Intelligent Systems : Unified approach to Knowledge Representation, Analysis and Implementation based on Fuzzy Petri Nets

From: Proceedings, Fourth Bar Ilan Symposium on Foundations of Artificial Intelligence. Copyright © 1995, AAAI (www.aaai.org). All rights reserved.

Dr. Valery Cherniaev

Department of Math.and Computer Sci.,
Bar- Ilan University,
Ramat- Gan, 52900, Israel
chernyae@bimacs.cs.biu.ac.il

Abstract. The design and implementation of Fuzzy Intelligent systems are studied. These systems operate in conditions of uncertainty, imprecision and ambiguity. Their behavior is described by Fuzzy Algorithms (FA) in the form of Fuzzy production rules. A new approach to design of such Fuzzy systems is suggested. The approach is based on a following : (a) Fuzzy Petri Nets (FPN) as new model and tools for a formal representation and modeling of given Fuzzy systems; (b) isomorphism between two representations of Fuzzy algorithms (in terms of its Fuzzy PN and its Fuzzy Finite Automata) as a base for their Hardware realization. The approach proposed here guaranties a simple and an efficient execution of following main stages of an arbitrary Fuzzy system design : (i) Formal description of corresponding FA rules in the form of Fuzzy PN; (ii) Analysis of FA rules; (iii) Hardware implementation of a given Fuzzy system using different element bases. The stages (ii) and (iii) use a structural analysis (synthesis) of the FA in terms of a functional decomposition (composition) of a corresponding Fuzzy PN from a set of elementary PN-components. The method of mapping from a given Fuzzy algorithm to its Hardware implementation is developed.

Introduction

The past 15 - 20 years demonstrate stable growth of Intelligence of control, management and decision- making systems in different applications such as Robotics, Flexible Manufacturing, Telecommunications, Transport, Medicine and Knowledge Bases. A special term MIQ(Machine Intelligence Quotient)even appeared to measure an intelligence of such systems [1,2]. At present there is a need in methods and tools for Knowledge and Data Representation to model such *Intelligent Systems* (IS) during the different steps of their design. Such modeling includes formal description, verification (validation), implementation and testing both for a software and a hardware realizations of IS.

Good candidates for such tools are well- known Petri Nets (PN) and their numerous extensions and generalizations such as a High-level PN(coloured,hierarchical, numerical, stochastic timed etc.)[3]. These PN have been widely and successfully used for a top-down methodology of design, analysis, synthesis and implementation of systems, which behavior is adequately described by a corresponding classes of algorithms (deterministic, non-deterministic, stochastic etc.).

From: Proceedings, Fourth Bar Ilan Symposium on Foundations of Artificial Intelligence. Copyright © 1995, AAAI (www.aaai.org). All rights reserved.

All of these algorithms utilize *Crisp Data*. However a lot of industrial applications including human-machine systems needs the modeling tools for systems that operate in conditions of *uncertainty, imprecision* and *ambiguity*. Furthermore, these conditions are possibilistic rather than random, probabilistic. The majority of these systems are successfully described by Fuzzy Algorithms. They are known as **Fuzzy Systems** (FS) and include a variety of Fuzzy Logic Controllers and Fuzzy Expert Systems with numerous applications [4 - 7] .

The theoretical fundamentals of FS representation and modeling are *Fuzzy sets* and *Fuzzy logic* theory. During the last few years, *Fuzzy logic* together with *Neural Networks* theory and *Probabilistic Reasoning* have been created a perspective direction in AI, named as *SOFT COMPUTING* [1,2].

Architecture of any FS must involve the following major components [4 - 7]: (1) **Preprocessor**, (2) **Fuzzy Inference Subsystem**, (3) **Postprocessor**.

Preprocessor provides the following functions :

- External Representation of current input/output information in a format that is suitable for an user - in linguistic or crisp forms.
- Translation of such forms into Internal Representation Language using Fuzzy Sets (FUZZIFIER).

The second component, i.e. **Fuzzy Inference Subsystem** (FIS), as a core of FS consists of a following parts :

(a) Inference Machine, (b) Knowledge Base, (c) Global (working) Memory. FIS provides decision-making and approximation reasoning procedures and corresponding results in fuzzy terms.

Postprocessor executes an opposite operation - conversion Fuzzy results into Linguistic or Crisp information (DEFUZZIFIER).

It is assumed naturally to use an extension of Petri Nets, i.e. *Fuzzy Petri Nets* (FPN), for Knowledge representation and modeling of such Fuzzy Systems. At present there are a several variants of FPN models [8 - 12].

A main goal of the article is to create simple and efficient procedures and methods of transition from an algorithm which is described in the form of *Fuzzy Production rules* to its *Hardware Implementation* using *a predefined Element Basis (the set of logical elements)*. Here Fuzzy systems are given as corresponding Fuzzy Algorithm (FA) rules. The article concentrates on the following basic stages of Design and Implementation of the FA :

A. *Knowledge Representation* of FA ;
B. Its *Analysis* ;
C. *Hardware Implementation* of a given FA.

A. Knowledge representation

We assume that the Fuzzy algorithm is given by a set of *FuzzyProduction Rules* (FPR) Π as follows :

From: Proceedings, Fourth Bar Ilan Symposium on Foundations of Artificial Intelligence. Copyright © 1995, AAAI (www.aaai.org). All rights reserved.

$$\Pi = \{\Pi i\}, \quad \Pi = \{\ \mathbf{IF}\ L = Li\ \ \mathbf{THEN}\ \ R = Ri\ ,(H = Hi)\ \}$$

More compactly,

$$\Pi = \{\Pi i\}, \quad \Pi i = \{\ Li \underset{Hi}{\Rightarrow} Ri\ \}, \quad \mathrm{i} = 1,2,\ldots,\mathrm{n} \qquad (1)$$

where

Πi is an i- th production of set Π rules - productions,

$L = \{ Li \}$, $R = \{ Ri \}$ are antecedent (condition, premise) and consequent (action, conclusion) parts of Π,

$H = \{ Hi \}$ is a set of *Certainty factors* $\{ CF\ i \}$, i.e. belief degrees for a Rule set Π; Hi is a value of $CF i$, $Hi \in [0,1]$,

$\Rightarrow$ is the " Fuzzy Implication " operator that corresponds with an expression " IF ... THEN ".

We consider a general case of Fuzzy Production rules (1) with the following parts : each of $Li \in L$ and $Ri \in R$ is a complex expression of Fuzzy terms $\{X_\alpha, X_\beta, \ldots, X_\phi, X_\omega\} \subset \mathbf{X}$ and $\{Y_\alpha, Y_\beta, \ldots, Y_\lambda, Y_\omega\} \subset \mathbf{Y}$ respectively, which are connected by a logical operations AND, OR, NOT and ALTERNATIVE (ALT, or Exclusive OR).

Further we restrict ourselves to the following base Fuzzy PR types :

Type 1 (conjunctive) $\Pi 1 :\ X_\alpha\ \text{AND}\ X_\beta\ \text{AND} \ldots \text{AND}\ X_\omega \underset{B1}{\Rightarrow} Y_\lambda$.

Type 2 (disjunctive) $\Pi 2 :\ X_\alpha\ \text{OR}\ X_\beta\ \text{OR} \ldots \text{OR}\ X_\omega \underset{B2}{\Rightarrow} Y_\lambda$.

Type 3 (alternative) $\Pi 3 :\ X_\phi \underset{B3}{\Rightarrow} Y_\alpha\ \text{ALT}\ Y_\beta\ \text{ALT} \ldots \text{ALT}\ Y_\omega$.

Type 4 (reproductive) $\Pi 4 :\ X_\phi \underset{B4}{\Rightarrow} Y_\alpha\ \text{AND}\ Y_\beta\ \text{AND} \ldots \text{AND}\ Y_\omega$,

where

$Z = \{ Z,\ \text{NOT}\ Z \}$;

Terms in **X** and **Y** are fuzzy subsets on the different universe of discourse sets $U_1, U_2, \ldots, U_n$ and $V_1, V_2, \ldots, V_m$, respectively .

It is evident that an elementary (atomic) production $\Pi 0 :\ X_i \underset{B0}{\Rightarrow} Y_i$ is a special case for each of the types $\{\Pi 1, \Pi 2, \Pi 3, \Pi 4\}$.

The method of representation any complex (combined) rule-production of a given Finite algorithm as a composition of Fuzzy productions types 1- 4 and a computation of its *CF* was developed [18] .

We suggest to use an extension of PN,named as *Fuzzy Petri Net* (FPN), for a formal representation of an arbitrary Fuzzy PR set. Our a general model is as follows:

$$\text{FPN} = (\ \text{PN},\ E,\ \Lambda,\ \text{FAS}\ ) . \qquad (2)$$

From: Proceedings, Fourth Bar Ilan Symposium on Foundations of Artificial Intelligence. Copyright © 1995, AAAI (www.aaai.org). All rights reserved.

Here

PN = (*P, T, I, O*) is an conventional *Place /Transitional* Petri Net [23- 24],

E is a finite set of expressions (propositions) which correspond with terms in left and right parts of set Π,

Λ means a finite set of threshold values : $\Lambda = \{ \Lambda\ i \}$, i = 1,2,...,n,

FAS means a fuzzy assignment of Petri Net PN :

$$\text{FAS} = \{ f(P), f(T), f(PE), f(P\Lambda\ ) \}, \qquad (3)$$

where

$f(P): P \rightarrow \Phi$ is an association function that assigns a Membership function (Fuzzy subset) to each place,

$f(T) : T \rightarrow [0, 1]$ is an association function that assigns a Certainty value to each transition,

$f(PE): P \rightarrow E$ is an association function as a bijective mapping from places to expressions, both for left and right parts of each Fuzzy production rule in Π,

$f(P\Lambda\ ): P \rightarrow \Lambda$ is an association function as a bijective mapping from places to threshold values.

Presence in FAS(3) the functions $f(T)$ and $f(P\Lambda\ )$ simultaneously permits to monitor "firing moments" (excitation beginning) of corresponding Fuzzy rules in Π.

In this article, an Fuzzy Pert net model with the following assumptions is used:

(i) FPM is a pure deterministic net, i.e. one with a *deterministic* (but not a stochastic) reachability graph ;

(ii) The FPN is slightly simplified version of the general model (2)of Fuzzy Petri net is used, i.e. one with more simple function in its part (3) $f(P) : P \rightarrow [0,1]$. The choice of such model allows significantly to simplify an execution of the next stages of a given Fuzzy system design such as its analysis and implementation including Hardware implementation.

B. Analysis of Fuzzy Algorithm

Here and further the Fuzzy PN (2) is used as a Formal language to represent given FA rules. Hence the FA analysis can be reduced to an analysis of its FPN. In a general case, an such analysis consists of the following procedures :

B1. *Analysis I,* B2. *Analysis II.*

B1. Analysis I

The main goal of this analysis of an arbitrary finite algorithm is the *Validation*, i.e. *Logical analysis of its interior correctness* . It means a checking of its the basic

From: Proceedings, Fourth Bar Ilan Symposium on Foundations of Artificial Intelligence. Copyright © 1995, AAAI (www.aaai.org). All rights reserved.

logical properties such as *liveness* (in a particular case, *deadlock- freedom*), *boundedness* (*safety*), *reversibility* (in case of a feedback -loops presence). Reachability Analysis (with generation and checking of a corresponding Reachability Graph) is a wide-spread validation method for PN different classes [13, 14]. The major problem of PN validation (including of its Reachability Analysis) is connected with a great complexity of an such validation in case of Large-scale Petri nets. This fundamental problem is a **"state explosion problem"**. For its decision (particularly, in area of Networking and Distributed systems Protocols) it have been devoted a lot of both theoretical investigations and program tools [15,16]. One of the perspective ways to overcome "dimension explosion" problem for an analysis of complex PN is to decompose a given Petri net on a set of more simple PN- components that needed more simple validation (see , for example, [17]).

The article suggests to generalize the decomposition approach on case of Fuzzy PN. Our suggestion is based on the decomposition method for any Place / Transition Petri net of a general type. Such PN is described by a graph that includes a set of multiple and inhibitory arcs and also feedback loops. This method executes the following main steps [18]:

(i) A choice of a finite non-redundant subset of *elementary PN-components* as a *functionally complete Element basis* ;

(ii) A choice of merging rules (*Composition rules*) for such PN- components;

(iii) A generation of *Covering* variants for a given Petri net. Each of these variants corresponds to desired PN decomposition using its elementary components.

The method proposed guarantees both a generation of non-redundant covering solutions for an arbitrary finite Petri net by elements of fixed bases and an its simple validation.

B2. **Analysis II**

The goal of analysis II of any algorithm is to check its *exterior correctness.* This correctness means that its rules- productions satisfy basic properties such as compatibility (consistency) and compactness (non-redundancy). Numerous methods to provide a exterior correctness of the FA rules can divide into two main groups :

- modification (changing) of *linguistic components* of these rules ,
- modification (changing) of *fuzzy components* of the rules .

Each of fuzzy algorithms which satisfied both the properties of exterior and of interior correctness is **a correct Fuzzy algorithm** . Corresponding problems concerning a " *tuning* " of different FA with the aim of formation their *correct* Fuzzy rules set, particularly for Fuzzy controllers, have been studied by many authors [5,19 - 22]. Investigation of exterior correctness problems and survey and comparison of existing approaches to their solutions are beyond the scope of this article.

From: Proceedings, Fourth Bar Ilan Symposium on Foundations of Artificial Intelligence. Copyright © 1995, AAAI (www.aaai.org). All rights reserved.

logical properties such as *liveness* (in a particular case, *deadlock- freedom*), ***boundedness*** (***safety***), ***reversibility*** (in case of a feedback -loops presence). Reachability Analysis (with generation and checking of a corresponding Reachability Graph) is a wide-spread validation method for PN different classes [13, 14]. The major problem of PN validation (including of its Reachability Analysis) is connected with a great complexity of an such validation in case of Large-scale Petri nets. This fundamental problem is a "**state explosion problem**". For its decision (particularly, in area of Networking and Distributed systems Protocols) it have been devoted a lot of both theoretical investigations and program tools [15,16]. One of the perspective ways to overcome "dimension explosion" problem for an analysis of complex PN is to decompose a given Petri net on a set of more simple PN- components that needed more simple validation (see , for example, [17]).

The article suggests to generalize the decomposition approach on case of Fuzzy PN. Our suggestion is based on the decomposition method for any Place / Transition Petri net of a general type. Such PN is described by a graph that includes a set of multiple and inhibitory arcs and also feedback loops. This method executes the following main steps [18]:

(i) A choice of a finite non-redundant subset of *elementary PN-components* as a *functionally complete Element basis* ;

(ii) A choice of merging rules (*Composition rules*) for such PN- components;

(iii) A generation of *Covering* variants for a given Petri net. Each of these variants corresponds to desired PN decomposition using its elementary components.

The method proposed guarantees both a generation of non-redundant covering solutions for an arbitrary finite Petri net by elements of fixed bases and an its simple validation.

B2. Analysis II

The goal of analysis II of any algorithm is to check its *exterior correctness.* This correctness means that its rules- productions satisfy basic properties such as compatibility (consistency) and compactness (non-redundancy). Numerous methods to provide a exterior correctness of the FA rules can divide into two main groups :

- modification (changing) of *linguistic components* of these rules ,
- modification (changing) of *fuzzy components* of the rules .

Each of fuzzy algorithms which satisfied both the properties of exterior and of interior correctness is **a correct Fuzzy algorithm** . Corresponding problems concerning a " *tuning* " of different FA with the aim of formation their *correct* Fuzzy rules set, particularly for Fuzzy controllers, have been studied by many authors [5,19 - 22]. Investigation of exterior correctness problems and survey and comparison of existing approaches to their solutions are beyond the scope of this article.

From: Proceedings, Fourth Bar Ilan Symposium on Foundations of Artificial Intelligence. Copyright © 1995, AAAI (www.aaai.org). All rights reserved.

- For a fixed set of Fuzzy elements to develop an efficient method for a generation of a set of compositions (decompositions) the FAPN . Each member of this set must provide an optimal (suboptimal) Hardware implementation of a fixed Fuzzy FA.

Some of possible solutions of these Hardware implementation tasks are discussed.

In conclusion we want to emphasize a fruitfulness and perspectiveness of the mixed methodology presented here for simultaneous design and implementation of Fuzzy algorithms. Our methodology is based on a **Hybrid Automata-Net approach with using both Finite Automata and Petri Nets and their Fuzzy generalizations**. Such approach permits in principle to use for analysis and synthesis of Intelligent Systems enough rich and detailed theoretical results and program tools in the area of Finite Automata and Petri Nets theories.

References

1. L. Zadeh (1994), Soft Computing and Fuzzy logic, *IEEE Software*, Nov., 11 (6), 48- 56.
2. *Communications of the ACM* (1994), March, 37 (3), 69- 111.
3. *High- level Petri Nets. Theory and Application* (1991) (*K.Jensen, G. Rosenberg, eds.),* Springer- Verlag.
4. *Fuzzy Expert Systems* (1991), (*A.Kandel, ed.*), CRC Press .
5. W. Pedrycz (1993), *Fuzzy control and Fuzzy systems (second ed.),* John Wiley & Sons .
6. T. Terano, K. Asai, M. Sugeno (1992), *Fuzzy systems theory and its applications,* Academic Press.
7. *Fuzzy logic in Artificial Intelligence* (1993), (*E. Klement, W. Slany, eds.), LNAI 695,* Springer- Verlag.
8. S.-M.Chen, J.-S.Ke, J.-F.Chang (1990), Knowledge representation using fuzzy PetriNets, *IEEE Trans. Knowledge & Data Engineering,* Sept., 2(3), 311- 319.
9. V. Cherniaev (1990), Petri Nets: applications possibilities for Fuzzy systems modeling, *Proc. Conf. on Development and Applications of Flexible Hybrid Expert Systems,* Riga.
10. V. Cherniaev (1991), Fuzzy systems and Petri Nets, *AI News (Artificial Intelligence), 4,* Nauka, Moscow (in Russian).
11. T.Cao, A.Sanderson (1993), Variable reasoning and analysis about uncertainty with Fuzzy Petri Nets, *Application and theory of Petri Nets (M.Marsan ed.), LNCS 691,* 126- 145.
12. D. Yeung, E. Tsang (1994), Improved Fuzzy Knowledge representation and rule evaluation using Fuzzy Petri Nets and degree of subsethood, *Int. J. Intelligent Systems* , Dec., 9(12), 1083- 1100.

From: Proceedings, Fourth Bar Ilan Symposium on Foundations of Artificial Intelligence. Copyright © 1995, AAAI (www.aaai.org). All rights reserved.

13. J.Peterson (1981), *Petri Net theory and the modeling of systems,* Prentice-Hall.
14. T.Murata (1989), Petri Nets: properties, analysis and applications, *Proc.of the IEEE,* 77 (4), 541- 580.
15. *Advances in Petri Nets* (1993), *LNCS 639, (G.Rozenberg, ed.),* Springer-Verlag.
16. *Application and theory of Petri Nets* (1993), *LNCS 691, (M.Marsan ed.),* Springer-Verlag.
17. F.Lin, M.Liu (1992), Protocol validation for Large-scale applications, *IEEE Software,* Jan., 23- 26.
18. V.Cherniaev .On the iterative decomposition of a Finite deterministic algorithms using their Petri net representation (in preparation).
19. M. Mamdani, B. Sembi (1980), Process control using Fuzzy logic, *Fuzzy sets: theory and applications to Policy analysis and Information systems (Wang, Chang, eds),* Plenum, NewYork, 249- 265.
20. V. Cherniaev (1979), On some properties of Fuzzy Algorithms, *Control on Communication Networks and Nodes,* Nauka, Moscow, 127-137 (in Russian)
21. V.Cherniaev (1980), Analysis and design problems of Fuzzy Algorithms, *Proc. II All-Union Conf."Choice models of alternatives in Fuzzy environment"* Riga (in Russian).
22. M. Sugeno (1985), *Industrial applications of Fuzzy controls,* North- Holland.
23. H.Watanabe, W.Dettloff, K.Yount (1990), A VLSI Fuzzy logic controller with reconfigurable, cascadable architecture, *IEEE J. Solid-state Circuits,* Apr. 25 (2), 376- 382.
24. T. Yamakawa (1993), A Fuzzy Inference Engine in nonlinear analog mode and its application to a Fuzzy logic control, *IEEE Trans. on Neural Networks,* May, 4 (3), 496- 521.
25. V. Cherniaev (1989), Automata-based approach to Petri Net study, *Control in Distributed Information systems,* Nauka, Moscow, 127-137 (in Russian).