Modeling of the Reactive Navigation of Autonomous Robot using the Discrete Event System Specification DEVS

Kadda Mostefaoui
Department Of Computer Science
University Of Tiaret
ALGERIA

Youcef Dahmani
Department Of Computer Science
University Of Tiaret
ALGERIA

ABSTRACT
This paper presents the modeling and simulation of reactive navigation of a mobile robot using the characteristics of discrete event systems (DEVS), which is a young approach in modeling and simulation. The mobility and the autonomy of robots pose complex problems, as regards generation of trajectory in strongly constrained and unstructured spaces, and decision-making based on inaccurate or incomplete information sensors. In this work we propose an architecture based on the DEVS (Discrete Event System Specification) and the theory of fuzzy logic for modeling and simulation of an intelligent agent observing, deciding, acting on a dynamic and uncertain environment in context of the DEVS formalism. This problem is at the border of two areas of research: artificial intelligence (Mobile Robotics) and modeling and simulation.

Keywords
Modeling & Simulation, DEVS, Fuzzy Logic, Soft Computing, Reactive navigation, Mobile Robots, Fuzzy Logic Controller.

1. INTRODUCTION
Modeling and simulation are essential tools to analyze the behavior of dynamic systems. Several methods have been proposed to improve the process of analyzing the behavior of these systems. These proposals attempt to achieve more realistic models, relatively simple and highly flexible. Our focus is on DEVS (Discrete Event System Specification) that has been widely used to study the dynamics of discrete event systems.

Since the 1970s, formal work has been done to develop the theoretical foundations of modeling and simulation of discrete event dynamic systems [1].

The DEVS formalism, Discrete Event System Specification, was introduced by Professor BP Zeigler [2] as an abstract formalism for modeling discrete events.

The DEVS formalism is a modeling approach based on general systems theory. Specifically, it is a modular and hierarchical formalism for modeling, centered on the notion of state [2].

The DEVS formalism can be viewed as a multi-modeling environment gathering into a coherent way other modeling formalisms based them also on general systems theory and centered on the states Its capacity of opening in the computer sense, in fact a formalism adapted for many application areas [3] [4] [5,6].

The main goal of this paper is to propose a model using the DEVS formalism integrated with the theory of fuzzy logic to describe and simulate a complex system of reactive navigation of mobile robot.

This paper is organized as follows : In the section 2 we introduce the basic concepts of DEVS and the theory of fuzzy logic. In the section 3 we propose a model of the system reactive navigation of mobile robot. Finally we conclude our work in the section 4.

2. BACKGROUND
In this section we introduce the formalism DEVS and the theory of fuzzy logic.

2.1 The Devs Formalism
BP Zeigler defined in [2], a formal specification of discrete event models. This formalism was introduced as an abstract universal formalism independent of the implementation. The DEVS formalism is based on the definition of two types of models: atomic models and coupled models. The atomic models used to represent the basic behavior of the system. Coupled models are defined by a set of sub atomic models and / or coupled models to represent the internal structure of the system through coupling between models.

2.1.1 Formal specification of a DEVS atomic model
The atomic model provides an autonomous description of system behavior, defined by states and input / output functions and internal transitions of the model. The evolution of the model is done by state change according to external stimuli (via an input) or internal (via a transition function). These state changes are intended to determine the behavioral response of the system to these stimuli.

Formally, an atomic model M is specified by a 7-uplet:

$$AM = (X, Y, S, \delta_{ext}, \delta_{int}, \lambda\ , ta)$$

X : is the set of external events (input).

Y : is the set of output events.

S : is the set of states.

δ_{int}: $S \rightarrow S$: is the internal transition function caused by the occurrence of internal events.

δ_{ext}: $Q \times S \rightarrow S$: is the external transition function caused by the occurrence of external events, $Q = \{(s,e) \mid s \in S.0 \leq e \leq ta(s)\}$: total states and ***e*** describes the elapsed time since the system made a transition to the current state s.

λ : is the output function..

ta: is the function of lifetime of state, represents the maximum time during which the model remains in a state $s \in S$.

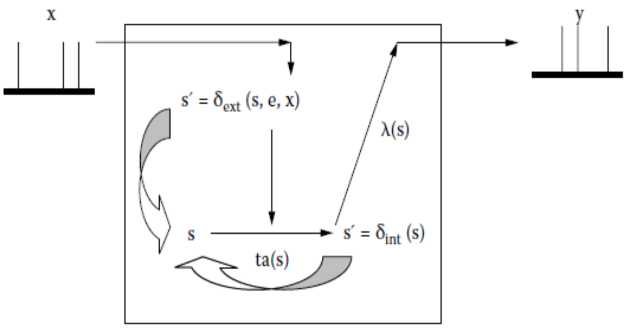

Fig 1:Description of a DEVS atomic model

2.1.2 Formal specification of a DEVS coupled model

To describe a more complex system we interconnect several atomic models to form a coupled model. This new model can be used as a base model in a higher level description is the hierarchical aspect of the formalism.

A coupled discrete event is defined by the following structure:
$CM = (X, Y, D, \{M_d / d \in D\}, EIC, EOC, IC)$
X: set of possible inputs of the coupled model.
Y: set of possible outputs of the coupled model.
D: set of names associated to the model components.
M_d: set of the coupled model components, these components are either atomic or coupled DEVS model,
EIC: set of External Input Coupling.
EOC: set of External Output Coupling.
IC: set of internal couplings.

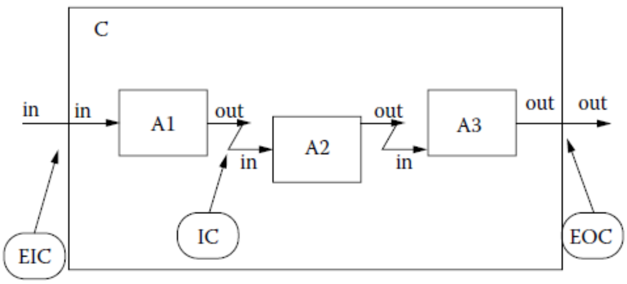

Fig 2: Description of a coupled DEVS

2.2 FUZZY LOGIC

Fuzzy systems are a technique that allows the definition of input-output mappings based on linguistic terms of every day common use. This approach has been particularly successful in the area of control problems.

The fuzzy set theory introduced by Zadeh [7] offers a set of tools to formalize the process of human reasoning. This is an effective way to take into account the imprecision in knowledge. .

Fuzzy Logic allows the formalization of imperfections due to inaccurate knowledge of a system and description of system behavior with words. It therefore allows the description of a system and data processing as well as digital language.

Fuzzy modeling is a difficult task, requiring the identification of many parameters. According to L. Zadeh it provides approximate but effective means to describe the behavior of systems that are too complex or poorly defined to allow the use of a precise mathematical analysis [1].

Fuzzy sets were introduced by Zadeh [7] as a generalization of the concept of a regular set. In a regular set, or crisp set, the membership function assigns the value 1 to the elements that belong to the set and 0 to the elements that do not belong to it. In fuzzy sets, elements can have different degrees of membership to the set.

Formally, a fuzzy set A is characterized by a membership function μ_A that assigns a degree (or grade) of membership to all the elements in the universe of discourse $U \rightarrow [0,1]$:

$$\mu(x) : U \rightarrow [0,1]$$

The membership value is a real number in the range [0,1], where 0 denotes definite no membership, 1 denotes definite membership, and intermediate values denote partial membership to the set. In this way, the transition from non membership to membership in a fuzzy set is gradual and not abrupt.

2.2.1 A fuzzy inference system FIS

One of the most successful areas of application of fuzzy systems is control, where fuzzy controllers have proved to be very effective in the context of controlling complex ill defined processes[12].

A fuzzy controller (FC) is a standard fuzzy system defined to control a target process, usually called a plant in control terminology. The fuzzy controller is usually designed by formulating the knowledge of a human expert into a set of linguistic variables and fuzzy rules [8].

Fuzzy inference systems (FIS) are composed of a collection of fuzzy rules which have the general form: "If p then q" [1].

The realization of an inference system consists of three operations: (1) fuzzification, (2) inference engine, and (3) defuzzification The operation of transforming a set of real into fuzzy set is called "fuzzification", and the reverse operation is called "defuzzification".

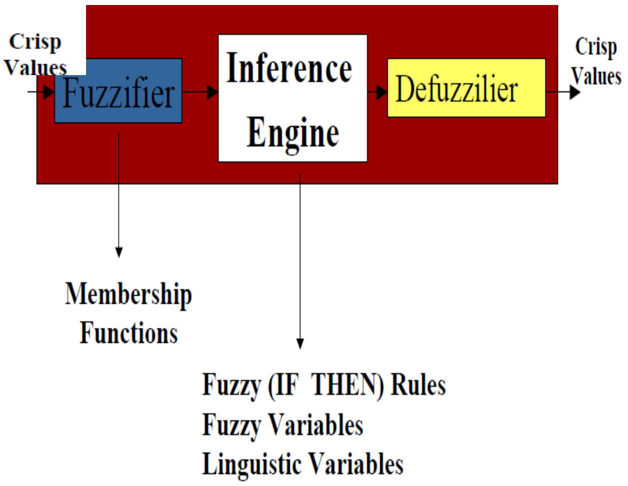

Fig3: Fuzzy inference system

3. A PROPOSED DEVS MODEL FOR REACTIVE NAVIGATION OF A MOBILE ROBOT

The problem addressed in this paper is the autonomous navigation in an uncertain dynamic environment. Our aim is to develop an approach using DEVS formalism integrated with fuzzy logic to model and simulate a robot moving autonomously in an environment which is not known a priori and in which obstacles are present. The task of the autonomous robot is to find each time a given position or goal avoiding collisions with the obstacles.

Modeling and simulation with DEVS has become a reference to the question of coupling of heterogeneous models. Today's modeling of decisionnel process is crucial with DEVS to study complex systems.

Autonomous robotic systems are deeply random because of their interaction with an unpredictable environment. Modeling of these systems under DEVS become difficult due to their random nature.

The classic DEVS formalism allows the modeling and analysis of complex systems with discrete event with a known structure and behavior.

It is possible to use this formalism for the specification of a system of the reactive navigation of mobile Robot. The mobile robot can be seen as an autonomous entity that perceives its environment and acts on it. Behavior and interactions of a mobile robot can be very complex. The principles of modularity and hierarchical decomposition outlined in our presentation of DEVS allows us to adopt a level of detail sufficient to express this complexity.

The behavior of the mobile robot can be summarized in a loop : Perception → Select_Action→ Execution (action). The choice of action is the result of reasoning.

3.1 Architecture of the mobile robot system in DEVS

The proposed model divides the system of navigation of a mobile robot into four subsystems: Localization, Perception, Controller, Actuator.

The subsystem Localization: permit to estimate the robot position and its own state with the environment or geographical objectives by proprioceptive sensors.

The subsystem perception : to elaborate the information coming from the sensors in order to acquire knowledge and understanding of the evolution of environment through exteroceptive sensors.

The subsystem controller: the robot must find a way to reach the goal from the current position satisfying some constraints as avoid collision with obstacles, the decision process must be based on the information gathered by subsystem perception.

The subsystem actuator: allows to execute the action selected by the controller subsystem, The robot puts in practice the decision taken with its actuators, which allow the robot to move in the environment.

Figure 4 shows the diagram of the architecture of the mobile robot system in DEVS.

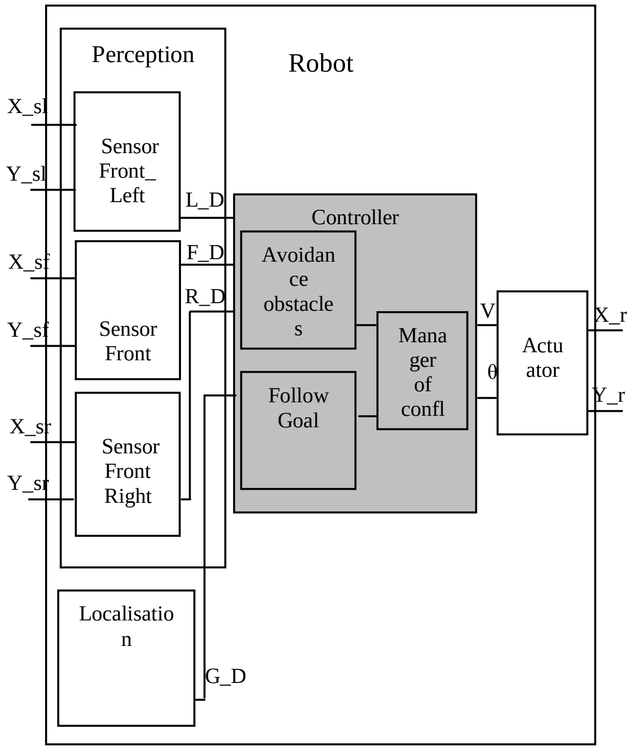

Fig 4: Mobile robot system in DEVS

3.2 The controller Fuzzy DEVS

Autonomous navigation in dynamic environment is an important challenge for robotics research. The detection of moving obstacles, predicting the future state of the robot presents major challenges in the reactive navigation of a mobile robot in an unknown dynamic environment. The classic DEVS formalism does not take into account inaccuracy and uncertainty on the events and the states. The multi-formalism DEVS enables the integration of many other formalisms or modeling methods.

In this work we propose a classical or traditional fuzzy controller combined with the DEVS formalism, Once the entries are known (sensor inputs), outputs are determined uniquely. Controllers of this type using the classical approach known as Mamdani in which each rule is represented by a conjunction (minimum) and aggregation of the rules by a disjunction (maximum). Figure 5 shows the principle design of our controller in DEVS.

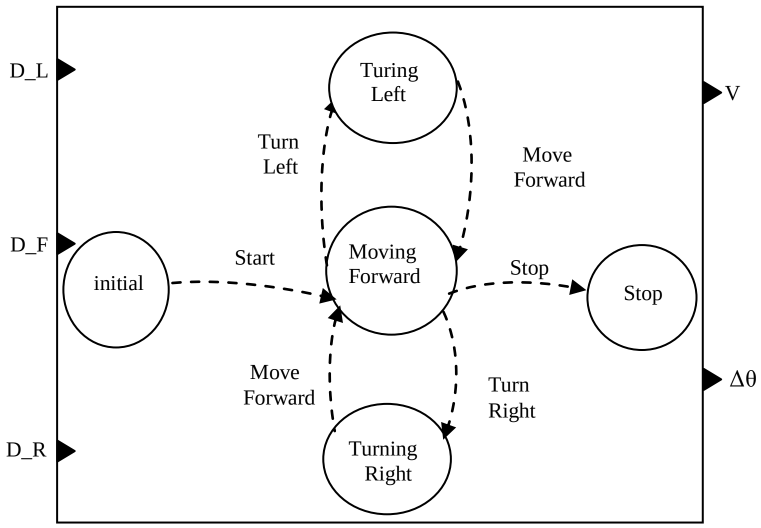

Fig 5: A fuzzy controller in DEVS

3.2.1 Formal specification of the controller

The formal specification of the fuzzy controller in DEVS is as follows:

$$\text{Controller} = (X, Y, S, \delta_{ext}, \delta_{int}, \lambda\ , ta) \quad \text{where :}$$

X: D (Dl, Df, Dr): distances given by the subsystem perception.

S: "Initial", "Turning left", "Turning right", "Moving Forward",
"Stop".

Y: V, Δθ which respectively correspond to the speed of movement and the angle of deviation of the robot.

δ_{ext} :FIS (Distance) fuzzy inference system, it receives input (D) from the subsystem perception.

δ_{int}: The internal transition function.

λ : Send results of FIS to the output ports.

ta : is the function of lifetime of state.

3.2.2 The classical fuzzy controller

The basic navigation strategy is a reactive approach, combining two basic behaviors:

Obstacle avoidance and Following the target. The fuzzy controller receives as inputs the distances measured by the left sensor "d_l", the front sensor "d_f" and right sensor "d_r" and "r_g" the angle corresponding to the existing angle between the orientation of the target and the orientation of the robot and returns as output the two commands Δθ and ΔV, which correspond respectively to the change of direction of the robot and the variation of its speed .

3.2.2.1 The fuzzification

The fuzzification is a process which associate the elements of the input vector to a fuzzy set according to a degree of membership.

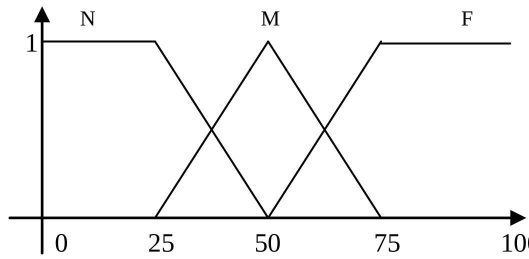

Fig 6: Membership function of the distance (Dl, Df, Dr)

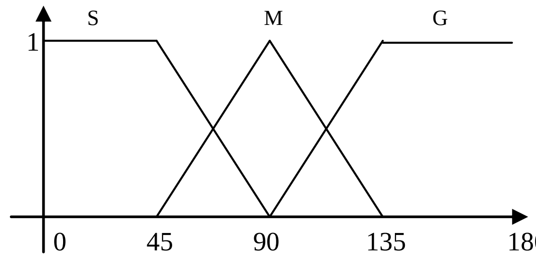

Fig 7: Membership function of the output Δθ

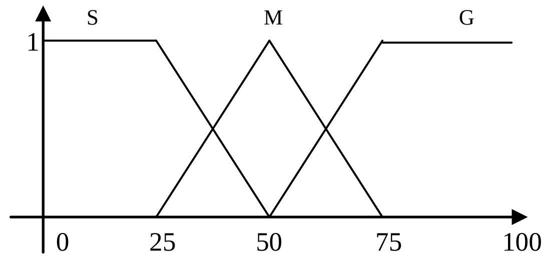

Fig 8: Membership function of the speed V

3.2.2.2 The inference

A fuzzy controller usually takes the form of a set of rules "if-then". The inference is based on min and max operations to perform the inference rules and the max operator for the aggregation rules.

The rules used in our system are universal rules in the navigation of mobile robots. The following table shows the rules used:

Table 1. The used rules in Fuzzy controller

Rule	1	2	3
Distance	F	M	N
Angle	S	M	G
Speed	G	M	S

3.2.2.3 The defuzzification

Transforms the command actions into crisp values useable directly by the process which is modeled. The most method used is the center of gravity. We calculate the center of gravity formed by the union of degrees of validity.

3.3 Simulations and results

3.3.1 The modeling and simulation Environment JDEVS

Simulation tools allow users to observe the structure and behavior of a real system. Currently there are several modeling and simulation environments based on the DEVS formalism allowing the representation of different atomic or coupled DEVS models. To specify, design and develop our system, we chose the environment JDEVS as a tool for modeling and simulation for our proposal model.

The need of a formal framework in environmental modelling extends beyond the choice of the DEVS formalism or the JDEVS implementation. Because it is based on a DEVS based formal framework, JDEVS provides a different approach than the existing tools. In terms of flexibility and genericity of use, it can provide the high-level approach of a general formalism. In terms of features, abstraction, components and interfaces, JDEVS provides the advantages of a domain specific modelling environment. With JDEVS, it is also possible to specify, store, retrieve, couple and simulate different kinds of models without having to specify how those models should be simulated[9].

3.3.1.1 Representation of the proposed model using the modeling tool JDEVS

Figure 9 shows the proposed model of reactive navigation of a mobile robot modeled on the environment JDEVS. This model consists of several atomic models.

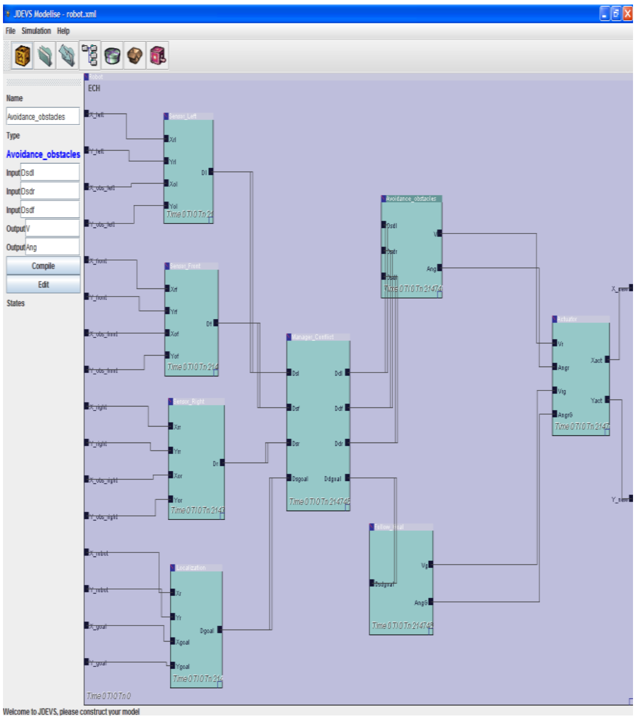

Fig 9: Representation of the model under JDEVS

To validate our approach coupling the DEVS formalism and fuzzy logic on the system of reactive navigation of a mobile robot in an uncertain environment, we conducted two simulation tests to verify the proposed approach, the simulation tool in JDEVS modeling and simulation system is analogous to real autonomous navigation of mobile robots.

Test 1

In the first test the robot starts in a free environment, there is no obstacle in its path which allows it to go straight and to move towards the target.

Fig 10: The trajectory in the first case

Test 2

In the second test the robot encounters an obstacle in its path. As shown in Figure 11 the robot makes a deviation on the left to avoid the obstacle then it continues to move straight and to move towards the target.

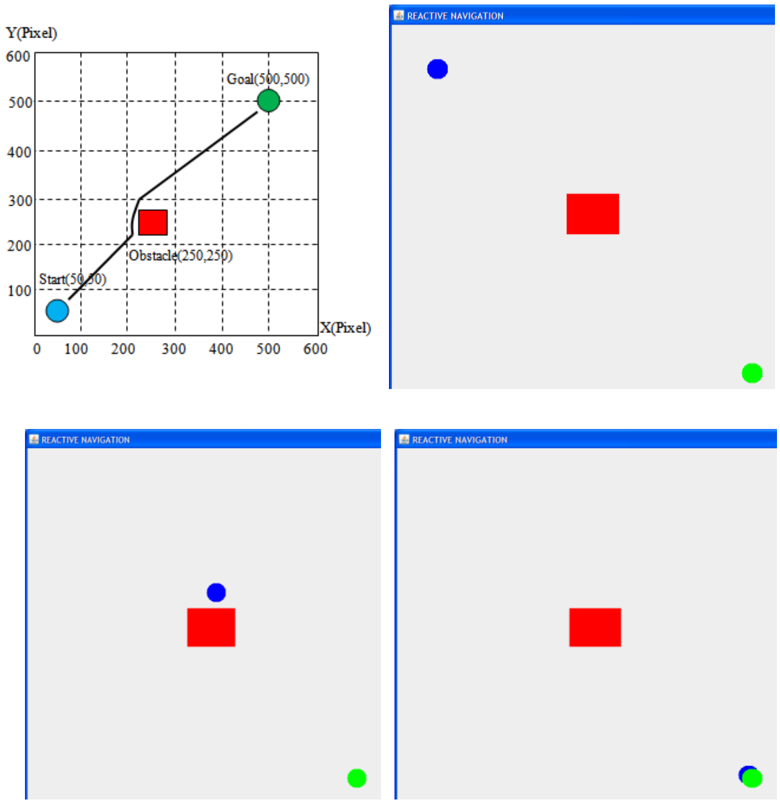

Fig 11: The trajectory in the second case

4. CONCLUSION AND FUTURE WORK

In this paper we presented the modeling and simulation of reactive navigation of a mobile robot in a dynamic and uncertain environment. Our contribution lies in the possibility of combining the DEVS formalism and the theory of fuzzy logic. We proposed a model that describes the studied system.

Based on this model, we developed an application using the tool for modeling and simulation JDEVS to simulate the behavior of autonomous mobile robot in an unknown environment.

The proposed approach works well in simple environments also in complicated environments.

The continuation of our work will concern several points. First, we consider the application of this proposal on more complex systems. Moreover, we wish to combine the DEVS formalism with other Soft computing tools (neural networks, genetic algorithms etc ...).

5.REFERENCES

[1] Bisgambiglia P. A., " Approximate modeling approach for discrete event systems: Application to the study of propagation of forest fires ", PhD Thesis, University of Corsica - Pasquale Paoli, France, 2008

[2] Bernard P. Zeigler. Theory of Modeling and Simulation. Academic Press, 1976.

[3] F. Barros. Dynamic structure discrete event system specification : a new formalism for dynamic structure modelling and simulation. In Proceedings of Winter Simulation Conference 1995, 1995.

[4] A. Uhrmarcher. Dynamic Structures in Modeling and Simulation : A Reflective Approach. ACM Transactions on Modeling and Computer Simulation vol. 11 2001, pages 206–232, 2001.

[5] Lewis Ntaimo and Bernard P. Zeigler. Expressing a forest cell model in parallel DEVS and timed cell-DEVS formalisms. Proceedings of the 2004 Summer Computer Simulation Conference, 2002.

[6] Alejandro Troccoli and Gabriel Wainer. Implementing parallel cell-DEVS. In IEEE, editor, Proceedings of the 36th Annual Simulation Symposium, 2003.

[7] L.A. Zadeh. Fuzzy sets. Information Control, 8 :338–353, 1965.

[8] Chuen Lee. Fuzzy logic in control systems: Fuzzy logic controller - part I. IEEE Transactions on Systems, Man and Cybernetics, 20(2):404–418, March/April 1990.

[9] Jean-Baptiste Filippi, Paul Bisgambiglia: JDEVS: an implementation of a DEVS based formal framework for environmental modelling, 14 August 2003

.