Computing and Informatics, Vol. 24, 2005, 563–576

FUZZY-PETRI-NET REASONING SUPERVISORY CONTROLLER AND ESTIMATING STATES OF MARKOV CHAIN MODELS

Georgi M. Dimirovski

Department of Computer Engineering
Dogus University of Istanbul
Zeamet Sk.1, Acibadem, Kadikoy
TR-34722 Istanbul
Republic of Turkey

Institute of Automation & Systems Engineering
SS Cyril and Methodius University
Karpos 2, Rugjer Boskovic BB
MK-1000 Skopje
Republic of Macedonia
e-mail: gdimirovski@dogus.edu.tr, dimir@etf.ukim.edu.mk

Revised manuscript received 12 October 2005

Abstract. Markov chain models are efficient tools for representing stochastic discrete event processes with wide applications in decision and control. A novel approach to fuzzy-Petri-net reasoning generated solution to initial or another state in Markov-chain models is proposed. Reasoning is performed by a fuzzy-Petri-net supervisory controller employing a fuzzy-rule production system design and a fuzzy-Petri-net reasoning algorithm, which has been developed and implemented in C++. The reasoning algorithm implements calculation of the degrees of fulfilment for all the rules and their appropriate assignment to places of Petri net representation structure. The reasoning process involves firing active transitions and calculating degrees of fulfilment for the output places, which represent propositions in the knowledge base, and determining of fuzzy-distributions for output variables as well as their defuzzified values. Finally, these values are transferred to assign the state of Markov-chain decision model in terms of transition probabilities.

Keywords: Fuzzy-Petri-net, control, Markov chains

1 INTRODUCTION

The synergy of advanced technologies in applied computing, communication, control, and decision during the last couple of decades has given rise to highly complex, technological, dynamic systems governed by some composition of time-driven and event-driven stochastic dynamics and referred to as discrete event systems. Their features have enhanced applications of Markov chain models and Petri-net models that complement each other, because they provide a framework for investigation studies of very many discrete event systems. Although in principle, a Markov chain model can be always solved, practice has demonstrated that this task is extremely difficult and steady-state (stationary) solutions in terms of state transition probabilities are sought and practically exploited. Even in the cease when steady-state solutions are sought there is no systematic way of how the initial state probability vector is to be chosen or determined [1]. It is known, on the other hand, that a Markov-chain decision model represents a discrete-event system in terms of sequence of state random values the probabilities of which at a time interval depend upon the random values at the previous time instant only. The controlling factor in a Markov chain is the transition probability, which is a conditional probability for the system to go to a particular next state given the current state of the system. In the present study we have explored a possible application of a two-level decision and control architecture, a supervisory controller that implements a fuzzy-Petri-net reformulation of Saridis' organizing intelligent controller [2, 3] following his principle of increasing intelligence with decreasing precision [4]. The supervisory controller has been constructed as a fuzzy-Petri-net production rule system [5–7]. Its software implementation in C++ is described in [6] by means of both programmer's and user's view. Petri net formalism provides considerable advantages when used in hybrid decision and control systems because of its great power for representation and modeling of parallel and concurrent processes, while fuzzy system formalism does the same with respect to reasoning by processing the respective fuzzy-rule inference systems [8–21]. The formalism of Petri nets (PN) can be used to model fuzzy-rule based systems by simply identifying some elements (places and transitions) and features (marking function) of Petri-net's formalism with the basic elements of a fuzzy-rule based knowledge base (KB) such as propositions, degree of truth and implication relationships. The fuzzy-Petri-net (FPN) formalism employed here has achieved this due to more specific terms such as association of the KB propositions and places in the FPN through introducing an appropriately defined bijective function, and association of the KB transitions and degrees of truth. Furthermore, a formal separation between the representational scheme (the FPN itself) and the associated discrete dynamic process (data driven evaluation algorithm) is established, yet it is not included as a part of the FPN model. A more adequate handling of multi-propositional rules has been introduced and implemented too. Still degrees of truth of the rules remained numerical values, and the chaining is still done at the value level and therefore some drawbacks have been identified to be present. This model includes the handling of true fuzzy production rule system by taking degree of truth of the implication rules.

2 ON MARKOV CHAIN MODELS OF DISCRETE EVENT SYSTEMS

In the category of discrete-event processes, there may be processes that cause different types of changes and usually cause leap changes of states following occurrence of some event(s). For instance, such are processes of planning and of forecasting and decision making analysis, which examine structural connections between state-transitions and event probabilities of the respective sets of attainable states and admissible events. Through this analysis it is possible to obtain the schedule of transitions and the basic units of time, which are needed for each transition. Solving problems like these requires the usage of state-transition and event-probability matrices embedded in Markov chain models, often a finite discrete-time Markov chain of events involving respective state transitions [1].

The work presented here is confined to discrete-time homogeneous Markov chain that is a stochastic process characterized by finite number of states; Markovian transitions (possessing Markov memory-less property), and stationary state-transition probabilities. Thus, a Markov chain is a sequence of random values such as the one known as a non-repeating random walk. The model of the problem at hand it suggest assuming an $n \times n$-matrix of probabilities, P, such that

$$p_{ij} \geqslant 0, \sum\nolimits_{j} p_{ij} \leqslant 1, i = 0, 1, 2, \dots \tag{1}$$

and thus generating the following array:

$$g_i = 1 - \sum\nolimits_{j} p_{ij}, i = 0, 1, 2, \dots \tag{2}$$

P can approximately describe a discrete Markov chain where the states of the chain are N integers. The element p_{ij} gives the transition probability for the random walk from state i to state j. The probability that the walk will terminate after state i is given by g_i. As long as the vector g, encompassing all these probabilities, is not zero the walk will eventually terminate. In fact, a Markov chain is a sequence of random values, whose probabilities at a next time instant of a given time interval depend upon the value at the immediately previous time instant (no state memory) regardless of the time spent in the current state (no age memory):

$$\begin{aligned} &P\left[X_{k+1} = x_{k+1} \middle| \; X_k = x_k, \, X_{k-1} = x_{k-1}, \dots, X_0 = x_0\right] \\ &\equiv P\left[X_{k+1} = x_{k+1} \middle| \; X_k = x_k\right]. \end{aligned} \tag{3}$$

The governing factor in a Markov chain is the transition probability, which is a conditional probability for the system to go to a particular new state, given the current state of the system with the assumption the initial one is known,

$P[X_0 = x_0] = q_{e_i}$, $\forall e_i \in E$, $i \in [0, N]$ with E being a set of admissible events. For many problems, the Markov chain obtains the much-desired importance sampling meaning fairly efficient estimates can be obtained if the proper transition probabilities are determined. In terms of stochastic timed automata formalism, in which E represents the set of admissible events, X the set of attainable states, $G(x)$ the set of enabled events defined with $G(x) \subseteq E$ for all $x \in X$, $G = \{G_i : i \in E\}$ a stochastic clock structure, $p = p(x^\nu;\, x, e_\nu)$ a state transition probability defined for all $x^\nu, x \in X$, $e_\nu \in E$ such that $p = p(x^\nu;\, x, e_\nu) = 0$ for all $e_\nu \notin G(x)$, and $p_0 = p_0(x)$probability mass function $P[X_0 = x]$, $x \in X$ of the initial state X_0, a Markov chain model is represented as

$$M_{SA} = (E, X, \ , p, p_0, C) \tag{4}$$

with $e_\nu \in G(x)$ representing the triggering event. In turn, the transition probability is an aggregate over all enabled events $e_i \in G(x)$ that may cause the transition from state $x \in X$ to state $x^i \in X$, which may depend on time, and it follows according to the rule of total probability

$$p_k(x^i;\, x) = P\left[X_{k+1} = x^i \middle| \, X_k = x\right] = \sum_{e_i \in G(x)} p(x^i;\, x, e_i) \cdot p(e_i, x) \tag{5}$$

where $p(e_i, x)$ is the probability that event e_i occurs at state x. Should the state space be represented by non-negative integers like in a random walk sequence, a Markov chain emanates from Chapman-Kolmogorov equation

$$p_{ij}(k, k+n) = \sum_{\text{all}\, r} p_{ir}(k, r) p_{rj}(r, k+n), k < r \leqslant k+n \tag{6}$$

for the event $P[X_{k+1} = j |\, X_k = i]$ conditioned on $P[X_r = r]$ with any r such that $k < r \leqslant k + n$.

It is therefore possible, in terms of an application of the FPS reasoning of the FPN supervisor that makes membership degrees evaluation largely satisfying memory-lees property, to assign the finally obtained values for output variables of the FPN supervisor to take the role of initial state probabilities of the observed Markov chain model. Thus, as a first step towards the desired resolution, a Markov chain with obligatory state probabilities, obtained from an expert system with a purpose after its evaluation processing has been completed, has been created. This way, it is believed, it even becomes possible to apply control actions on events in the Markov chain models, thus influencing its discrete-event transition evolution. This is discussed in the two subsequent sections.

3 ON FUZZY PRODUCTION RULE SYSTEM AND PROCESS OF FUZZY-PETRI-NET REASONING

Although different conceptualization of fuzzy production rule systems may be found in the literature, in here a rather generic representation of the rules is considered [5]. Namely, the fuzzy-rules that make up the knowledge base (KB) in a fuzzy-rule production systems (FPS) are outlined as follows:

$$\begin{array}{c}\text{Rule } R^r: \text{ If } \mathbf{X}_1^r \textbf{ IS } \mathbf{A}_1^r \text{ and}\ldots\text{and } \mathbf{X}_{Mr}^r \textbf{ IS } \mathbf{A}_{Mr}^r \\ \text{Then } \mathbf{X}_{Mr+1}^r \textbf{ IS } \mathbf{B}_1^r \text{ and}\ldots\text{and } \mathbf{X}_{Mr+Nr}^r \textbf{ IS } \mathbf{B}_{Nr}^r(\tau^r).\end{array} \tag{7}$$

Here, the bold presented symbols are the fuzzy propositions and τ^r, which are defined as functions of the type $[1,0] \to [1,0]$, represent linguistic values of the truth-variable that qualifies the rules.

The set of fuzzy rules, constituting a fuzzy-rule knowledge base in a fuzzy production rule system, is considered next through its projection onto a Petri-net (PN) representation structure, the respective bipartite graph $N = (T, P, A)$, where $P = (p_1, p_2, p_k, p_K)$ is the set of places, $T = (t^1, t^2, t^j, t^J)$ is the set of transitions, and $A \subseteq \{T \times P\} \cup \{P \times T\}$ is the respective set of directed arcs. For this purpose, the places of the PN are being identified with the propositions of the KB by means of the following bijective function :

$$P \to PR, p_k \to \alpha(p_k) = pr_k, k = 1, \ldots, K, \tag{8}$$

where $PR = \{pr_k\}$ is the set of propositions and K is the number of propositions in the KB. In the case where a proposition is found several times in different fuzzy rules of the KB, a different place will be assigned to it for each of these appearances in the KB.

The meaning of the transitions is more complex and involved to interpret because of the linking rules. In this work, basically the representation is in terms of union of two types of transitions: $T = T^R U T^C = \{t_1, \ldots, t_R, t_{R+1}, \ldots, t_{R+C}\}$. Subset T^R includes the transitions associated with each one of the rules that make up the fuzzy-rule KB, whereas subset T^C includes the transitions that are associated with the existence of links between propositions. Hence the input and the output functions over set T ought to be defined

$$I : T \to \phi(P), \tag{9}$$

$$O : T \to \phi(P), \tag{10}$$

such that these associate to each transition the set of places which constitute its input and output, respectively. It is these functions that can have a different interpretation depending on the subset of T in which they are considered:

$$\text{If } t^j \in T^R, \forall p_i \in P, p_i \in I(t^j) \iff a(p_i) \in \text{ Antecedent part of } R^j \tag{11}$$

$$\text{If } t^j \in T^R, \forall p_i \in P, p_i \in O(t^j) \iff a(p_i) \in \text{ Consequent part of } R^j \tag{12}$$

$$\text{If } t^j \in T^C, p_i \in I(t^j), p_k \in O(t^j) \iff a(p_i) \text{ is linked with } \alpha(p_k) \tag{13}$$

Therefore a single transition $t_j \in T^C$ will exist for each of the intermediate variables X_j of the fuzzy-rule knowledge base.

In terms of graphs, the representation of the fuzzy Petri-net is defined as follows:

$$A = \bigcup_{t^j \in T} \left\{ t^j \times O(t^j) \right\} \cup \left\{ I(t^j) \times t^j \right\} \tag{14}$$

where also a truth function, f, that assigns to each $t^j \in T^R$ the linguistic truth value associated with the respective rule R^j has been defined as follows:

$$f : T^R \to V, t^j \to f(t^j) = \tau^j. \tag{15}$$

Here, V represents the set of linguistic values of the linguistic truth variable. It should be noted, in addition, that in this fuzzy Petri-net model the place p_l is immediately reachable from place p_k if:

$$\exists t^j \in T / p_k \in I(t^j) \textit{ and } p_l \in O(t^j). \tag{16}$$

The adjacent transitions associated with chains will represent the multiple link situations: several rules establish inferences over the same variable and one or more later rules make use of this variable in its (their) antecedent part. In this case, a transition will be associated with each chain.

4 PROCESS OF FUZZY-PETRI-NET REASONING IN MARKOV MODEL STATE ESTIMATION

It should be noted that the fundamental notion of executing a fuzzy-rule knowledge base represented by the respective Petri-net bipartite graph does coincide with is that of a marked PN model. Marking indicates that the degree of fulfilment (DOF) of the associated proposition is known, so this proposition can be used in the process of obtaining new references. It will be necessary for the DOFs of the different propositions to be available all the time and be handled as appropriate. The latter required a well defined fulfilment function

$$g : P \to [0, 1] \tag{17}$$

be introduced such that it assigns to each place a real value:

$$g(p) = DOF(\alpha(p)). \tag{18}$$

In the above PN representation structure, tokens are transferred from some places to others by means of the activation of transitions, following a basic rule: A transition $t_j \in T$ is active (and will fire) if every $p_i \in I(P)$ has a token.

When during the process of firing a transition the token of the input places is removed, the information obtained about the DOF of that propositions is preserved in the fulfilment function. The firing of an active transition $t_j \in T^R$ is equivalent to the application of a rule in the process of evaluating the KB. The activation of $t_j \in T^C$ is equivalent to knowing (whether it will be through previously performed inferences or through observation), the DOF of propositions $\alpha(p_i), \forall p_i \in I(t^j)$. In this case, the DOF for propositions $\alpha(p_k), \forall p_k \in O(t^j)$ is determined not by the application of rules of the KB, but by essentially the same method as the one used to determine the DOF of a proposition with observed input distribution values. Most of the operations participating in this calculation can be carried out a priori, leading to a significant simplification of the execution process. When all DOF's of the antecedent part of a rule are known and it is executed, the marking function will have tokens placed in all of the input places of the corresponding transition, activating it and causing it to fire, which will produce new markings generated by PN marking function.

The initial marking map M in the PN representation of the KB of the fuzzy production rule system can be defined as follows:

$$M : p \to \{0, 1\},\ p_i \to M(p_i) = \{0, if\ g(p_i) \text{ is unknown; and } 1,\ \text{otherwise}\}. \quad (19)$$

The marking mapping function makes explicit the requirement that the DOF of a set of propositions must be known before an evaluation of the KB can be carried out. From a given marking map M, the firing of a transition t^j may produce a new marking map M^*. The evolution of the marking mappings of a PN, hence of a FPN too, is represented by the respective transition function, t_{FPN}, defined as follows:

$$t_{FPN} = M \times T \to M, (M, t^j) \to M^* \quad (20)$$

$$M^* = \{0,\ \text{if}\ \ p_i \in I(t^j);\ 1,\ if\ \ p_i \in O(t^j);\ \text{and}\ M(p_i)\ \text{otherwise}\}. \quad (21)$$

In mapping function (20–21), M represents the set of all possible marking maps of the PN and FPN model, respectively. The process of executing a KB can be understood as the "propagation" of possibility distributions through the KB, via implication operations (which permit "propagating" distributions from the antecedent part of a rule to the consequent part of the same rule) and via links (which "connect" the consequent part of one or several rules to the antecedent part of other(s)). This evaluation process is carried out following a certain order, which determines at any moment in time the rule(s) that may be applied. The process finishes with the operation of aggregating all the possibility distributions inferred for each output variable into a single final possibility distribution.

Without loss of generality, for a fuzzy-rule KB consisting of only two chained rules, R^S and R^C, which are linked by one of fuzzy-set defined antecendents, to this end by making use of the above defined bijective function the related places and propositions are obtained:

$$P = \{p^r_{mr} | mr = 1, \ldots, M_r + N_r,\ r = S, T\} \quad (22)$$

$$PR = pr^r_{mr} = \{"X^r_{mr}\ IS\ A^r_{mr}",\ mr \leq M_r;\ "X^r_{mr}\ IS\ B^r_{mr-Mr}", mr > M_r\}. \quad (23)$$

Furthermore, given the relative simplicity of the KB, the transitions for the rules and links are obtained as follows:

$$T^R = \{t^S, t^T\} \quad (24)$$

$$T^C = \{t^3\}. \quad (25)$$

In the sequel, the focus is on the process of obtaining the DOF that corresponds to proposition $\alpha(p^T_1)$ from the DOF of $\alpha(p^s_{Ms+1})$, i.e. $g(p^T_1)$ from $g(p^s_{Ms+1})$. Then it is observed that

$$\underline{b}^s_{1,i} = \tau^s(g(p^s_{Ms+1}) \wedge b^s_{1,i}), i = 1, \ldots, I \quad (26)$$

where $B^s_1 = \{b^s_{1,i}\}$ is the possibility distribution associated with linguistic value B^s_1 in proposition $\alpha(p^s_{Ms+1})$. The DOF will be

$$g(p^T_1) = V[\tau^s(g(p^s_{Ms+1}) \wedge b^s_{1,i}) \wedge a^T_{1,i}]; \quad (27)$$

$a^T_{1,i}$ is the possibility distribution of linguistic value in propositions.

Let now the more general case of FKB be observed in which several rules $R^1, \ldots, R^S$ perform inference over a variable, and the same variable is in the antecedent part of at least one later rule R^T. Typically, one obtains

$$\begin{aligned} R^1 &: \text{IF } X^1_1 \text{ IS } A^1_1 \text{ AND}\ldots \text{THEN } X^1_{M1+1} \text{ IS } B^1_1 \text{ AND}\ldots(\tau^1) \\ R^2 &: \text{IF } X^2_1 \text{ IS } A^2_1 \text{ AND}\ldots \text{THEN } X^2_{M2+1} \text{ IS } B^2_1 \text{ AND}\ldots(\tau^2) \\ R^S &: \text{IF } X^S_1 \text{ IS } A^S_1 \text{ AND}\ldots \text{THEN } X^S_{MS+1} \text{ IS } B^S_1 \text{ AND}\ldots(\tau^S) \\ R^T &: \text{IF } X^T_1 \text{ IS } A^T_1 \text{ AND}\ldots \text{THEN } X^T_{MT+1} \text{ IS } B^T_1 \text{ AND}\ldots(\tau^T) \end{aligned} \quad (28)$$

with the linking relationship

$$X^1_{M1+1} = X^2_{M2+1} = \cdots = X^S_{Ms+1} = X^T_1. \quad (29)$$

Following a procedure that is analogous to the previous one we will obtain the DOF for proposition pr^T_1:

$$g(p^T_1) = V[V[\tau^s(g(p^s_{Ms+1}) \wedge b^s_{1,i})] \wedge a^T_{1,i}]. \quad (30)$$

Now it is possible to outline the actual reasoning algorithm. Basically, it comprises a two-stage computing process: the stage of defining the marking function, and that of producing the DOFs of the corresponding propositions and firing of the active transitions. These stages are sequentially repeated until there are no more active transitions; at this moment the inference process will be ended. Finally, an aggregation-assignment of a single possibility distribution to each output variable is performed.

Assume that IP and OP represent the sets that group input and output places, respectively. Then the outline of the reasoning algorithm is as follows:

Step 1. Initially, only the DOFs of the propositions that operate on input variables are assumed, that is, those associated with input places, are known. Therefore the initial marking function will be:

$$M(p_i) = \{0,\ if\ p_i \notin IP;\ and\ 1,\ if\ p_i \in IP. \tag{31}$$

Step 2. We fire the active transitions. Let t^j be any active transition; that is,

$$t^j \in T | \forall p_k \in I(t^j),\ M(p_k) = 1. \tag{32}$$

In fact, the transition function $t_{FPN} = M \times T \to M$, as defined with (21), defines the successive marking functions as processing the algorithm evolves. In turn, the corresponding DOFs are obtained as follows:

$$If\ t^j \in T^R,\ g(p_i) = \Lambda g(p_k), \forall p_i \in O(t^j) \tag{33}$$

$$If\ t^j \in T^C, g(p_i) = V\{[\tau^{rk}(g(p_k))o^{rk}\mu_{pk,pi}]\}, \forall p_i \in O(t^j). \tag{34}$$

Step 3. Go back to step 2, while:

$$t^j \in T | M(p_i) = 1, p_i \in I(t^j). \tag{35}$$

Step 4. For each output variable X, its associated possibility distribution $\underline{B} = \{\underline{b}_i\}$, $i = 1, \ldots, I$, is found

$$\underline{b}_i = V\{\tau^r(g(p_n^r))o^r\tau^r(b_{n,i}^r) \tag{36}$$

with the set P_X of places associated with propositions

$$P_X = p_n^r \in P | \alpha(p_n^r) = \text{"}X\ IS\ B_n^r\text{"} \tag{37}$$

in which inferences over X are carried out.

The next set of simulation results is given for the purpose of illustration of the above-described fuzzy-Petri-net reasoning process and the kind of results that may be obtained. These are obtained via reasoning with the following five-rule fuzzy-rule knowledge base involving seven fuzzy membership grades as given bellow:

$$R^0 : \text{IF } X1 = LP\backslash 0.12 \text{ AND } X6 = SP\backslash 0.10 \text{ THEN } X2 = ZO,$$
$$\Rightarrow dof(A00) = 0.64; dof(A01) = 0.70$$

$$R^1 : \text{IF } X1 = LN\backslash 0.06 \text{ AND } X3 = SN\backslash 0.12 \text{ THEN } X2 = ZO,$$
$$\Rightarrow dof(A10) = 0.82; dof(A11) = 0.64$$

$$R^2 : \text{IF } X2 = SN\backslash 0.07 \text{ AND } X5 = LN\backslash 0.20 \text{ THEN } X7 = ZO,$$
$$\Rightarrow dof(A20) = 0.79; dof(A21) = 0.40$$

$$R^3 : \text{IF } X2 = SN\backslash 0.15 \text{ AND } X5 = SN\backslash 0.25 \text{ THEN } X4 = ZO,$$
$$\Rightarrow dof(A30) = 0.55; dof(A31) = 0.25$$

$$R^4 : \text{IF } X4 = SP\backslash 0.05 \text{ AND } X3 = SN\backslash 0.20 \text{ THEN } X7 = ZO,$$
$$\Rightarrow dof(A40) = 0.85; dof(A41) = 0.40.$$

The resulting fuzzy-Petri-net chaining is depicted in Figure 1 whereas the computed aggregate fuzzy distribution is presented in Figure 2. Final values will depend, of course, on the method of defuzzification employed as indicated with the numerical results in Figure 2.

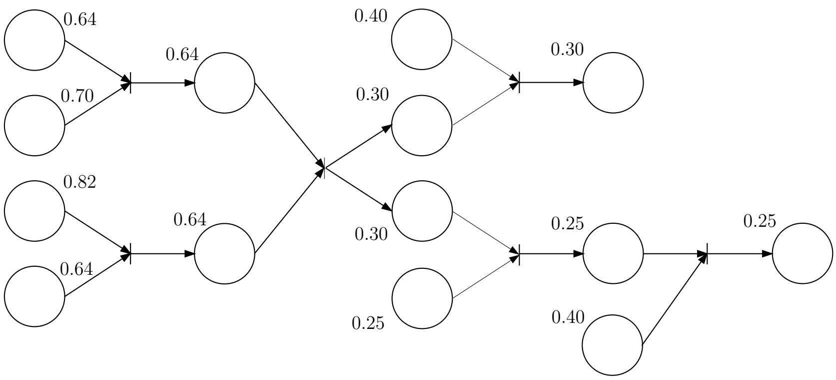

Fig. 1. Graphical representation of the fuzzy-Petri-net structure with calculated DOFs of propositions (places) representing the rule chaining in the course of inference reasoning

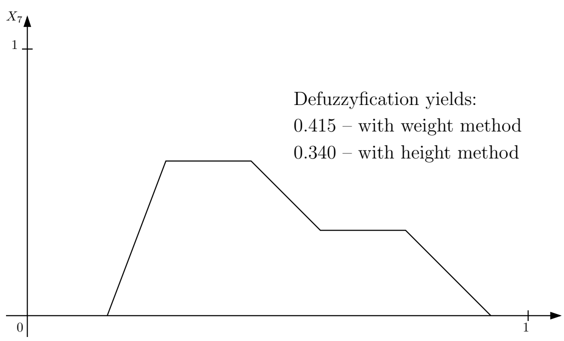

Fig. 2. Graphical presentation of the resulting membership grade function for the output variable and two cases of numerically resulting singleton to assign state probability in Markov chain

5 CONCLUSION

The controlling factor in a Markov chain model is the transition probability that is a conditional probability for the system to go to a particular new state, given the current state of the system. It was shown in this paper that distribution values obtained from the fuzzy-Petri-net reasoning system can be assigned to probabilities of the states, in particular for unique definition of the initial state of the Markov chain model, which makes it solvable. Albeit it is believed that a similar procedure may be developed to assign state transition probabilities, this case is not well understood and it is a task for future research. To some extent, this may serve the purpose of controlling the feasible events in Markov chain models. In was shown in this work that by making use of the complementing formalism of Petri-net bipartite graphs, a model of fuzzy-rule production system as a model for inference and conclusion chaining can be constructed. Furthermore, this system is compatible with well-structured algorithms for data-driven execution of fuzzy-rule knowledge bases. This process is based on a FKB execution approach through the compositional rule of inference, so that most of the computational load is put to the design stage, and not to execution stage. This allows the complexity of the execution algorithms to remain independent of the discretization of the discourse universes over which the linguistic variables to be manipulated in the fuzzy production systems are defined. Despite the fact that the analysis of the whole process and the description of the algorithms is carried out for a sup-min compositional rule of inference, the same results are valid for the sup-prod rule, although with less flexibility in the definition of the linguistic truth values that qualify the rules. We have also used the Petri net formalism in order to obtain a formal structure that permits the definition of algorithms for carrying out inferences in different situations.

Acknowledgment

The author gratefully acknowledges the contributions of his former doctoral student Zoran M. Gacovski, who has developed the experimental application software [7] in C++, and implemented the author's ideas demonstrating therewith their correctness and working potential.

REFERENCES

[1] Cassandras, C. G.—Lafortune, S.: Introduction to Discrete Event Systems. Boston, MA: Kluwer Academic Publishers, 1999, chapters 4 and 7.

[2] Saridis, G. N.—Valavanis, K. P.: Analytical Design of Intelligent Machines. Automatica, Vol. 24, 1988, No. 2, pp. 123–133.

[3] Saridis, G. N.: Analytic Formulation of the Principle of Increasing Precision with Decreasing Intelligence for Intelligent Machines. Automatica, Vol. 25, 1989, No. 3, pp. 461–467.

[4] Stellakis, H. N.—Valavanis, K. P.: Fuzzy-Logic Based Formulation of the Organizer of Intelligent Robotic Systems. J. of Intell. & Robot. Syst., Vol. 4, 1991, No. 1, pp. 1–24.

[5] Dimirovski, G. M.: Fuzzified Petri-Nets and Their Application to the Organizing Controller. In Computational Intelligence: Soft-Computing and Fuzzy-Neuro Integration, O. Kaynak, L. A. Zadeh, B. Turksen and I. J. Rudas, Eds., NATO Series on Computer and Systems Sciences, Vol. F 162. Berlin, DE: Springer-Verlag, 1998, Chapter 3.5, pp. 260–282.

[6] Dimirovski, G. —Gacovski, M. Z. M.—Vukobratovic, M. K.: Reformulation of Organizing Supervisory Controller via Parameterized Fuzzy-Petri-Nets. In Proceedings of 14$^{\text{th}}$ World Congress of IFAC – Volume A, T. Y. Chai, M. Zaremba, P. Bernus and A. Villa, Eds., Oxford, UK: Pergamon Elsevier Science Ltd, 1999, pp. 391–396.

[7] Gacovski, Z. M.: Intelligent Supervisory-Organizing Controllers Based on Parameterized, Fuzzy and Stochastic Petri Nets. Ph. D. thesis; supervisor G. M. Dimirovski. SS Cyril and Methodius University, Skopje, MK, 2002.

[8] Zadeh, L. A.: Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. IEEE Trans. on Syst., Man & Cybern., Vol. 3, 1973, No. 1, pp. 28–44.

[9] Zadeh, L. A.: Knowledge Representation in Fuzzy Logic. IEEE Trans. on Knowledge Data Eng., Vol. 1, 1988, No. 1, pp. 96–105.

[10] Looney, C. G.: Fuzzy Petri Nets and Applications. In Fuzzy Reasoning in Information, Decision and Control Systems, S. G. Tzafestas and N. Venetsanopoulos, Eds. Norwll, MA: Kluwer Pub., 1994, pp. 511–527.

[11] Y. Yuan—Shaw, M. J.: Induction on Fuzzy Decision Trees. Fuzzy Sets & Syst., Vol. 69, 1995, pp. 125–139.

[12] Chen, S. M.—Ke, J. S.—Chang, J. F.: Knowledge Representation Using Fuzzy Petri Nets. IEEE Trans. on Knowledge Data Eng., Vol. 2, 1990, No. 3, pp. 311–319.

[13] Grag, M. L.—Ahson, S. I.—Gupta, P. V.: A Fuzzy Petri Net for Knowledge Representation and Reasoning. Inf. Process. Lett., Vol. 39, 1991, pp. 165–171.

[14] Bugarin, A. J.—Barro, S.: Fuzzy Reasoning Supported by Petri Nets. IEEE Trans. on Fuzzy Systems, Vol. 2, 1994, No. 5, pp. 135–150.

[15] Pedrycz, W.—Gomide, F.: A Generalized Fuzzy Petri Net Model. IEEE Trans. on Fuzzy Systems, Vol. 2, 1994, No. 4, pp. 295–301.

[16] Yeung, D. S.—Tsang, E. C. C.: Fuzzy Knowledge Representation and Reasoning Using Fuzzy Petri Nets. Expert Syst. Appls., Vol. 7, 1994, pp. 281–290.

[17] Scrapelli, H.—Gomide, F.—Yager, R. R.: A Reasoning Algorithm for High-Level Fuzzy Petri Nets. Vol. 4, 1996, No. 3, pp. 282–293.

[18] Chen, S. M.: A Fuzzy Reasoning Approach for Rule-Based Systems Based on Fuzzy Logic. IEEE Trans. on Syst., Man & Cybern. B, Vol. 26, 1996, No. 5, pp. 769–778.

[19] Janikow, C. Z.: Fuzzy Decision Trees: Issues and Methods. IEEE Trans. on Syst., Man & Cybern. A, Vol. 28, 1998, No. 1, pp. 1–14.

[20] Yeung, D. S.—Tsang, E. C. C.: A Multilevel Weighted Fuzzy Reasoning Algorithm for Expert Systems. IEEE Trans. on Syst., Man & Cybern. A, Vol. 28, 1998, No. 2, pp. 149–158.

[21] Li, X.—Rosano, F. L.: Adaptive Fuzzy Petri Nets for Dynamic Knowledge Representation and Inference. Expert Syst. Appls., Vol. 19, 2000, No. 3, pp. 281–290.

Georgi M. Dimirovski is a Professor of Automation & Systems Engineering at SS Cyril & Methodius University, Skopje, and of Computer Science & Information Technologies at Dogus University, Istanbul, and also an Invited Professor at Istanbul Technical University. He is a Foreign Member of Academy of Engineering Sciences in Belgrade, Serbia and Montenegro, and Senior Member of the IEEE, USA. He obtained his degrees Dipl.-Ing. in EE (Control) from SS Cyril & Methodius University of Skopje, Rep. of Macedonia, and M. Sc. in EEE from University of Belgrade, Rep. of Serbia – then both in the former S. F. R. of Yugoslavia – in 1966 and 1974, respectively. In 1977 he obtained Ph. D. by Research from University of Bradford, UK, where he also held a postdoctoral position, in 1979. He was a Visiting Professor at University of Bradford, in 1984 and 1986, and at Wolverhampton University, in 1988 and 1991, as well as a Senior Fellow & Visiting Professor at Free University of Brussels, in 1994, and at Johannes Kepler University of Linz, in 2000. He has had longer or shorter term academic visits with seminars to more than 20 universities across Europe and in Shenyang, P. R. of China. He has developed a number of undergraduate and graduate courses in various areas of control and applied computing at universities in Skopje and in Bradford, Istanbul, Linz, and Zagreb too. He has successfully supervised 2 postdoctoral, 15 Ph. D. and 27 M. Sc. as well as more than 300 graduation students' projects. He was invited member of Steering Committee of European Science Foundation's Scientific Programme on Control of Complex Systems led by K. J. Astroem and M. Thoma. He has edited 4 proceedings volumes of the IFAC and one of the IEEE series. He has contributed 4 invited papers in research monographs, and published more than 30 journal articles and more than 200 conference papers in proceedings of the IFAC and the IEEE alone. He served on the Editorial Boards of Proceedings of Inst. Mech. Engineers J. of Systems & Control Engineering (UK), Automatika (former YU), and Information Technologies & Control (BG), and was Editor-in-Chief of the J. of Engineering (MK). Currently, he is serving on the Editorial Boards of Facta Universitatis Series EE (University of Nis, SR-MN) and IU J. of Electrical & Electronics Engineering (University of Istanbul, TR). In 1985, he founded the Institute of Automation & Systems Engineering at Faculty of Electrical Engineering of SS Cyril & Methodius University. In 1981 he has founded the ETAI Society – Macedonian IFAC NMO – and was its first President, and during 1985–1991 was President of former Yugoslav IFAC NMO, Yugoslav Association ETAN. Also, he was among the founders of IEEE R. of Macedonia Section in 1995, and Turkish Chapter of IEEE Computational Intelligence Society in 2002. During 1988–1993, he served on Executive Council of European Science Foundation (ESF) in Strasbourg. He served the IFAC in capacity TC Vice-Chair (1996–2002) and Chair of TC 9.3 on Developing Countries (2002–2005). He has served on the IPC's for many IEEE and IFAC conferences and several congresses. Currently, he is a Member of Technical Board and Chair of CC-9 on Social Systems of the IFAC.