

Cognitive Simulation-Based on Knowledge Evolution in Fuzzy Discrete Event Systems

Paul-Antoine Bisgambiglia
University of Corsica and
CNRS UMR SPE 6134
Quartier Grossetti, Bat 018
France, Corte 20250
Email: bisgambiglia@univ-corse.fr

Paul Antoine Bisgambiglia
University of Corsica and
CNRS UMR SPE 6134
Quartier Grossetti, Bat 018
France, Corte 20250
Email: bisgambi@univ-corse.fr

Jean-Sebastien Gualtieri
University of Corsica and
CNRS UMR SPE 6134
Quartier Grossetti, Bat 018
France, Corte 20250
Email: gualtieri@univ-corse.fr

Abstract—Discrete event systems (DES) theory has been developed and widely used to study the behavior of complex systems. In this paper, from a formalism based on this theory (DEVS), we propose to present a formal approach for taking into account the knowledge evolution. Indeed, knowledge is not static; it can evolve in time, or during the simulation. Few tools offer to take into account these developments, which may have a significant impact on the validity of results. To deal this knowledge, we suggest adding a lifespan to the modeled data, the latter is used to define the validity degree of knowledge; it is a kind of confidence measure.

Keywords—DES; FDES; DEVS; fuzzy sets; knowledge; modeling; simulation

I. INTRODUCTION

In this article we will present works that aim to improve the knowledge consideration in numerical models.

The process of modeling and simulation (M&S) can be divided into three phases (Figure 1):

- (1) the acquisition of knowledge about the system, and proposing hypotheses;
- (2) model the system based on the proposed assumptions to simplify it and make it executable by computer;
- (3) simulate the model to validate and confirm the hypotheses.

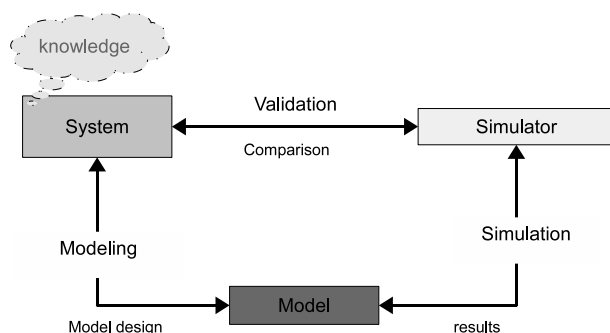


Figure 1. Modeling and simulation (MS) process

These three phases are more or less based on the need to correctly represent, i.e. in a legible and understandable, the

knowledge. This knowledge can come from field observations and laboratory. They can be inferred or learned from an expert, etc. It is therefore important to identify any issues arising. What types of knowledge are needed to realize a good model? How to represent and express them? How to take into account their evolution?

Knowledge can be defined as a system for explaining the relationship between thoughts and objects. Knowledge representation amounts to establishing a correspondence between the studied system and a symbolic representation that can be processed by a computer: our model (see Figure 1). The cognitive sciences are the source of the work on knowledge; they have contributed to the development of multiple theoretical approaches [1]. Philosophers, psychologists, linguists, sociologists, computer scientists, mathematicians seek to analyze and represent knowledge. The ambition is to allow to the computer to use when calculating the same type of information that man. The difficulty of validating the information gained is the fact that we do not control the knowledge of an individual but an expression of that knowledge. The representation problem arises and causes problems of understanding, representation, expression and communication. This problem has led us to use fuzzy logic [2], [3], [4] because it allows to represent and manipulate imperfect knowledge (imprecise and uncertain). Methodological tools it offers can take into account the information so gradually, expressed in natural language, that is to say, very close to the mode of reasoning and expression of the human being. These properties are very interesting.

Recently, to deal imperfect knowledge in our models, we proposed several extensions that couple the computational intelligence methods with the DEVS formalism. iDEVS [5] is based on fuzzy set theory [2], it can simulate systems with imprecise parameters. DEVFIS to represent the FIS as a DEVS model [6]. Articles [7], [8] introduce a high-level language, based on the MDA approach and fuzzy logic for the specification of DEVS models. The experience gained during this work led us to think, and see, differently, some stages of the knowledge representation, and in particular the modeling phase. Generally modeling is defined as a step for creating a simplified representation (equation, experience, black box) of the studied system: a model (Figure 1). (1) A model is

an accurate or simplified description of the system in its environment. This definition is most commonly used. It is illustrated in Figure 1, which details the steps of modeling and simulation.

We want to propose an alternative definition, more focused on cognition: (2) “a model does not describe the system, but a state of knowledge, knowledge gained from the study or observation of system”. This interpretation seems more consistent with the presented knowledge definition. The work presented thereafter, will be based on the definition (2). In this paper, our objectives are to introduce the concept of cognitive simulation, and explain and elaborate our modeling approach. We can define cognition as a process of complex information, such as reasoning, decision making, etc. A cognitive process is to acquire information about its environment, and relates everything to do with knowledge. Unlike a classical simulation, which evolves over time a model based on static knowledge, we propose the concept of cognitive simulation [9], [10], [11]: it aims to take into account the knowledge evolution over time. A state of knowledge may vary depending on the expert who utters (interpretability problem), it can also vary over time and become less relevant (temporality or confidence problems), it is knowledge that we can describe as dynamics.

In the first section, we present the general framework of the discrete event systems (DES) theory, then the DEVS formalism. In the second part, we detail the mechanisms to implement to apply our approach to cognitive simulation. Before concluding and presenting the limitations and perspectives of our work, in the last part, we will describe a sample application.

II. DISCRETE EVENT MODELING

Modeling and simulation (M&S) have emerged as essential tools for analyzing the behavior of complex systems. Several methods have been proposed to improve the analysis process. These proposals attempt to achieve more realistic models, relatively simple and highly flexible. Our interest focuses on DEVS (Discrete Event System Specification [12]) is a formalism for modeling systems based on the DES theory [13].

A. DES and FDES

A Discrete Event System (DES) is a discrete-state, event-driven dynamical system in which the state space is described by a discrete set, and states evolve in terms of asynchronous occurrence of discrete events over time [13]. In DES theory, states, events and transition functions have crisp values and they are defined as: a DES can be modeled by a five-tuple [13], [14], [15]:

$$G = (S, \Sigma, \delta, \lambda, s_0) \quad (1)$$

where:

- S : is the set of states;
- Σ : is the set of events containing detectable (i.e. an event is detectable if it produces a measurable change in the system output) and undetectable events, which are generally fired asynchronously;

- $\delta : S \times \Sigma \rightarrow S$ is the state transition function;
- $\lambda : S \times \Sigma \rightarrow \Sigma_d$ is the output function, where Σ_d and Σ_{ud} are the set of detectable and undetectable events, respectively $\Sigma = \Sigma_d \cup \Sigma_{ud}$;
- and, s_0 is the initial state vector whose elements are zero or one.

Actually, DESs represent many technological and engineering systems such as communication networks, computer networks, manufacturing systems, transportation systems, natural systems, and more. So far, many modeling approaches of DESs have been proposed and developed, including the notably finite automata [13], [16], fuzzy automata [17], [18], Petri nets [13], [19], DEVS formalism [12].

DES theory is used to model systems that cannot be described by differential or difference equations. However, dynamic models that cannot be easily represented by the ordinary and classical approaches might be modeled by sequences of events recording significant qualitative changes in the state of the system. Although DES theory has been applied in many engineering fields, it is not adequate for all fields. Recently, in order to make it possible to effectively represent deterministic uncertainties and vagueness as well as human subjective observation and judgment inherent to many real-world problems especially those in complex systems, several works introduce the concept of fuzzy states and fuzzy events, and generalize (crisp) discrete event systems (DES) to fuzzy DES (FDES). Fuzzy discrete event systems (FDESs) were proposed recently by [20], [14], [21], which may better cope with the real-world problems with fuzziness, impreciseness, and subjectivity such as those in biomedicine.

In summary in [20], [14], [21], a fuzzy DES is represented by a fuzzy automaton:

$$\tilde{G} = (\tilde{S}, \tilde{\Sigma}, \tilde{\delta}, \tilde{s}_0) \quad (2)$$

where $\tilde{S}$, $\tilde{\Sigma}$, $\tilde{\delta}$, $\tilde{s}_0$ are fuzzy extensions of the crisp S , Σ , δ , s_0 respectively. Formally:

- $\tilde{S}$ is a set of fuzzy states $\tilde{s}$, where $\tilde{s}$ is represented by a vector over $[0, 1]$;
- $\tilde{\Sigma}$ is a set of fuzzy events $\tilde{\alpha}$, where $\tilde{\alpha}$ is represented by a matrix over $[0, 1]$;
- $\tilde{\delta} : \tilde{S} \times \tilde{\Sigma} \rightarrow \tilde{S}$ is a transition function, where $\tilde{\delta}(\tilde{s}, \tilde{\alpha}) = \tilde{s} \circ \tilde{\alpha}$;
- and, $\tilde{s}_0$ is the initial state.

The integration of fuzzy information and the definition of FDES is a plus to improve the knowledge processing. This is a first step towards defining approaches to cognitive simulations. However, data to be processed (imprecise or uncertain) can occur at all levels of the system (see equation 1), their proper representation and their inclusion is a major issue. Despite this, most approaches only take into account that the vagueness or uncertainty of the states and/or system events. Inputs and the knowledge evolution over time are often omitted. We propose to extend the DEVS formalism for the inclusion of such information.

B. DEVS formalism

For several years our team working on the DEVS formalism [22], we work to develop and expand this M&S tool. This formalism provides a formal foundation to M&S that combines the advantages of a simulation-based approach with the rigor of a formal methodology. DEVS formalism is based on a mathematical paradigm with well-defined concepts of coupling of components, hierarchical, modular model construction, support for discrete event approximation of continuous systems and support of object-oriented paradigms. The DEVS formalism is based on the definition of two types of components: atomic models and coupled models.

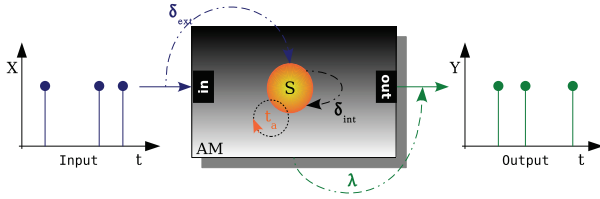


Figure 2. Behavior of atomic DEVS model

The atomic model (eq. 3) provides an autonomous description of the behavior of the system, defined by states, input/output functions and transition functions (see Figure 2). 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.

Atomic DEVS model:

$$AM :< X, Y, S, t_a, \delta_{ext}, \lambda, \delta_{int} > \quad (3)$$

where:

- X : is the set of input events, is characterized by a couple (port, time, value), where the port means the input on which the event occurs, the time is the date of occurrence of the event, it is blank for internal events, and the value symbolizes the data from the event;
- Y : is the set of output events;
- S : is the set of partial or sequential states, which includes the state variables;
- $t_a : S \rightarrow T^\infty$: is the time advance function which is used to determine the lifespan of a state;
- $\delta_{ext} : Q \times X \rightarrow S$: is the external transition function which defines how an input event X changes a state of the system, where $Q = \{(s, t_e) | s \in S, t_e \in (T \cap [0, t_a(s)])\}$ is the set of total states, and t_e is the elapsed time since the last event;
- $\lambda : S \rightarrow Y^\iota$: is the output function where $Y^\iota = Y \cup \{\iota\}$ and $\iota \notin Y$ is a silent event or an unobserved event. This function defines how a state of the system generates an output event, when the elapsed time reaches to the lifetime of the state;

- $\delta_{int} : S \rightarrow S$: is the internal transition function which defines how a state of the system changes internally, when the elapsed time reaches to the lifetime of the state.

Every state S is associated with a lifetime t_a , which is defined by the time advance function. When an event receives an input event X , the external transition function δ_{ext} is triggered. This function uses the input event, the current state and the time elapsed since the last event in order to determine what the next model's state is. If no events occur before the time specified by the time advance function for that state, the model activates the output function λ (providing outputs Y), and changes to a new state determined by the internal transition function δ_{int} .

The DEVS models are executed by abstract simulators that are independent from the models themselves. Consequently, separated concerns between models and implementations of simulation can be achieved and enhance the verification of each layer independently. DEVS is a popular method to simulate a variety of systems. However, since its introduction by Zeigler in [12], significant efforts were taken to adapt this formalism to different fields and situations. The many proposed extensions [23], [24], [25], [26], [5], [6], proved its ability to extend and openness. In this large niche, we look towards the issue of knowledge evolution over time.

C. Proposal

As for the extension from DES to FDES, there are extensions to the DEVS formalism to deal fuzzy data, but they only address the uncertainty associated with events or states transitions [23], [26]. As used, the models are the result of the expression of imperfect knowledge and static over time. As we present them, they are not based on knowledge of an individual but on the expression of such knowledge, that is to say, on a state of knowledge, state variable over time.

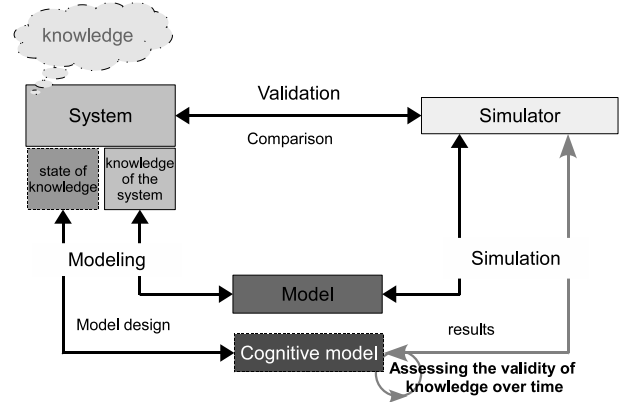


Figure 3. Integration of our cognitive model in MS process

Using Figure 3, we want to introduce the differences between the classic approach of M&S, presented in the introduction, and our approach. Between the cognitive model and the simulator, we added an evaluation mechanism that can evolve during the simulation the validity of the model parameters (eq. 3). Our approach makes sense, if we consider

that to give correct results must be taken into account in the models, the evolution and validity of data.

For example, a transport company that uses a predictive model to estimate the cost of its investments will not fix the price of a Brent barrel in advance. The evolution of the price of Brent is greatly influenced by complex factors: consumption, geopolitics, etc. Consider its price as imprecise can execute the model taking into account possible fluctuations in the price. Failing to have an accurate result, from our method, the company may have a tool for decision-making. Our proposal does not provide accurate results, it even adds to the imprecision of knowledge, but the results thus obtained allow having a broad estimate that has a good chance of being correct. We can consider this as a tool for decision support.

III. DEVS COGNITIVE APPROACH

From the DEVS formalism, which is based on the DES theory, we propose a cognitive simulation approach to knowledge representation whose validity may decrease over time. Such data or information are not based on supposed knowledge of the expert, but are inferred from the knowledge expression. They are subjective, subject to human interpretation, and their relevance may vary. The information we want to take into account, can be likened to adding a confidence measure in the expert knowledge.

A. Cognitive DE S models

We start from a simple observation, knowledge expressed and validated at a time t can, and is often the case; loses its relevance over time. To represent this loss of information, we propose to define a mechanism to data control and integrate it into the atomic DEVS model. Our goal is to extend the DEVS formalism to take into account what we call dynamic data. During the simulation, as data may lose their relevance and precision, we add a knowledge evaluation function called (Γ), during the simulation, it assessed the validity of data models and transforms some data to imprecise interval. This transformation, which can be seen as a fuzzification, will be performed by a new function added in the atomic DEVS model.

B. iDE S method

The integration mechanism is based on our previous work, they aim to represent and manipulate imprecise data in DEVS models [27], [5]. The mechanism is simple and intuitive; we defined a new data type (a class), which allows to represent and to manipulate a fuzzy interval (Fuzzy Set) as an integer or a float. It is possible to perform calculations: basic operations (+, -, *, /) or implement functions (sin, cos, log). With the advantages of object-oriented programming (OOP), the use of our data type is transparent to the user. For example, let an integer A , and a fuzzy interval B , $A+A$ = an integer, $A+B$ = a fuzzy interval, $B+A$ = a fuzzy interval, and $B+B$ = a fuzzy interval. This is possible thanks to operator overloading to the right and left. Our data type is used to represent imprecision in various forms described in Figure 4.

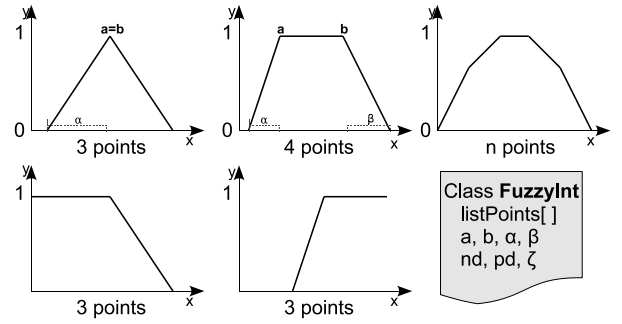


Figure 4. Example of imprecise data described from our class

Our data structure consists of a list of point (type (x, y)), 4 points defining the kernel (a and b) and the support (ψ and ω). We also added the left and right radius: $\alpha = a - \psi$ and $\beta = \omega - b$, and a linguistic label. Added to this, different coefficients to represent the validity of the information (pd), its necessity (nd), and so on. We also added a temporality variable (lifespan τ) which is evaluated and used to define the validity degree. The addition of this new information can describe the state of knowledge.

C. Operation method

We offer two application modes:

(1) a classic data (crisp) will be transformed into imprecise data (fuzzy interval), and we add a validity coefficient, a coefficient of necessity, and a lifespan. Interval support ($\omega - \psi$) or fuzzification coefficient (radius) is fixed by the user. All data are transformed into an object of *FuzzyInt* type, in order to add our coefficients. During the first transformation, there is really no fuzzification, that is to say, that we do not add imprecision, the aim of the transformation is to add the coefficients, e.g. a value $A = 5$, will be transformed into interval $I_A = [5; 5; 5; 5]$, then depending on the validity degree and the simulation time, the bounds of the interval I_A will grow;

(2) imprecise data is tested and evaluated. Depending on its lifespan, its validity coefficient is decreased, and its support is increased (see Table I). In conclusion, we add the imprecision when data validity is low.

Table I
PROPOSED CHANGES IN THE VALIDITY DEGREE BASED ON THE SIMULATION TIME

Simulation time (%)	0 to 40	50 to 60	70 to 100 %
Validity degree	1 to .6	.5 to .4	.3 to 0.0
Fuzzification data	$n \times 1$	$n \times 2$	$n \times 3, a-1, b+1$

D. Cognitive function

These two modes are grouped in a function added in our atomic DEVS model:

For state (S):

If S is imprecise, then we evaluate the lifespan of S compared to simulation time (lifespan equal simulation time minus current time), and we update the validity degree and the interval radius with the correlation table (Tab I).

Else, we transform S (fuzzification) using data from the user; the default fuzzification radius is 1.

For input (X):

If X is imprecise, then we evaluate the lifespan of X compared to simulation time, and we update the validity degree and the interval radius with the correlation table (Tab I).

Else, we transform X (fuzzification) using data from the user; the default fuzzification radius is 1.

Algorithm 1 Cognitive function algorithm

```

 $\Gamma(Q, X)$ :
  N = Q (state) or X (input);
  if type(N) == fuzzy interval:
    N.update(N.listPoints = newList, N.to = SIM_TIME -
c_Time, N.pd = tab[N.to]) ;
  else
    N.fuzzification(data_user);
end;
```

In Table I we present an example of evolution of the state of knowledge. When the simulation advances of 10 %, decreases the validity degree of 0.1. Depending on the data validity, we increase the imprecision. It is the user to give the fuzzification data. By default, if the validity degree is above 0.7, we do not change; between 0.5 and 0.6 we double the interval radius, and under 0.4, we are tripling the interval radius and we are moving from a triangular representation ($a = b$) to a trapezoidal representation ($a \neq b$).

E. Integration into DE S

Our cognitive function may be launched either in simulation algorithms, either directly in the atomic models. Its integration in the simulation part implies modification in the simulation algorithms. The method loses its genericity, because it will only work with a specific algorithm. In this case, add a line of code to run this function before launch the external transition function (δ_{ext}). We chose to integrate in the modeling part, at the level of atomic models. The function is called and then executed in the code of the external transition function. This is the first thing that must be δ_{ext} . Thus, it is not necessary to modify the simulation algorithms, and our method can be deployed in existing DEVS environments. So we keep the genericity of the approach.

F. Atomic Cognitive DE S model

Our atomic cognitive model has the same features as the classic atomic DEVS model. Almost all functions have the same role, and the same behavior. Level simulation algorithms, their sequence is identical. The equation 4 describes our atomic model. The elements identified as being imprecise are described by a tilde. The additional element that appears is the function Γ , it defines the knowledge behavior over time.

Atomic cognitive model:

$$AcM : < \tilde{X}, \tilde{Y}, \tilde{S}, t_a, \delta_{ext}, \Gamma, \lambda, \delta_{int} > \quad (4)$$

with $\tilde{X}, \tilde{Y}$ and $\tilde{S}$ are a fuzzy extensions of the crisp X, Y, S , and where:

- $\tilde{X} \supset X$: is the set of input events, is characterized by a couple (*port*, *time*, *value*), where:
 - the *port* means the input on which the event occurs;
 - the *time* is the date of occurrence of the event, it can be imprecise, that is to say, defined as an object of *FuzzyInt* type. It is blank for internal events;
 - and, the *value* symbolizes the imprecise data from the event;
- $\tilde{Y} \supset Y$: is the set of output events, if one element of the model is imprecise, the outputs are imprecise;
- $\tilde{S} \supset S$: is the set of partial or sequential states, which includes the state variables. States are not directly imprecise, the state variables can be of *FuzzyInt* type;
- $t_a : S \rightarrow T^\infty$: is the time advance function which is used to determine the lifespan of a state. The lifetime of the state may not be accurate, in this case solutions have been proposed in [27], [28], [5];
- $\delta_{ext} : Q \times \tilde{X} \rightarrow \tilde{S}$: is the external transition function which defines how an input event $\tilde{X}$ changes a state of the system. In our model δ_{ext} has the same objectives and a new role, which is to trigger the function Γ ;
- Γ : is our cognitive function, it allows to control and assess the confidence we have in information (see 1);
- $\lambda : \tilde{S} \rightarrow \tilde{Y}$: is the output function. This function defines how a state of the system generates an output event, when the elapsed time reaches to the lifetime of the state. In the case of imprecise data, it is used to send additional information such as membership degrees;
- $\delta_{int} : \tilde{S} \rightarrow \tilde{S}$: is the internal transition function which defines how a state of the system changes internally, when the elapsed time reaches to the lifetime of the state.

This model takes into account imprecise data, it is by no means a general proposal detailing the fuzzy DEVS model. To generalize it to fuzzy information (imprecision and uncertainty), one should take into account the uncertainties associated with the transition functions, the state changes, and events.

IV. CASE STUDY

To illustrate the interest of our approach, we propose to present a pedagogic example of application. We will present the difference in results between a traditional model and a model that performs our cognitive function.

The system is simple, a random generator (I) sends digital information (time, value) to two summer (adder) (A and B) responsible for adding the input value with a value stored in state variables. The summer A has a classical behavior, B run our cognition function. Two output models (O_1 and O_2) to print the results. In Model A, there is no distinction between knowledge of the system and state of knowledge. By cons, this is the case in Model B. For the latter, knowledge defined the

model behavior, that is to say, the addition of an input and a state variable. The state of the knowledge is characterized by the fuzzification of the value of state variable.

Table II
MODELS A AND B SIMULATION RESULTS

time	input	Output A	Output B
0	0	0	[-2; 0; 0; 2]
1	5	5	[2; 5; 5; 8]
8	2	7	[2; 7; 7; 12]
11	6	13	[5; 13; 13; 21]
27	3	16	[4; 15; 17; 28]
32	7	23	[7; 21; 25; 39]
40	2	25	[5; 22; 28; 25]
45	8	33	[9; 29; 37; 57]
50	5	38	[10; 33; 43; 66]

Table II presents the simulation results of the models A and B. We can see that the vagueness increases much during the simulation; this is due to the loss of data validity.

Model A makes only an addition of the input value with the value stored as a state variable. For example, at time 27: $13+3 = 16$.

Before each addition, model B evaluates the lifespan of its state variable, and updates its validity degree, then it converts input value to imprecise variable and finally, adds its imprecise state variable with the imprecise input data. For example, at time 27: state model is [5;13;13;21], $pd=1$, $\tau=30$, after its evaluation, we get [2;12;14;24], $pd=0,3$ and $\tau=3$. The input data is transformed (fuzzification) [2;3;3;4] the lifespan is 23 (Simulation final time minus current time 50-27), validity degree is set at 1. The result is well [4;15;17;28], $pd=1$ ($\max(0,3;1)$) and $\tau=3$ ($\min(3;23)$).

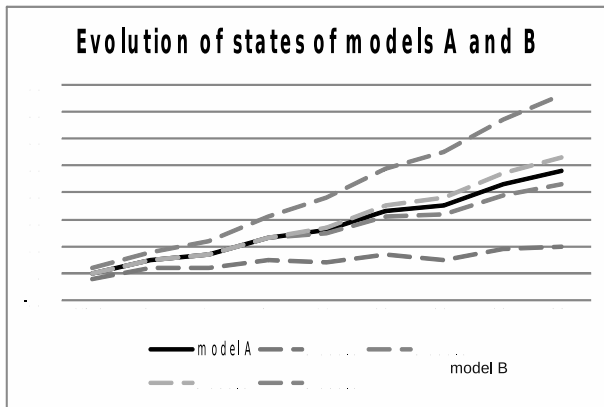


Figure 5. Models A and B state variables evolution

For this example, we chose to use the max operator for operations between validity degrees, and the min operator for the operations between lifespan. Curve 5 shows that the imprecision increases over time, and the results obtained from the classical model are bordered by the results of the cognitive model.

We are aware that the proposed case study is limited, its purpose is to validate the theoretical approach on a simple and concrete case. As shown by the example, and as one might expect, over time the imprecision of the data increases. All the advantages of this method are not clearly exposed from this application, we will soon propose other implementations. We can still highlight the originality of our approach, few studies focus on the impact of knowledge evolution over time, and therefore on the simulation results.

V. CONCLUSION

In this paper, from a formalism based on DES theory (DEVS) and a method for modeling imprecision (iDEVS), we presented a formal approach for taking into account the knowledge evolution: from our point of view, knowledge is not static, it can evolve in time, or during the simulation. To take into account this knowledge, we added a lifespan to the modeled data, the latter is used to define the validity degree of knowledge. It is a kind of confidence measure.

This work has as main objective to take into account the evolution over time of the knowledge validity, that is to say, a state of knowledge and a confidence measure. A mechanism of control and fuzzification was added at the atomic models of DEVS formalism (Atomic Cognitive DEVS model). The integration of this mechanism can respond to our problems while retaining all the properties of the DEVS formalism.

This work complements our research in the fields of FDES. They should be seen in a broader context, which aims to develop a multi-modeling framework for the study of complex systems, and more particularly a fuzzy toolbox for DES [29], [6].

Our approach is in validation phase, and it has to be improved to take into account to the problems of combinatorial explosion associated with the use of some mathematical operators in interval arithmetic. Another avenue of reflection is to study the impact of the operator used in operations between the validity degrees (pd). We propose three methods max, min, average, but we have not evaluated their impact in real situations. Finally, returning to Figure 3, and complete our reflection, we believe that the traditional model may be regarded as derived from necessary knowledge, and not necessarily sufficient. The cognitive model rather describes a possible behavior of the system. This reflection allows us to do a parallel with the possibility theory [30], [31], [32], and a link between possibility measure and necessity. We plan thus to complete our approach from the development of possibility theory.

REFERENCES

- [1] B. Russell, *Introduction to Mathematical Philosophy*. 1993.
- [2] L. Zadeh, "Fuzzy sets," *Information Control*, vol. 8, pp. 338–353, 1965.
- [3] L. A. Zadeh, "Fuzzy logic," *Computer*, vol. 21, no. 4, pp. 83–93, 1988.
- [4] D. Dubois and H. Prade, "Fuzzy set in approximate reasoning," *part 1, Fuzzy set and Systems*, vol 40, 1993.
- [5] P-A. Bisgambiglia, E. de Gentili, P. Bisgambiglia, and J-F. Santucci, "idevs: new method to study inaccurate systems," in *Proceedings of the The Annual IEEE International Conference on Fuzzy Systems, Jeju Island (Korea)* (IEEE, ed.), IEEE, 8 2009.

- [6] P.-A. Bisgambiglia, L. Capocchi, P. Bisgambiglia, and S. Garredu, "Fuzzy Inference Models For Discrete Event Systems," in *Proceedings of the 2010 IEEE World Congress on Computational Intelligence (CCI 2010) - IEEE International Conference on Fuzzy Systems (FZZ-IEEE), Barcelona (SPAIN)* (IEEE, ed.), IEEE, 7 2010.
- [7] S. Garredu, P.-A. Bisgambiglia, E. Vittori, and J. Santucci, "Towards the definition of an intuitive specification language," in *Proceedings of the International Modeling and Simulation Multiconference (IMSM) - Conceptual Modeling Simulation (CMS)* (SCS, ed.), pp. 187–192, 2 2007.
- [8] S. Garredu, P.-A. Bisgambiglia, E. Vittori, and J. Santucci, "A new approach to describe dev models using both uml state machine diagrams and fuzzy logic," in *Proceedings of the 2010 International Mediterranean and Latin American Modelling Multiconference, I3M 2010, Fez, (Morocco)* (EMSS, ed.), EMSS, 10 2010.
- [9] P. Cacciabue, F. Decortis, B. Drozdowicz, M. Masson, and J.-P. Nordvik, "COSIMO: A cognitive simulation model of human decision making and behavior in accident management of complex plants," vol. 22, no. 5, pp. 1058 – 1074, 1992.
- [10] P. Mayenobe, L. Trassoudaine, T. Bellet, and H. Tattegrain-Veste, "Cognitive simulation of driver and cooperative driving assistance," in *Proceedings of Intelligent Vehicle Symposium*, vol. 1, p. 234, IEEE, 2002.
- [11] A. Das and W. Stuerzlinger, "A cognitive simulation model for novice text entry on cell phone keypads," in *Proceedings of the 14th European conference on Cognitive ergonomics: invent! explore!*, ECCE '07, (New York, NY, USA), pp. 141–147, ACM, 2007.
- [12] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation, Second Edition*. 2000.
- [13] C. Cassandras and S. LaFortune, *Introduction to Discrete Event Systems*. 1999.
- [14] F. Lin and H. Ying, "Modeling and control of fuzzy discrete event systems," vol. 32, no. 4, pp. 408 – 415, 2002.
- [15] E. Kilic, C. Karasu, and K. Leblebicioglu, "Fault diagnosis with dynamic fuzzy discrete event system approach," vol. 3949, pp. 117 – 124, 2005.
- [16] J. Thistle, "Supervisory control of discrete event systems,"
- [17] W. G. Wee and K. S. Fu, "A formulation of fuzzy automata and its applications as a model of learning systems," *IEEE transaction on SSC*, vol. 5, pp. 215–223, 1969.
- [18] M. Doostfateme and S. C. Kremer, "New directions in fuzzy automata," *International Journal of Approximate Reasoning*, vol. 38, pp. 175–214, 2005.
- [19] L. Holloway, B. Krogh, and A. Giua, "A survey of petri net methods for controlled discrete event systems,"
- [20] F. Lin and H. Ying, "Fuzzy discrete event systems and their observability," in *Proceedings of IFSA World Congress and 20th NAFIPS International Conference, Vancouver, (Canada)* (IEEE, ed.), vol. 3, pp. 1271 – 1276, 2001.
- [21] F. Lin, H. Ying, R. D. MacArthur, J. A. Cohn, D. Barth-Jones, and L. R. Crane, "Decision making in fuzzy discrete event systems," *Inf. Sci.*, vol. 177, pp. 3749–3763, September 2007.
- [22] B. P. Zeigler, T. G. Kim, and H. Praehofer, *Theory of Modeling and Simulation*. Orlando, FL, USA: Academic Press, Inc., 2000.
- [23] Y. Kwon, H. Park, S. Jung, and T. Kim, "Fuzzy-DEVS Formalism : Concepts, Realization and Application," *Proceedings AIS 1996*, pp. 227–234, 1996.
- [24] N. Giambiasi and S. Ghosh, "Min-Max-DEVS: A new formalism for the specification of discrete event models with min-max delays," pp. 616–621, 13th European Simulation Symposium, 2001.
- [25] L. Ntamo and B. P. Zeigler, "Expressing a forest cell model in parallel DEVS and timed cell-DEVS formalisms," *Proceedings of the 2004 Summer Computer Simulation Conference*, 2002.
- [26] S. Sheikh-Bahaei, J. Liu, M. Jamshidi, and P. Lino, "An intelligent discrete event approach to modeling, simulation and control of autonomous agents," vol. 10, no. 4, pp. 337 – 348, 2004.
- [27] P.-A. Bisgambiglia, E. de Gentili, P. Bisgambiglia, and J. Santucci, "Fuzzy simulation for discrete events systems," in *Proceedings of the 2008 IEEE World Congress on Computational Intelligence (CCI 2008) - IEEE International Conference on Fuzzy Systems (FZZ-IEEE)*, pp. 688–694, 06 2008.
- [28] P.-A. Bisgambiglia, E. de Gentili, P. Bisgambiglia, and J. Santucci, "Discrete events system simulation-based defuzzification method," in *Proceedings of The 14th IEEE Mediterranean Electrotechnical Conference (MECON)*, pp. 132–138, 05 2008.
- [29] P.-A. Bisgambiglia, E. de Gentili, P. Bisgambiglia, and J.-F. Santucci, "Fuzz-idevs: Towards a fuzzy toolbox for discrete event systems," in *Proceedings of the SIM Tools'09, Rome (Italie)* (ACM, ed.), SIMU-Tools, 3 2009.
- [30] L. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, pp. 3–28, 1978.
- [31] D. Dubois and H. Prade, "Representation and combination of uncertainty with belief functions and possibility measures," *Comput. Intell.*, no. 4, pp. 244–264, 1988.
- [32] D. Dubois, H. T. Nguyen, and H. Prade, "Possibility theory, probability and fuzzy sets : Misunderstandings, bridges and gaps," pp. 343–438, 2000.