Fuzzy output system on the basis of the modified fuzzy Petri nets

Dmitry Kochkin[1], and *Aleksei* Sukonschicov[1]

[1]Department of Automation and Computer Engineering, Vologda State University, Lenina street 15, 160000, Vologda, Russia

Abstract. The development of information technologies requires improvement of simulation methods and mathematical apparatus. The mathematical apparatus of Petri nets is used for simulation of parallel asynchronous systems and has a broad scope. The modified fuzzy Petri nets expand the modeling capabilities of Petri nets by combining the properties of different extensions. Modified extension can serve for construction of models with a complex structure and logic of operation with the use of fuzzy logic apparatus for control on the basis of the system of production rules.

1 Introduction

Modeling of information processes and systems becomes an urgent task for business and industry. Information support of production processes is a guarantee of effective performance and determines the success of the enterprise to a large extent. Stable operation is provided by the application of simulation on every stage of the life cycle of information and technical systems. The urgency of developing new approaches to modeling and development of the existing mathematical apparatus increases, considering the complexity of modern software products and technical means.

The article introduces the mathematical apparatus of the modified fuzzy Petri nets (MFPN). A description of the structure of the MFPN apparatus is provided. The possibility of constructing a system of fuzzy output on the basis of MFPN, which can enable fuzzy control implementation in Petri net models, is described.

2 Mathematical apparatus

The mathematical apparatus of the modified fuzzy Petri nets (PN) is developed on the basis of various extensions of Petri nets. The basis of the MFPN are coloured [1, 2] and fuzzy PN. The modeling process includes the properties of priority, hierarchical and timed PN in order to simplify the process and to increase visibility and modeling capacity of the modified extension (Fig. 1).

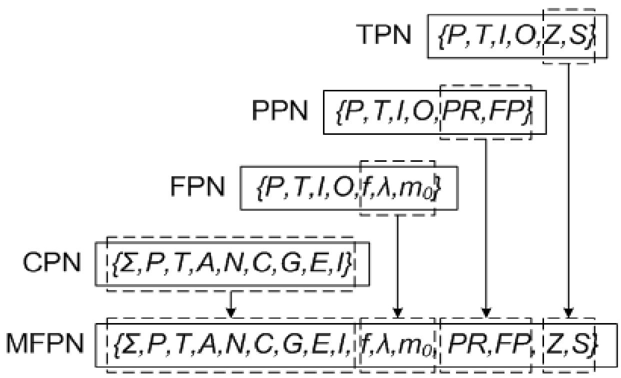

Fig. 1. The structure of the modified fuzzy Petri nets.

MFPN is the modified PN extension, CPN is the coloured PN, FPN is the fuzzy PN, PPN is the priority PN, TPN is the timed PN.

Using the definitions of basic PN extensions [1-7], MFPN can be presented in the following way:

$$MPN = (\Sigma, P, T, A, N, C, G, \\ , E, I, f, \lambda, m_0, PR, FP, Z, S) \quad (1)$$

where:

1. Σ is the finite set of types;
2. P is the finite set of positions, $P = P_c \cup P_f$, where P_c is the finite set of CPN positions, P_f is the finite set of FPN positions, $P_c \cap P_f = \varnothing$;
3. T is the finite set of transitions, $T = T_c \cup T_f \cup T_{cf} \cup T_{fc}$, where T_c is the finite set of CPN transitions, T_f is the finite set of FPN transitions,

T_{cf} is finite set of CPN to FPN transitions, T_{fc} is the finite set of FPN to CPN transitions, the sets of T_c, T_f, T_{cf}, T_{fc} are pairwise disjoint;

4. A is the finite set of arcs;
5. N is the incidence function $N : A \to A_c \cup A_f \cup A_{cf} \cup A_{fc}$,

$$A_c = P_c \times T_c \cup T_c \times P_c$$
$$A_f = P_f \times T_f \cup T_f \times P_f$$
$$A_{cf} = P_c \times T_{cf} \cup T_{cf} \times P_f \; ;$$
$$A_{fc} = P_f \times T_{fc} \cup T_{fc} \times P_c$$

6. C is the function of position type definition $C : P_c \to \Sigma$;
7. G is the function of transitions triggering conditions such as $\forall t \in T_c : [Type(G(t)) = \mathrm{B} \wedge Type(Var(G(t))) \subseteq \Sigma]$;
8. E is the function of arc expressions $\forall a \in A_c : [Type(E(a)) = C(p)_{ms} \wedge Type(Var(E(a))) \subseteq \Sigma]$, p is position incidental to the given arc;
9. I is initialization function (initial marking), $\forall p \in P_c : [Type(I(p)) = C(p)_{ms}]$;
10. $f = (f_1, f_2, ..., f_n)$ is the vector of values of the membership function of the fuzzy transitions functioning $t_i \in T_f$, $f_j \in [0,1](\forall j \in \{1,2,...,u\})$;
11. $\lambda = (\lambda_1, \lambda_2, ..., \lambda_n)$ is the vector of values of transitions functioning threshold $t_i \in T_f$, $\lambda_i = [0,1](\forall j \in \{1,2,...,u\})$;
12. $m_0 = (m_1^0, m_2^0, ..., m_n^0)$ is the vector of initial marking $p_i \in P_f$, each component of which is determined by the value of the membership function of the fuzzy existence of one of the marker at the position of the FPN Cf, $m_i^0 \in [0,1](\forall i \in \{1,2,...,u\})$;
13. PR is the finite set of priorities, which determine the order in which transitions functions in the case of several active transitions;
14. FP is the function transition priority $FP : T \to PR$;
15. Z is the vector of parameters of time delays in the positions of markers $Z = (z_1, z_2, ..., z_n)$, $z_i \in N \cup \{0\}$;
16. S is the vector of time stamps of MFPN transitions $S = (s_1, s_2, ..., s_n)$, $s_i \in N \cup \{0\}$;

The inclusion of new types of positions and transitions does not allow to use the existing rules of CPN or FPN functioning. The MFPN rules take into account the presence of coloured and fuzzy positions and transitions in one model.

3 MFPN application

The possibility of using the marks of a complex format is achieved by combining the properties of various extensions in MFPN in order to enable modeling of information exchange. Setting the time stamps to transitions and labels can help to adequately simulate objects functioning in time. The application of complex triggering conditions and the setting of expressions on arcs allow simulation of the complex logic of the simulated objects. The use of fuzzy PN as a basic extension allows simulation of fuzzy production rules and control on the basis of fuzzy output.

Coloured Petri nets are used for modeling of telecommunication equipment, data transmission protocols. Based on this we can define the area of MFPN application.

Models based on apparatus can be built using CPN Tools, or any other tool that has the possibility of constructing models on the basis of Petri nets with assignment of different types of marks, expressions on the arcs, the trigger conditions of transitions and time transitions stamps. Integrated CPN Tools programming language CPN-ML allows implementation of fuzzy transitions triggering rules.

4 MFPN-based representation of fuzzy output system

The work [3] presents the models based on fuzzy Petri nets for modeling stages of aggregation of sub-condition and activation of sub-conclusions. The MFPN allows simulating of all stages of fuzzy output by extending the capabilities of fuzzy PN.

Fuzzification is the procedure of finding values of membership functions in terms of input variables on the basis of their values. Before fuzzification stage we need to define the set of input variables $V = \{\beta_1, \beta_2, ..., \beta_m\} \mid m \in N$, the set of values of input variables $V' = \{a_1, a_2, ..., a_m\} \mid m \in N$ and the set of output variables $W = \{w_1, w_2, ..., w_s\} \mid s \in N$ $a_i \in X_i$, X_i is the universe of linguistic variable β_i. The purpose of this stage is to define the values of vector $B = \{\beta_i'\} \mid \beta_i' = \mu(a_i)$. B is the range of values of membership functions of all terms of each input linguistic variable.

MFPN-based fuzzification procedure of some input variable is presented in figure 2.

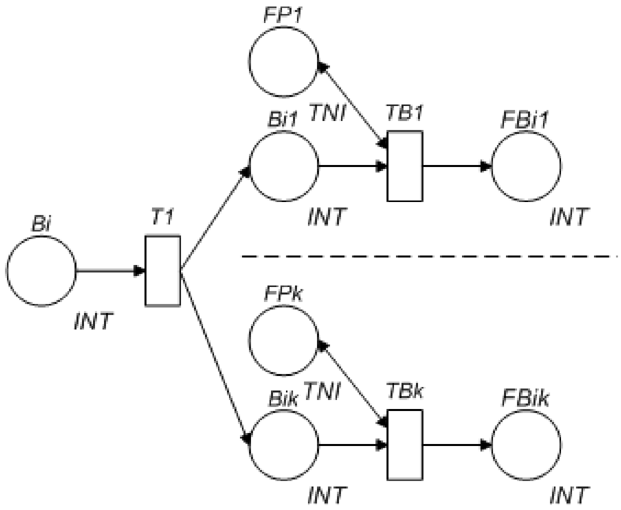

Fig. 2. MFPN-based fuzzifacation procedure.

The initialization function *I(p)* $\{FP_1, FP_2,...,FP_k\} \mid k \in N$ is determined in the form of trapezoidal fuzzy interval.

*colset TNI = record a:INT*b:INT*c:INT*d:INT;*

var tni:TNI; var i:INT;

E(Bik,TBk)= if (i<=(#a tni-(#c tni)) orelse i>=(#b tni+(#d tni))) then 1`0

else if i<(#a tni) then 1`floor((Real.fromInt(i + (#c tni) - (#a tni))/Real.fromInt(#c tni)))

else if i>(#b tni) then 1`floor((Real.fromInt((#b tni) + (#d tni) - i)/Real.fromInt(#d tni)))

else 1`1

The vector of $B = \{\beta_i'\}$ values is determined by the marked positions $\{FBi_1, FBi_2,...,FBi_k\} \mid k \in N$, as $B = \{\beta_i'\} \mid \beta_i' = m(FBi_i)$.

Accumulation is a procedure of finding of the membership function for each linguistic variable output $W = \{w_1, w_2,..., w_s\} \mid s \in N$. The membership function of the output variable consists of membership functions of the output conclusions and of sub-conclusions from the rule database. The necessity of accumulation is explained by the fact that sub-conclusions relating to one and the same output variable can be located in different rules. The procedure of accumulation for some of the output variable is shown in figure 3.

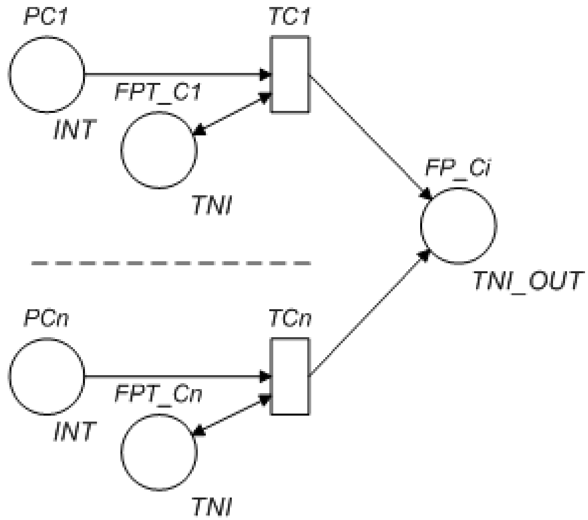

Fig. 3. The accumulation procedure based on MFPN.

Marking positions *FP_Ci* represent the accumulated value of the membership function of the output variable w_i. Marking positions $\{PC_1,...,PC_n\} \mid n \in N$ indicate the degree of truth of an individual sub-conclusion in the output of the production rules. Marking positions $\{FPT\_C_1,...,FPT\_C_n\} \mid n \in N$ represent the membership functions of the terms of some output variable w_i, in various sub-conclusions.

*colset TNI_OUT = record tni:TNI*f:INT;*

E(PCj, TCj) = i;

E(FPT_Cj, TCj) = tni;

E(TCj, FP_Ci) = 1`({tni=tni,f=i});

Accumulation is carried out according to the formula of max-association of fuzzy sets $\{PC_1,...,PC_n\} \mid n \in N$.

$$\mu_c(x) = \max\{\mu_a(x), \mu_b(x)\} \qquad (\forall x \in X)$$

Formally, the stage of accumulation is considered to be completed when all the membership functions of all the output variables are calculated, taking into account all the sub-conclusions related to these variables.

Defuzzification is a procedure of finding the crisp values for each of the output of linguistic variables $W = \{w_1, w_2,..., w_s\} \mid s \in N$ on the basis of accumulated values of the membership function. The results of defuzzification procedure $Y = \{y_1, y_2,..., y_s\} \mid s \in N$ can be used by external devices in relation to the system of fuzzy output.

It is proposed to use a combination of methods of the left $y = \min\{x_m\}$ and right $y = \max\{x_m\}$ modal values as a method of defuzzification.

$$y = \min(x_m) + \left(\frac{\max(x_m) - \min(x_m)}{2}\right) \qquad (2)$$

where x_m is the modal value of the fuzzy set of some output variable w_i. Defuzzification of the output variable w_i based on MFPN is shown in figure 4.

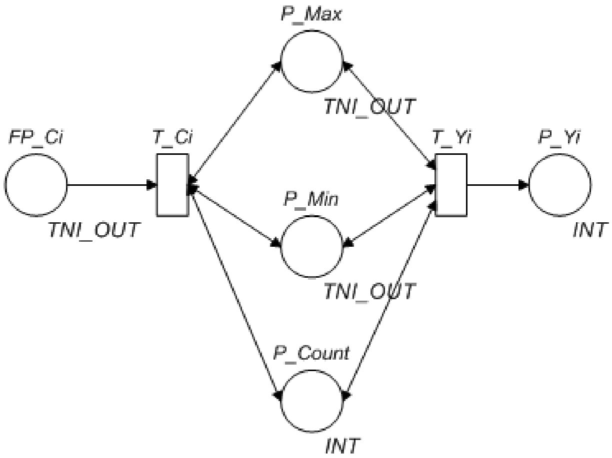

Fig. 4. Defuzzification procedure based on MFPN.

Marking the position *FP_Ci* represents the accumulated value of the membership function of the output variable w_i. Marking the positions *P_Yi* represents the crisp value y_i of the output variable w_i. Position *P_Max* and position *P_Min* represent the maximum and minimum modal values, respectively. Position *P_Count* stores the value of the number of accumulated membership functions for the output variable w_i.

var tni_out:TNI_OUT;
var tni_out_max,tni_out_min:TNI_OUT;
E(FP_Ci, T_Ci) = tni_out;
E(P_Max, T_Ci) = tni_out_max;
E(T_Ci, P_Max) =
if ((#f tni_out) > 0) andalso (#b (#tni tni_out)) > (#b (#tni tni_out_max)) then tni_out
else tni_out_max
E(P_Min, T_Ci) = tni_out_min;
E(T_Ci, P_Min) =
if ((#f tni_out) > 0) andalso (#a (#tni tni_out)) < (#a (#tni tni_out_min)) then tni_out
else tni_out_min
E(P_Count , T_Ci) = i; E(T_Ci, P_Count) = i+1; E(P_Count, T_Yi) = I; E(T_Yi, P_Count) = 0;
E(P_Max, T_Yi) = tni_out_max;
E(P_Min, T_Yi) = tni_out_min;
E(T_Yi, P_Max) = 1`{tni={a=0,b=0-100000,c=0,d=0},f=0};
E(T_Yi, P_Min) = 1`{tni={a=100000,b=0,c=0,d=0},f=0};
E(T_Yi, P_Yi) = (#a (#tni tni_out_min)) + (((#b (#tni tni_out_max))-(#a (#tni tni_out_min))) div 2)

Formally, the defuzzification stage is completed when the value y_i is calculated for each output variable w_i.

5 Conclusions

The PN apparatus is widely used for solving problems connected with the simulation of information, production and business processes. Such popularity is explained by the presence of a large number of extensions, designed to address specific applied tasks, by the developed analytical apparatus and the availability of software modeling.

The MFPN continues the trend for the extension of the modeling capabilities of PN. By combining the properties of basic expansions, it allows to simulate various technical and informational processes with the use of the apparatus of fuzzy logic for control.

References

[1] K. Jensen, *Coloured Petri Nets with Time Stamps*, Computer Science Department, Aarhus University. Denmark (1991).

[2] K. Jensen, *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*, Vol. 1, Basic Concepts, Monographs in Theoretical Computer Science, Springer-Verlag, 2nd corrected printing (1997).

[3] A.V. Leonenkov, *Nechetkoe modelirovanie v srede MATLAB i fuzzyTECH*, SPb.: BHV-Peterburg, 736 (2005).

[4] K. Jensen, L. Kristensen, Coloured Petri Nets: modelling and validation of concurrent systems, New York: Springer-Verlag, 384 (2009).

[5] K. Jensen, L.M. Kristensen, L. Wells, *Coloured Petri Nets and CPN Tools for Modeling and Validation of Concurrent Systems*, Aarhus: Department of Computer Science. University of Aarhus, 302 (2008).

[6] A.A. Sukonshchikov, D.V. Kochkin, A.N. Shvetsov, *Fuzzy and neural Petri nets*, Kursk, 209 (2019).

[7] A.N. Shvetsov, A.A. Sukonschikov, D.V. Kochkin, I.A. Andrianov, *Situational intelligent systems to support decision-making*, Kursk, 251 (2018).