

Use of Fuzzy Clustering for Discrete Event Simulation Model Construction

Yan Wang*, Grégory Zacharewicz*, Mamadou Kaba Traoré** and David Chen*

*IMS, University of Bordeaux, 33405 Talence Cedex, France.

**LIMOS, University of Blaise Pascal, 63173 Aubiere Cedex, France.

Abstract: System inference is widely recognized as a critical challenging issue. Discrete event simulation model construction is the relevant approach regarding of this issue. Indeed, it allows to discover a Fuzzy Discrete Event System Specification (Fuzzy-DEVS) model from data using process mining. However, this approach lacks of modularity. In this paper, the objective is to propose a new method for Fuzzy-DEVS coupled model. This method extends discrete event simulation model construction by integrating fuzzy clustering. This later is implemented as a plugin in the Process Mining Framework (ProM). In order to evaluate the relevance, a case study is presented. In this case study, a real life data of business process is inferred and the SimStudio tool is used for its simulation.

Keywords: System Inference, Process Mining, Fuzzy-DEVS, Discrete Event Simulation Model Construction, Fuzzy Cluster, ProM, SimStudio.

1. INTRODUCTION

Recently, as the data explode, systems are becoming more and more complicated. These systems can be represented by system theory (Simon H.A. 1991) which provides a fundamental and rigorous mathematical framework. In such framework, a system is characterized by its structure and its behaviour. All the knowledge about the system is organized in a 4-level hierarchy proposed by Klir (Klir G. 2013) as depicted in figure 1. System structure is at the top levels of the hierarchy and system behaviour is at the bottom levels. In other words, the more one goes down the hierarchy, the less knowledge is acquired. Moving between these levels of system knowledge, can be interpreted in three basic ways:

- In system analysis, we know the existing or hypothetical system structure and we try to generate its data;
- In system design, the system does not exist yet and we are investigating the alternative structures for a completely new system. In other words, we know what data we are expecting and we try to build a structure that can produce this data;
- In system inference, the system exists and we are trying to generate its structure. This has been called “climbing the hill” by Zeigler (Zeigler B.P., et al. 2000). A slight but significant difference between system design and system inference is its existence or not, prior to the study.

To solve the problem of system inference, we propose to use discrete event simulation model construction. This later comprises three stages: (1) the extraction of event logs from real data by the System Entity Structure; (2) the mining of transition system from event logs by process mining; (3) the transformation from transition system to Fuzzy-DEVS model.

Our choices of techniques are motivated by the following: System Entity Structure (SES) (Zeigler et al., 2007) provides a formal ontology framework as interpretations and applications; Process Mining (Van der Aalst W.M.P. 2011) provides the methods to discover monitor and improve actual processes by extracting knowledge from event logs readily available in today’s systems; DEVS (Zeigler B.P., et al. 2000) not only has general framework for modeling and simulation of complex systems, but also has formal temporal and coupling features; Unlike DEVS models, Fuzzy-DEVS models (Kwon Y.W., et al. 1996) have the possibility property as they are able to solve imprecise and uncertainty. Compared with most process models, time is defined as one of the basements in the DEVS rather than an extension or perspective which is handled in process mining. Most of the discovery algorithms ignore the actual time information. Process mining does not consider about hierarchical problem so it is hard to handle when considering about big and complex systems.

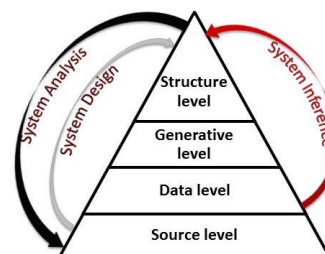


Fig. 1. System Problems in the Hierarchy of Knowledge.

However, this approach lacks of modularity, i.e., the possibility to couple Fuzzy-DEVS models in order to obtain larger models. In this paper, we propose to achieve this goal by integrating fuzzy clustering to the discrete event simulation model construction. Fuzzy clustering (Kaufman L, et al. 2009) is one of the clustering methods. The advantage

of fuzzy clustering not only produces a hard clustering but also allows for some ambiguity in the data. The proposed approach is implemented through a case study in ProM. In order of evaluation, the simulation tool SimStudio is used. The simulation result shows an optimal business process.

The paper is organized as follows. Section 2 gives some related studies. Section 3 shows the background of fuzzy sets and Fuzzy-DEVS formalism. Section 4 presents the methodology of discrete event simulation model construction integrating with fuzzy clustering. Section 5 uses a case study to show the result of using methodology. At last, Section 6 concludes this paper.

2. STATE OF THE ART

In the process mining, Petri net (Peterson J.L. 1977) is frequently identified as the direct resulting process models. To discover Petri net, α -algorithm and region-based mining (Van der Aalst W.M.P. 2011) are proposed. α -algorithm is able to discover concurrency but unable to manage the trade-offs. Region-based mining (two phase approach) first learns Transition system (Robert M.K. 1976) by state representation. Then it discovers Petri net by dividing regions. A drawback of the later is that the representational bias cannot be exploited during discovery.

In discrete event simulation, Zeigler proposes a hierarchical, modular approach at the multicomponent and the network of systems level (Zeigler B.P., 1984). Zeigler also makes a relation to testing, reusability, and distributed simulation (Zeigler B.P., 1990).

Fuzzy-DEVS model is one of the extensions of DEVS models. In this paper, it is selected as the target model. If we compare with Stochastic DEVS model (Castro R., et al. 2008), Fuzzy-DEVS is more adaptable and feasible:

- Possibility which is always used in Fuzzy-DEVS emphasizes the likelihood in an objective manner rather than probability in Stochastic DEVS model. Fuzzy-DEVS can make user focus on the mainstream behaviour of the business process;
- Fuzzy-DEVS can provide more semantics by simply integrating subjective data and using linguistics;
- Fuzzy-DEVS can provide defuzzification methods which can be used for simulation.

Moreover, Fuzzy-DEVS can be evolved in a lot of disciplines such as fuzzy if-then rules, fuzzy reasoning, fuzzy modeling, fuzzy inference system and fuzzy control language. Dahmani and Hamri (Dahmani Y., et al. 2014) use if-then rule to fuzzy reasoning rules obtained from observers or expert knowledge and specify a Fuzzy-DEVS model which computes this duration. They apply the method on forest fire propagation in the simulator to specify the new value in the model. Bigambiglia et al. (Bigambiglia P.A., et al. 2010) use fuzzy inference systems (FIS) with DEVS formalism in order to perform the control or the learning on systems described incompletely or with linguistic data. They also present a case study to support this approach. The limitation of these studies is that the model is not coming from data in reality.

3. BACKGROUND

The concept of fuzzy sets (Zadeh L.A. 1965), can be used to change the crisp set of the characteristic function. A fuzzy set \tilde{F} is equivalent to giving a reference set Ω and a mapping $\mu_{\tilde{F}}: \Omega \rightarrow [0, 1]$. For $\omega \in \Omega$, $\mu_{\tilde{F}}(\omega)$ is interpreted as the degree of membership of ω in the fuzzy set \tilde{F} . So the fuzzy set can be defined as:

$$\tilde{F} = \{(\omega, \mu_{\tilde{F}}(\omega)) | \omega \in \Omega, 0 \leq \mu_{\tilde{F}}(\omega) \leq 1\} \quad (1)$$

The Fuzzy-DEVS formalism (Kaufman L, et al. 2009) extends the DEVS formalism by applying fuzzy set theoretic formalism into the characteristic functions. It consists of two parts of the formalism according to the levels of the models: atomic model and coupled model. A fuzzy atomic model \tilde{M} is characterized by:

$$\tilde{M} = \langle X, Y, S, \tilde{\delta}_{int}, \tilde{\delta}_{ext}, \tilde{\lambda}, \tilde{ta} \rangle$$

- X: the set of input values;
- Y: the set of output values;
- S: the set of states;
- $\tilde{\delta}_{int}: S \times S \rightarrow [0, 1]$, fuzzy internal transition function;
- $\tilde{\delta}_{ext}: Q \times X \times S \times S \rightarrow [0, 1]$, fuzzy external transition function, $Q = \{(s, e) | s \in S, 0 \leq e \leq ta(s)\}$ where $ta(s)$ is the defuzzified value of \tilde{ta} ;
- $\tilde{\lambda}: S \times Y \rightarrow [0, 1]$, fuzzy output function;
- $\tilde{ta}: S \times \tilde{A} \rightarrow [0, 1]$, fuzzy time advance function, \tilde{A} = the set of fuzzy linguistic numbers.

A coupled model DN is defined:

$$DN = \langle X, Y, D, EIC, EOC, IC, SELECT \rangle$$

- X: input event sets;
- Y: output events sets;
- D: DEVS components set;
- $EIC \subseteq \{((N, ip_N), (d, ip_d)) | ip_N \in IPorts, d \in D, ip_d \in IPorts_d\}$.
- $EOC \subseteq \{((d, op_d), (N, op_N)) | op_N \in OPorts, d \in D, op_d \in OPorts_d\}$.
- $IC \subseteq \{((a, op_a), (b, ip_b)) | a, b \in D, op_a \in D, \in OPorts_a, ip_b \in IPorts_b\}$.
- SELECT: $2^D \rightarrow \{\} \rightarrow D$, tie-breaking selector.

4. METHODOLOGY

As levels of knowledge are discussed in the first chapter, the goal of discrete event simulation model construction is to climb the hills of the levels. The main structure of it is shown in figure 2. We will make a brief introduction of the previous part and then put emphasis on constructing Fuzzy-DEVS coupled model by using fuzzy cluster.

4.1 Observing event logs from data

The starting point is the data. They are measured and observed from the source by human knowledge. The data is selected by conditions at first. The conditions are based on

XES standard (Van der Aalst W.M.P. 2011). In this part of methodology, there are several steps:

- Set up Goals: the goals are investigated to make all the following steps well guided;
- Identify Relationships of Activities: The data generate SES structure to identify relationships, public and private activities can be found;
- Select a Process Instance: Choose the interesting level of the SES structure;
- Identify Activities: Add, delete or replace the activities;
- Mapping to XES file: Transform into event logs based on XES standard.

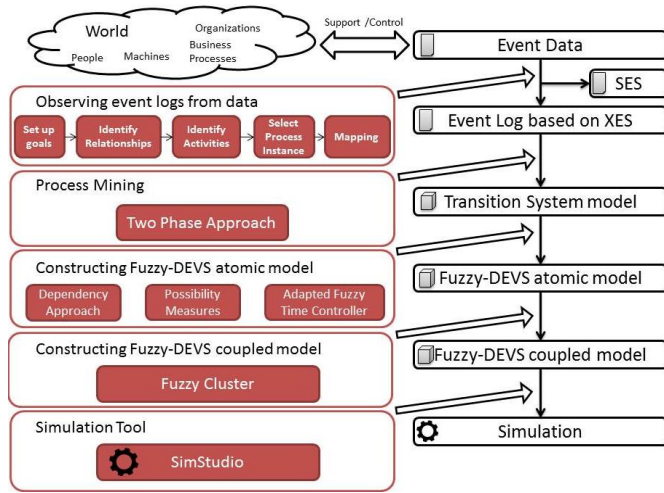


Fig. 2. General structure of the methodology.

4.2 Two Phase Approach

In this paper, we use one part of Two Phase approach in process mining (Van der Aalst W.M.P. 2011) to transform event logs into transition system. In this part of methodology, there are different methods in different dimensions to capture states. By combining different methods, we can get different kinds of transition systems. The core idea of this part of methodology is to discover regions that correspond to places.

4.3 Constructing Fuzzy-DEVS atomic model

When we get transition system, based on the previous work (Wang Y. et al., 2015), we propose to transform it into Fuzzy-DEVS atomic model. An improved region-based approach is defined to specify state in DEVS atomic model. Let $TS = (S^T, A, T)$ be a transition system and $R \subseteq S^T$ be a subset of states. P_a is a period time for each activity $a \in A$. R is a region if for each activity $a \in A$ and one of the following conditions holds:

- All transition $(s_1^T, a, s_2^T) \in T$ enter R , i.e. $s_1^T \notin R$ and $s_2^T \in R$;
- All transition $(s_1^T, a, s_2^T) \in T$ exit R , i.e. $s_1^T \in R$ and $s_2^T \notin R$;

- All transition $(s_1^T, a, s_2^T) \in T$ do not cross R , i.e. $s_1^T, s_2^T \in R$ or $s_1^T, s_2^T \notin R$;
- For all the transitions $a_1 \in T_1, a_2 \in T_2, \dots, a_n \in T_n$ enter $R, P_{a_1} \approx P_{a_2} \approx \dots \approx P_{a_n}$.

Let p_a be the private activity and u_a be the public activity. According to Fuzzy-DEVS formalism in chapter 3, the transformation follows the rules:

$$R \rightarrow S \quad (2)$$

Where the state of DEVS atomic model $s \in S$.

$$u_a \rightarrow x \text{ or } y \quad (3)$$

Where the input value $x \in X$ and the output value $y \in Y$.

$$\tilde{t}a = \begin{cases} 0 & \text{if } \exists s_0 \in S \\ T^F & \\ \text{Infinite} & \text{if } \exists S = S_1^I \end{cases} \quad (4)$$

Where s_0 is the initial state, T^F is the result coming from Adapted Fuzzy Time Controller, S_1^I is the input states of all internal transition.

$$(s_1^T, ua, s_2^T) \xrightarrow{T \rightarrow \tilde{\delta}_{int}} (s_1, s_2, \mu_{int}) \quad (5)$$

Where $s_1 \in R_1$ and $s_2 \in R_2$, μ_{int} is the result coming from dependency method.

$$\tilde{\lambda} : (y, \mu_{int}) \quad (6)$$

$$(s_1^T, pa, s_2^T) \xrightarrow{T \rightarrow \tilde{\delta}_{ext}} (s_1, e, x, s_2, \mu_{ext}) \quad (7)$$

$$\mu(s_i \rightarrow s_j) = \begin{cases} \frac{F(s_i \rightarrow s_j) - F(s_j \rightarrow s_i)}{F(s_i \rightarrow s_j) + F(s_j \rightarrow s_i) + 1} & \text{if } i \neq j \\ \frac{F(s_i \rightarrow s_j)}{F(s_i \rightarrow s_j) + 1} & \text{if } i = j \end{cases} \quad (8)$$

Where the elapsed time $e: 0 \leq e \leq \tilde{t}a$, μ_{ext} is the result coming from the dependency method. The first step of the dependency method is to calculate the frequency of every transition from event logs. Then we use equation (8) to calculate the possibility of every transition. The structure of Adapted Fuzzy Time Controller is shown in figure 3 inherited from Fuzzy Time Controller (Khan M.S. 2008). The main idea is to fuzzify time and makes the inference rule between time duration and remaining time. The final output is inferred by using the weighted average method for defuzzification.

4.4 Constructing Fuzzy-DEVS coupled model

The data can be organized in an n-by-k matrix, where the rows correspond to the objects (or cases) and the columns correspond to the variables. In the data, the variables can be identified as clusters. The aim is to form groups in such a way that variables in the same group are similar to each other. We propose to use membership coefficients to assign these variables. As the clusters come from data, the algorithm

to calculate the membership coefficients is the dependency method. Due to the dynamic modelling and simulation environment, the membership coefficients are also related to the state or the time. As the time elapses, the membership coefficients may change. Furthermore, we extend the formalism of Fuzzy-DEVS coupled model:

$$\tilde{E}TC \subseteq \{((N, ip_N), (d, ip_d), (\mu_{EIC}, e)) | ip_N \in IPorts, d \in D, ip_d \in IPorts_d\} \quad (9)$$

$$\tilde{E}OC \subseteq \{((d, op_d), (N, op_N), (\mu_{EOC}, e)) | op_N \in OPorts, d \in D, op_d \in OPorts_d\} \quad (10)$$

$$\tilde{I}C \subseteq \{((a, op_a), (b, ip_b), (\mu_{IC}, e)) | a, b \in D, op_a \in OPorts_a, ip_b \in IPorts_b\} \quad (11)$$

Where μ_{EOC} , μ_{EIC} and $\mu_{IC} \rightarrow [0,1]$ are the membership coefficients of EOC, EIC and IC and they are limited by the elapsed time e .

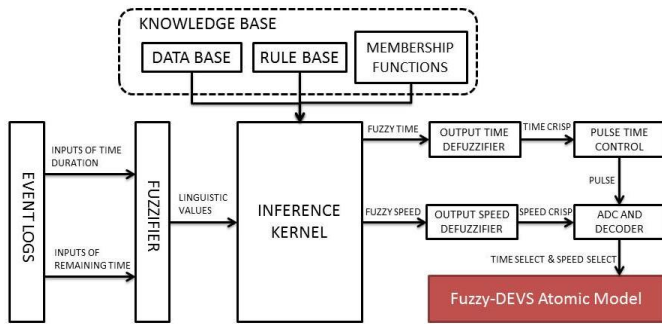


Fig. 3. The structure of Adapted Fuzzy Time Controller.

4.5 Implementation and Simulation

To implement these methods, we use ProM which is designed and developed by the process mining group. ProM is an open-source framework for collecting tools and applications of process mining (Van der Aalst W.M.P. 2011). We are trying to create a new plugin called “Convert to Fuzzy-DEVS using Regions” which is synchronized on the server of the process mining group (Wang Y., the subversion server).

The simulation engine of Fuzzy-DEVS atomic model is SimStudio (Traoré M.K. 2008). Model class is inherited by Atomic Model and Coupled Model and contains abstract methods. Simulator class initialises and runs the simulation. In this paper, we propose to use Atomic Model class, Coupled Model class and Simulator for simulation.

5. CASE STUDY

This case study comes from the problem in the website (BPI 2016). The first step is to transform from csv file to event logs based on XES standard. From website, we capture two main goals: how the channels are being used; when customers move from one contact channel to next. Following by these two steps, we choose “Question.csv” and “Werkmap-message.csv”. In the “Question.csv”, we can

generate the corresponding SES structure as shown in Figure 4 (the words are translated into English). In the “Werkmap-message.csv”, we can only get “channel” as the aspect of “Workbook: message”. We identify the variable “channel” between 1 and 2 as private activities and others as public activities. Eventually, these two files are converted into two event logs.

We use plugin “Mine Transition System” to construct a Transition System model for each event logs. And then we use the proposed plugin “Convert to Fuzzy-DEVS using Regions” to construct Fuzzy-DEVS atomic model (see in Figures 5 and 6). The model in figure 5 can be shown more clearly in the full paper. The coupling rule by using fuzzy cluster is calculated (see Table 1). According to different state (or time), we have different membership coefficients. We choose the biggest membership coefficient as output event. This event is sent to the model in Figure 6 to trigger the external transition.

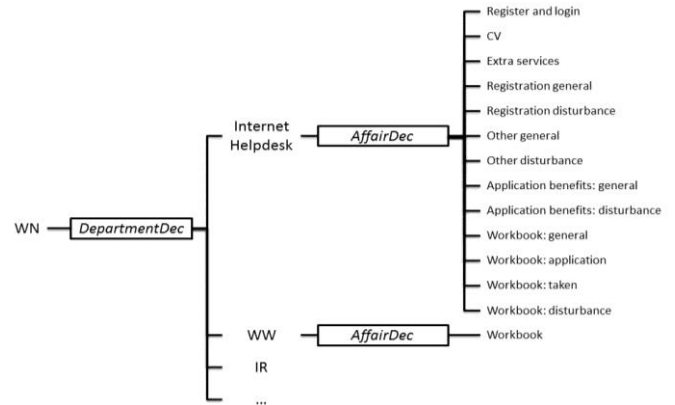


Fig. 4. The SES structure of Question document.

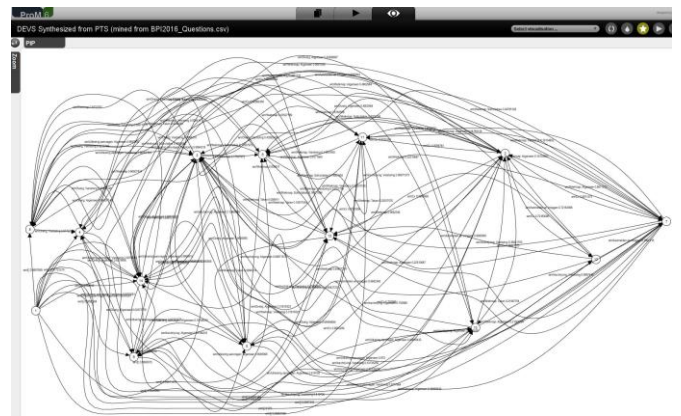


Fig. 5. Fuzzy-DEVS atomic model from Question file.

The simulation result (see in Figure 7) shows the mainstream of business process from the model in Figure 5. As the time elapses, state is changed by internal transition in Figure 5. Meanwhile, the events are sent through the port wm to the model in Figure 6. The model in Figure 6 recognizes the events by using the rule of fuzzy clustering and executes the external transition. The final output provides insight into the goal of this case study.

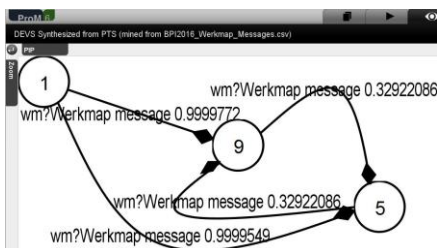


Fig. 6. Fuzzy-DEVS atomic model from Werkmap file.

Table 1. Part of the coupling rule between two models

Activity in Question Document	Activity in Werkmap Document	Membership Coefficient in state []	Membership Coefficient in state [Werkmap]
Workbook: general	Workbook: message	0.995192	0.843691
Workbook: application	Workbook: message	0.989796	0.923379
Workbook: taken	Workbook: message	0.98	0.960352
Workbook: disturbance	Workbook: message	0.994444	0.863342

```

1 : start []
Internal transition: [] & [Inschrijving: Verstoring] ---- 0.98597464
Internal transition: [] & [Uitkering aanvragen: Verstoring] ---- 0.9714286
Internal transition: [] & [Inschrijving: Algemeen] ---- 0.9878049
Internal transition: [] & [Werkmap: Taken] ---- 0.98
Internal transition: [] & [Overig: Algemeen] ---- 0.99300706
Internal transition: [] & [Werkmap: Algemeen] ---- 0.99519235
Internal transition: [] & [Werkmap: Verstoring] ---- 0.99444444
Internal transition: [] & [Uitkering aanvragen: Algemeen] ---- 0.98611111
Internal transition: [] & [Aanmelden en inloggen] ---- 0.975
Internal transition: [] & [Werkmap] ---- 0.9995915
Internal transition: [] & [Cv] ---- 0.9959184
Internal transition: [] & [Overig: Verstoring] ---- 0.9896907
Internal transition: [] & [Werkmap: Sollicitaties] ---- 0.9897959
420 : start [Werkmap]
Internal transition: [Werkmap] & [Cv] ---- 0.8183506
Internal transition: [Werkmap] & [Werkmap: Algemeen] ---- 0.8436912
Internal transition: [Werkmap] & [Inschrijving: Algemeen] ---- 0.9355477
Internal transition: [Werkmap] & [Werkmap: Verstoring] ---- 0.8633422
Internal transition: [Werkmap] & [Werkmap: Taken] ---- 0.9603524
Internal transition: [Werkmap] & [Aanmelden en inloggen] ---- 0.9682348
Internal transition: [Werkmap] & [Inschrijving: Verstoring] ---- 0.94709635
Internal transition: [Werkmap] & [Overig: Algemeen] ---- 0.88996136
Internal transition: [Werkmap] & [Overig: Verstoring] ---- 0.9241352
Internal transition: [Werkmap] & [Uitkering aanvragen: Verstoring] ---- 0.97219986
Internal transition: [Werkmap] & [Werkmap: Sollicitaties] ---- 0.9233792

```

Fig. 7. Part of Simulation result by SimStudio.

6. CONCLUSION

This paper shows a general view of discrete event simulation model construction. The practice of process mining is all pervasive but discrete event simulation model construction provides a new way to discover Fuzzy-DEVS model. This model is able to represent time and complex system. Secondly, this approach is enhanced by integrating fuzzy cluster. Thanks to fuzzy cluster, Fuzzy-DEVS atomic models are coupled together. A case study illustrates this methodology with reality in ProM. The future work will focus on the implementation of coupling function automatically in ProM and model validation by expert participation.

REFERENCES

Bisgambiglia P.A., Capocchi L., Bisgambiglia P., et al., (2010) Fuzzy inference models for Discrete Event systems. Fuzzy Systems (FUZZ), IEEE International Conference on. IEEE, pp. 1-8.

Business Process Intelligence (BPI) (2016) https://data.4tu.nl/repository/collection:event_logs_real.

Castro R, Kofman E, Wainer G., (2008) A formal framework for stochastic DEVS modeling and simulation. Proceedings of the 2008 Spring simulation multiconference. Society for Computer Simulation International, pp. 421-428.

Dahmani Y., Hamri M.E.A., (2014) Specification of the State Lifetime in the DEVS Formalism by Fuzzy Controller. arXiv preprint arXiv:1401.5638.

Kaufman L, Rousseeuw P J. (2009) Finding groups in data: an introduction to cluster analysis[M]. John Wiley & Sons.

Khan M.S., (2008) Fuzzy Time Control Modeling of Discrete Event Systems. International Conference on Intelligent Automation and Robotics, WCECS, U.S.A., pp. 683-688.

Klir G., (2013) Architecture of systems problem solving. Springer Science & Business Media.

Kwon Y.W., Park H.C., Jung S. and Kim T.G., (1996) Fuzzy-DEVS Formalisme: Concepts, Realization and Application. Proceedings AIS, pp. 227-234.

Peterson J.L., Petri Nets. ACM computing surveys, vol. 9(3), pp. 223-252, 1977.

Robert M.K., (1976) Formal verification of parallel programs. Communications of the ACM, 19(7), 371-384.

Simon H A. (1991) The architecture of complexity. Facets of systems science. Springer US, 457-476.

Traoré M.K., (2008) SimStudio: a next generation modeling and simulation framework. Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Van der Aalst W.M.P., (2011) Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer Science & Business Media.

Wang Y., the subversion server at the Technical University of Eindhoven, <https://svn.win.tue.nl/repos/prom/Packages/TS2DEVS>.

Wang Y., Zacharewicz G., Chen D. and Traoré M.K., (2015) a proposal of using DEVS model for process mining. Proceedings of the European Modeling and Simulation Symposium, Bergeggi, pp. 403-409.

Zadeh L.A., (1965) Fuzzy Sets. Inform. Control, vol. 8, pp. 338-353.

Zeigler B.P., Multifaceted Modeling and Discrete Event Simulation. Academic Press, London, 1984.

Zeigler B.P., Object-Oriented Simulation with Hierarchical Modular Models. Academic Press, Boston, 1990.

Zeigler B.P., Praehofer H. and Kim T.G., (2000) Theory of Modeling and Simulation. Academic Press, New York.

Zeigler B P, Hammonds P E. (2007) Modeling & simulation-based data engineering: introducing pragmatics into ontologies for net-centric information exchange[M]. Academic Press.