Mapping Fuzzy Petri Net to Fuzzy Extended Markup Language

Zahra Masoudinezad*, Ali Haroonabadi, Mohammad Teshnehlab***
* Departement of Computer, Science and Research branch, Islamic Azad University, khouzestan-Iran
** Asistent Professor, Computer Department, Islamic Azad University, central Tehran branch
***Control Departmant K. N. Toosi University of Technology Seyed Khandanbridge, Tehran, Iran

Article Info

Article history:

Received Apr 14, 2013
Revised Jul 28, 2013
Accepted Aug 10, 2013

Keyword:

FPNML
Fuzzy Petri Net
FXML
Petri Net Markup Language
XML

ABSTRACT

Use of model gives the knowledge and information about the phenomenon also eradicates the cost, the effort and the hazard of using the real phenomenon.Characteristics and concepts of Petri nets are in a way that makes it simple and strong to describe and study the information processing system; especially it is shown in those which are dealing with discrete, concurrent, distributed, parallel and indecisive events. Yet, due to Petri nets inability to face with systems working on obscure data and continues events, the interest to develop fundamental concept of Petri nets has been raised which is led to new style of presented model named "fuzzy Petri nets". The difference in Petri nets is in the elements that have been fuzzed. Transitions, places, signs and arcs can be fuzzed. PMNL, on the other hand as a markup language has been engaged in uttering Petri nets in previous researches. Fuzzy markup nets can model the uncertainty of concurrent scenarios different from a dynamic system by a board of parameters and use of fuzzy membership dependencies. Therefore, in order to define these uncertain data, it is vital to use a formal language to describe fuzzy Petri nets. To support this version in this thesis, a markup language will be presented stating the structure and grammar of markup language and covering fuzzy concepts in Petri nets as well. Presenting the suggested grammar accommodates the support of fuzzy developed markup language and includes the combination of uncertainty and XML.

Corresponding Author:

Zahra Masoudinezad,
Departement of Computer, Science and Research branch,
Islamic Azad University, khouzestan-Iran
Email: z.masoudinezad@yahoo.com

1. INTRODUCTION

Petri nets were discussed by Carl Adam Petri in his PHD's thesis in 1962 [1] & [14]. They are generally used as a device to study and model systems. In fact he presented the relationship among the system components by a graph or a network. First and foremost a Petri net is a mathematical description but it gives a graphical or visual display of system as well. A Petri net can be used to determine important information about the model system structure. Normal Petri nets are used to model system described in details, but to those which have uncertain and obscure data they are inappropriate, while in practice we are dealing with complicated systems that have a degree of uncertainty and they are not subject to precise mathematical modeling. Regarding the ostensibly of obscure data with fuzzy logic, the coalescence of fuzzy theory in Petri nets is useful to enhance the ability of Petri nets modeling. It has been done with the work of Looney and some authors of Petri nets and association of artificial intelligence in 1988. Different types of fuzzy Petri nets compatible with the theory of Petri nets were designed [2]. These different ways which synthesize Petri net and fuzzy sets are called fuzzy Petri nets [3] & [4] & [5]. The difference is in the elements that have been fuzzed. Transitions, places, signs and arcs can be fuzzed. All these things can be fuzzed. PNML, on the other hand, as a markup language has uttered Petri nets. Fuzzy markup networks can model the uncertainty of concurrent scenarios different from a dynamic system by a board of parameters and use of fuzzy membership dependencies. Therefore using a formal language is urgent to describe fuzzy Petri nets to define these uncertain data. To support this version in this article the related work will be reviewed in the first part, in the second a markup language will be presented expressing the grammar and structure in markup language as well as covering fuzzy concepts in Petri nets, in the third a case study will be done and conclusion and future works will be presented in the last part.

2. RELATED WORK

Fuzzy set theory, proposed by Zadeh in 1965, is not to represent non deterministic situation of uncertainty such as randomness or stochastic process, but rather to represent deterministic uncertainty by a class or classes which do not possess sharply defined boundaries [6]. In crisp set, the characteristic function assign a value of either 1or 0 to each individual in the universal set, thereby discriminating between members and non-members of the crisp set under consideration. The concept of fuzzy set however allows a given element to have its membership function between non-membership and full-membership in a given class or fuzzy set. In deterministic uncertainty of fuzzy set, one may subjectively determine membership function of a given element by his knowledge. In other words, knowledge plays important roles in determining or defining a fuzzy set. Similarly, some basic concepts such as equality, containment, complementation, union and intersection are redefined. In addition by fuzzy conditional probability relation as proposed in [8] & [15], granularity of knowledge is given in two frameworks, crisp granularity and fuzzy granularity. Petri nets are a graphical and mathematical modeling tool applicable to many systems. They are a promising tool for describing and studying information processing systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic. As a graphical tool, petri nets can be used as a visual-communication aid similar to flow charts, block diagrams, and networks. In addition, tokens are used in these nets to simulate the dynamic and concurrent activities of systems [9]. Petri Net (PN) (also known as a place/transition net or P/T net) is one of several mathematical modeling languages for the description of Discrete Event Systems (DES). PNs were emerged in 1962 from the PhD thesis of Carl Adam Petri and proved to be quite effective tool for graphical modeling, mathematical modeling, simulation, and real time control by the use of places and transitions. However, there was an intuitive need for a system, which would be able to address uncertainties and imprecision of the real world systems, because of increase in the complexity of industrial and communication systems. Fuzzy logic proved to be an appropriate complement because of its possiblistic nature to handle vague data. Up till the date, numbers of ways have been proposed for ombining PN with fuzzy logic, according to different applications. But with the increasing applications of these nets, there is an increase in the ambiguity about their types and structures. Almost in every new research on Fuzzy Petri Nets (FPN), researchers claim to have come up with new type of FPN. Therefore, for the ease of future researchers and engineers, it was essential to categorize FPN on the basis of some criteria. Owing to this fact, in current research FPN are classified according to their structures, and algorithms. Further, literature review of the applications of FPN has been done in the light of their classifications. As PN can be timed and/or colored, similarly FPN can also be timed and/or colored to include the temporal effect and/or enhance their visibility. Like that of Neural Networks (NN), FPN can also do learning, and can be trained in order to get adapt to the changing situations. And as Fuzzy logic is being combined with PN to get FPN, in the same way FPN can be combine with other Artificial Intelligence (AI) tools, and mathematical models to become more efficient, and powerful. In [14] on the basis of structures and algorithms FPNs have been classified as; Basic Fuzzy Petri Nets (BFPN), Fuzzy Timed Petri Nets (FTPN), Fuzzy Colored Petri Nets (FCPN), Adaptive Fuzzy Petri Nets (AFPN), and Composite Fuzzy Petri Nets (CFPN)[10]&[11]. Mapping between IF-THEN rules and fuzzy Petri nets is obvious. Any IF-THEN rule of the previous defined form:

IF X1 is A1 AND … AND Xn is An THEN Y is B

can be expressed by the following petri net[7]:

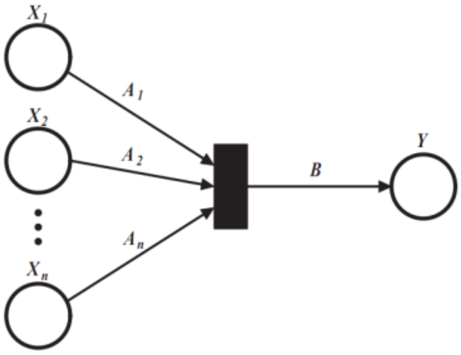

Figure 1. Simple petri net

The Petri Net Markup Language (PNML) is an XML-based interchange format for Petri nets. In order to support different versions of Petri nets, its focus is on universality and flexibility, which is achieved by a technique for defining new Petri net types. For presenting and precisely defining the XML-syntax, PNML uses UML meta models: The PNML Core Model defines the concepts shared by all kinds of Petri nets; additional Petri net type definitions are UML meta models for defining the concepts that are specific to particular kinds of Petri nets. The concrete XML-syntax is then defined by mapping the concepts of these UML meta models to XML elements.

Currently, PNML is standardized as Part 2 of the International Standard ISO/IEC 15909 as the transfer syntax for three particular versions of Petri nets: Place/Transition-Nets, High-level Petri Nets, and Symmetric Nets [12] & [13].

To support of this version in this paper will be presented a markup language. FXML having tags (labels) to support the fuzzy concepts enable us to support the fuzzy formal model. In this paper in addition to support of the fuzzy concepts in Petri Nets, to express the structure and grammar in markup language, in other words can be support the elements of Fuzzy Petri Nets with FXML.

3. THE PROPOSED GRAMMAR

Grammar proposal that will be discussed in this paper called FPNML (Fuzzy Petri Net Markup Language) which is a new syntax based on FXML for describing petri nets.

Tags and labels in this language and their child are listed below:

<net>: Petri Nets

A Petri net is defined with the element <net>. This element has the following attributes:

a) **id:** Unique identifier .allows the net to be referenced from other nets.

b) **type**: The Petri net type:

The net's places, transitions, arcs and subnets are children of the <net> element; and it may further contain the following elements:

c) **graphics:** This element is optional; it is used if an overview page exists with an item for all nets in the document. It specifies the position and optionally the size of the item within the overview.

d) **name**: Optional, The element <name> contains the name of the element. A name is optional and not necessarily unique. It has no attributes and the name is inserted in the element <text>. Optionally, graphical information can be added with <graphics>.

e) **description**: Optional. The element <description> contains the description of an element. It is optional and not necessarily unique. This element doesn't have attributes and the value of the description is inserted into the element <text>. Optionally, graphical information can be added with the element <graphics>.

f) **D**:{d1,d2,…,dn} was a finite set of propositions.

………………………………………………

1. **<place>**: places

A place is defined with the element <place>. This element has the following attributes:

a) **id:** Unique identifier; allows the place to be referenced. Such references are made in <referencePlace>s and <arc>s. A <place> may further contain the following elements:

b) **graphics**: Optional;

c) **name**: Optional; The element <name> contains the name of the element. A name is optional and not necessarily unique – it is legal for two different elements to carry the same name. It has no attributes and the name is inserted in the element <text>. Optionally, graphical information can be added with <graphics>.

d) **description**: Optional; The element <description> contains the description of an element. It is optional and not necessarily unique. This element doesn't have attributes and the value of the description is inserted into the element <text>. Optionally, graphical information can be added with the element <graphics>.

e) **initialMarking**: Optional. The <initialMarking> element provides the initial marking of a place. Its required sub element <text> specifies the value; for uncolored nets, it is a nonnegative decimal number. This element may have a sub element <graphics> to specify a relative position for displaying the content in a diagram.

f) **type:** a place can either be a store or a channel. A channel is a regular Petri net place, while a store serves as a data storage component. Optional; a place is assumed to be a channel. The element <text>, which is required for <type>, holds the type information.

g) **α:** P [0,1] was an association function, a mapping from places to real values between zero and one.

h) **β:** P→D was an association function, a objective mapping from places to propositions.

…………………………………………………

<transition>: Transitions

A transition is defined by the element <transition>. This element has the following attributes:

a) **id**: Unique identifier, allows the transition to be referenced. Such references are made in <arc>s and <referencePlace>s. A <transition> may further contain the following elements:

b) **graphics** :

c) **description** :Optional.

d) **transformation**: The <transformation> element specifies the token value transformation performed by the transition. This element is optional. The transformation is the content of the required sub element <text>. Optionally, information about relative positioning can be added with the element <graphics>.

e) **CF (μ):** CF was the "Certainty Factor"; a larger CF value indicated a higher certainty of the rule.

…………………………………………………

<arc>: Arcs

An arc is defined by the element <arc>. This element has the following attributes:

a) **id:** Unique identifier, allows the arcs to be referenced. the id of a <transition>, <place> within the same <net> or <page> as the <arc> itself.

b) **Source:** {X_1 , X_2 ,…, X_n} was a independent variables and {P_1 , P_2 ,…, P_n} was a input places.

c) **target**: {Y1 , Y2 ,…, Yn} was a dependent variables and {P1 , P2 ,…, Pn} was a output places An <arc> may further contain the following elements:

d) **graphics:** For an <arc>, this element can occur 0, 1, or multiple times; it specifies intermediate support points for layout purposes. It is up to the drawing tool to interpret the values. Start and end points are not specified; the layout algorithms must be deducing them from source and target.

e) **name**: Optional; The element <name> contains the name of the element. A name is optional and not necessarily unique – it is legal for two different elements to carry the same name. It has no attributes and the name is inserted in the element <text>. Optionally, graphical information can be added with <graphics>.

f) **description** :Optional; The element <description> contains the description of an element. It is optional and not necessarily unique. This element doesn't have attributes and the value of the description is inserted into the element <text>. Optionally, graphical information can be added with the element <graphics>.

g) **inscription**: Optional; its value must be in the sub element <text>. A sub element <graphics> may be added with relative positioning for layout purposes.

…………………………………………………

<IF>: if … then

a) **THEN:** the element <THEN> contains the main part of <IF>. This element doesn't have attributes and the value of the description is inserted into the element <text>.

……………………………………………

<THEN>

Newinitialmarking.place ID: this element is new initialmarking after firing.

Newα.place ID: this element is new α after firing.

……………………………………………

Sub nets:

Nets can contain subnets. Different variants of Petri nets support different notions of subnets; one difference between them is how the subnet can be connected with its environment.

1) **<page>:** Subnet definitions
   It has the following attributes:

a) **id:** Unique identifier, allows the subnet to be referenced.
   A <page> may further contain the following elements:
b) **description**: Optional; The element <description> contains the description of an element. It is optional and not necessarily unique. This element doesn't have attributes and the value of the description is inserted into the element <text>.Optionally, graphical information can be added with the element <graphics>.
c) **THEN**: Required; this element have one attribute and this is β.
d) **Newα:** this element is new α after firing and it is calculate follows: $\alpha.p_{source}$* CF(μ). its value must be in the sub element <text>.

…………………………………………………

2. Common elements:

1)) <graphics>: Graphics

The <graphics> element is optional.

It can have the following sub elements: <position> and <dimension>.

These elements, when they occur, have two required attributes:

x The x coordinate of the element.

y The y coordinate of the element.

They do not have any further content.

- Coordinate values denote numbers of screen pixels.
- Sizes are absolute, i.e. relative to the including <net>.
- Positions are always relative to the position of the nearest containing element.
- The x coordinate increases to the right; the y coordinate increases downward; the origin is the containing element's upper left hand corner.
- The <position> sub element is used for <place>s, <transition>s, and <page>s. It is required. It is not used anywhere else.
- The <offset> sub element is used for node attributes that may be displayed in the diagram, It is optional; when used, it is relative to the object to which it refers.
- The <dimension> element is used to denote the size of an element within its containing element, even for <page>s and <net>s.

…………………………………………………

<name>: Name

The element <name> contains the name of the element. A name is optional and not necessarily unique it is legal for two different elements to carry the same name. It has no attributes and the name is inserted in the element <text>. Optionally, graphical information can be added with <graphics>.

…………………………………………………

<description>: Description

The element <description> contains the description of an element. It is optional and not necessarily unique. This element doesn't have attributes and the value of the description is inserted into the element <text>. Optionally, graphical information can be added with the element <graphics>.

…………………………………………………

<**id>:** Identifiers and references

Every <net>, <page>, <transition>, <place> and <arc> must have an id attribute; The source and target attributes of <arc>s IDREF attributes: their values must occur as the value of some id attribute in the document.

4. CASE STUDY

In this paper, a petri net is being studied in which design rules for system bank facilities using fuzzy scenario. To describe this network, used the proposed grammar in the previous section and all the component of the network and fuzzy behavior is expressed using FPNML language.

Figure 2: Fuzzy Petri net system of bank facilities

Fuzzy rules describing the petri net in figure 2:

- If p1 and p2
- If p4 and p5 then p6
- If p2 then p3
- If p3 and p6 then p7
- If p2 and p6 then p9
- If p9 then p10
- If p7 and p10 then p8

In this figure, all the places as the element <place> and their properties by sub element using proposed tags in FPNML language. Transitions and Arcs and their properties are expressed as well as places using tags <transition> and <arc>. Fuzzy rules that are defined in figure 2, are described using tags <IF> and <THEN>.

5. CONCLUSION

What is highlighted in this paper is presented a markup language to support the fuzzy formal model.Although so far is done activities on XML tags and also injection uncertainty in fuzzy petri net, but mapping a fuzzy formal model to fuzzy markup language is innovations that discussed in this paper.

REFERENCES

[1] Fellow T. *Petri Nets: Properties, Analysis and Application*". Proceeding of the IEEE. 1989; 77(4).

[2] Intan R, Mukaidono M and Emoto M. Knowledge-based Reprisentation of Fuzzy sets. *IEEE Internatonal conference on 2002*. 2002; 1: 590-595.

[3] Intan R, Mukaidono M. *Conditional Probability Relations in Fuzzy Relational Database*. Proceedings of RSCTC'00, Springer. 2000: 251-260.

[4] Intan R, Mukaidono M, Yao YY. *Generalization of Rough Sets with α- coverings of the Universe Induced by Conditional Probability Relations*. Proceedings of International Workshop on Rough Sets and Granular Computing, LNAI- 2253, Springer-Verlag, y. 2001: 311-315.

[5] Tsuji K. A New Type of Extended Petri Nets and its Applications. *International symposium on circuits and systems*. 2000; 1: 192-195.

[6] Chen SM. Weighted Fuzzy Reasoning Using Weighted Fuzzy Petri Nets. *IEEE Transaction on knowledge and data engineering*. 2002: 14.

[7] Pavliska V. Petri nets as fuzzy modeling tool. *Submitted/to appear: Internal publication University of Ostrava Institute for Research and Applications of Fuzzy Modeling*. 2006.

[8] Aziz MH, Erik LJ Bohez, Manukid P and Chanchal S. Classification of Fuzzy Petri Nets, and Their Applications. *World Academy of Science, Engineering and Technology*. 2010.

[9] Kindler E. *Concepts, status, and future directions Paper for the invited talk at EKA 2006*. In E. Schnieder (ed.): Entwurf Komplexer Automatisierungs systeme, EKA 2006, 9. Fachtagung, Braunschweig, Germany. 2006, pp. 35-55.

[10] KonarA, Mandal AK. Uncertainty management in expert system using fuzzy petri nets. *Transaction on Knowledge and Data Engineering, IEEE*. 1996: 96-105.

[11] Paul B, Konar A, Mandal AK. *Estmation of certanty factor of knowledge with fuzzy petri nets.* International Conference on Fuzzy Systems Proceeding, IEEE. 1998: 951-955.

[12] Cardoso J, Pradin-chezalviel B. *Logic and fuzzy petri nets.homepages.iaas.fr/ Robert/workshop.d/janete.ps.gz*. 1996.

[13] Chaoui A, Hadjadj I. *PNTools: a multi-language environment to Integrate petri nets tools.* 2009.

[14] *http://www.bibsonomy.org/bibtexkey/conf/ciia/2009.*

[15] *www.esoa.ir*, Iran׳s Information Artichecture committee, [online]2010.