Embedding the Fuzzy Inference Engine into Petri Net Models

Jih-Fu Tu
The Department of Electronic Engineering, St. John's University, Taiwan

Abstract: - We offer a hybrid system, which is integrated with the fuzzy system and Petri net models of discrete event system. It is convenient to users or students for building the model-level of system while in learning. We exploit the characteristics of fuzzy theory, imprecise or ambiguous information, to map the algorithm of crisp inputs and outputs. Furthermore, model problems, product a nonlinear function, and final predict the next state transition of Petri net graphs. In order to quickly and precisely predict the result state, we use the *more or less* hedge mathematic $[\mu_A(x)]^{0.5}$ in this practical application, and use the *Center of gravity* (*COG*) defuzzification method to infer the state transition for this hybrid system. Finally, we issue an example of washed-machine to illustrate this proposed hybrid system.

Keywords: Discrete event system (DES), fuzzy associative memory, hybrid system, *more or less* hedge mathematic, finite state machines (FSM), and Petri net graphs.

1. Introduction

Several new control theories and finite state machines (FSM) had been brought about and applied to control system, in the recently years. Among them, such as fuzzy theory was widely famous by means of the inference-engineer of logical mathematical model that was the one of theories to successfully predict the results of one system by uncertainty data, without previously to notify the clear conditions of system. On the other hand, Petri Nets (PNs) [9,12] was used for system model analysis and development tool, and was combination of graphic and mathematic properties. Nowadays, They are applied in different kinds of model construction, analysis, and simulation [1, 3, 13].

Still now, lacking for the integrating researches and papers of Petri net graph and fuzzy system, and no common rules are defined and used to both. By aforementioned reasons, this paper will investigate how to exploit the fuzzy theory to infer the state transition in Petri net graph, and embed the inference rules in the transition nodes of Petri net graph to predict the transformation of *transactions* or *tokens*. Anywhere, we make the two different inference models as a new novelty graphic integrator in which the interpret inference rules of fuzzy system is build into the transition nodes of Petri net graph to predict and analyze the moving tokens for processing the control systems.

In recent years, because of the change of sciences and technologies, make all kinds of systematic structures and functions complicating day by day. In the tradition, the analysis method of the complicated system is to find out the systematic structure prior to the systematic physical characteristic, and then utilize the physics formula to analyze the control-system with modeling. The control-system has already getting complicated and use the physics formula and system-model, which uses one layer analytic approaches of model (Model-Level analysis approaches) [8] and must input some materials to initial and get in the system, to describe.

For the output of result productions, which are the relation of analyzing, condition defining, output arrangement, structure systematically, and process reasonable. They are set up and performed by the algorithm, and modeled to control the systems with the proper mathematics, respectively. In addition, the behavior of the proposed discrete event system (DES) for modeling, predicting, and calculating is used the clear (crisp), precise, and confirmed (certain) characteristics of fuzzy theory. The meaning of clear (crisp) represents at binary logic, judge one of the binary logic that is distinguished only into 'denying' and 'really'. The meaning of accurate (precise) represents for the phenomenon and characteristics of a modeled system, which expresses with accurately parameters. The meaning of confirm that represents the structure and parameters of model via clear learning, no matter how the outputs do not have any queries. In the new era, this sciences and technologies are developed, the systems are becoming to complication and unable to use quantitative to analyze already, thus make the traditional system analytic approaches face the serious bottleneck.

This proposed hybrid system has the following specifications: The fuzzy rule is implemented the Mamdani method with the AND logic operation to explicate the fuzzy logic, the mathematical expression is *more or less* hedge, and the defuzzification is *COG*. The structure of this paper is organized as follows. In Section 2, we describe the characteristics of fuzzy logic and Petri net, individually. Section 3 illustrates the inference methods of place and transaction of the proposed system. In Section 4, we implement this hybrid simulation system and give an example such as washing machine to verify our assumption. Finally, remark the conclusion in Section 5, in which summarizes the research results, and put forward the suggestion to the research direction in the future.

2. Embedding Fuzzy Inference Engineer into Petri Nets

The process of defining the Petri net graph involves two steps. First, we define the Petri net structure, which is analogous to the state transition diagram of automaton. Then, we adjoin to this graph an initial state, a set of marked states, and a transition labeling function [1,2, 4, 5]. Petri net structure is a weighted bipartite graph: (P, T, F, w), where, $P = \{P_1, P_2, \ldots, P_n\}$ is the finite set of places, $T = \{T_1, T_2, \ldots, T_m\}$is the finite set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is the set of arcs from places to transactions and from transactions to places, w: $F \rightarrow \{1,2,3,\ldots\}$ is the weight function on the arcs. In automata systems, the state transition mechanism is directly connected to the arcs in the state transition diagram. For state transition mechanism, the Petri net is provided by moving tokens through the net and hence changing the state. The state of Petri net is changed while a transaction fires or occurs. Then one token is moved from each of the input places to the one of output places [6, 11].

The fuzzy systems are structured numerical estimators. They start from highly formalized insights bout the structure of categories found in the real word and then articulate fuzzy IF-THEN rules as a kind of expect knowledge [7, 10, 14, 15].

In light of their similarities, Petri net and fuzzy systems are suitable for solving many of the same problems, achieving some degree of machine intelligent, and graphing the properties of a matrix relation of systems. The same characteristics of both are vertex graph to illustrate the state transition, relation matrix to identify the relation of input and output state equation, and the tree path to describe the state model.

In this paper, we will embed the fuzzy logic into the transition nodes (abbreviated to T) of Petri net to infer the state transfer to implement a hybrid intelligent system, which has the following characteristics: 1) the fuzzy set is convex, 2) the fuzzy numbers are defined by mixed function, which is combination of the S function and π function, 3) the fuzzy set-relation equations focus on max-min composition, $\mu_X(y)=\mu_{YoR}(y)$ = max min [$\mu_X(x)$, $\mu_R(x,y)$], 4) use the method of *Center of gravity* (*COG*) to defuzzy.

When drawing Petri net graphs, we need to differentiate between the two types of nodes, places(abbreviated to P) and transitions. Among places and transitions are connected by an arc, we allow multiple arcs to connect two nodes and assign a weight to each arc representing the number of arcs, for instance Figure 1. A token is to move among two nodes from place (or transition) to transition (or place) while transaction happens, the moving path of token is decided to the inference rules of transition.

$$P = \{P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12\}$$
$$T = \{t1, t2, t3, t4, t5, t6\}$$

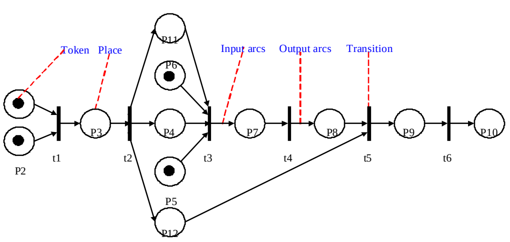
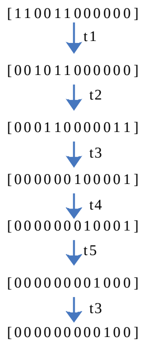


Fig. 1. The structure of Petri net.

Refer to Figure 1, we construct the coverability tree of the Petri net model and its input/output matrix, both are shown in Figure 2.



$$D^- = I = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Input matrix

$$D^+ = O = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

output matrix

Fig. 2. Coverability tree and input/output matrix of the example of Figure 1.

3. The Inference Engineer in Petri Net

The inference engineer of fuzzy system is embedded in the transaction node of this proposed Petri net to predict where the targets, i.e. the next places, of a moving token. For the proposed Petri net, in which the transition node with the function of inference engineer is achieved by fuzzy logic. We had divided the proposed Petri net into three levels, such as initial level, hidden level, and terminal level.

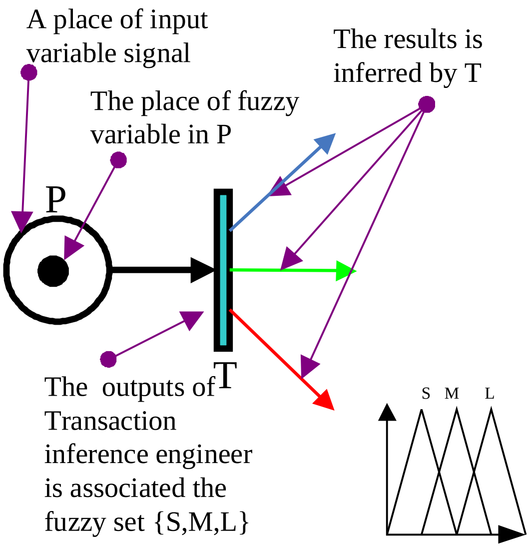


Fig. 3. The initial level of Petri net with fuzzy logic.

The initial level (also called the fuzzy level) is used to fuzzy the input variables, and combination of start node, which maybe place or transition. If the start node is place, which must be masked some tokens, then connected to transition node. If the start node is transition, which must be defined a fired condition by fuzzy logic, then connected to place node. The relation of place and transition of this Petri net mixed fuzzy theory in initial level is illustrated in Figure 3.

Second-level is hidden level (also called the inference level), which is combination of a serial of transition nodes and places nodes. The transitions is used to infer the next place (state) of a moving token, the inference rule is decided to the fuzzy logic of the system. The relation of place and transition of hidden level is illustrated in Figure 4. The set of input and output places from transition T_i is represented to I (Ti) = {P1, P2, …, Pn} and O (T_i) = {P3, P4, …}, where i = 1,2,…,n, individually.

The inference process is progressed by transition Ti according to the fuzzy rules with the AND Boolean logic operation. The Producing output of each transition has different behalf of system parameter and attitude, which are turned into token that is the incident will gather and change each setting value and comparing in the attitude to excite. The token of Petri net represents the systematic signal parameter in each fuzzy value while assembled.

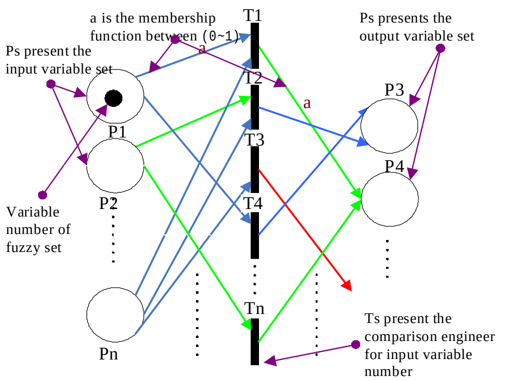


Fig. 4. The hidden level of proposed Petri net.

The third-level is the terminal level, which is used to store the results by defuzzing (shown in Figure 5). The set of output places from transition node (T) is represented to O (T_t) = {Pt}.

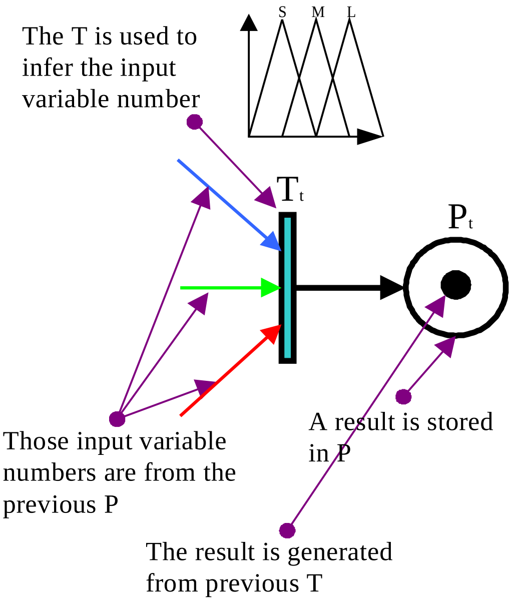


Fig. 5. Terminal level of proposal Petri net.

In a word, the place is used to narrated state, transition is used to change the attitude becoming, and the token that can indicate every place after transiting of different levels in the above description.

We will describe the implemented steps or this novel Petri net, which is combination of three interfaces there are fuzzifciation, fuzz-rule base and inference engineer, and defuzzifcition, individually. In this paper, we will construct a novel Petri net.

The structures of graph user interface (GUI), in which has four sequence blocks: the graphical object and route table, the fuzzification definition table, the path recognizing, and the inference algorithm. The graph user interface (GUI) supports the platform to user for using this proposed new simulator. The functions of each block are described as follows:

1. The block graphical object and route table offers an interactive environment to build the place, transition, token, and directed arc for Petri net graph, and set the α (cut value), fuzzy rules of fuzzy systems, which is combined to the *Image*, *Line*, *Label*, *ComboBox*, and *ListBox* objects of VB. The *Image* object is used to create the place, transition, and token elements of Petri net, The *Line* object is used to create the directed arc. The *Label* object is used to name the label of the elements of Petri net. The *ComboBox* object is used to control the transform of place and transition. Finally, the *ListBox* object is used to store the recorder and the traced path of Petri net.
2. The block fuzzification definition table is used to create the fuzzy set and the membership set of the desired fuzzy systems via the *more or less* hedge mathematical expression $[\mu_A(x)]^{0.5}$.
3. The block path recognizing can automatically derive a relative table of place-to-transition from the created Petri net model.
4. The block inference algorithm is used to infer the fuzzy logic and represent the results with animation.

This novel Petri net graph systems had created with Chinese version, the graphical user interface (GUI) is shown in Figure 6. The menu bar offers the following functions: such as file management, edit management, tool, view, help, and example. We also have supported several icons, there are *place*, *transition*, *token*, *arc*, *run* for the new created Petri net graph, and other execution this simulation tools, respectively.

The proposed Petri net environment is developed using the Virtual Basic 6.0(VB) software and suit to execute in the personal computer with the Microsoft windows 98, 2000, or XP operation system.

Fig. 6. The GUI of Proposed Petri net.

4. The Example of Proposed Simulator

In this section, we will give an example of washing machine in which the model is defined to this hybrid intelligent system. The example is described as follows.

Example: for a washing machine, the output variable of washing time (T) is depended upon two input variables, the weight of clothes (X) and stream of water (Y). Those variables are departed three degrees, which are small (S), middle (M), and large (L) defined as follows:

- The weight of the clothes is X, which range is from 0 to 10 (shown in Figure 7) and the unit is kilogram (Kg).
- The stream of water is Y, which range is from 0 to 80 (shown in Figure 7) the unit is liter per minute (l/min.).
- The washing time is T, which range is from 0 to 100 (shown in Figure 7) and the unit is kilogram (min.).

There are two input and one-output variables in our example. The inference engineer is constructed of “ If <*input variable*> AND <*input variable*> THEN

<*output variable*>” fuzzy rule. According to the above fuzzy rule, the square fuzzy associative memory (FSM) of X, Y, and T variables are listed in Table 1 and Table 2 contains these rules.

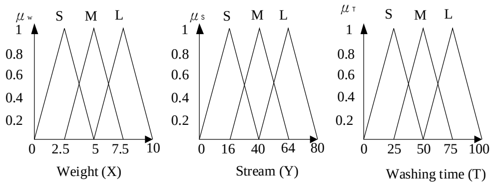


Fig. 7. Membership functions of W, Y and T.

Table 1. Fuzzy associative memory (FSM) of example

Washing time (T)		Weight (X)		
		S	M	L
Stream (Y)	S	M	L	L
	M	S	M	L
	L	S	S	L

If we want to wash the clothes, while turn on the power, the washing machine automatic detects the weight of clothes (X) is 3.2 Kgs, and adjusts the stream water (Y) to 32 l/min., individually. Refer to Figure 7 gets the input fuzzy set of $X_{3.2\ Kg}$ and $Y_{32\ l/min.}$ as follows:

The fuzzy set of $X_{3.2\ Kg}$ = {*0.8/S, 0.2/M, 0/L*} (1)
The fuzzy set of $Y_{32\ l/min.}$= {*0.4/S, 0.8/M, 0/L*}(2)

Exploiting the min operation of Mamdanl has a result shows in Figure 9. Furthermore, using the max operation of Mandanl has a result shows in Figure 9, too.

For the above definition example, we assume and set the input value of X and Y to 3.2 Kg and 32 liter, respectively. The initial nodes of the proposed Petri net graph are corresponding to the P1 and P2 place, individually. Furthermore, the 3.2 Kg and 32 liter places fire the marks and been turned into the fuzzy value on state P1 (the weight of washed clothes, X) and P2 (the water stream, Y) those are the first part of systems inputs at the beginning, respectively.

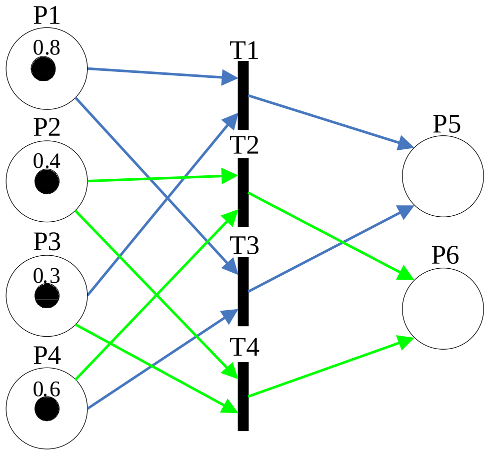


Fig. 8. The result using the min operation of Mamdanl method.

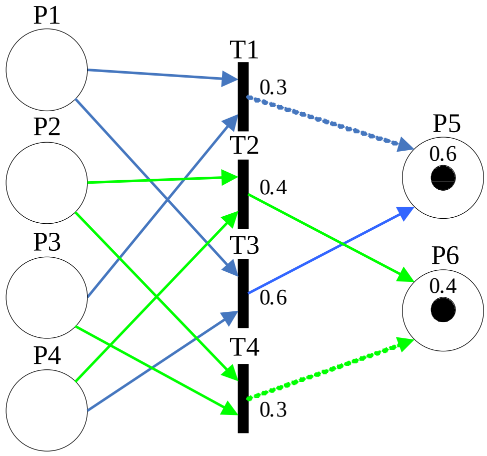


Fig. 9. The result using the max operation of Mamdanl method.

Then the exciting attitudes have became more corresponding to the membership of the fuzzy function and correspond to next states of ownership by inferring the transaction nodes, T1 and T2. Sequentially, the fuzzy regular inference of the second part will turn into the exciting results, which are produced from the attitude of every second-part places of the first level. The tokens are delivered to the corresponding next place from the previous place via the connection line. For instance the changed X, the product places of P3 (0.8/small), P4 (0.2/middle), and P5 (0/large) are derived from P1 place. In other words, the P6 (0.4/small), P7 (0.8/middle), and P8 (0/large) places are derived from P1 place of the changed Y variable parameter.

For the transaction nodes, T, of the proposed Petri net graph are constructed with the capacity of inference. The embedded inference engineer is implemented with rule-based, which is defined as ***IF*** < *the crisp value of input*> ***THEN*** < *the corresponded memberships value of fuzzy set*>. Then, a token is delivered to the corresponded place via the connection arc.

The third-level is used to solve and defuzzy the T of second-level. We get the maximum value of each fuzzy value, which are produces from the transaction nodes of second-level. The attitudes of third-level is defuzzied by the *COG* defuzzification formula, which is the fuzzy analytical method for the weight to obtain a real output value of system. In this example, the tokens of P9, P10, and P11 are defuzied by the T12 transaction. Final, it is turned into 41 minutes and shown in P12 place. The simulation results are illustrated as Figure 10.

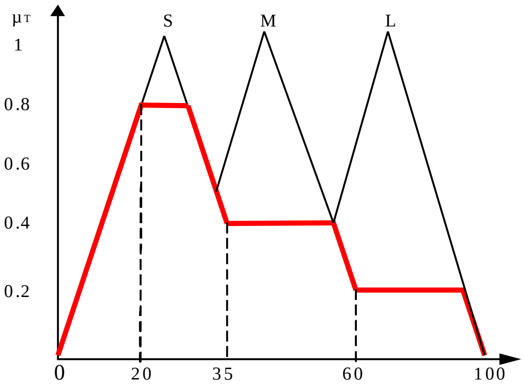


Fig. 10. The result of washing machine of example.

The real washing time is defuzzied by the *Center of gravity* (*COG*) defuzzification formula. The washing time is calculated as following steps,

$z_{COG} = \Sigma \mu_C(z_j) z_j / \Sigma \mu_C(z_j)$,

where j= 1,…, n, and n is the number of quantization levels of the output,

z_j is the number of control output at the quantization level j,

$\mu_C(z_j)$ represents its membership value in the output fuzzy set.

Exploiting the formula of *COG*, we get the fuzzy set of washing time as *w* = {0.8/20, 0.4/35, 0.2/60}, and calculate the washing time exploit *COG* formula,
The washing time (T) :

T= 41.025 min.

5. Conclusions

In this paper, we attempt mixing two different characteristics theories, the fuzzy system and Petri net system, to create a novel Petri net graph, in which the transition nodes have the capability to infer and predict the state change. This proposed hybrid system has the following specifications: The fuzzy rule is implemented the Mamdani method with the *AND* logic operation to explicate the fuzzy logic, the mathematical expression is *more or less* hedge, and the defuzzification is *COG*. To indicate that the proposal Petri net is usage, we also have created a Petri net system, which is a hybrid system for the Petri net and fuzzy system.

In this paper, we also have issued a completed novel Petri net, which implement the inference in transition with the fuzzy logic algorithm and have the Chinese graphical user interface (GUI). In the future, we will attempt to embed the inference engineer into the place node of the Petri net system and exploit the Neuron network technology.

References:

[1]. A. J. Bugarin, and S. Barro, “Fuzzy reasoning supported by Petri nets,” IEEE Transactions on Fuzzy Systems, Vol.2, No.2, pp.135-150, 1994.

[2]. C. G. Cassandras and S. Laforture,

"Introduction to Discrete Event Systems," Kluwer academic Publishers, 1999, pp. 225-267.

[3]. S. M. Chen, "Weighted Fuzzy Reasoning Using Weighted Fuzzy Petri Nets," IEEE Transaction on Knowledge and Data Engineering, Vol.14, No.2, 2002, pp. 386-397.

[4]. S. M. Chen, J. M. Ke, and J. F. Chang, "Knowledge representation using fuzzy Petri nets," IEEE Transaction on Knowledge and Data Engineering, Vol. 2, No. 3, 1990, pp. 311-319.

[5]. Design CPN Web, http://www.daimi.au.dk/CPnets/.

[6]. R, Johansson, "Systems Modeling and Identification," Prentice-Hall, 1993.

[7]. J. Lee, K.F.R. Liu and W. Chiang, "A Fuzzy Petri Net Based Expert System and Its Application to Damage Assessment of Bridges," IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 29, No. 3, June 1999, pp. 350-370.

[8]. P. Lindskog, " Fuzzy identification from a grey box modeling point of view" in Fuzzy Model Identification, H. Hellendoorn and D. Driankov, Editor, Springer, 1997.

[9]. C. G. Looney, "Fuzzy Petri nets for rule-based decision making," IEEE Transaction on Systems, Man, and Cybernetics, Vol. 18, No. 1, 1988, pp. 178-183.

[10]. J. O. Moody and P. J. Antsaklis, "Supervisory Control of Discrete Event Systems Using Petri Nets," Kluwer Academic Publishers, Boston, 1993.

[11]. Petri net Editor Web, http://www.cs.uct.ac.za/Research/DNA/DaNAMiCS/.

[12]. C. A. Petri, "Communication with Automate, " New York: Griffiss Air Force Base, Tech. Rep. RADC-TR-65-377, Vol. 1, Sept. 1964.

[13]. H. Scarpelli, F. Gomide, and R. Yager, "A reasoning algorithm for high level fuzzy Petri net," IEEE Transaction on Fuzzy Systems, Vol. 4, No. 3, 1996, pp. 282-294.

[14]. Jih-Fu Tu, "Using Coloured Petri net for Modeling a High-Performance of Processor Communication," T*he Journal of St. John's university*, September 2005, St. John's University, Taiwan, pp. 12-20.

[15]. J. -F. Tu and Yin.-Tuei Hsu, "Exploiting Petri Net Graph to Model the FSM of Modulo-P Counter," 2004 IEEE Conference on CIS, Dec. 1-3, 2004, Singapore, pp. 1089-1093.