# Discrete events system simulation-based deFuzzification method

P.-A. Bisgambiglia, E. de Gentili, P. Bisgambiglia and J.-F. Santucci

*Abstract*— **This article presents our approach to discrete event simulation for system with inaccurate parameters. For such systems, simulation runs according to events whose dates are known; but in fuzzy modeling, it may be that the date of the events are inaccurate. To solve this problem, we suggest using a method that converts a inaccurate value into accurate value. This method is incorporated into a modelling and simulation formalism.**

*Index Terms*— **Fuzzy Sets, Defuzzification, Interval Analysis, Modeling, Simulation, Discrete Events Systems.**

## I. Introduction

FOR several years our laboratory to study discrete events system, the formalism of modeling and simulation DEVS (Discrete EVent system Specification) introduced by professor B. Zeigler [1], and followed by an international community of scholars ,[2], [3], [4], [5] is the focus of this work. DEVS facilitates phases of modeling, simulation, and validation, in the study of complex systems.

At different study we found that the incorporation of information could lead to accurate final results wrong. Most of the modeling approaches do not take into account imperfect information.

We propose in this paper a new method for modeling and simulation based on the DEVS formalism taking into account the inaccurate information. This approach is based on fuzzy interval arithmetic, with such arithmetic, it is possible to take into account the inaccuracy on the data and return a bound containing a sudden the result of a calculation: the strength of by interval arithmetic is the reliability of the results.

Furthermore, it was shown by Professor L. Zadeh [6] a large part of the functions to classical intervals can be extended to fuzzy intervals.

The first part of this article presents the work of Professor L. Zadeh on the fuzzy set and DEVS formalism by Professor B. Zeigler. The second part deals our approach to modeling and simulation. Before concluding finally we present an application.

## II. Background

For several years our laboratory is working on the modeling and simulation of complex systems. We distinguish three stages in the process of modeling and simulation:

1) Model, it's to describe a way synthetic and digital the system behavior to make it interpretable by a computer;
2) Simulate, it's to activate of the numerical model to obtain the results of the model behavior according to a given situation.
3) Validate, it's comparing between digital model data, after simulation, and the real data of the system to confirm their validity.

During these phases information gathered on the ground are transformed into a model and then simulated. Depending on the field or source, the information may be inaccurate. The aim of our research is to allow the incorporation of imperfect information in a modeling and simulation formalism. After presenting various techniques for handling of uncertain we revisit DEVS [1] formalism, and the reasons that have pushed to work on it.

### A. 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, the handling of interval is made possible using several methods gathered under the name of fuzzy arithmetic

*1) Fuzzy Intervals:* In a reference set $X$, a fuzzy set of this reference is characterized by a membership function (fig.1) $\lambda$ of $X$ in the interval of the crisp number $[0, 1]$ [6]. 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 $\tilde{A}$.

A fuzzy number (fig.1), as *Dubois* and *Prade* defined it in [7], is a fuzzy interval at compact support having only one modal value. 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 $[0, 1]$, symmetrical, not decreasing on $[0, +\infty[$; such as:

University of CORSICA - CNRS UMR SPE 6134 - Faculté des Sciences et Techniques Batiment 018 Quartier Grossetti 20250 Corte, Corse (bisgambiglia/gentili/bisgambi/santucci)@univ-corse.fr

*978-1-4244-1633-2/08/.00 ©2008 IEEE*

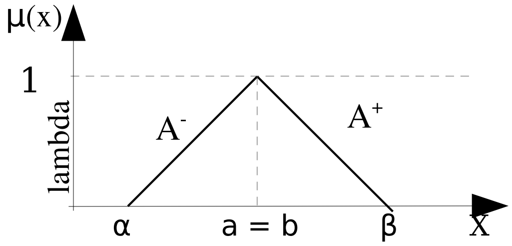


Fig. 1. Membership function example

$L(0) = R(0) = 1$, $L(1) = R(1) = 0$ where $L(x) > 0\,\forall\, x$ with $lim_{x\to\infty} L(x) = 0$ and $R(x) > 0\,\forall\, x$ with $lim_{x\to\infty} R(x) = 0$.

They are noted : $\mu_A(x) = \begin{cases} L(\frac{a-x}{\alpha})\, if\, x \leq a \\ 1\, if\, a < x < b \\ R(\frac{x-b}{\beta})\, if\, x \geq b \end{cases}$ for $A = [a, b, \alpha, \beta]$ (see fig.1).

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

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

$A^-$ representing the equation of the half-line $(alpha, a)$ defined by equation $A^-(\lambda) = \lambda \times (m - \alpha) + \alpha$.

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

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

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

*2) Fuzzy Intervals handling:* The extension principle (eq.1), proposed originally by Pr. L.A. Zadeh [6], 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 $\phi$ be an relation between a universe $E$ and $F$. Where $A$ is a definite fuzzy subset of $E$. The principle of extension stipulates that the image by $\phi$ of $A$, is a fuzzy subset $B$ of $Y$ which membership function membership is defined by:

$$\mu_B(y) = sup\{min(\mu_\phi(x, y), \mu_A(x)|x \in E\} \tag{1}$$

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 [8], 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.

### B. Discrete Event System Specification

DEVS formalism [1] can be defined as a universal and general methodology which provides tools to model and simulate systems whose behaviour is based on events. It is based on the systems theory, the concept of component and allows the specification of discrete events complex systems in a modular and hierarchical form. Nevertheless DEVS must be adapted and extended when it is replaced in the specific context of applicability field. DEVS is based on the definition of two types of components: atomic components or atomic model and the composition model or coupled model. These components communicate between them thanks to various types of events.

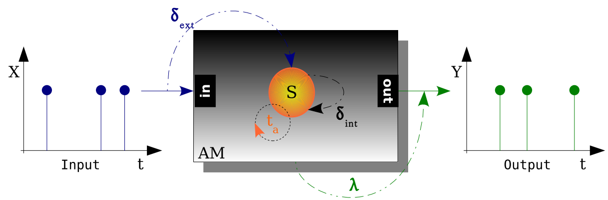


Fig. 2. Behaviour of an atomic model

*1) Atomic model:* The atomic model (AM, fig.2, eq.2) provides an autonomous description of the behaviour of the system, defined by states, by input/output functions, and by internal/external transitions functions of the component. It 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;
- $t_a$: $S \to \mathbb{R}^+$ the time advance function;
- $\delta_{int}$ : $S \to S$ the internal transition function;
- $\delta_{ext}$: $Q \times X \to S$ 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;
- $\lambda : S \to Y$ the output function.

*2) Coupled model:* The coupled model (eq.3) is a composition of atomic models and/or coupled models. It is modular and presents a hierarchical structure, which allows the creation of complex models from basic models. It is described in the form:

$$CM =< X, Y, C, EIC, EOC, IC, L > \quad (3)$$

with :

- $X$ the input ports set;
- $Y$ the output ports set;
- $C$ the set of all component models;
- $EIC$ the external input coupling relation which connects the input ports of the coupled model to one or more of the input ports of its internal components;
- $EOC$ the external output coupling relation which connects the output ports of the internal components to the output ports of the coupled model;
- $IC$ the internal coupling relation which connects the output ports of the internal components to the input ports of other components;
- $L$ the list of priorities between components.

In DEVS each component is independent and can be regarded as a whole entity with of the system, or as the component of a larger system. It is shown in [1], [10] that DEVS formalism is closed under coupling, i.e. that for each atomic or coupled DEVS model, it is possible to build an equivalent DEVS atomic model.

*3) Events:* 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:

$$E = (time; port; value) \quad (4)$$

In formula 4, 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 [1].

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.

*4) DEVS simulation:* Establish a simulation requires the precise definition of behavior and the description of the interactions between entities in the model.

One of the important properties of DEVS formalism is automatically provided a simulator for each of the models. DEVS draws a distinction between the modeling and simulation of a system as any DEVS model can be simulated without the need to implement a specific simulator. Each atomic model is associated with a simulator to manage the component behavior, and each coupled model is associated with a coordinator in charge of time synchronization underlying components. All of these models is managed by a coordinator specifically called *Root* [1].

Each model communicates with the sending and receiving of several messages. The principle is described in [1]. Each message generating events that are listed in a schedule, which is a data structure consisting of events classified according to a chronological order, the head of the timeline representing the immediate future, and tail the more distant future. The simulation is to change the time and cause changes in state foreseen by the events.

DEVS is supported by a large scientific community, there are many extension, the following section presents a first DEVS extension that aims to take into account the fuzzy transitions between states.

*5) Fuzzy-DEVS :* DEVS was developed for the study of electronic systems, and its use in many other areas has led the development of extensions as:

– DSDE [2] or DynDEVS [4] for dynamical systems, that is, systems whose structure changes over time ;
– Cell-DEVS [11] for systems with cellular interface ;

In this section we look specifically at Fuzzy-DEVS, an initial extension of DEVS takes into account the fuzzy transitions between states.

The Fuzzy-DEVS formalism introduced by Pr. Y. Kwon in [3], drift DEVS formalism while keeping its semantics, some of its concepts and its modularity. It is based on fuzzy logic, the "Min-Max" rules and the methods of fuzzification and defuzzification. To allow the simulation, imprecise parameters must be transformed into crisp parameters (defuzzification) ; to be exploited, the output data are again transformed into fuzzy data (fuzzification).

The fuzzy atomic structure of the model described in [3] is : $\tilde{AM_F} =< X,\, Y,\, S,\, \tilde{\delta}_{int},\, \tilde{\delta}_{ext},\, \tilde{\lambda},\, \tilde{t}_a >$

As described Fuzzy-DEVS model does take into account the different possibilities of transition $(\tilde{\delta_{int}}, \tilde{\delta_{ext}})$ between state. The inputs and outputs of the model are not represented as fuzzy. Moreover, the Fuzzy-DEVS atomic model, unlike DEVS atomic model, is not deterministic, which means it does not meet the following two conditions :

1) The internal transition function is launched $(\delta_{int}(s_t) = s_{t+1})$ when the lifespan of the state is passed $(t_a = 0)$ and the external transition function $(\delta_{ext}(s_t, X_t) = s_{t+1})$ is carried out when an external event is received before time is passed.
2) The output function $(\lambda(s_t) = Y_t)$ is launched when the lifespan of a state is finished $(t_a = 0)$.

In Fuzzy-DEVS the following state ($S \leftarrow S_{t+1}$ is not determined with $\delta_{int}$ and $\delta_{ext}$ but with the rule "Max-Min" [3]. The various possibilities of input, output and state update are represented by matrices and the evolution of the model by possibilities trees [3], [12]. Fuzzy-DEVS does not address the fuzzy values of a model, but proposes a methodology that provides a tree of options describing various transitions between states of the system.

Fuzzy-DEVS is a theoretical formalism still in research phase. This approach does not appear fully consistent with the DEVS formalism, but it provides avenues for good work, as the ability to define the lifespan of a state $(t_a)$ with a linguistic label ; but it does not provide an answer to our problems, namely the definition of a method of taking into account the fuzzy at all parameters of the mode

## III. Methodology

Our goal is to provide a tool for modeling and simulation based on the DEVS formalism allowing the study of systems with fuzzy parameters. Therefore, we must make changes to classical DEVS formalism, define a method for modeling fuzzy information and finally define an approach to interpreting the results. To interpret the results we want to use fuzzy logic in a module to decision support.

At the level of representation of the information we have chosen to model in the interval form. These intervals are handled using two techniques from the fuzzy arthmetics; vertex method introduced by Professor Dubois [9] for the basic operation, and the extension principle by professor Zadeh [13].

To enable a fully taken into account fuzzy information at all levels of DEVS formalism is problematic, formalism presented in part II-B.5 does not allow elsewhere. As a result we propose in this part a new method for modeling and simulating systems with fuzzy parameters. From a classical DEVS event 4 we note that inaccurate information can be of two types: fuzzy on the time of the event or its value. The fuzzy value leads to a problem of modeling while the vagueness leads on the time a modeling and simulation problem. The simulation is driven by events, if one does not know the exact time of event simulation does little more progress.

This section deals our modeling and simulation method introduced in [14], [8].

### A. Fuzzy DEVS modeling

Having chosen as a mode of data representation intervals, it must be integrated into DEVS formalism to allow the creation of models to be imprecise parameters. For this we use two approaches. We have defined a new data type "Fuzzy" [15], which allows you to define and manipulate information in the form of an fuzzy interval. This new type of data includes a large number of handling function, vertex method, extension principle, and so on. As well as overload operators to allow bases to conduct operations between different types of data.

The second approach, outlined in [8] is the definition of a DEVS model library for the creation of coupled models from the basic atomic model. We can quote fuzzy addition model, fuzzy subtraction model, fuzzy number generator model.

The use of these two approaches allow a modeler to define its own fuzzy models, as well as to resolve the problem in the modeling. The value of a fuzzy event is represented in the interval form, sent as the model. With this method we can handle imprecise information at the input values of the models and the models in DEVS classic functionality (except $t_a$), the output generated course they are well represented in the form of interval.

DEVS time advance function $(t_a)$ can be represented in the interval form but it leads to a problem in the simulation, because its role is to change the status of the model, and these changes are triggered at the simulation schedule, if the we do not know exactly the execution time there remains much in the same state and simulation fails.

### B. Fuzzy DEVS simulation

The imprecise regarding the time may be involved in the time advance function $(t_a)$, and on the date of execution events, in order to solve the problem of fuzzy simulation time, we propose to use a Expected Existence Measures method (EEM). This method is to calculate the expected existence measures [16], it spend through a phase of defuzzification, defuzzification aims to move from a fuzzy value to a crisp value; several methods can be used, but we have chosen to submit a starting two examples :

1) $C_1$: the value $x_1$ sent to the time $t = 4$, this data is modeled from the interval form $[4, 4, 2, 8]$ and from equation function of the $\lambda$ and function of the time, representing the two interval fronts (fig.3).
   a) $C_1^-(\lambda) = 2 + 2 \times \lambda$ et $C_1^+(\lambda) = 8 - 4 \times \lambda$
   b) $C_1^-(t) = (1/2) \times t - 1$ et $C_1^+(t) = -(1/4) \times t + 2$
2) $C_2$ : the value $x_2$ send to the time $1 < t < 8$, this proposal is modeled in the interval form $[3, 4, 1, 8]$, and in the form of an equation, according to $\lambda$ and function of time, representing two fronts of the interval (fig.4).
   a) $C_2^-(\lambda) = 1 + 2 \times \lambda$ et $C_2^+(\lambda) = 8 - 3 \times \lambda$
   b) $C_2^-(t) = (1/2) \times t - (1/2)$ et $C_2^+(t) = -(1/3) \times t + (8/3)$

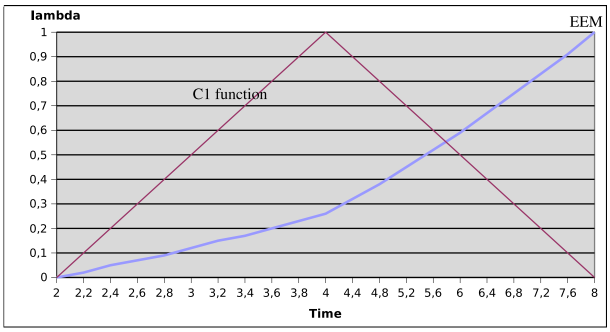


Fig. 3. Membership function C1 and EEM function

EEM takes place in two stages :

1) The first step we calculate a integral report from the function representing the interval between $\alpha$ and $t$ ($\int_\alpha^t$), and between $\alpha$ and $\beta$ ($\int_\alpha^\beta$). This integral report $(\frac{\int_\alpha^t (C_n)}{\int_\alpha^\beta (C_n)})$

is shown by the function $C_1$ and $C_2$: EEM (fig.3 and 4), it represents the expected existence measures.
2) The second step is the definition of a trust factor between 0 and 1, 0 event has not yet taken place, 1 it occurred. We seek the value of the interval as $EEM = coef$.

It enables us to add to this defuzzification technique a coefficient decision support, and if it is defined very small means that we seek to trigger the event early, if we want to be sure that the event has taken instead, we can set a higher coefficient. Indeed over the coefficient is close to 1 and the greater the chance that the event has occurred. This method does not currently refuzzification, but as the others it can maintain the level of validity of the condition: $\lambda$.

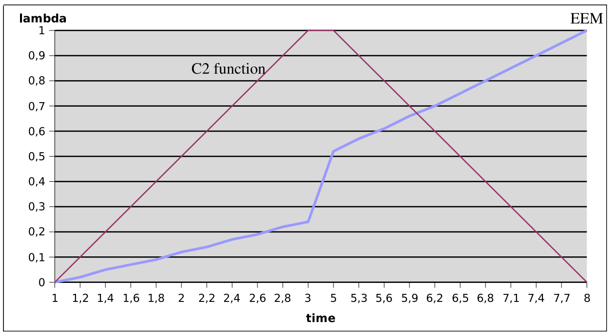


Fig. 4. Membership function C2 and EEM function

For a fixed coefficeint to $(coef = 0.5)$ :

$C_1 : \begin{cases} t = 5.5 \\ \lambda = 0.62 \end{cases}$ and $C_2 : \begin{cases} t = 4.9 \\ \lambda = 1 \end{cases}$ (show fig.3 and 4)

This method is added to our data type, one can only add in DEVS formalism and modify the basic atomic model to reflect our changes.

The problem is related to time, we changed the time advance function $(t_a)$ she did not return to crisp time. So we no longer have a problem with fuzzy simulation time, it is resolved in the design of the model and therefore the modeling. Inside the model, time can be defined as an interval but it will be defuzzifier before being sent to the simulator. Function $(t_a)$ was amended to test if time to return is accurate or inaccurate, in the second case, the time has changed. We do not see the function $(t_a)$ simply as a function which defines the lifespan of a state, but as a feature that lets you set the time of execution of internal events. If the events come from a generator, so a model atomic be managed with the function $(t_a)$ fuzzy values on the timing of events, such management requires the definition of a fuzzy atomic model; the difference between the function $(t_a)$ DEVS classical model and function $(t_a)$ DEVS fuzzy model remains transparent to the final user.

In the DEVS fuzzy model (cf.5) we apply with the function $(t_a)$ a defuzzification function (EEM), and we save the validity degree of the condition (of the orderly function of belonging to the interval: $\lambda$) form coefficient, average $(\lambda)$ defuzzifier. This new variable can be stored as sytem state variable $(\tilde{S})$ or added as a class variable in the class atomic model, it is the same for the time interval representative at the end of the simulation.

### C. *Fuzzy atomic model*

Our fuzzy atomic model is shown below:

$$\tilde{AM} :< \tilde{X}, \tilde{Y}, \widetilde{S}, \widetilde{t_a}, \widetilde{\delta_{int}}, \widetilde{\delta_{ext}}, \widetilde{\lambda} > \qquad (5)$$

with:

- $\tilde{X}$ and $\tilde{Y}$ : the input and output ports, receive and send the fuzzy intervals;
- $\widetilde{S}(tps, coef)$ : The states sets of the system, the state variables are fuzzy.
- $\widetilde{t_a}$: The time advance function;
  *If* $\sigma$ is crisp // We test the type of $\sigma$.
  $t_a = \sigma$
  $tps+ = \sigma$ // We increment the state variable *tps*.
  *Else*
  $t_a = EEM(\sigma)$ // We apply a defuzzification method (cf.III-B) on the time advance function.
  $coef = \lambda(EEM(\sigma))$ // We keep the validity coefficient $\lambda$.
  $tps+ = \sigma$ // We increment the state variable tps .
- $\widetilde{\delta_{ext}}$ : The external transition function, update the state.
- $\widetilde{\delta_{int}}$: The internal transition function, update the state if $t_a = 0$.
- $\widetilde{\lambda}$ : The output function, send the result on $\tilde{Y}$.

## IV. Example

To correctly place wind turbines must be studied before the area. The location is important for profitability.

This example shows the use of our defuzzification method on measurements of wind speed.

Almost everywhere in the world, the wind blows stronger the day that night. Data in the first column "DEFAULT" of table 5, are stored measurements over a 24-hour three-hour intervals through a sensor-type wind anemometer.

| DEFAULT | | COEF = 0.2 | | COEF = 0.4 | | COEF = 0.6 | | COEF = 0.8 | |
|---|---|---|---|---|---|---|---|---|---|
| time | wind | time | wind | time | wind | time | wind | time | wind |
| 0 | 3.4 | 22.2 | 3.4 | 23.4 | 3.4 | 0.6 | 3.4 | 1.8 | 3.4 |
| 3 | 3.4 | 1.2 | 3.4 | 2.4 | 3.4 | 3.6 | 3.4 | 4.9 | 3.4 |
| 6 | 4 | 4.2 | 4 | 5.5 | 4 | 6.7 | 4 | 7.9 | 4 |
| 9 | 4.8 | 7.3 | 4.8 | 8.5 | 4.8 | 9.7 | 4.8 | 10.9 | 4.8 |
| 12 | 5.4 | 10.3 | 5.4 | 11.5 | 5.4 | 12.7 | 5.4 | 13.9 | 5.4 |
| 15 | 5.2 | 13.3 | 5.2 | 14.5 | 5.2 | 15.7 | 5.2 | 16.9 | 5.2 |
| 18 | 4.6 | 16.3 | 4.6 | 17.4 | 4.6 | 19.6 | 4.6 | 20.8 | 4.6 |
| 21 | 3.9 | 19.2 | 3.9 | 20.4 | 3.9 | 21.6 | 3.9 | 22.8 | 3.9 |
| Result validity | | 0.4 | | 0.81 | | 0.79 | | 0.39 | |

Fig. 5. Wind measure and simulation results

This information comes from the European Wind Atlas.

We use our method on a faulty wind sensor, anemometer type. It sends data at times inaccurate, may meadows every 3 hours more or less three hours around. In order to have a precise value for progress in the simulation we have defuzzifier a time membership function (fig.6).

We present in the table 5 the results from several validities coefficient (0.2, 0.4, 0.6, 0.8). With a validity coefficient large or small, which corresponds to the execution of the event early, respectively very late, we have little chance that the value is correct. But in some cases, such as fire spread, it may

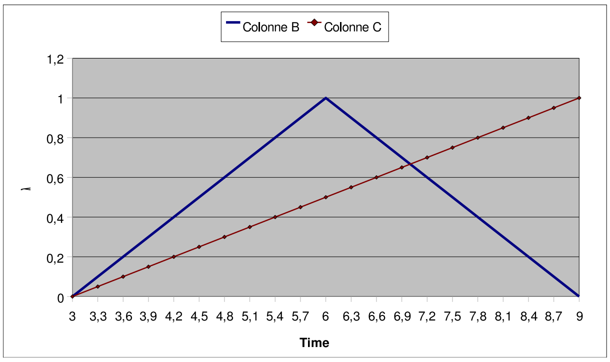


Fig. 6. Time membership function for the time 6, and EEM function.

be interesting to trigger events very early to see the different scenarios spread.

In this example functions are symmetrical, so this is where the validity coefficient is close to 0.5, we get the results most likely.

It should to add this method a tool for decision support. It will help to choose the good validity coefficient depending to the needs or the field and to interpret the results.

## V. Conclusions

This paper presents our work on the modeling and simulation of system with inaccurate parameters. We have heightened our communication on simulation part (modeling part [17], [8]), presenting the problems associated to the simulation event with an inaccurate time. To solve this problem we propose to add a defuzzification method in a DEVS atomic model. This method allows defining a new type of DEVS atomic model which takes into account the inaccurate parameters without having to change the simulation party of DEVS formalism. The final section presents a sample application in the measurement of wind speed.

Our approach is generic; in order to improve we are working on several tracks. After integration in the DEVS formalism the fuzzy sets theory, to deal with inaccurate parameters, we drive our works to the possibilities theory to deal uncertain and inaccurate parameters. The inaccuracy is relative to the value of an event, and the uncertainty in its occurrence.

Furthermore, as it was shown in the last part, we want to improve our approach by providing a tool for decision support; this, in order to assist in the results interpretation. This part has to be seen in the context of a pluridisciplinary project in the domain of computers which aims to integrate DEVS formalism and several other modeling techniques such as, SMA, the SIG, and WEB services, so as to validate a software environment of modeling and simulation of complex (dynamic and/or fuzzy) spatialized systems.

## References

[1] B. P. Zeigler, *Theory of Modeling and Simulation*. Academic Press, 1976.

[2] 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.

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

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

[5] A. Hamri, N. Giambiasi, and C. Frydman, "Min-max devs modeling and simulation," *Simulation Modelling Practice and Theory (SIMPAT)*, vol. 14, pp. 909–929, October 2006. Ed. Elsevier, ISSN 1569–190X.

[6] L. Zadeh, "Fuzzy sets," *Information Control*, vol. 8, pp. 338–353, 1965.

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

[8] P.-A. Bisgambiglia, L. Capocchi, E. de Gentili, and P. Bisgambiglia, "Manipulation of incomplete or fuzzy data for DEVS-based systems," pp. 87–92, International Modeling and Simulation Multiconference (IMSM) - Conceptula Modeling Simulation (CMS), 2 2007.

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

[10] H. Vangheluwe, "The Discrete EVent System specification DEVS Formalism," tech. rep., 2001. http://moncs.cs.mcgill.ca/people/hv/teaching/MS/COMP522A2004/.

[11] A. Troccoli and G. Wainer, "Implementing parallel cell-DEVS," in *Proceedings of the 36th Annual Simulation Symposium* (IEEE, ed.), 2003.

[12] A. Anglani, P. Caricato, A. Grieco, F. Nucci, A. Matta, G. Semeraro, and T. Tolio, "Evaluation of capacity expansion by means of fuzzy-devs," (citeseer.ist.psu.edu/499458.html), 14th European Simulation MultiConference, May 2000. Ghent, Belgium.

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

[14] P.-A. Bisgambiglia, E. de Gentili, J. Filippi, and P. Bisgambiglia, "DEVS-Flou: a discrete events and fuzzy logic-based new method of modelling," SCSC SummerSim, 7 2006.

[15] P.-A. Bisgambiglia, E. de Gentili, P. Bisgambiglia, and J. Santucci, *Modélisation floue basée sur le formalisme DEVS et sur la méthode des fronts*. cépaduès editions ed., 11 2007.

[16] 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*, vol. 2, pp. 923–928, 2000.

[17] P.-A. Bisgambiglia, E. de Gentili, J. Santucci, and P. Bisgambiglia, "DEVS-Flou: a discrete events and fuzzy sets theory-based modeling environment," pp. 95–100, Systems and Control in Aerospace and Astronautics, (ISSCAA), 1 2006.

**Paul-Antoine Bisgambiglia** born on 14 September 1981 at Peri in Southern Corsica. Graduates from the University of Corsica - Research Master of Computer Science option Integrated Information Systems, currently in the third year doctorate in modeling and simulation team. IEEE Student member since 2007.

**Emmanuelle de Gentili** was born in Bastia and is Associate Professor in Computer Sciences at the University of Corsica. She holds a PhD in computer science. Her research activities concern the techniques of modelling and simulation of complex systems and the Fuzzy Logic. She makes his researches in the laboratory of the UMR CNRS 6134.

**Paul Antoine Bisgambiglia** is a professor in Computing Sciences at the University of Corsica. He is responsible of the modelling and simulation team of the UMR CNRS 6134. His research activities concern the techniques of modelling and simulation of complex systems and the test of systems described at high level of abstraction. He makes his researches in the laboratory of the UMR CNRS 6134.

**Jean-François Santucci** is Professor in Computer Sciences at the University of Corsica since 1996. His main research interests are modelling and simulation of complex systems. He has been author or co-author of more than 100 papers published in international journals or conference proceedings. He has been the scientific manager of several research projects corresponding to European or industrial contracts. Furthermore he has been the advisor or co-advisor of more than 20 PhD students and since 1998 he has been involved in the organization of more than 10 international conferences.