# Specification of the State's Lifetime in the DEVS Formalism by Fuzzy Controller

1st Author
1st author's affiliation
1st line of address
2nd line of address
Telephone number, incl. country code
1st author's E-mail address

2nd Author
2nd author's affiliation
1st line of address
2nd line of address
Telephone number, incl. country code
2nd E-mail

## ABSTRACT
This work aims to improve simulation performance by introducing fuzzy logic in the discrete event simulation. This paper attempts to develop a new approach to assess the duration of dynamical systems, especially the natural processes. To do, we have used the fuzzy logic theory and particularly the fuzzy inference system combined to the discrete event simulation (DEVS). The process simulation is adjusted by the fuzzy controller which interacts with the input variables of the system. In fact, the fuzzy rules reflect the inputs influence on the duration of processes. The idea is to define a set of fuzzy rules obtained from observers or expert knowledge and to specify a fuzzy model which computes the duration of the process. This duration is then directed towards the simulated model in order to get a new trajectory. In the conventional model, each state is defined by a mean lifetime value whereas our method, calculates for each state the new lifetime according to input variables values. A case study is presented at the end of the paper which is a wildfire spread. It is a challenging task due to its complex behavior, dynamical weather condition, and various variables involved. A global specification of the fuzzy controller and the forest fire model are presented in the DEVS formalism and comparison between conventional and fuzzy method is illustrated.

## Categories and Subject Descriptors
D.3.2 [**Java**]: Language Constructs and Features – *abstract data types, classes, polymorphism, control structures.*

## General Terms
General terms of this work are: Performance, Design, Reliability and Experimentation.

## Keywords
Simulation, Modeling, Fuzzy Controller, DEVS Formalism, State Lifetime, Forest Fire.



## 1. INTRODUCTION
The modeling and simulation formalisms are used in order to understand, to represent, and specify the dynamic of complex systems [14]. However, these systems are very complex to be described formally due to their partial knowledge, and demand a great effort to represent the whole system with its overall variables.

Understand the dynamics systems, particularly the natural systems is a hard job. Different methods and techniques were created in order to improve their formulation. We distinguish two main categories: Analytic methods, which are difficult to grasp, and modeling and simulation methods [5]. Formally, a large variety of dynamic behaviors can be formulated mathematically. To learn about such systems, we must take into account all involved variables. Although their dynamics reveals a complex formulation involved by these variables, the corresponding equations are unable to provide accurate results due to a lack of information for such systems and the complexity of their combination. To overcome this issue, modeling and simulation methods were proposed.

The modeling and simulation is based on an experimental frame, the likeness between experimentations and the modeling and simulation was the essence of this twinning [1,4], offering the possibility of predicting the behavior of complex systems. Various approaches were defined to treat the two phases of modeling and simulation, depending on either time-driven or event-driven systems [19].

Model and simulate discrete events deal with systems whose temporal and spatial behaviors are complex to be treated analytically. The DEVS formalism (Discrete EVent system Specification) is one of the common formalism used in the simulation of dynamical systems [23]. It is known for its modularity, expressivity. This formalism offers, compared to the other ones, a general purpose [20].

However, the DEVS formalism is based on constant piecewise input-output trajectories to simulate continuous dynamic systems [24,25]. In order to overcome this issue, many variants on DEVS were adopted by introducing appropriate theories such as the cellular automata [19], the fuzzy logic etc.

The incomplete knowledge of certain systems involves vagueness and incompleteness. This point was studied by fuzzy logic [13,18,21]. The main difference with the conventional analytic methods is, firstly, it doesn't require a rigorous mathematical model to control a system. In the most cases, it uses knowledge of human operators to develop the controller, synthesizing the human operator actions. Secondly, its characteristic is the simplicity integration of subjective data in the controller. Its utilization is recommended when the drive system is uncertain and imprecise, or when its variables are inaccurate or expressed in natural language.

This work aims to assess the states lifetime of a natural system introduced by a fuzzy controller. A case of forest fire propagation is studied to apply this approach. This assessment is affected by the involved variables of the process.

We focus on the influence of the inputs variables to approximate the duration of these process. Our example is based on this remark: "the duration of a wildfire spread at dry and windy time is necessarily shorter than that of a rainy and calm weather". Starting from this remark, we have tried to translate this observation by a fuzzy controller and simulated it via DEVS formalism. Thus, the example that we took to validate our approach is based on forest fire spread. The rest of this work is organized as follows: After this introduction, a background resumes some basic theories on fuzzy logic, and the DEVS formalism. The third part is devoted to the specification of the fuzzy controller in DEVS formalism. The fourth section illustrates our example of forest fire growth; we present its different variables and its formal description in DEVS. The fifth part presents results and at the end, in the sixth section, a conclusion, with future works, is given.

# 2. THE FUNDAMENTAL CONCEPTS

## 2.1 The Fuzzy Logic

### *2.1.1 Linguistic Variable*

The linguistic variable is a variable whose values are words or sentences in a natural or artificial language. It is characterized by quintuple (L,T(L),U,G,M). where L is the name of the variable, T(L) is the set of fuzzy sets (linguistic values), U is the universe of discourse, G is the syntactic rule and M its semantic [12,23]. The Figure 1 illustrates an example of the linguistic variable "velocity" with three terms: slow, middle and fast.

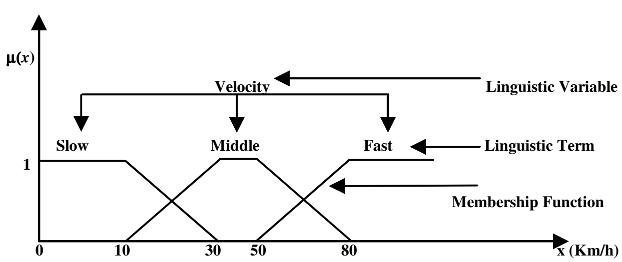


**Figure 1. The membership functions of linguistic variable "Velocity".**

### *2.1.2 Fuzzy Rules*

The fuzzy rules [15] are expressions of this general form:

$$\mathbf{R_i} : \textbf{If } x_1 \text{ is } \mathbf{X_1^i} \text{ and ......... and } x_n \text{ is } \mathbf{X_n^i} \ \ \textbf{Then } y \text{ is } \mathbf{Y} \qquad (1)$$

Where $\mathbf{X_j^i}$ is a label of fuzzy set of the input $j$ ($j \in \{1..n\}$) and linguistic variable i ($i \in \{1..N\}$). Each linguistic term is characterized by its own membership function. Many forms can be used, trapezoidal, triangular, Gaussian (Figure 2).

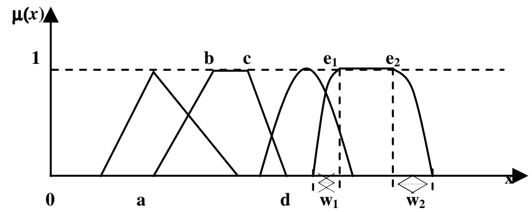


**Figure 2. Common shapes of the membership functions.**

This kind of rules is called Mamdani's rules in which the consequent part is given as symbolic and fuzzy member. Another form of the fuzzy rules is given by Takagi and Sugeno [18]. In this type of rules, the conclusion part is described as a function of the input variables of the premise part.

### *2.1.3 Fuzzy Inference System*

A fuzzy inference system (FIS) also known as fuzzy controller aims to build a control law from linguistic and qualitative description of system's behavior via fuzzy base rules [22].

A Fuzzy controller is described by five main elements (Figure 3):

- Rule Base: Expresses the knowledge processes introduced by intuition and experimentation with Human operators.
- Data Base: Represents the properties of fuzzy sets.
- Fuzzification: Numerical values are transformed into linguistic variables with appropriate membership functions.
- Defuzzification: Transforms the command actions into crisp values useable directly by the controlled process.
- Inference Engine: Makes decisions through the activated fuzzy rules. It is the core of the controller.

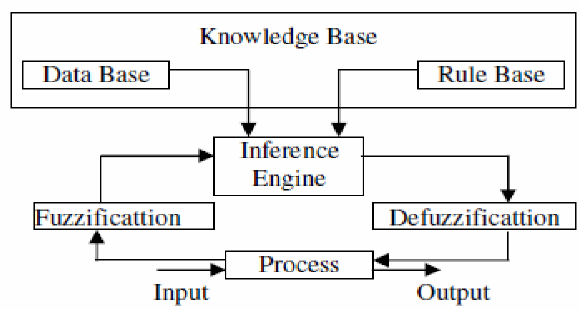


**Figure 3. Fuzzy Inference System.**

### *2.1.4 Inference*

The inference is often reduced to the deduction in which the truth of the premises guarantees the completely truth of the conclusion. It is the decision-making mechanism; it gives the final conclusion for all activated rules according to the input data [15].

For an input vector $x=(x_1,\ldots,x_n)^t$, the fuzzy reasoning consists of 5 steps (Figure 4):

1. Obtain the membership degrees which match the appropriate membership function of each input.

$$\mu_{A_j^i}(x_j) \qquad (2)$$

2. Calculate the truth value of each rule.

$$\alpha_i(x) = \min_j \left( \mu_{A_j^i}(x_j) \right) \quad (3)$$

3. Generate the contribution of each rule.

$$\mu(y) = \min \left( \alpha_i(x) \, , \, \mu_{B^i}(y) \right) \quad (4)$$

4. Aggregate the qualified rules.

$$\mu(y) = \max_i \left( \mu_{B^i}(y) \right) \quad (5)$$

5. Produce the numerical value of the fuzzy output.

$$y = \frac{\int u \mu(u) du}{\int \mu(u) du} \quad (6)$$

Where min stands for minimum function, max for maximum, n is the number of inputs whereas N is the number of fuzzy subsets.

This implementation is called "min, max, center of gravity". It is the Mamdani inference method [13].

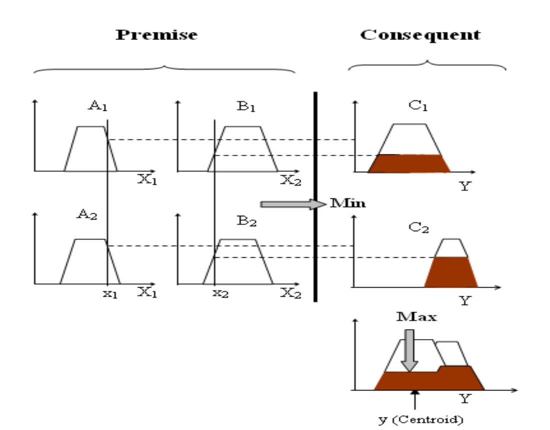


**Figure 4. Fuzzy Inference.**

We recall that in the literature, many kinds of fuzzy reasoning exist. They depend on the sort of the fuzzy rules employed and also on the nature of calculation of the crisp output [11,15].

## 2.2 DEVS Formalism

### *2.2.1 Introduction*

The DEVS formalism "Discrete EVent system Specification", was developed by Professor B.P. Zeigler [23]. It is based on mathematical theory of dynamic systems [24]. It is known for coupling heterogeneous models and separates the modeling process from the simulation one [25]. In fact this formalism is well adapted to represent a continuous system and describes the paradigm "event" in its overall features [12]. This formalism was applied in a great number of applications. It offers a general framework, and known as multi-formalism model [20].

Each system is characterized by two features: functional (behavioral) and structural aspects [19]. Similarly, the DEVS formalism authorizes two levels of description. At the lowest level, a basic part called atomic DEVS describes the behavior of a discrete-event system. At the highest level; a coupled DEVS describes a system as modular and hierarchical structure [23,25].

### *2.2.2 The DEVS Atomic Model*

The atomic models are the fundamental elements of the formalism; they describe the functional aspect of the system (Figure 5). They operate as "state-machines" [8]. Formally, a DEVS atomic model is described by seven-tuple (Equation 7):

$$AM = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle \quad (7)$$

Where

X: the set of input events;

S: the set of partial states;

Y: the set of output events;

$\delta_{int}: S \rightarrow S$ : internal transition function, models the states changes caused when the elapsed time reaches to the lifetime of the state;

$\delta_{ext}: Q \times S \rightarrow S$ : external transition function, defines how an input event changes a state of the system;

$Q = \{(s,e) \mid s \in \mathbf{S}. 0 \leq e \leq t_a(s)\}$ : total states and $e$ describes the elapsed time since the last transition of the current state $s$;

$\lambda$: $S \rightarrow Y$: when the elapsed time reaches to the lifetime of the state, this function generates an output event;

$t_a$: $S \rightarrow \Re_0^+ \cup \infty$: time advance function, which is used to determine the lifespan of a state describing how long the system will stay in unchanged state if external events doesn't occur.

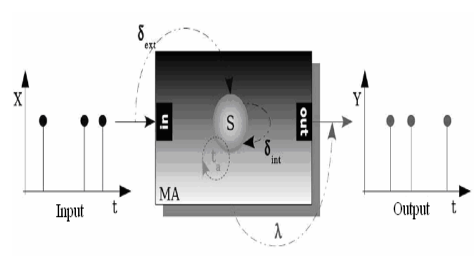


**Figure 5. DEVS Atomic Model.**

### *2.2.3 The DEVS Coupled Model*

The DEVS coupled model defines which sub-components belong to it and how they are connected to each other. It allows the creation of complex models starting from atomic and/or coupled models. Thus, it is modular and presents a hierarchical framework.

A DEVS coupled model is defined as an eight-tuple (Equation 8). A sample of coupled model is depicted on the Figure 6:

$$CM = \langle X_{self}, Y_{self}, D, \{M_d \mid d \in \mathbf{D}\}, EIC, EOC, IC, select \rangle \quad (8)$$

Where

$X_{self}$ : set of possible inputs of the coupled model;

$Y_{self}$ : set of possible outputs of the coupled model;

D : is the name set of sub-components;

$M_d \mid d \in \mathbf{D}$: is the set of sub-components, these components are either DEVS atomic or coupled model;

EIC: set of External Input Coupling;

EOC: set of External Output Coupling;

IC: defines the Internal Coupling;

Select: $2^D \rightarrow D$: tie-break selector which defines how to select the event from the set of simultaneous events.

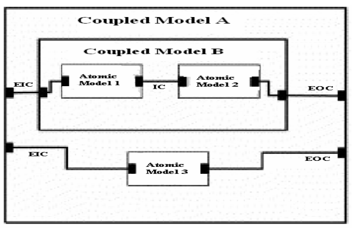


**Figure 6. A simple DEVS Coupled model.**

# 3. FUZZY CONTROLLER BY DEVS

## 3.1 Fuzzification DEVS Atomic Model

We consider the following assumptions: We have two variables $x_1$ and $x_2$ and one output y.

The linguistic terms of the variable $x_1$ are $A_1$ and $A_2$; $x_2$, are $B_1$ and $B_2$ and those of y are $C_1$ and $C_2$.

Therefore, the fuzzy rules are defined as follows:

**Rule** $_i$: If $x_1$ is $A_j$ and $x_2$ is $B_j$ Then y is $C_j$

with $j \in \{1,2\}$ and $i \in \{1,4\}$

As a base rules, we assume the table below (Table 1).

**Table 1. Sample of Fuzzy Base Rules**

| $x_1$ / $X_2$ | $A_1$ | $A_2$ |
|---|---|---|
| $B_1$ | $C_1$ | $C_2$ |
| $B_2$ | $C_2$ | $C_1$ |

In the present work, every fuzzy set is depicted as trapezoidal shape (Figure 7).

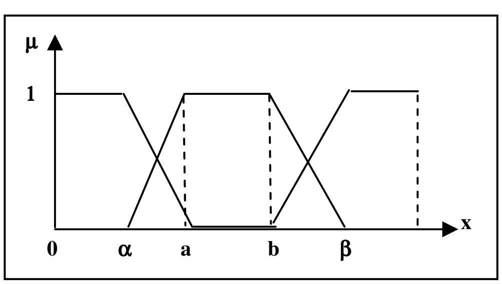


**Figure 7. Trapezoidal membership function.**

Every membership function of the fuzzy inference system is considered as an atomic model. Its DEVS specification is defined by (Equation 9) and depicted on the Figure 8:

$$FuzzificationAM=<X,S,Y,\delta_{int}, \delta_{ext},\lambda,t_a> \quad (9)$$

Where

InPorts = {'InNum'}, $X_{InPorts}=\Re$

OutPorts = {'OutNum'}, $Y_{OutPorts} = [0, 1]$,

$X = \{(in, x) / in \in InPorts, x \in X_{InPorts}\}$,

S = {'passive', 'active'}$\times \Re$,

$Y = \{(out, y) / out \in OutPorts, y \in Y_{OutPorts}\}$,

$\delta_{int}$ ('active',0) = ('passive', ∞),

$\delta_{ext}$ ( ('passive', ∞), e, ('InNum'?x)) = ('active', μ(x) ),

λ('active',m)=OutNum!m

$t_a$(phase,m) = 0 if phase=active

∞ if phase = passive

μ(x) is the membership function associated with this fuzzifier (Equation 2). The initial state of this model is (passive,∞).

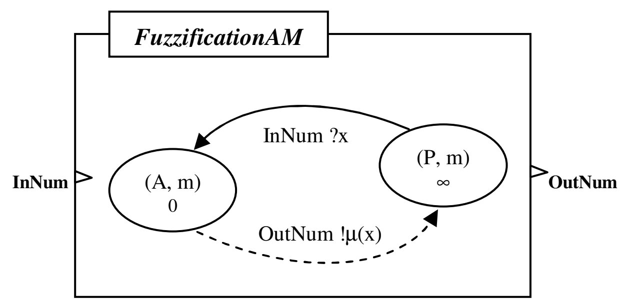


**Figure 8. Fuzzification DEVS Atomic Model.**

For each input value, FuzzificationAM performs a calculation. The result represents the degree of membership to the corresponding fuzzy set. FuzzificationAM is independent from fuzzy inference, but it depends naturally from the typical shapes of the membership functions (Figure 2).

## 3.2 Fuzzy Rule DEVS Atomic Model

According to the assumptions of section 3.1, each fuzzy rule has two inputs variables. Thus, the fuzzy rule is described as an atomic model (RuleAM) and its specification is illustrated as below (Equation 10):

$$RuleAM=<X,S,Y,\delta_{int}, \delta_{ext},\lambda,t_a> \quad (10)$$

Where

InPorts = {'InNum1', 'InNum2' }, $X_{InPorts}=[0, 1]$

OutPorts = {'OutFuz'}, $Y_{OutPorts} = [0, 1]\times\Re^4$,

$X = \{(in, x) / in \in InPorts, x \in X_{InPorts}\}$,

$S = \{$‘passive’, ‘active’$\} \times [0, 1] \times \Re^4$,

$Y = \{(out, y) / out \in OutPorts, y \in Y_{OutPorts}\}$,

$\delta_{int}$ ( ‘active’, (α,a,b,β),0 ) = (‘passive’, (α,a,b,β), ∞) ,

$\delta_{ext}$( (‘passive’, (α,a,b,β), ∞), e , ( (‘InNum1’ ? x1) & (‘InNum2’ ? x2) ) ) = (‘active’, x, (α’,a’,b’,β’) ) )

λ(‘active’,m, (α,a,b,β) ) = OutFuz! (m, (α,a,b,β) )

$t_a$(phase, m, (α,a,b,β) ) = 0 if phase = active

= ∞ if phase = passive

x=min(x1,x2) which is given by Equation 2., while (α',a’,b’,β’), is calculated by Equation 4.

The initial state of this model is (passive, ∞,(α,a,b,β)).

When RuleAM receives x1 and x2 from FuzzificationAM, it transitions to active state otherwise it remains in wait state. The transition to the active state is conditioned by the arrival of both inputs. The RuleAM depends on the base rules. It begets the contribution of each rule (step 3 of fuzzy inference) based on the outputs value of the FuzzificationAM.

## 3.3 Defuzzification DEVS Atomic Model

A defuzzification atomic model (DefuzzificationAM) outputs y. This value corresponds to crisp value which will be used to control the system. It is formally defined as:

$$DefuzzificationAM = <X,S,Y,\delta_{int},\delta_{ext},\lambda,t_a> \quad (11)$$

Where

InPorts = {‘InFuz’ }, $X_{InPorts}=[0, 1] \times \Re^4$,

OutPorts = {‘OutNum’}, $Y_{OutPorts} = \Re$,

$X = \{(in, x) / in \in InPorts, x \in X_{InPorts}\}$,

$S = \{$‘passive’, ‘active’$\} \times \Re$ ,

$Y = \{(out, y) / out \in OutPorts, y \in Y_{OutPorts}\}$,

$\delta_{int}$ (‘active’,0) = (‘passive’, ∞) ,

$\delta_{ext}$ ( (‘passive’, ∞), e, ‘InFuz’ ? (x1&x2&x3&x4) ) = (‘active’, u)

λ(‘active’,m)=OutNum!m

$t_a$(phase,m) = 0 if phase = active

= ∞ if phase = passive

‘u’ is obtained by Equation 6, corresponding to defuzzification method. The DeffuzificationAM generates a final conclusion of the fuzzy controller based on the fired rules of the base rules. It begins with the passive state (passive, ∞) until receives all RuleAM outputs (four contribution rules, see section 3.1) otherwise none output will be done and the model remains in wait state.

The DefuzzificationAM depends on the type of fuzzy inference adopted [10]. In our situation, the inference employed is the center of gravity (Figure 4).

## 3.4 FIS DEVS Coupled Model

As mentioned in section 3.1, we have used Mamdani rules type, where the consequent part of each rule is fuzzy. Thus the fuzzy inference system coupled model (FIS_MamdaniCM) has four FuzzificationAM, four RuleAM, one DefuzzificationAM, two inputs and a single output. It is formally depicted on the Figure 9.

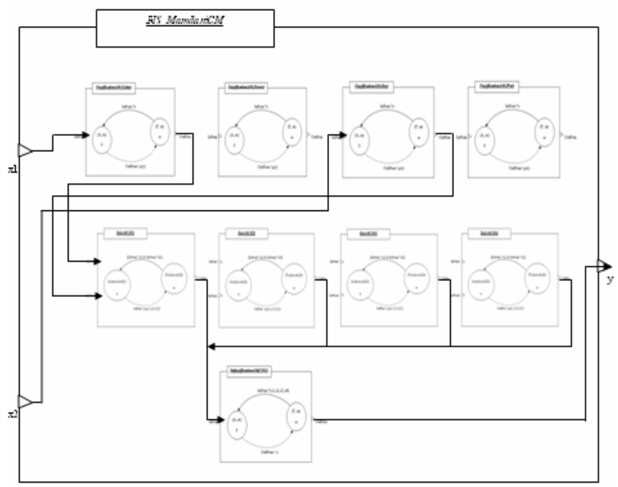


**Figure 9. FIS DEVS coupled model.**

# 4. APPLICATION

## 4.1 Problem Identification

Due to the dynamic and complex nature of wildfire, it is impossible to identify, capture and model all influential parameters with absolute accuracy [6,9,16]. Thus, its formulation is very complex in terms of taking all its parameters. DEVS seems a useful tool and appropriate solution for this dynamic process. However in this formalism, each lifetime is a constant over the time, therefore any evolution in the environment will not appear on our modeled system. In this work, we try to give a solution for this issue by introducing a fuzzy controller to assess modification when the input events occur on the system.

The literature distinguishes three classes of parameters which set the fire spread ratio: vegetation type (caloric content, density...); fuel properties (vegetation size) and environmental parameters (wind speed, humidity and slope...) [17]. The flaming fire evolves mainly according to the direction of the wind, its velocity and the relative humidity.

The present work uses two relevant baseline parameters: wind velocity (V) and relative humidity (H). The humidity influences the wildland fire behavior by increasing the risk factor. Low relative humidity is an indicator of high fire danger. A dry and powerful wind, associated with a dry ground, enormously increase the fire propagation.

Firstly, we identify five possible states that a cell can take (Figure 10). Each cell represents a limited area of the forest [3]:

- Nonflammable area (N): It can be a road, a surface of water or just an empty surface.
- Unburned area (U): It’s a passive state; it represents any fuel which is not consumed yet by fire.
- Burning area (B): represents a consuming fire.
- Ember area (E): A small, glowing piece of coal or wood, as in a dying fire.
- Ash area (A): It is afterburning state; it is the final combustion process state. At this stage, the nonvolatile products and residue were formed when matter is burnt.

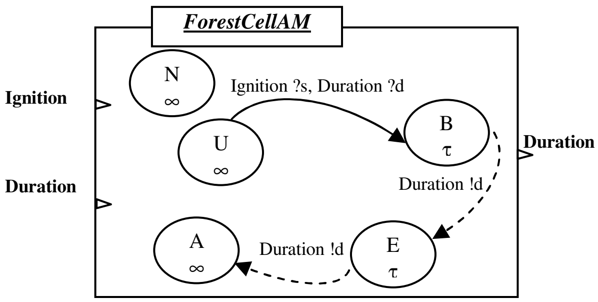


**Figure 10. Forest cell DEVS atomic model.**

Each state's lifespan depends on the ignition and duration inputs values. The Ignition port indicates the fire start time (at what time the fire was triggered?), while the port Duration, it brings the consumption time of each forest cell.

## 4.2 Fuzzy Reasoning

According to our forest cell atomic model (Figure 10), we note H the relative humidity parameter, whereas V the wind velocity. The fuzzy logic controller describes the structure of the fuzzy rules as follows:

**Rule$_i$**: **<u>If</u>** H is **A** **<u>and</u>** V is **B** **<u>Then</u>** $\tau_f$ is **C** (12)

Where A, B and C are linguistic variables and $\tau_f$ stands for fuzzy lifetime (fuzzy consumption time).

The variables are fuzzified as below (Figure 11):

The variable humidity H is divided into two fuzzy sets (linguistic term): Dry (D), and Wet (W). The wind velocity V is also fuzzified into two fuzzy sets: Calm (C), and power (P). The output variable $\tau_f$ is also fuzzified into two sets: Slow (S), and Fast (F).

The universe of discourse of each variable is given by:

- H: its values belong to [0%, 100%];
- V: is the interval [0,100km/h];
- $\tau_f$: The firefighters estimate the fire consumption of each cell at approximately 3 to 8% of the wind speed [2].

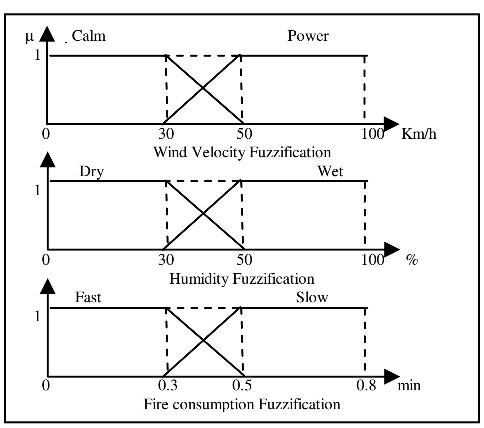


**Figure 11. Fuzzification of variables H, V and $\tau_f$.**

The fuzzy rules base is given by Table 2. This table is filled by firefighters. It is obtained by their experiences.

**Table 2. The Fuzzy Base Rules**

| $x_1$ / $x_2$ | D | W |
|---|---|---|
| **C** | S | S |
| **P** | F | S |

The fuzzy inference system uses the min-max center of gravity method. It calculates the consumption lifetime of each state and the result is given to forest fire coupled model (Figure 12).

## 4.3 Forest Fire DEVS Coupled Model

The proposed architecture is a classical DEVS framework. Our challenge is to keep the DEVS formalism unchanged and to improve it without modifying its components.

Our contribution is the addition of the FIS module whose function is to assess the lifetime of each state according to the input parameters, wind velocity (V) and humidity (H).

Initially, we fill the fuzzy rules base gotten from firemen reasoning. Each fuzzy rule is composed of two parts. The premise part, initially obtained from a data generator, and the consequent part which represent a state variable of the rule's DEVS atomic model.

The generator is a DEVS atomic model; it provides two kinds of values: spatial-temporal and environmental data. The spatial-temporal data are directed towards the forest coupled model, they supply the fire trigger event while the environmental data, pointed at the FIS coupled model to calculate the fire consumption duration (Figure 12).

The forest coupled model is a grid composed of n lines and m columns. Each cell represents a forest cell atomic model (Figure 10). Each cell is connected to its neighbors and provides the duration time obtained from the FIS coupled model. Each cell represents a DEVS atomic model which is associated to one simulator.

The dynamic system of the flaming front propagation speed is given by the simulator. It is based on the current cell position, consumption period and the wind direction. The wildland fire is considered as a propagation process that all burning cells ignite their unburned neighboring cells.

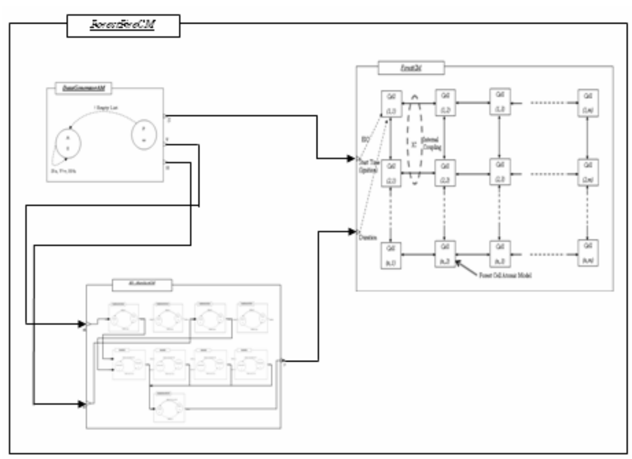


**Figure 12. Forest fire DEVS coupled model.**

# 5. IMPLEMENTATION AND RESULTS

The simulator used in this work is implemented in Java. It is developed in LSIS laboratory. Much functionalities are inspired from its predecessor LSIS-DME [7]. This version lacks of visual modeling tool however its utilization is very simple. The different paradigms of DEVS are defined as classes like root, simulator, atomic model, coupled model and so on. Each model inherit these classes and every implementation is easy to model despite the manually construction manner.

## 5.1 Variables Setting

In order to test our approach, two kinds of simulations are done. In the first one, we assume the lifetime of each state is a constant value. In the second simulation, the lifetime is obtained by the fuzzy controller.

In these simulations, the different values are:

- Wind velocity: Its value is 35 km/h.
- Humidity coefficient: (45%).
- Wildland: Closely spaced.
- The fuzzy controller outputs the propagation velocity. For each cell, $\tau$ is obtained as an output of the atomic model described by Equation 11.
- Virtual forest is constructed as a grid of 90×90 cells where each cell represents an area of 1.2×0.8 m².
- Each cell is connected to 8 neighbors to form a coupled model. Nearest neighbors are defined as grid.
- The initial ignited cell is the cell (1,1) (Figure 12).
- We assume uniform parameters characterize the cell space, i.e. the direction and wind speed, and the humidity are constant along the forest fire area.

## 5.2 Results and Discussion

To compare the simulation performance between the conventional DEVS lifetime state and the fuzzy one, two experiments on forest fire propagation are executed using the parameters described in section 5.1. The difference concerns the manner to obtain the duration of each state.

The simulations were carried out on a Dell System GX280 with Intel ® Pentium (R) IV, CPU 2.80GHz processor,2G DDR2 SDRAM memory and Linux 2.6.32-5-686 operating system .

The Table 3 summarizes some important results. The model ForestFireSimZ uses a conventional lifetime while ForestFireSim uses our approach. In the latter model, an atomic model was added in order to compute the duration of the cells fire consumption. This addition ensures the obtaining of the duration depending on weather changes.

**Table 3. Comparative Results**

| **Results** | **Conventional DEVS lifetime** | **Fuzzy DEVS lifetime** |
|---|---|---|
| Start time (Ignition) | Line1,Col1 | Line1,Col1 |
| Cell consumption time (Duration ($\tau$)) | 0.5 minutes | 0.556 minutes |
| Forest consumption time | 64.5 minutes | 69.6 minutes |
| Duration of the simulation | 616.29 seconds | 639.75 seconds |

To get better results, we have used additional free software which is Jconsole. It is a JMX-compliant monitoring tool. The table 4 resumes some important performances analysis between both models.

**Table 4. Performances Results**

| **Performance** | **ForestFireSimZ Model** | **ForestFireSim Model** |
|---|---|---|
| Uptime | 10 minutes | 10 minutes |
| Process CPU time | 3 minutes | 4 minutes |
| Total compile time | 18.819 seconds | 3.688 seconds |
| Live threads | 16 | 16 |
| Peak | 8,118 | 8,129 |
| Daemon threads | 12 | 12 |
| Total threads started | 183,553 | 199,785 |
| Current classes loaded | 1,912 | 1,909 |
| Total classes loaded | 1,937 | 1,946 |
| Total classes unloaded | 25 | 37 |
| Current heap size | 14,345 kbytes | 9,083 |
| Maximum heap size | 253,440 kbytes | 253,440 |
| Garbage collector Name: Copy | Collections:390 1,400 seconds | Collection:360 1,791 seconds |
| Garbage collector MarkSweepCompact | Collection:1 0,123 seconds | Collection:2 0,397 seconds |
| Committed memory | 17,380 kbytes | 18,428 |
| Pending finalization | 0 objects | 0 objects |
| Committed virtual memory | 517,808 kbytes | 519, 032 |
| Total physical memory | 2,065,076 kbytes | 2,065,076 |
| Free physical memory | 616,392 kbytes | 607,060 |
| Total swap space | 1,662,688 kbytes | 1,662,688 |
| Free swap space | 1,662,688 kbytes | 1,662,688 |

According to these results, we remark our approach brings some computation overhead compared to the traditional one. However, this method can add an interactive aspect by modifying the trajectory of the process without a great effort. It is sufficient to adapt the rules base and the lifespan of each state is modified immediately. However, a statistical study may be of interest to determine the compatibility of this comparison results and the viability of this approach.

Our view is that the model presented here to calculate the state lifetime by a fuzzy controller can complement rather than compete with the more popular deterministic or stochastic DEVS models. In absence of a formal model, this process can be possible. Also the fuzzy lifetime function proposed in this paper is tentative, providing a satisfactory model for the forest fire is beyond the scope of this work.

## 6. CONCLUSION AND FUTURE WORK

For dynamic processes whose modeling accuracy requirements surpasses the classic discrete event specification that uses mean state lifetime, this work has presented an approach without modifying the core of the DEVS formalism and introduces the concept of interactive lifetime by showing the relationship between the input values and the duration of the states. This method allows adjusting the trajectory of the process even if the input values change. Also, it can ensure a dynamic structure of the model.

The structural and behavioural framework was developed and implemented. Some relevant results were presented at the end of this work.

We have applied this method on forest fire propagation. An overview was presented on the relevant parameters whose influence is considered important. We have adapted the DEVS formalism by taking into account uncertainties without modifying the structure of the classic DEVS specification.

Thereby, the resulting application simulates forest fire propagation, including imperfect data. A comparison between the traditional simulation and our approach was given. However, this work needs to be tested in real environment to judge its efficiency.

Many parameters remain to be introduced in this model as topology, inflammability etc. This addition will help in affirming the validity of our approach.

## 7. REFERENCES


[1] Bertalanffy L.V. 1973.*General system theory*. Dunod edition.

[2] Bisgambiglia P.A. Capocchi L. Bisgambiglia P. and Garredu S. 2010. Fuzzy Inference Models for Discrete Event System. *IEEE world congress on computational intelligence*, Barcelona, Spain, July 18-23, 2010.

[3] Dahmani Y. and Hamri M. 2011. Event Triggering Estimation for Cell-DEVS Wildfire Spread Simulation Case. *European Symposium on Computer Modeling and Simulation* EMS 2011, 144-149, Madrid, Spain.

[4] Fishwick P.A. 1995. *Simulation model design and execution: building digital worlds.* Prentice Hall.

[5] Giambiasi N. Escude B. and Ghosh S. 2000. GDEVS: A Generalized Discrete Event Specification for Accurate Modeling of Dynamic Systems. *Transactions of the Society for Computer Simulation International*, 17 (3), 120-134.

[6] Grishin A.M. 1997. *Mathematical modelling of forest fires and new methods of fighting them.* House of the Tomsk State University.

[7] Hamri M. and Zacharewicz G. 2007. LSIS-DME: An Environment for Modeling and Simulation of DEVS Specifications. *AIS-CMS International modeling and simulation multiconference*, Buenos Aires Argentina.

[8] Hopcroft J.E. Motwani R. and Ullman J.D. 2000. *Introduction to Automata Theory, Languages, and Computation.* 2nd Edition. Pearson Education. ISBN 0-201-44124-1.

[9] Iliadis L.S. 2005. *A decision support system applying an integrated fuzzy model for long term forest fire risk estimation*. Elsevier, Environmental Modelling and Software, 20, 613–621.

[10] Jamshidi M. Sheikh-Bahaei, S. Kitzinger J. Sridhar P. Beatty S. Xia S. Wang Y. Song T. Dole U. and Lie J. 2003. *V-LAB–A distributed intelligent discrete-event environment for autonomous agents simulation*. Intell Autom Soft Comput, 9, 181–214.

[11] Lee C.C. 1990. Fuzzy logic in control systems: fuzzy-logic controller-part 1 and 2. *IEEE Transaction on Systems, Man, and Cybernetics*, 20(2), 404-435.

[12] Milivojević N. Grujović N. Stojanović B. Divac D. and Milivojević V. 2009. *Discrete Events Simulation Model Applied to Large-Scale Hydro-Systems*. Journal of the Serbian Society for Computational Mechanics, Vol. 3, No.1, 250-272.

[13] Mamdani E.H. 1975. Application of fuzzy algorithm for control of simple dynamic plant. *IEE, vol. 121*, 1585–1588.

[14] Papadopoulos G.D. and Pavlidou F.N. 2011. A comparative review on wildfire simulators. *IEEE Sys. Journal*, Vol 5, N 2, 233-243.

[15] Roger J. and Shing J. 1993. ANFIS : Adaptative-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol 23, issue 3, 665-685.

[16] Rothermel R.C. 1972, *A mathematical model for predicting fire spread in wildland fuels.* Research Paper Int, 115, Ogden, UT : U.S. Department of Agriculture, Forest Service.

[17] Scot J.H. and Burgan R.E. 2005. *Standard fire behavior fuel models: a comprehensive set for use with Rothermel's surface fire spread model.* USDA Forest Service General Technical Report RMRS-GTR-153.

[18] Takagi T. and Sugeno M. 1985.Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems Man and Cybernetics*, 15(1): 116–132.

[19] Wainer G.A. and Mosterman P.J. 2011. *Discrete-Event Modeling and Simulation: Theory and Applications.* CRC Press, Taylor & Francis Group, LLC.

[20] Zacharewicz G. 2006. *An environment G-DEVS/HLA : Application to modeling and simulate distributed workflow*". Doctoral thesis, University Aix-Marseille III, France.

[21] Zadeh L.A.1975. *The concept of a Linguistic variable and its Application to Approximate Reasoning*. Information Sciences 8, 199-249.

[22] Zadeh L.A. 1992. *The Calculus of Fuzzy If/Then Rules.* Artificial Intelligence Expert, Vol 7, N° 3, 23-27.

[23] Zeigler B.P. 1976. *Theory of modelling and simulation.* Wiley & Sons, New York.

[24] Zeigler B.P. 1984. *Multifaceted modeling & discrete event simulation.* Academic Press.

[25] Zeigler B.P. Praehofer H. and Kim T.G. 2000. *Theory of Modeling and Simulation 2nd Edition. Integrating Discrete Event Continuous Complex Dynamic Systems.* Academic Press.