTime, of **event** structure, allows defining the time when the signal (carrying the event) must reach its destination. Other elements are: (ii) *Priority*, the priority of the event, used to break a possible simultaneity of events. (iii) *CreationTime*, representing the time when the event is created. (iv) *Id*, an identifier of the event. (v) *Time*, the clock of the process. (vi) *Destination*, the final destination (process PID) of the signal. Not all the parameters of **event** structure must be defined, only those needed to fully define the behavior of the model.

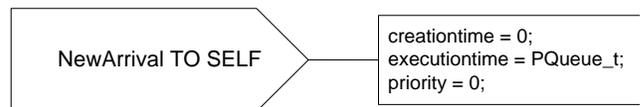


Figure 3: Defining the executionTime, and other parameters, of the signal using the SDL time extensions.

Regarding the implementation for the cellular automaton remark that all the cells have the same behavior. This implies that is not needed to represent all the cells, but only one cell. Also is needed to represent the relation that have with the neighborhood (that of course is specific of the cellular automata), and the relation with the other layers. That means define the **combination**, **vicinity** and **nucleus** functions (Fonseca i Casas, et al., 2005).

In SDL language we can use **types** to define blocs that have the same behavior (as is usual in any OO language). In order to represent the communication between the different cells, are needed to represent all the communication

channels between the cells. This implies the need of represent at least a set of cells that belongs to the vicinity (and the nucleus) of a cell. Also if we want a complete description of the cellular automaton (if we want an automatic generation of code), we need to represent all the cells of the cellular automaton.

To simplify the representation (avoiding the representation of all the cellular automaton cells) we decide to add a new kind of agent to SDL language, the **mnca**. This agent have 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). Also the OUTPUT SDL elements can carry a parameter to define what is the cell that receives the signal (Fonseca i Casas, et al., 2010). 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.

3. FIBONACCI FUNCION MODEL

As a first example we represent cellular automata that calculate the well known Fibonacci function. First is needed to define the different elements that compose our simulation model.

This first level of the SDL diagrams, in this case, only contains a single block, representing the cellular automata that implements the Fibonacci function. Always this first level of the model helps to understand the different elements that must be combined inside the model and the interaction of the model with the environment (users, computers, etc).

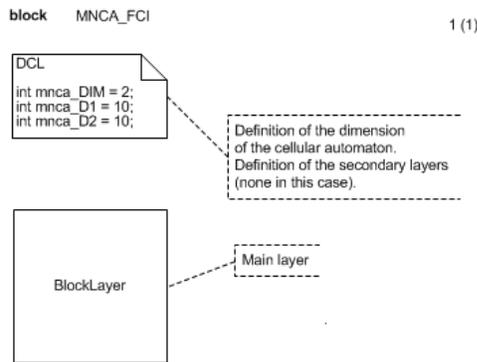


Figure 4: m:n-CAk cell representation. Here we can see the relation between the different layers.

Next, we must define the structure of this m:n-CA^k cellular automata. First the number of cells (the dimensions), using the DCL (declaration block). On this block, the mnca_DIM

variable defines the number of dimensions of the cellular automata, and mnca_D1 to mnca_Dn defines the size of each one of these dimensions. In that case we have a matrix (10x10) as is represented in Figure 4.

Evolution function is defined in the *ProcessLayer*, to see its representation (Fonseca i Casas, et al., 2010) can be consulted.

4. WILDFIRE MODEL

As a second example we show the first diagrams of a wildfire model. In that case we are following behave model to represent the fire spread (Andrews, et al., 1989). Is interesting to remark that since in this model we need different layers we must define them on the DCL block on the block mnca (Figure 5).

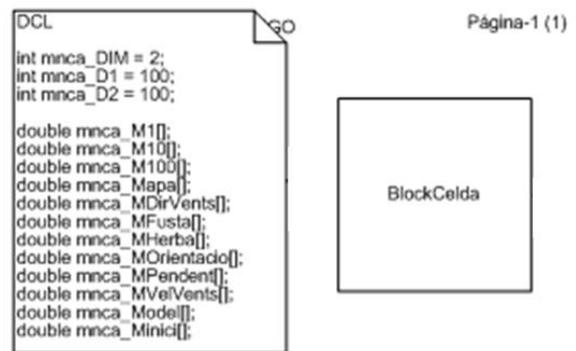


Figure 5: m:n-CAk cell representation for wildfire model.

Inside this we can find the definition of the behavior of the model. For space reasons we only show the states diagram that define the transition on each one of the cellular automaton cells (Figure 6). Also, on Figure 7 we show the aspect of the SEND signal that represents the propagation of the fire to other cells of the cellular automaton.

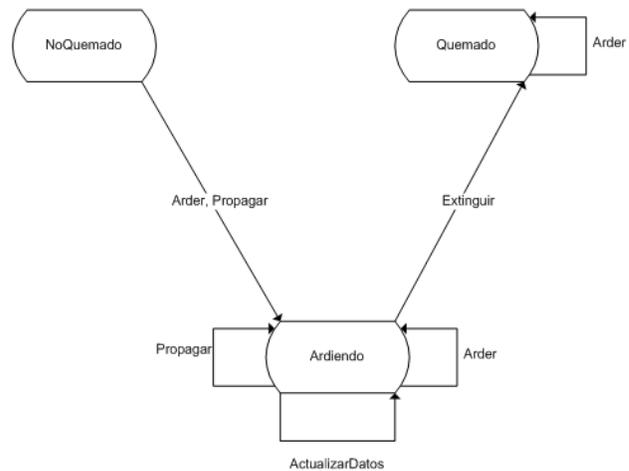


Figure 6: states diagram for a cell.

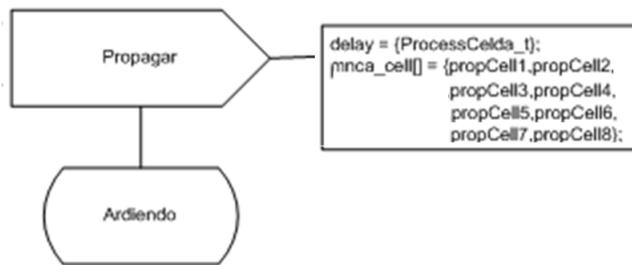


Figure 7: propagation of the wildfire.

5. CONCLUSIONS AND FUTURE WORK

This paper shows how we can model environmental systems using Specification and Description Language. To do this the main concern is how model the behavior of cellular automata graphically using SDL, and how to manage time.

Two examples are quoted, a representation of a Fibonacci function over a cellular automaton, and the fire spread following the Behave model.

Two proposed extensions to SDL are shown, one to manage time on the SDL signals and other to simplify the representation of the cellular automaton structures.

The future work is focused in the verification of the implemented structures on SDLPS and the use of this system on some ongoing real projects involving industrial models or environmental models.

6. REFERENCES

- Andrews P.L. and Chase C.H.** BEHAVE: Fire behavior prediction and fuel modeling system-BURN subsystem, part 2 [Report] : Gen. Tech. Rep. INT-260.. - Ogden, UT : U.S. Department of Agriculture, Forest Service, Intermountain Research Station, 1989. - p. 93.
- Benenson Itzhak and Torrens Paul M.** Geosimulation, Automata-based Modeling of Urban Phenomena [Book]. - West Sussex PO19 8SQ : John Wiley & Sons Ltd., 2004. - ISBN: 0-470-84349-7.
- Brade D.** Enhancing modeling and simulation accreditation by structuring verification and validation results [Conference] // Winter Simulation Conference / ed. Joines J. A. [et al.]. - 2000.
- CINDERELLA SOFTWARE** Cinderella SDL [Online]. - 2007. - 03 31, 2009. - <http://www.cinderella.dk>.
- De Lara J. and Vangheluwe H.** ATOM³: A Tool for Multi-formalism Modelling and Meta-modelling [Conference] // 4th International Conference on Enterprise Information Systems ICEIS 2002. - 2002.
- Doldi Lauren** Validation of Communications Systems with SDL: The Art of SDL Simulation and Reachability Analysis [Book]. - [s.l.] : John Wiley & Sons, Inc., 2003. - p. 310. - ISBN: 978-0-470-85286-6.
- Emmeche Claus** Vida Simulada en el ordenador [Book]. - Barcelona : Gedisa, 1998. - ISBN: 84-7432-616-8.
- Fonseca i Casas Pau and Casanovas Josep** Simplifying GIS data use inside discrete event simulation model through m_n-AC cellular automaton [Conference] // Proceedings ESS 2005. - 2005.
- Fonseca i Casas Pau** SDL distributed simulator [Conference] // Winter Simulation Conference 2008. - Miami : INFORMS, 2008. - <http://wintersim.org/abstracts08/POS.htm#fonsecaicasasp84590>.
- Fonseca i Casas Pau, Colls Màxim and Casanovas Josep** Representing Fibonacci function through cellular automata using Specification and Description Language [Conference] // Proceedings of the 2010 Summer Simulation Multiconference.. - Ottawa, ON, Canada : [s.n.], 2010.
- IBM TELELOGIC** [Online]. - 2009. - 03 31, 2009. - <http://www.telelogic.com/>.
- IEC** SDL Tutorial [Online] // IEC. - May 2010. - <http://www.iec.org/online/tutorials/sdl/topic04.html>.
- Praehofer H. and Pree D.** Visual modeling of DEVS-based multiformalism systems based on higraphs [Conference] // Winter Simulation Conference / ed. Evans G. W. [et al.]. - Los Angeles, California, United States : [s.n.], 1993. - pp. 595-603. - DOI= <http://doi.acm.org/10.1145/256563.256737>.
- Telecommunication standardization sector of ITU** Specification and Description Language (SDL) [Online] = ITU-T Z.100. 1999 // Series Z: Languages and general software aspects for telecommunication systems.. - International Telecommunication Union, 1999. - April 2008. - <http://www.itu.int/ITU-T/studygroups/com17/languages/index.html>.
- Wainer G. and Chen W.** A framework for remote execution and visualization of Cell-DEVS models [Journal] // Simulation: Transactions of the Society for Modeling and Simulation International. - November 2003. - pp. 626-647.