iDEVS: new method to study inaccurate systems

P.-A. Bisgambiglia, E. de Gentili, Pr. P.A. Bisgambiglia and J.-F. Santucci
University of Corsica - CNRS UMR SPE 6132
Quartier Grossetti, bat 018, 20250 CORTE, FRANCE
(bisgambiglia,gentili,bisgambi,santucci)@univ-corse.fr

Keywords: Discrete event simulation, iDEVS, Fuzzy Sets Theroy, Inaccurate System, Fire Spreading

Abstract

Our recent research in the fields of modeling and simulation of complex systems, led us to study fuzzy systems. A system is fuzzy, because its parameters are inaccurate, or its behavior is uncertain. We propose in this paper to describe a new modeling method based on the association of DEVS formalism and the fuzzy set theory. Combining its two approaches we have permit to define a method of inaccurate modeling, whose goal is to study systems with inaccurate parameters.

1. INTRODUCTION

The work presented in this article concerns the modeling approach and simulation applied to the study of complex natural systems. The purpose is to elaborate a software environment and to propose generic tools adapted to a large number of situations.

The approach proposed is conducted in the framework of research undertaken in the field of modeling and discrete event simulation. The DEVS (Discrete Event system Specification) formalism introduced by Pr. B.P. Zeigler [23] has been developed and upheld for over thirty years by an International community of researchers [13, 3, 18, 11, 17, 1]. The work undertaken is part of the effort to develop an approach which will facilitate the modeling, simulation and validation phases of the study of complex systems.

This approach is based on the development of a software architecture enabling us, on the one hand to use the same multi-modeling environment to analyze different systems and fields and on the other hand to implement generic simulation techniques in order to simulate the corresponding models.

The modeling of natural systems leads to the processing and analysis of information and variables, the values of which are badly defined (fuzzy: inaccurate, uncertain, etc.).

The classical approach consists in approximating the values of the fuzzy variables which can create mistaken results during the simulation. We propose to integrate into DEVS formalism the use of tools arising from the fuzzy sets theory, so allowing the representation, the handling and the processing of this data. We can group together under the title "fuzzy sets theory": all mathematical tools that allow to reason on inaccurate data, for example fuzzy sets theory, fuzzy arithmetic introduced by Pr. Zadeh [20, 21], or the vertex method introduced by Pr. Dubois [9], etc.

We have chosen to validate the incorporation of the concepts of processing the inaccurate into the DEVS environment through the study of forest fire propagation. In this field there are many uncontrolled parameters (wind, vegetation) that can be modeled from an inaccurate representation.

In the first section, we present the concepts at the base of our approach: fuzzy modeling and DEVS formalism. In the second section, we present the iDEVS method. Before concluding, we propose the application of this approach to define a model of fire spreading to high level of abstraction.

2. BACKGROUND

In the fields of decision support, or the study of natural phenomenon, the data to address are very important. Most often, they are from ground study or measuring instrument unreliable. In these fields the methods of fuzzy modeling are adapted. They can represent and manipulate data. For this we are working on combining two modeling approach, approximate methods (fuzzy) and systemic methods (DEVS). In this section, we present the founding concepts of our new method of fuzzy modeling.

2.1. Fuzzy modeling

Fuzzy modeling, i.e. the design of fuzzy systems, is a difficult task, requiring the identification of many parameters. According to the Pr. L.A. Zadeh: "fuzzy modeling provides approximate but efficient means to describe the behaviour of the systems which are too complex or too badly defined to admit the use of a precise mathematical analysis".

To model a system with fuzzy parameter, we chose to represent these parameters in the form of fuzzy interval. A fuzzy interval is a generalization of the concept of fuzzy set. It is a simplified representation, to describe denumerable quantities. The handling of interval is made possible using several methods gathered under the name of fuzzy arithmetic.

2.1.1. Fuzzy Intervals

In a reference set X, a fuzzy set of this reference is characterized by a membership function (fig.1) λ of X in the interval of the crisp number $[0,1]$ [20]. This function is the extension of the characteristic function of a traditional set. The purpose of the concept of fuzzy set is to authorize an element to, belong more or less strongly, to a class.

A fuzzy set $\tilde{A}$ on the field of variation X of x is defined by the triplet: $(\tilde{A}, \tilde{a}, \lambda_{\tilde{A}})$, where:

- $\tilde{A}$ is a subset of X;
- $\tilde{a}$, a linguistic label, characterizing qualitatively part of the values of X;
- $\lambda_{\tilde{A}}$, the function x of X $x \in X \rightarrow \lambda_{\tilde{A}}(x) \in [0;1]$, which gives the degree of membership of an observation of X to fuzzy set $\tilde{A}$.

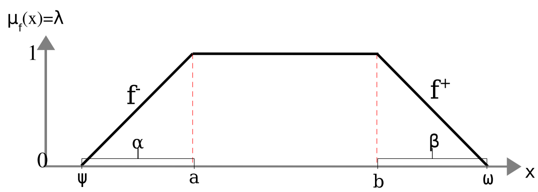

Figure 1. Membership function example

A fuzzy number (fig.1 if a=b) defined in [10], is a fuzzy interval at compact support. To make it simplier and more effective their handling, certain classes of numbers and fuzzy intervals were defined using a parametrical representation known as $L-R$. We take other two functions of form, L (left) and R (right), of $\mathbb{R}^+$ in $[0,1]$.

Data can be represented in two forms : an interval type $[(a,1);(b,1);\alpha;\beta]$ or two profil $[f^-(left), f^+(right)]$, a profil is a function of $[0,1]$ in $\mathbb{R}$ to model the right or left boundary of a fuzzy intervals.

f^+ representing the equation of the half-line (b,ω) defined by equation : $f^+(\lambda) = -\beta \times \lambda + (\beta + b)$ with $\beta = \omega - b$.

f^- representing the equation of the half-line (ψ, a) defined by equation $f^-(\lambda) = \lambda \times \alpha + (a - \alpha)$ with $\alpha = a - \psi$.

These two types of representation are shown on the figure 1.

A data type "between x and y" is modeled by the interval $[a = b = ((x+y)/2, 1); \psi = (x,0); \omega = (y,0)]$.

A data type "Approximately z" is modeled by the interval $[a = b = (z,1); \psi = (z - coef, 0); \omega = (z + coef, 0)]$ with $coef$ a confidence coefficient.

2.1.2. Fuzzy Intervals handling

The extension principle (eq.1), proposed originally by Zadeh [20], is one of the fundamental tools of the theory of the fuzzy sets. It allows to obtain the image of fuzzy sets by a function. Let ϕ be an relation between a universe E and F, where A is a fuzzy set of E and $B \in F$. The extension principle stipulates that the image by ϕ of A and B, is a fuzzy set which membership function is defined by:

$$\mu_{\phi(A,B)}(z) = sup\{min(\mu_A(x), \mu_B(y)) \,|\, \phi(x,y) = z\} \qquad (1)$$

With this principle may have generalized classical intervals functions at the fuzzy intervals.

The data handiling of the interval form is call fuzzy arithmetic. There are many methods for handling of such interval, presented in [4], in particular the Vertex method [9] or the interval analysis. For two fuzzy interval A and B.

- $A + B = [A^+ + B^+, A^- + B^-]$
- $A - B = [A^- - B^+, A^+ - B^-]$

– $A \times B = [min(A^- \times B^-, A^+ \times B^-, A^- \times B^+, A^+ \times B^+), max(A^- \times B^-, A^+ \times B^-, A^- \times B^+, A^+ \times B^+)]$

– if $A^- > 0$ then $ln(A) = [ln(A^-), ln(A^+)]$

The multiplication gives an approximate result, and if the calculate function is not monotonous, the functions are a little more complex, there is a difference cases $A > 0, A < 0, A = 0$.

Depending on the operations complexity performed by manipulating our information we can use these three methods; the extension principle and the interval analysis will allows the extension of any transaction at fuzzy intervals but is more complex to use. The vertex method is more intuitive and allows for manipulation of interval in the form of equation, but does not perform all the operations, such as the multiplication of two gives a result interval approached.

2.2. DEVS formalism

Since the seventies, formal work has been conducted to develop the theoretical foundations of modeling and the simulation of dynamic discrete event systems. DEVS (Discrete Event system Specification) [23] was introduced as abstract formalism for modeling discrete events. It allows the modeler to totally isolate himself from the implementation of simulators using the modeling of the system and is based for the simulation on the events and not on the time.

2.2.1. Principle of DEVS modeling

DEVS formalism can be defined as a universal and general methodology which provides tools to model and simulate systems, the behavior of which is based on events. It is based on the systems theory, the notion of components and enables the specification of complex discrete event systems in modular and hierarchical form. DEVS formalism is based on the definition of two types of model: atomic models and coupled models.

The atomic model provides an autonomous description of the behavior of the system, defined by states, input/output functions and internal transitions of the component.

The coupled model is a composition of atomic models and/or coupled models. It is modular and presents a hierarchical structure which enables the creation of complex models from basic models.

In DEVS, each model is independent and can be considered as its own entity or as a model of a larger system. It was shown in [23, 18] that DEVS formalism is closed under composition, that is to say that for each atomic or coupled DEVS model it is possible to build an equivalent DEVS atomic model.

Atomic model is characterized by:

$$AM = < X, Y, S, t_a, \delta_{int}, \delta_{ext}, \lambda > \tag{2}$$

with :

- X the input ports set, through which external events are received;
- Y the output ports set, through which external events are sent;
- $S$the states set of the system;
- t_a: $S \rightarrow \mathbb{R}^+$ the time advance function;
- δ_{int} : $S \rightarrow S$ the internal transition function;
- δ_{ext}: $Q \times X \rightarrow S$ the external transition function, with :
  - $Q = \{(s,e)|s \in S, 0 \leq e \leq t_a(s)\}$ state set;
  - $e =$ the time passed since the last transition;
- $\lambda : S \rightarrow Y$ the output function.

We have just seen, DEVS formalism is based for modeling on two types of components: coupled and atomic models. These components have input ports, output ports and variables. The exchange of the information is established through the ports of the various elements of a model, thanks to two types of fundamental events: external events and internal events.

An external event expected at the moment t represents a modification of the value of one or several input ports belonging to an element given M. This has as a consequence a modification of the variables of M, at the moment t.

An internal event expected at the moment t represents a modification of the variables of M, without any external event intervening. Moreover, the arrival of an internal event causes, at the moment t, a change of value on one or more output ports of the model M.

An event DEVS can be characterized by:

$$E = (time; port; value) \tag{3}$$

In formula 3, the first field represents the *time* of occurrence of the event, the second indicates the *port* on which the event happens, and the third symbolizes the *value* of the event.

In DEVS an event happens at a given time, it modifies the state of only one variable. If the state of several variables must be modified, several events are generated at the same date, which are treated by the algorithms of simulation according to a list of priority. For example if three variables must be modified by an event E which happens at time t it is fragmented in three events E1, E2, E3 still taken into account always at time t but according to a list of priority defined by the user [22]. As we have just specified it, in DEVS formalism an event must be treated with a quite precise date t. As the concept of events is at the base of the process of simulation.

2.2.2. Principle of simulation

Establishing a simulation requires the precise definition of behavior as well as the description of interactions existing between the entities of the model.

One of the important properties of DEVS formalism is that it automatically provides a simulator for each one of the models. DEVS establishes a distinction between the modeling and the simulation of a model in such a way as any DEVS model can be simulated without it being necessary to implement a specific simulator. Each atomic model is associated with a simulator in charge of the temporal synchronization of the underlying components. The totality of these models is managed by a specific coordinator called Root [23].

Each model communicates thanks to the sending and the reception of several types of messages. The principle is described in [23]. Each message generates events which are stocked in a schedule, which is a structure of data composed of events classified in chronological order, the head of the schedule representing the immediate future and the tail the more distant future. The simulation consists in making time evolve and provoking the changes of state predicted by the events.

2.2.3. Observation and proposal

DEVS formalism allows a separation of modeling and simulation phases. In a DEVS framework, the user only has to worry about design of its model, the simulation algorithms are automatically generated. This property coupled with modular and hierarchical aspects of formalism are a very powerful tool for studying complex system of any type.

However, depending on the studied system it is necessary to define specific models. One of the main advantages of DEVS formalism is its capacity for openness, so it can be easily extended to many fields of application.

This led him to be described as multi formalism: it brings together in a consistent manner, several methods or modeling formalisms.

Our goal is to rely on these properties to define a new modeling approach to take into account the inaccurate data models. To do this, we want to integrate into the DEVS formalism of tools related to the fuzzy sets theory. This theory introduced in the sixties provides a set of mathematical methods to represent and manipulate inaccurate data. Data is inaccurate when it is difficult to express clearly (approximately, almost, about, etc.). We chose to treat this type of data because they are widespread in the study of natural system high level of abstraction.

Two approaches based on DEVS formalism to take into account the imperfections on model parameters.

1. Min-Max-DEVS [11] formalism is too specific to a field, and deals only delays the time for triggering events;

2. Fuzzy-DEVS [13] formalism only deals in the uncertainties in transitions between states.

Although both formalisms do not meet our problem, namely the inclusion of inaccuracy on all parameters DEVS models, they have served as basis for defining some of the specifications of the iDEVS method.

3. IDEVS METHOD

In this part we present a new method of modeling and simulation for discrete event system that allows specification systems fuzzy parameters.

A system can be considered fuzzy if its parameters are known but not accurate (inaccurate), if the achievement of its parameters is not sure (uncertain), if his behavior is partially known (incomplete). This new approach called iDEVS was developed to be complementary with formalisms Fuzzy-DEVS and Min-Max-DEVS. From the study of these two formalisms we defined constraints and objectives to be met.

3.1. Constraints and objectives

iDEVS method is based on the fuzzy sets theory for representation and the manipulation of fuzzy quantities, a fuzzy quantity is an inaccurate (fuzzy) interval or number, it is a generalization of the concept of fuzzy set as applied to the denumerable parameters. Thanks to the fuzzy arithmetic, extension of some functions to handling the real numbers fuzzy quantities, we can model and manipulate to DEVS format systems with inaccurate parameters.

To make the link between DEVS formalism and the fuzzy sets theory, we created a library (object class) to build object representing of inaccurate variables. This library was subsequently incorporated into the DEVS formalism to give birth to iDEVS.

To take into account of inaccuracies in all DEVS model parameters without having to modify the simulation algorithms as in Min-Max-DEVS and Fuzzy-DEVS, we had to define new types of models. They were designed so that the changes remain imperceptible for the final user, unless it wants to program its models. iDEVS models incorporate DEVS concepts and tools developed to take account of inaccuracy. iDEVS is therefore an extension of DEVS formalism, it respects all its constraints, an iDEVS model which all parameters are defined as accurate has the same behavior a DEVS classical model.

In the DEVS formalism, an inaccuracy of the lifespan of a state leads a simulation problem. If you do not know precisely the end of lifespan of the state, simulation can not to execute. In answering this problematic, in modeling part, we've added in the time advance function of the atomic model a specific function. With this simulation algorithms do not have to be changed. As a result iDEVS method can be imported into any DEVS framework without having to reprogram, just using the data structure that has been defined. The coupling between the DEVS formalism and our data structure used to simulate systems inaccurate parameters. In the following, we present the first step towards the creation of the library.

3.2. Inaccuracy on the parameters

Our thinking was initially focused on the identification parameters can be inaccurate. At the level of atomic models, all parameters can be inaccurate: inputs X, outputs Y, states S, transition functions δ, output function λ, and time advance function t_a. In fact, the functions are not really inaccurate, their achievement may be uncertain, i.e. they can be executed or not, but they are not inaccurate. Thereafter we see them as inaccurate because they handle inaccurate data. The parameters that are most prone to inaccuracies are the state variables, lifespan of the states, values input and output models.

State variables that do level models, lifespan and values inputs and outputs are either at the origin of events, either directly manipulated by events. We therefore logically turned to the concept of events to take into account the inaccuracies in the DEVS formalism. It is essential, events run throughout the simulation; they distribute the information to models. In an event, the inaccuracy can be at the time and / or value.

In the proposed approach, the inaccuracy of value can be treated without having to change the DEVS formalism. An event inaccurate value can be inserted in the schedule of DEVS as a standard event, only the data type changes. An inaccuracy of value leads to a modeling problem, i.e. that it is the designer of the model to define its data in an appropriate type, but there is no change in the classical DEVS formalism. In fact, we give the possibility the designer to specify its data so inaccurate.

At the time level, an inaccuracy on a date induced modeling and simulation problems. An event is sent and placed in the schedule simulation on a given date if it does not know the precise date; the event can take place and therefore can not be taken into account in the classic DEVS schedule.

At the models level taking into account inaccuracies induced behavioural change and not structural, it is treated by various characteristics function of the DEVS models ($\delta_{int}, \delta_{ext}, \lambda, t_a$). We let the designer can describe the behavior of models from a library of programmed functions (+, -, ×, /, sin, cos, etc.), or reusing iDEVS models defined and stored in a library.

The identification of the parameters was an important step; it lays the foundation of the iDEVS approach. The next step must allow the representation of inaccurate parameters, why it is necessary to choose an appropriate data type.

3.3. Data representation

To describe the settings, it is important to choose a suitable mode of representation, we have based on the fuzzy sets theory and the settings description as a fuzzy quantity. A fuzzy quantity allows to model a fuzzy interval or number, it is a fuzzy set on the real (later we will use only the term fuzzy interval).

This mode of representation is adapted to dialogue with specialist's designers. Moreover, it allows to take into account the inaccuracy of the proposal, and thanks to the membership functions, to combine digital and linguistic representations To describe a fuzzy interval, and to provide adapted tools to the representation of specialists and their study fields, we offer two methods of construction.

The first method can quickly and simply describe an inaccuracy. It is based on a description of the interval from reference points, i.e. four couples value membership degree, type $[(a,1);(b,1);(\psi,0);(\omega,0)]$. They are presented in figure 1: a and b are the vertex of the membership function; $\psi = a - \alpha$ is the lower limit of the interval; $\omega = b + \beta$ is the upper limit of the interval.

The second method can represent the interval in the equation form. This style is practical, simple, intuitive, and can quickly translate an inaccuracy in graphic form. The use of a visual description is easy to understand and assimilate. Two forms of equations are presented, according to lambda (y-axis, ordinate $f^-(\lambda) = \alpha \times \lambda + (a - \alpha)$, and $f^+(\lambda) = -\beta \times \lambda + (\beta + b)$) or function of x (x-axis, abscissa $\mu^-(x) = \frac{x-\psi}{a-\psi}$ and $\mu^+(x) = \frac{x-\omega}{b-\omega}$). According to implement operations on intervals, it may be necessary and appropriate to switch from one to another.

These representations are a generalization of the concept of classical interval. The basic principle is to replace any inaccurate number by an interval container and perform calculations on intervals, all calculated interval contains the result of exact calculation. These concepts are included in the fuzzy sets theory, with the addition of a membership degree. With the extension principle it was proved that a large part of the operation defined for classical intervals are usable for fuzzy intervals.

3.4. Data handling

Once the chosen mode of representation, the aim being to simulate data, it should be possible to perform operations and handle. The fuzzy intervals are commonly used in approximate reasoning; there is a lot of research to extend basic operations. We rank these operations into several categories: classic (+, - ,×, /, etc.) monotonous functions, functions whose variation sense does not change on the interval (sqrt, expo, ln, etc.), and non-monotonous functions (sin, cos, tan, etc.). Each of these cases requires the use of specific methods, from either the extension principle either the calculation of classical intervals, either the vertex method.

These different methods can perform operations on the fuzzy intervals as if it were classical intervals. The data form, interval or equation, is not important in choosing manipulation functions. As required we switch from one to another.

Depending on the operations, we use the methods that we consider most appropriate and most intuitive. The aim is to promote understanding of operations. For example, the vertex method is used for simple operations (+,-), calculation intervals and the extension principle for more complex operations (×, /, expo, ln, sin, cos, tan, etc.). Much of these methods and structure of data are integrated into a class to allow the definition and manipulation of data fuzzy interval type.

In the algorithm 1 we present this library functions as a class, called FuzzyInterval. It is possible to create an instance of the class from the values a, b, ψ, ω may have to add a linguistic label characterizing the interval.

In this class we have implemented various classic operations handling fuzzy intervals and have chosen to use the most appropriate methods to meet our objectives. For instance, for the operation +, we work on the bound of the interval, from the vertex method. A user who wants to test other methods should overrun our functions. If for much of the conventional operators different methods give the same results, an overrun is not of great interest for other functions, which have been programmed specifically for our needs, their uses can provide results. In this case, it is advisable to adapt to the context.

A user who wants to test other methods should redefine our functions. If for much of the conventional operators different methods give the same results, a redefine is not of great interest for other functions, which have been programmed specifically for our needs, their uses can provide results. In this case, it is advisable to adapt to the context.

To take account of inaccuracy parameters, we have identified a new type of data, several handling functions, and then we have grouped the all in a class. This class implements most of the classic functions of arithmetic and provides tools to define new functions using the most appropriate methodology (vertex method, fuzzy arithmetic, the extension principle). With this approach a designer has the option to use in these models variables inaccurate to describe the behavior of

Algorithm 1 Fuzzy Interval class

list $function = ln, expo, \surd, sin, cos, tan, etc.$
list $operator = +, -, \times, /, etc.$
$class$FuzzyInterval{
int $\lambda \in [0,1]$
float a,b
float ψ, ω
float $\alpha = a - \psi$, $\beta = \omega - b$
$function\ left(int\,\lambda) = \alpha \times \lambda + (a - \alpha)$
$function\ right(int\,\lambda) = -\beta \times \lambda + (\beta + b)$
$function\ left(float\,x) = \frac{x-\psi}{a-\psi}$
$function\ right(float\,x) = \frac{x-\omega}{b-\omega}$
FuzzyInterval$(a, b = a, \psi = a, \omega = b, label = 'I')$ // constructor
FuzzyInterval $operator*$(FuzzyInterval) // operator with FuzzyInterval
FuzzyInterval $operator =$ (FuzzyInterval)
FuzzyInterval $operator +$ (FuzzyInterval)
...
FuzzyInterval $operator + (int)$ // operator with integer
...
FuzzyInterval $operator + (float)$ // operator witg float
...
float $defuzzificationfunction(FuzzyInterval)$ // defuzzification methods
...}
// friends function
FuzzyInterval sin(FuzzyInterval)
...

a system, it is sufficient to use the FuzzyInterval type.

At the level of the modeling formalism, simulation algorithms are automatically generated in accordance with DEVS formalism, changes to keep this property are presented in the next section.

3.5. iDEVS simulation

The definition of inaccurate parameters in models can cause problems simulation. If the lifespan of the state is inaccurate, it is impossible to define the date of update, and we remain always in the same state. To solve this problem, we have studied both. The first is, just change the simulation algorithms by adding a fuzzy ranking [16, 14]. The goal is to evaluate and rank the different time in terms of their execution date, so it is possible to choose which event trigger. The disadvantage of this solution is that it involves changes to the simulation algorithms which lead us to amend part simulation of classical DEVS formalism; we set a priority to use the iDEVS approach in any DEVS framework without having to make major changes on the simulation. The second solution is the use of defuzzification function (the defuzzification is a decision-making phase that can transform a fuzzy value of a variable from crisp value). This method seems more suited to our problem.

To validate our choices, we implemented various defuzzification functions in the FuzzyInterval class and have tested different iDEVS models, the results were the subject of two publications [6, 5]. The function that we believe has the greatest advantages is the EEM method [2], it allows you to add a decision support; defining a confidence coefficient, the user can choose when transform inaccuracies on time (if the confidence coefficient is small, between 0 and 0.5, we trigger the event early, more confidence coefficient is high, between 0.5 and 1, more the event is triggered later). In the end we lose information on the x-axis, but we keep the coefficient of the validity of the proposal.

We believe it can be useful, in addition to the average validity coefficients defuzzifed, to provide final result the interval of time simulation. This interval represents all the time that the simulation could be terminated. It is calculated through a state variable type FuzzyInterval which increments the variable imprecise time (before defuzzification) to each execution of the function.

Once the method chosen must include in DEVS formalism and modify the basic atomic model to reflect our changes. This step we passed DEVS to iDEVS because the DEVS atomic model needs evolve. This evolve can handle all problems at the level of modeling. It is not necessary to change the simulation algorithms, our starting constraints are met. The problem is linked to time; we modified the time advance function (t_a defuzzification method class 1) atomic model for returns accurate time. Thus we have no problem of inaccurate time simulation is set in the design of the model and therefore modeling part.

Behavioural level (in the model), the time can be defined as inaccurate, but it will be defuzzifed before being sent to the simulator. The time advance function was amended to test if time to return is accurate or inaccurate, in the second case, time is transformed. If the events from a generator model, thus an atomic model, we manage with the time advance function values inaccuracy on the time of triggering events, such management requires defining a fuzzy atomic model, The difference between the function of classic DEVS model and function of your iDEVS model remains imperceptible to the final user. It is important to note that the first message sent to the simulator, initialization message (i-message that sets the simulation time), does not contain inaccurate time. Generally, in DEVS formalism, all events are a source. This remark is to take into account that in the case where an event would be a source outside the system.

In the iDEVS atomic model, we apply in the time advance function a defuzzification function (EEM), and we keep the degree of validity of the condition (ordinate membership function to the interval: λ) in the form of coefficient average oh the defuzzification lambda. This new variable may be stored as a state system S or added as a class variable in the class atomic model, it is the same for the interval from the time at the end of the simulation. These changes are presented in the next section.

3.6. iDEVS models

To take into account the inaccuracies iDEVS models must derive or instantiate class FuzzyInterval. The goal is to provide the designer ways to define 'inaccurate' models regardless of platform or an application, just important class in modeling framework. The designer can model its fuzzy system from the combination of iDEVS models derived from the class FuzzyInterval and a classic DEVS atomic models. It is noted that the coupled model resulting from a combination of 'inaccurate' models and conventional models is a model that automatically returns inaccurate results. All iDEVS models handle inaccuracies data (data for FuzzyInterval type), they have the opportunity to use the functions defined in FuzzyInterval class. In this section, we describe the changes made to DEVS models to allow consideration of inaccuracies.

3.6.1. Atomic model

iDEVS atomic model is similar to the DEVS atomic model. Its uniqueness is that it can manipulate variables accurate or inaccurate. Its role is to describe the behavioural aspect of part of a system parameters inaccuracy. If all parameters of the iDEVS model are accurate, it has the same behavior classic DEVS model. In terms of DEVS and iDEVS atomic models, the behavior of the internal and external transitions functions (δ_{int}, δ_{ext}) is strictly identical. As against, for handling inaccuracy variables, we use the overloaded functions in the class FuzzyInterval (+, -, / cos, etc.). To describe δ_{int} and δ_{ext} the user need only indicate that the variables are inaccurate, employing the appropriate functions automatically. For the time advance functions and output function of structural changes were introduced, but they remain imperceptible to the final user. The time advance function must return a crisp value so that the corresponding event can be inserted into the schedule. To this we added a defuzzification function. This function is automatically activated and the final user does not have to worry about its operation. A user can modify the implementation of this function from the class FuzzyInterval. Another structural change appears in the output function; in the end it returns the simulation results, and two class variables that contain the validity degree of the result and the time interval simulation.

$$AM_{iDEVS} :< \tilde{X}, \tilde{Y}, \tilde{S}, \tilde{t_a}, \tilde{\delta}_{int}, \tilde{\delta}_{ext}, \tilde{\lambda} > \quad (4)$$

With:

- $\tilde{X} = \{(p,\tilde{v}) \,|\, p \in input\ ports, \tilde{v} \in \tilde{X}_p\}$: the list of input ports, each port is characterized by a couple (port number/value), where the value can be defined as accurate or inaccurate ;
- $\tilde{Y} = \{(p,\tilde{v}) \,|\, p \in output\ port, \tilde{v} \in Y_p\}$: the list of output ports, each port is characterized by a couple (port number/value), where the value is accurate or inaccurate depending on the behavior of the model ;
- $\tilde{S}$: all state or state variables accurate S or inaccurate $\tilde{S}$ system $S \in \tilde{S}$;
- $\tilde{t_a}(\tilde{S}) \rightarrow R^+$: time advance function, algorithme 2 show this function $\tilde{t_a}$;

Algorithm 2 time advance function $\tilde{t_a}$

// declaration of class variables
FuzzyInterval $\tau = [0,0,0,0]$ *// interval representing the time to end simulation*
real Λ *// the sum of membership degrees λ defuzzification*
real $nbrDefuz \leftarrow 1$ *// variable that counts the number of defuzzification*
real $moy\Lambda = \frac{\Lambda}{nbrDefuz}$ *// variable that keeps the average λ, t is returned at the end of each simulation model*
// time advance function
function réel $\tilde{t_a}$(état $\tilde{S}$){
  σ the lifespan of the state $\tilde{S}$
  if σ is accurate *// σ is tested, if σ is accurate the function t_a has a classic behavior*
    $t_a \leftarrow \sigma$
    $\tau \leftarrow \tau + \sigma$ *// interval τ increases in σ, end simulation will provide a interval time*
  else
    $t_a \leftarrow \sigma.coefEEM()$ *// σ is a instance of the class FuzzyInterval we apply the defuzzification method $coefEEM()$[5]*
    $\Lambda \leftarrow \Lambda + \mu(\sigma.coefEEM())$ *// Λ is the sum of defuzzification λ, function $\mu(x)$ return the value λ for x*
    $nbrDefuz \leftarrow nbrDefuz + 1$
    $\tau \leftarrow \tau + \sigma$ *// we add to the interval τ to the interval σ*
  return t_a
}

- $\tilde{\delta}_{ext} : \tilde{Q} \times \tilde{X} \rightarrow \tilde{S}$: external transition function, where :
  - $\tilde{Q} = \{(\tilde{S}_i, e) \,|\, \tilde{S}_i \in \tilde{S}, 0 \leq e \leq t_a(\tilde{S}_i)\}$: the set of the accurate or inaccurate states $\tilde{S}_{\{1,2,\ldots,n\}}$;
  - e : is the time elapsed since the last transition, the role of external transition specifies how the atomic model changes state (from $\tilde{S}_1$ to $\tilde{S}_2$ when a accurate or inaccurate input occurs (external event) before $t_a(\tilde{S}_1)$ has expired ;
- $\tilde{\delta}_{int} : \tilde{S} \rightarrow \tilde{S}$: internal transition function. It allows to switch between a state $\tilde{S}_2$ to the date t_1, to a state $\tilde{S}_1$ at the moment t_2 when external event happens during the

lifespan of the state $t_a(\tilde{S}_2)$;
– $\tilde{\lambda} : \tilde{S} \rightarrow \tilde{Y}$: output function, it returns the model outputs and the class variables τ and $moy\Lambda$.

The data handled by iDEVS atomic model are represented by a quadruple $[a, b, \psi, \omega]$, defined in the class called FuzzyInterval. If a = b and $\alpha = \beta = 0$ iDEVS model becomes a classic DEVS model (not fuzzy) handling accurate data. Equation 4 presents in detail the general iDEVS atomic model. The tilde (~) on a parameter means that it is inaccurate or manipulating inaccurate variables. The input values may be inaccurate X; upon receiving a input value, fuzzy external transition function is triggered δ_{ext}, it updates the state S of and its lifespan ta according to the specifications defined by the designer. If no entry is found before the end of lifespan ($t_a = 0$), fuzzy internal transition function and output function are triggered. δ_{int} updates the state of the system according to specifications set by the designer and lambda generates simulation results Y.

3.6.2. Coupled model

iDEVS coupled model has the same form of the DEVS coupled model; both are described in the same way. The only difference is that the iDEVDS coupled model may consist of classic DEVS models or iDEVS models. Therefore the input variables and output are defined as inaccurate. Inaccurate data includes accurate data.

The main advantage of the iDEVS method is consideration of inaccuracies in defining the parameters of the models in the form of intervals. The intervals may be described from numerical values or linguistic. This description is simple and intuitive, close to the mode of human representation. Moreover the simulation results are guaranteed, less precise but with great opportunity to be fair.

These models can be used to describe specific systems, i.e. that the designer himself shall encode its model to represent the system to simulate. After introducing the general forms Idev models in the next section, we present some of these models in their application contexts.

4. APPLICATION

These last few years have reminded us with force that the fight to combat forest fires has not been won in advance. Several methods for the study of the propagation of forest fires exist. Some are used to describe in a more or less in-depth manner, with the help of physical and mathematical equations [15, 19], all the mechanisms implemented. Others closer to a more in the field level of reasoning consider that a large number of parameters may not be taken into account [12, 7].

In this perspective and in order to conform to the realities in the field, we have undertaken work in collaboration with the SDIS (Service Départementale d'Incendie et de Secours in English Depatmental/County Fire Rescue Service) of Northern Corsica. Several courses of action, remaining very close to their needs and concerns, have emerged from this cooperation.

The model presented in this part transcribes in a data processed manner the empirical reasoning of the SDIS firemen undertaken in the field. The information presented in linguistic form has been translated into models with as objective the carrying out of a system in real time. One of the problem areas advanced by the SDIS is the necessity to rapidly predict the possible progression of the fire in order to implement an adequate policy to fight it.

The model describes the evolution of the fire front in terms of zone (vector propagation [8, 7]). The terrain is modeled in terms of its influence on the fire, it is divided into zones, and each zone has its own characteristics. We do not calculate the spread of fire with time, but in terms of changes in zones. On a given zone we consider that the parameters influencing the fire are invariants. The aim of the model is to provide firefighters the ability to predict different scenarios of propagation, and to take into account the structural or behavioral changes on the ground.

The evolution of the model takes place in three stages: (1) it calculates the points of intersection between the fire front and the next affected zones; (2) it assesses the distances traveled by the fire; (3) and it calculates the likely time before the next zone does not is reached.

The model is defined on the following parameters:

- coordinates of departure set by use;
- coordinates delimiting each zone;
- spread coefficient given by a fuzzy inference system corresponding to the characteristic of each zone (flammability, height and density of vegetation, wind speed on the area, topology field, etc.). The fuzzy inference system was established in collaboration with firefighters;
- wind speed and direction premises;
- percentage of the wind speed that set the speed of fire spread over a zone. It is equal to more or less 3 to 8% of the wind speed, and was defined by firefighters.

4.1. System

The system is based on all these parameters, the fire start to coordinates (10;10) set by the user. The field is divided into three zones:

1. the first zone is a rectangle. It is described from the coordinates [(0.0) (80,200)]. The slope is 28° (defined in the fuzzy inference system as "mounted"), the degree of flammability of the vegetation is set at 0.7 (highly flammable), the height of vegetation is 1.60m (timber bottom), ground is dirty (maintenance = 1.6), and the

wind comes from the south west (type of corsica wind "Lebecciu"), it blows between 50 and 70km / h (defined as "large wind"). From these parameters, the spread coefficient returned by the fuzzy inference system is equal to 1.8 (high);

2. the second zone coordinates [(80,200) (140,200)], the wind comes from the south west (type of corsica wind "Lebecciu"). The spread coefficient returned by the fuzzy inference system is equal to 1 (medium);

3. the third zone coordinates [(140,200) (300,200)], the spread coefficient is equal to 1.4 (high). Winds between 30 and 50km / h south (corsica wind "Sciroccu").

4.2. Model

To represent the system we have identified four DEVS or iDEVS atomic models. The first model (ground model) contains the ground parameters. It returns the start coordinate of the fire and the parameters of the affected zone, and when it receives a message from the propagation model, for each change of zone. The second model (model weather) is a generator that transmits meteorological data, wind speed and direction. The third model (propagation model) is the most important; it calculates the points of intersection between a zone and the fire front. The final model (model display) displays the results of the propagation model.

4.3. Results

The propagation model calculating the time and coordinates of new zone of impact. The data returned are FuzzyInterval type. The table 1 shows the data obtained. We have for each zone coordinates, distance travelled, time before and after defuzzification and the degree of validity time defuzzification. We can see that the results are relatively good, the degree of membership is always greater than 0.5. All these results are from our simulation algorithm (t_a 2). Indeed, it helps to advance the simulation defuzzification inaccurate times, but also to return the entire fuzzy interval. To get these results, we chose a very little confidence coefficient (in our defuzzification function EEM). This helps trigger the event very early in time. In the case of fires forest this possibility of setting is very important. It allows to predict critical cases, for example, the fires spread very quickly.

From these data we have recreated the spread of the fire. In this figure 2 the four zones are visible, the coordinates of fire start (10.10) and different points of intersections. We can see that more simulation lasts, the greater the impact interval is high. The angle also plays an important role; it is easy to notice the change in the direction of propagation between the first two zones and the third. From these data simulation, we can conclude that the fire front to reach the four zone in 21 hours with a certainty above 0.6. The fire will have travelled about 244 kilometres in 21 hours. These results may provide fire-fighters a good database to position their men on the field long before the fire happens. Thus, without fear of endangering men, it is possible to prepare the ground to slow at best the fire front.

5. CONCLUSION AND PERSPECTIVES

In this document we have presented part of our work on fuzzy modeling. Notably, we have detailed our approach based on the integration of the fuzzy sets theories into multi-modeling DEVS formalism. This method has as objective to help experts in a domain, such as fire-fighters for forest fires, to specify in a simple way the behavior of a complex system characterized by badly defined parameters.

Our approach can be used both in the domain of help for decision-making as well as for crisis management

The basic idea in our methodology is to enable the modeler to specify the parameters of fuzzy models, in the form of intervals or variable linguistics. In order to make the simulation of this data possible, a library of fuzzy functions has been added to the DEVS formalism.

Furthermore, with a view to its improvement, we are working on several courses of action, such as the adding of new iDEVS models and, the definition of other functions in the class

Finally, it will also be necessary to see how the fire brigade services will be able to use this method in the field. Moreover, we are working on a pluridisciplinary project in the domain of computers which aims to integrate DEVS formalism and several other modeling techniques such as, MAS (Muli-agent system), the GIS (Geographic information system), and WEB services, so as to validate a software environment of modeling and simulation of complex (dynamic and/or fuzzy) spatialized systems. In this context we are currently working on developing a fuzzy toolbox for DEVS formalism, called fuzz-iDEVS. It would take into account in the same modeling formalism inaccuracies and uncertainties.

REFERENCES

[1] *Transformation of VHDL Descriptions into DEVS Models for Fault Modeling and Simulation*, 2003.

[2] A. Anglani, A. Grieco, F. Nucci, G. Semeraro, and T. Tolio. A new algorithm to rank temporal fuzzy sets in fuzzy discrete event simulation. *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on*, 2:923–928, 2000.

[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.

	X	Y	Distance	Time	Accurate time	membership degree
Zone	80	93.4	108.9	40.3		
	80	80.0	98.9	20.3	17.1	0.7
1	80	68.7	91.3	9.1		
Zone	140	164.9	93.3	172.8		
	140	140	84.8	94.2	75.5	0.6
2	140	119.1	78.3	44.5		
Zone	200	175.5	60.9	48.3		
	200	145.2	60.2	23.9	21.2	0.8
3	200	119.1	60.0	10.7		

Table 1. Simulation result.

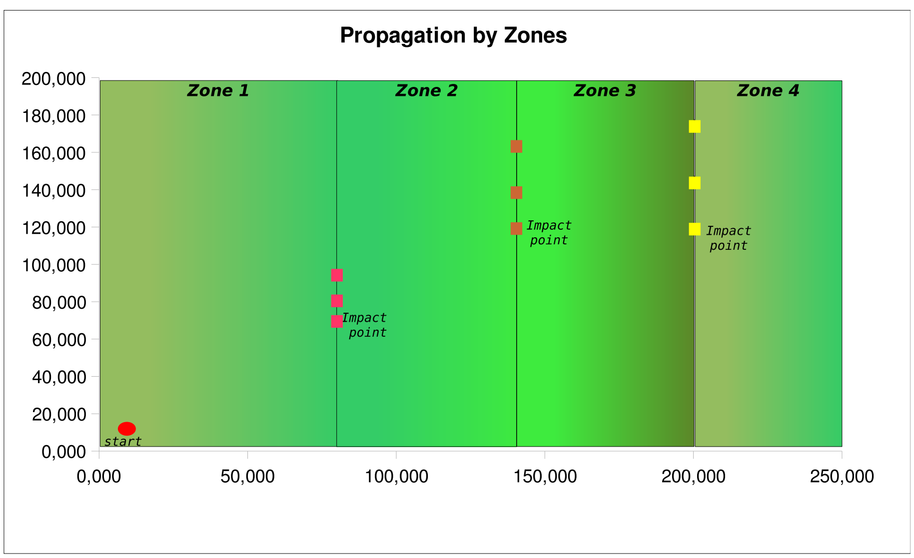

Figure 2. Results sample interpretation

[4] P.-A. Bisgambiglia, L. Capocchi, E. de Gentili, and P.A. Bisgambiglia. Manipulation of incomplete or fuzzy data for DEVS-based systems. In SCS, editor, *Proceedings of the International Modeling and Simulation Multi-conference (IMSM) - Conceptual Modeling Simulation (CMS)*, pages 87–92, 2 2007.

[5] P.-A. Bisgambiglia, E. de Gentili, P.A. Bisgambiglia, and J.F. Santucci. Discrete events system simulation-based defuzzification method. In *Proceedings of The 14th IEEE Mediterranean Electrotechnical Conference (MELECON)*, pages 132–138, 05 2008.

[6] P.-A. Bisgambiglia, E. de Gentili, P.A. Bisgambiglia, and J.F. Santucci. Fuzzy simulation for discrete events systems. In *Proceedings of the 2008 IEEE World Congress on Computational Intelligence (WCCI 2008) - IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 688–694, 06 2008.

[7] P.-A. Bisgambiglia, E. de Gentili, J.B. Filippi, and P.A. Bisgambiglia. DEVS-Flou: a discrete events and fuzzy logic-based new method of modelling. *SIMULATION SERIES, VOL 38, PART 4*, pages 83–90, 7 2006.

[8] P.-A. Bisgambiglia, J.B. Filippi, and E. de Gentili. A fuzzy approach of modeling evolutionary interfaces systems. In IEEE, editor, *Proceedings of the ISEIM 2006, Corte (France)*, pages 98–103, 6 2006.

[9] D. Dubois, H. Fargier, and J. Fortin. A generalized vertex method for computing with fuzzy intervals. In IEEE, editor, *Proc. of the International Conference on Fuzzy Systems*, pages 541–546. IEEE, July 2004. Budapest, Hungary.

[10] D. Dubois and H. Prade. Fuzzy set in approximate reasoning. *part 1, Fuzzy set and Systems, vol 40*, 1993.

[11] N. Giambiasi and S. Ghosh. Min-Max-DEVS: A new formalism for the specification of discrete event models with min-max delays. pages 616–621. 13th European Simulation Symposium, 2001.

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

[13] Y. Kwon, H. Park, S. Jung, and T. Kim. Fuzzy-DEVS Formalisme : Concepts, Realization and Application. *Proceedings AIS 1996*, pages 227–234, 1996.

[14] Seungsoo Lee, Kwang H. Lee, and Doheon Lee. Ranking the sequences of fuzzy values. *Inf. Sci. Inf. Comput. Sci.*, 160(1-4):41–52, 2004.

[15] R. C. Rothermel. A mathematical model for predicting fire spread in wildland fuels. *Research Paper INT- 115*, Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station.:40p, 1972.

[16] Liem Tran and Lucien Duckstein. Comparison of fuzzy numbers using a fuzzy distance measure. *Fuzzy Sets Syst.*, 130(3):331–341, 2002.

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

[18] H. Vangheluwe. The Discrete EVent System specification DEVS Formalism. Technical report, 2001. http://moncs.cs.mcgill.ca/.

[19] R.O. Weber. Modeling fire spread through fuel beds. *Prog. Energy Combust. Sci, vol. 11*, pages 67–82, 1991.

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

[21] L.A. Zadeh. A fuzzy-set theoretic interpretation of linguistic hedges, 1972.

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

[23] Bernard P. Zeigler, Herbert Praehofer, and Tag Gon Kim. *Theory of Modeling and Simulation, Second Edition*. 2000.