Published in final edited form as:

Eng Appl Artif Intell. 2008 March ; 21(2): 157–170. doi:10.1016/j.engappai.2007.04.007.

A Fuzzy Reasoning Design for Fault Detection and Diagnosis of a Computer-Controlled System

Y. Ting[✉], W.B. Lu, C.H. Chen, and G.K. Wang

Department of Mechanical Engineering, Chung Yuan Christian University, 200, Chung Pei Rd., Chung Li, Taiwan 32023, R.O.C

Abstract

A Fuzzy Reasoning and Verification Petri Nets (FRVPNs) model is established for an error detection and diagnosis mechanism (EDDM) applied to a complex fault-tolerant PC-controlled system. The inference accuracy can be improved through the hierarchical design of a two-level fuzzy rule decision tree (FRDT) and a Petri nets (PNs) technique to transform the fuzzy rule into the FRVPNs model. Several simulation examples of the assumed failure events were carried out by using the FRVPNs and the Mamdani fuzzy method with MATLAB tools. The reasoning performance of the developed FRVPNs was verified by comparing the inference outcome to that of the Mamdani method. Both methods result in the same conclusions. Thus, the present study demonstrates that the proposed FRVPNs model is able to achieve the purpose of reasoning, and furthermore, determining of the failure event of the monitored application program.

Keywords

Error detection; Error diagnosis; Petri nets; Fuzzy rule

1. Introduction

The configuration of an error detection and diagnosis mechanism (EDDM) for a fault-tolerant computer-controlled system has been described in the previous researches (Ting et al., 2002; Ting et al., 2004). The detection mechanism employs the hook process to capture the message in and between the various APs and the OS, and detects whether the monitored AP is failed. Establishment of error classification and standardization was developed in the previous research (Lu et al., 2003). The diagnosis mechanism identifies the failure type and the location of error message, and makes predictable estimation on the executing APs. As been known, while the application program (AP) is failed, the operating system (OS) of computer sends an error message with illustration of the failure event. However, most displayed error messages are difficult for the user to understand, letting alone for them to know the damage level, so that one can hardly deal with the failure appropriately. Also, when different APs are executed in the PC, there may be similar error message but with different error definition and illustration in different programming language (Inprise Corporation, 2002; Richter, 1999). Therefore, it is demanding to investigate how to unify the illustration of error message and examine the closeness degree of the same failure event but with different error description, so that a unified

[✉]Corresponding author. Tel.: 886-3-2654319; fax: 886-3-2654096, E-mail address: yung@cycu.edu.tw.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

error knowledge database could be established, and then the error symptoms could be better inferred via the reasoning algorithm to acquire a final decision-making. The proposed EDDM, which is a fault-tolerant computer-controlled system, is designed and aimed to satisfy the above needs. Petri nets (PNs) is an ideal candidate for investigating and modeling of systems, and it can represent the inference process as a discrete-event dynamic system. The advantages of using PNs in rule-based systems include (1) the graphical formalism, which can visualize the inference states step by step; (2) the transparent modeling, which has well-established formal mechanisms for modeling and structure inconsistency checking; (3) the analyzing capability, which can express dynamic and structural behaviors of a rule-based system via algebraic forms (Scarpelli et al., 1996; Tsang et al., 1999; Yang et al., 2003). The fuzzy Petri nets combines the graphical technique of PNs, the fuzzy sets theory, and the fuzzy production rule, so it has the advantage of the graphical power of PNs and the capability of fuzzy to model rule-based decision system effectively (Fay, 2001). Thus, fuzzy Petri nets would outperform the Petri nets and improve the efficiency of fuzzy reasoning (Yang et al., 1997). In general, fuzzy rule-based system consists of knowledge base, database of event facts, rule base, and inference engine. In order to deal with the input information from the non-fuzzy system, the fuzzy inference mechanism will carry on fuzzification, rule matching, and defuzzification in advance to allow the likely incomplete and imprecise information to match the antecedent proposition of the fuzzy rule and then obtain an inference result. Quite a few researches use PNs to construct the fuzzy rule-based system (Chen et al., 1990; Koriem, 2000; Chen, 2002). The input and output places of each transition in a PNs can be used to represent the knowledge of the antecedent and consequent propositions by the fuzzy production rule. Many researches have investigated on the extension of PNs to fuzzy Petri nets, and the modeling as well as the reasoning of fuzzy rule-based system. For instance, Gao et al. used the fuzzy reasoning Petri nets model to describe the production rule-based system and verify its performance as a diagnostic expert system on the turbine (Gao et al., 2003). Chen used fuzzy Petri nets to construct knowledge description and fuzzy reasoning, and developed a weighted fuzzy Petri nets method (Chen et al., 1990; Chen, 2002). Looney developed an algorithm of rule-based decision-making by using fuzzy Petri nets (Looney, 1998).

In this study, it is attempted to use the PNs, the fuzzy sets theory, and the fuzzy production rule to establish the Fuzzy Reasoning and Verification Petri Nets (FRVPNs) model for the diagnosis mechanism. The FRVPNs is designed based on the fuzzy rule decision tree (FRDT) with the merit of hierarchical structure. It provides different level of abstraction, which can be used to represent the sub-model construction and the rule decision for independent level of the FRVPNs. Regarding the efficiency of hierarchical design, it has been discussed in several research articles. For example, Delgado et al. proposed a multi-objective decision making scheme, allowing the evolutionary parameters based on a hierarchical genetic fuzzy system adjustable, so it not only improves the models performance (accuracy), but also guarantees the interpretability of the resulting fuzzy models (Delgado et al., 2002); Joo et al. also demonstrated a scheme of hierarchical fuzzy system with constraints on the fuzzy rules approximating with high accuracy and fewer fuzzy rules (Joo and Lee, 2005); Wang proposed a hierarchical structure in a fuzzy system achieving good approximation accuracy (Wang, 1999). According to these works, it is therefore believed that improvement of inference accuracy can be achieved by use of the hierarchical design.

Since different types of AP may have different definition of the failure event, it is likely to describe the error symptoms in terms of linguistic variables via the fuzzy set theory, and construct the inference contexts by use of the fuzzy production rule. The fuzzy production rule makes use of the “*IF...THEN...*” rule to describe the conditions of the antecedent and consequent propositions. The membership degrees (MDs) of the propositions on the fuzzy rule are calculated via the membership function (MF). Then, the damage level decision-making is determined by using the fuzzy reasoning method. In addition, the rule-checking process and

the verification and modification module are included in the FRVPNs. The former one is used to confirm the correctness of the winning rule. The latter one is used to deal with the problem of redundancy, conflict, circularity, and incompleteness while new fuzzy rules are added. The rule verification and modification module will address in another article. Simulations are carried out with several sets of examples by using the developed FRVPNs and the fuzzy logic toolbox of MATLAB. As compared with the inferred results, both methods draw the same conclusions.

2. Fuzzy Diagnosis Reasoning Structure of EDDM

The EDDM is proposed for a fault-tolerant computer-controlled system and is a middleware structured in and between the user applications of the user mode and the kernel mode for error detection and diagnosis of the AP. The EDDM consists of the retrieve process, the detection mechanism, the diagnosis mechanism, the **information record database**, and the **knowledge base** (Ting et al., 2004). Several important parameters such as the total central processing unit (CPU) usage (TCPUU), process CPU usage (PCPUU), total memory usage (TMU), handling time (HT), and error hazard level (EHL), etc. are considered for fuzzy reasoning. Figure 1 demonstrates the procedure: the EDDM measures these parameters of the monitored AP, and sends them to the **fuzzification module**. And then the **fuzzification module** handles the measured parameters of the monitored AP by fuzzification. By using the fuzzified parameters, the **fuzzy reasoning module** will search a corresponding fuzzy rule, read an object knowledge unit from the **knowledge database**, and then infer an outcome of the consequent proposition according to the input of the antecedent proposition of the fuzzy rule and the contents of the object knowledge unit. Once the parameter fuzzification fails to acquire appropriate matching, the fuzzified parameters are sent to the **rule self-learning module**. The **rule self-learning module** collects the new available information from the user or the **information record database**, and then carries on self-learning before constructing new rules. If it is impossible to acquire a corresponding fuzzy rule from the **fuzzy rule base**, the **domain expert** will be allowed to input the domain knowledge to the **knowledge acquisition module**. The **rule verification and modification module** will detect whether the new rules have problems of redundancy, conflict, circularity, and incompleteness by using the input/output transition function of the place and the inhibit condition. Hence, error of inference and unnecessary expense of inference due to inaccuracy or redundancy of fuzzy rule can be reduced; in other words, inference accuracy can thus be improved. The fuzzy inference results will be saved to the **information record database** and the **knowledge base** of the EDDM, and be sent to the **defuzzification module** for defuzzification and decision-making. Finally, the **defuzzification module** delivers an output of damage affection to the monitored AP and the OS.

3. Representation of Fuzzy Sets and Rules

3.1. Definition of Fuzzy Sets and Linguistic Variables

The input variables of the antecedent proposition considered for the fuzzy rule include the TCPUU, the PCPUU, the TMU, the HT, and the EHL. The output variables of the fuzzy reasoning include the response time (RT) and the damage level decision-making (DLDM). In reference to Negnevitsky (2002), linguistic value, notation and normalized numerical range of the linguistic variable are included in the fuzzy rule and illustrated in Table 1. The values of linguistic variables are fuzzified to obtain the MD by MF. Values of all the linguistic variables are normalized in the range of [0, 1], so the MD is bounded in the range of [0, 1]. The MDs of the concerned variables are all designed of the trapezoid shape of fuzzy sets, as listed in Table 1.

3.2. Fuzzy Rules Representation

The implication relationship of the antecedent and consequent proposition is used to establish the elements for fuzzy rules. Assuming FR is the set of fuzzy rules represented by $FR = \{FR_1, FR_2, FR_3, \dots, FR_{s-1}, FR_s\}$, $FR_k \in FR$. The k^{th} fuzzy rule is denoted by

$$FR_k: IF A_a THEN C_b \quad (CF_k) = \mu_k \quad (1)$$

where a, b, k and s are positive integer, and $1 \leq k \leq s$; A_a is the antecedent proposition defined in a fuzzy region of input space and described by the linguistic variables such as "Small", "Medium", and "Large"; C_b is the consequent proposition defined in a fuzzy region of specified output and described by linguistic variables such as "Low", "Medium", and "High"; $CF_k \in [0, 1]$ is called the certainty factor (CF); μ_k is the value of the CF that represents the strength of the belief in the rule FR_k .

In general, composite fuzzy production rule is classified into several rule types (Gao et al., 2003). The rule type of "AND" connection is used to describe the antecedent proposition in this study. The CF value is designed to reflect the way the experts think (Negnevitsky, 2002; Qu and Shirai, 2003). In this article, assuming that the belief strength of a fuzzy rule assigned by human expert is $CF=1$, which represents that the fuzzy rule is completely believable, and denoted by

$$IF A_1 AND A_2 AND \dots AND A_a THEN C_b \quad (CF=1) \quad (2)$$

4. Structure and modeling of FRVPNs

In Figures 2 and 3, an example of a two-level FRVPNs including the rule-checking process as well as the verification and modification module is presented. The proposed FRVPNs is constructed based on the FRDT with the merits of hierarchical design and PNs technique. It uses the rule-checking transitions and places to confirm the correctness of the firing rule so as to protect the accuracy of inference result of the winning rule. The verification and modification module is another important function. It uses the input/output transition function of the place and the inhibit condition to detect the inconsistent fuzzy rules (conflict, redundancy, incompleteness, and circularity rules) so that inference accuracy is improved.

4.1. Hierarchical structure

FRVPNs is designed based on the FRDT with the hierarchy structure. On the basis of the hierarchical design, the inference accuracy of FRDT is expected to improve (Joo and Lee, 2005; Delgado et al., 2002; Wang, 1999). Figure 4 shows that there are three inputs considered in the Level_I, including the TCPUU, the PCPUU, and TMU, by which the RT can be inferred. In Figure 5, there are three inputs considered in the Level_II, including the RT (the output of the Level_I), the HT, and the EHL, by which the DLDM can be inferred in terms of three degrees (Normal, Slightly High, High) to the AP and the OS.

4.2. Dynamic Decision-Making Procedure

The dynamic decision-making procedures of the Level_I and Level_II are shown in Figure 4 and Figure 5 respectively and described as follows:

(1) Level_I:

Step 1. enter the required new object variables for the fuzzy rule in the Level_I.

Step 2. calculate the MD of the proposition of new object variables via the MF.

Step 3. calculate the firing strength by the composition operator *MIN*.

Step 4. calculate the maximum firing strength by the composition operator *MAX*, and find the winning rule among the activated rules in the Level_I.

Step 5. draw a conclusion of the output sub-object in the Level_I from the selected winning rule.

(2) Level_II:

Step 6. import the three object variables, the RT, HT, and EHL, to the Level_II.

Step 7. calculate the MD of the new object variables by the MF.

Step 8. calculate the firing strength of each rule in the Level_II by the composition operator *MIN*.

Step 9. search for the maximum firing strength of the winning rule among the activated rules in the Level_II by the composition operator *MAX*.

Step 10. draw a conclusion of the output in the Level_II from the selected winning rule.

4.3. Petri Nets for FRVPNs Modeling

The FRVPNs takes advantage of the PNs to model the reasoning structure and represents the dynamic behavior of the fuzzy rule. An ordinary PNs structure is a 4-tuple (P, R, I, O) defined in (Cardoso and Camargo, 1999; Peterson, 1981; Girault and Valk, 2003; Zhou and Venkatesh, 1998), but the ordinary PNs cannot satisfy the fuzzy reasoning design for the EDDM. Therefore, the proposed FRVPNs applied 11-tuples, defined by

$FRVPNs = (P, R, I, O, WF, M, H, Pro, ProMF, RFS, WRFS)$

Related definitions are described as below.

Definition 4.1—*WF*: $AS \subseteq (P \times R) \cup (R \times P) \rightarrow +N$ is a weight function associated with the weight of the arc or the number of the direct arcs, where *AS* is a set of the arcs, and the current relation holds only between the places and the transitions.

Definition 4.2—*H*: $P \times R \rightarrow N$, $H(P_k) = \{P_k \in P: H(P_k, R_m) > 0\}$, is an inhibitor function. An inhibitor arc from a place P_k to a transition R_m has a “circle” symbol rather than an arrowhead at the transition. A transition represented by an “arrow” symbol is enabled when tokens are in all of its normal inputs and no tokens are in all of its inhibitor inputs. The transition is fired by removing tokens from all of its normal inputs.

Definition 4.3—*Pro* = $\{Pro_{I1}, Pro_{I2}, \dots, Pro_{ij}\}$ is a finite set of propositions. A proposition $Pro_{ij} \in Pro$ is mapped on a place $P_n \in P$ in the FRVPNs.

Definition 4.4—*ProMF* is a MF of a proposition. The input object of the proposition can be mapped to a corresponding MD through the *ProMF*. The MD of a proposition of a fuzzy rule FR_k is represented by $\mu(x) \in [0, 1] \rightarrow FR$, where $x \in X$ is the input object of the proposition.

Definition 4.5—a fuzzy composition operation represents a MD of the required proposition calculated by the composition operator *MAX/MIN*. The fuzzy composition is defined by $MAX/MIN \rightarrow R$.

Definition 4.6— RFS_k : fuzzy composition $\rightarrow \mathbf{R}$, represents the firing strength of a fuzzy rule. The larger firing strength indicates the larger belief strength of a rule. Based on the fuzzy operators *AND/OR* shown in the antecedent proposition of a rule FR_k , the RFS_k is calculated by the composition operator *MIN* or *MAX*.

Definition 4.7— $WRFS$: a winning-rule firing strength, is calculated by the *MAX* composition operator. The FRDT structure consists of *L*-levels, e.g., Level_I and Level_II in this article, and each level includes *s*-rules, e.g., $FR_1, FR_2, \dots, FR_k, \dots, FR_s$. Let RFS_k be the firing strength of each k^{th} rule in the Level_I. Let $WRFS_k: MAX(RFS_1, RFS_2, \dots, RFS_k, \dots, RFS_s) \rightarrow R$ be the firing strength of the winning rule $FR_k \rightarrow FR$. The $WRFS_k$ is used to compose the firing strengths of the *s*-rules in the Level_I or Level_II and select the winning rule FR_k with the highest confidence.

5. FRVPNs Reasoning Strategy

As illustrated in Figure 6, the examples of FR_1 and FR_2 are used to describe the FRVPNs modeling and the dynamic reasoning behavior. Figure 6(a) shows the contents of the fuzzy rule in the Level_I and Level_II. Figure 6(b) shows part of the FRVPNs model by means of the PNs technique. Figure 6(c) shows the dynamic reasoning behavior of the rules FR_1 and FR_2 . The properties of the proposition set of places and the firing transitions are described as follows.

1. I_{ij}^{APro} and II_{ij}^{APro} : represents the “antecedent proposition” place of a rule FR_k in the Level_I and Level_II respectively, where *i* represents the i^{th} input object of a fuzzy rule FR_k , and *j* represents the j^{th} antecedent proposition of a fuzzy rule FR_k .
2. I_{ij}^{AProMF} and II_{ij}^{AProMF} : represents the “MF” transition of an antecedent proposition of a fuzzy rule FR_k in the Level_I and Level_II respectively.
3. I_{ij}^{AProMD} and II_{ij}^{AProMD} : represents the “MD” place of an antecedent proposition of a fuzzy rule FR_k in the Level_I and Level_II respectively.
4. I_k^{RFS} and II_k^{RFS} : represents the “firing strength” transition of a fuzzy rule FR_k in the Level_I and Level_II respectively.
5. I_k^{RFS} and II_k^{RFS} : represents the “firing strength” place of a fuzzy rule FR_k in the Level_I and Level_II respectively. The place I_k^{RFS} with a token represents the firing strength of a fuzzy rule FR_k .
6. $R^{MAX()}$: represents the “firing strength” transition of a winning rule in the Level_I or Level_II. It is calculated by the composition operator *MAX*.
7. P_k^{WRFS} : represents the “winning fuzzy rule” place. It models the firing strength FRS_k of the winning rule FR_k in the Level_I or Level_II.
8. I_k^{CProMF} and II_k^{CProMF} : represents the “MF of a consequent proposition” transition in the Level_I and Level_II respectively. The MD of a consequent proposition of the winning rule FR_k is calculated by the centroid of the aggregate output MF.
9. I_k^{CProMD} : represents the “MD of a consequent proposition” place. The place I_k^{CProMD} with a token represents the MD of a consequent proposition of a rule FR_k in the Level_I.

10. P^{FDM} : is a “final decision-making” place. The place P^{FDM} with a token represents the final result in the Level_II.

5.1. FRVPNs Reasoning Algorithm for the Level_I

The FRVPNs model of the Level_I is illustrated in Figure 2, and described as below.

Step 1. Import the parameters of the proposition of the rules by the following places and transitions:

1. $I_{P}^{AProIO} = (I_{P_1}^{AProIO}, I_{P_2}^{AProIO}, \dots, I_{P_k}^{AProIO}, \dots, I_{P_s}^{AProIO})$ is a set of “input object” places. Each place has an input variable.
2. $I_{R}^{AProD} = (I_{R_1}^{AProD}, I_{R_2}^{AProD}, \dots, I_{R_k}^{AProD}, \dots, I_{R_s}^{AProD})$ is a set of “input object distribution” transitions. It represents the transition distribution of linguistic variables for each antecedent proposition of a rule.

Step 2. Establish the antecedent propositions for each rule and calculate the MD for each antecedent proposition by the following places and transitions:

1. $P^{APro} = (I_{P_{11}}^{APro}, I_{P_{12}}^{APro}, I_{P_{13}}^{APro}, \dots, I_{P_{ij}}^{APro})$ is a set of “antecedent proposition” places for each rule.
2. $R^{AProMF} = (I_{R_{11}}^{AProMF}, I_{R_{12}}^{AProMF}, \dots, I_{R_{ij}}^{AProMF})$ is a set of “MF of an antecedent proposition” transitions. $I_{R_{ij}}^{AProMF}$ represents the MF of the j^{th} fuzzy set corresponding to the i^{th} input object of the antecedent proposition, thus the MD of the i^{th} input object of the antecedent proposition is computed by the j^{th} MF.
3. $I_{WF_{ij}}$ represents the weighted function corresponding to each transition arc of the antecedent propositions.
4. $P^{AProMD} = (I_{P_{11}}^{AProMD}, I_{P_{12}}^{AProMD}, \dots, I_{P_{ij}}^{AProMD})$ is a set of “MD of an antecedent proposition” places for each rule.

Step3. Perform *MIN* composition operation and calculate the firing strength for the activated rules by the following places and transitions:

1. $R^{RFS} = (I_{R_1}^{RFS}, I_{R_2}^{RFS}, \dots, I_{R_k}^{RFS}, \dots, I_{R_s}^{RFS})$ is a set of “firing strength of rule” transitions for each rule in the Level_I. $I_{R_k}^{RFS}$ uses the fuzzy operator “AND” or “OR” to perform *MIN* or *MAX* composition operation.
2. $P^{RFS} = (I_{P_1}^{RFS}, I_{P_2}^{RFS}, \dots, I_{P_k}^{RFS}, \dots, I_{P_s}^{RFS})$ is a set of “firing strength of rule” places. While the token exists in the place $I_{P_k}^{RFS}$, the transition $I_{R_k}^{RFS}$ is disabled by the inhibitor arc to ensure that the new result of firing strength is transferred into the place $I_{P_k}^{RFS}$ after the previous firing strength in the place have been transferred out.

Step4. Perform *MAX* composition operation on the firing strengths of the activated rules, and select the winning rule from the available rules in the Level_I by the following places and transitions:

1. $R^{MAXO} = MAX(I_{P_1}^{RFS}, I_{P_2}^{RFS}, \dots, I_{P_k}^{RFS}, \dots, I_{P_s}^{RFS})$ is a transition of *MAX* composition operation. The transition R^{MAXO} is used to calculate the maximum firing strength among the activated rules.

2. P_k^{WFR} is a “WRFS_k” place, and it represents the firing strength RFS_k of the winning rule FR_k.

Step5. Determine the winning rule for rule-checking purpose by the following places and transitions:

1. $R^{FSC}=(I_1^{FSC}, I_2^{FSC}, \dots, I_k^{FSC}, \dots, I_s^{FSC})$ is a set of “firing strength comparison” transitions. The transition is used to compare the firing strength of an activated fuzzy rule with that of the WRFS_k of a FR_k. If the firing strengths are not same, the reasoning process will be terminated.
2. $P^{WFR}=(I_1^{WFR}, I_2^{WFR}, \dots, I_k^{WFR}, \dots, I_s^{WFR})$ is a set of “winning firing rule” places. While the place I_k^{WFR} contains a token, the place I_k^{WFR} represents FR_k is the winning firing rule and is selected to fire.

Step6. Determine the final decision rule for rule-checking purpose by the following places and transitions:

1. $I_{[-k]}^{RFS}$ is a set of “firing strength of rule” places, except the I_k^{RFS} place.
2. I_{PCPrO} is a “linguistic variable of output object of a consequent proposition” place.
3. $R^{FDR}=(I_1^{FDR}, I_2^{FDR}, \dots, I_k^{FDR}, \dots, I_s^{FDR})$ is a set of “final decision of the winning rule” transitions. The transition is used to delete the rules with the firing strength less than the rule with the highest firing strength.
4. I_{PDMI} is a “decision-making identification” place. It represents the decision result of the selected winning rule FR_k with the highest firing strength.

Step7. Construct the consequent propositions of the final decision rule by the following places and transitions:

1. $P^{CProM}=(I_1^{CProM}, I_2^{CProM}, \dots, I_k^{CProM}, \dots, I_s^{CProM})$ is a set of “marking of the consequent proposition” places.
2. $R^{DMI}=(I_1^{DMI}, I_2^{DMI}, \dots, I_k^{DMI}, \dots, I_s^{DMI})$ is a set of “decision-making identification” transitions. It represents the transition distribution of linguistic variables for a consequent proposition of a fuzzy rule.
3. $P^{CPro}=(I_1^{CPro}, I_2^{CPro}, \dots, I_k^{CPro}, \dots, I_s^{CPro})$ is a set of “consequent proposition” places. It describes the conclusions of the fuzzy rules.

Step8. Calculate the MD for each consequent proposition by the following places and transitions:

1. $R^{CProME}=(I_1^{CProME}, I_2^{CProME}, \dots, I_k^{CProME}, \dots, I_s^{CProME})$ is a set of “MF of a consequent proposition” transitions for each consequent proposition. When the transition I_k^{CProME} is fired, the MD is computed by the k^{th} MF.
2. $P^{CProMD}=(I_1^{CProMD}, I_2^{CProMD}, \dots, I_k^{CProMD}, \dots, I_s^{CProMD})$ is a set of “MD of a consequent proposition” places. While the place I_k^{CProMD} contains a token, it receives an MD from the transition I_k^{CProME} .

5.2. FRVPNs Reasoning Algorithm for Level_II

The FRVPNs model of the Level_II is illustrated in Figure 4, and described as below.

Step1. Establish the antecedent propositions of the rules and calculate the MDs for each antecedent proposition as follows.

1. The output object from the Level_I is used as the input place to the Level_II.
2. The new input objects of the antecedent proposition of the rules in the Level_II can be modeled by repetition of the Step1 in Section 5.1.
3. The MDs of the new input objects are calculated by repetition of the Step2 in Section 5.1.

Step2. Describe the decision behaviors of the rules as follows.

1. Use Step3 in Section 5.1 to calculate the firing strength of each activated rule.
2. Use Step4 in Section 5.1 to find out the maximum firing strength from the available activated rules.
3. Use Step5 and Step6 in Section 5.1 to decide the winning rule among the activated rules.
4. Use Step7 in Section 5.1 to determine the consequent propositions of the Level_II.

Step3. Determine the final decision-making by the following places and transitions:

1. $R^{FDM} = (R_1^{FDM}, R_2^{FDM}, \dots, R_k^{FDM}, \dots, R_s^{FDM})$ is a set of “final decision-making” transitions, which determines R_k^{FDM} .
2. P^{FDM} is a “final decision-making” place of entire FRVPNs.

6. Simulations

As shown in Figures 2 and 3, the fuzzy rule base of the reasoning processor in the EDDM is constructed of two levels in this study. The measurement values [TCPUU, PCPU, TMU, HT, EML] = [73%, 33%, 65%, 10.5s, 72%], are assumed to be the inputs variables to the antecedent propositions of FRVPNs model, which is the example_1 with the inference results in the Level_I and Level_II listed in Table 2. With this example, the reasoning process is described in detail through the following steps.

Step1. Normalize the measurement inputs (e.g. [TCPUU, PCPU, TMU, HT, EHL] = [73%, 33%, 65%, 10.5sec, 72%]) by the fuzzy sets in Section 3.1 and distribute the normalized inputs to the FRVPNs model. The measurement values are normalized to be

$$\begin{aligned} & \left[\frac{TCPUU}{100\%}, \frac{PCPUU}{100\%}, \frac{TMU}{100\%}, \frac{HT}{30\text{sec}}, \frac{EHL}{100\%} \right] \\ &= [(73/100), (33/100), (65/100), (10.5/30), (72/100)] \\ &= [0.73, 0.33, 0.65, 0.35, 0.72] \end{aligned}$$

where the maximum HT is defined to be 30 seconds.

Step2. Assume the Level_I of the reasoning processor has the following rules:

FR₁: IF “TCPUU is Medium” AND “PCPUU is Low” AND “TMU is Low” THEN “RT is Fast”.

FR₂: IF “TCPUU is High” AND “PCPUU is Low” AND “TMU is Medium” THEN “RT is Medium”.

FR₃: IF “TCPUU is High” AND “PCPUU is Medium” AND “TMU is High” THEN “RT is Slow”.

Then, TCPUU= 0.73, PCPUU= 0.33, and TMU= 0.65 are imported to the places $I_{P_1}^{AProIO}$, $I_{P_2}^{AProIO}$, and $I_{P_3}^{AProIO}$ in the Level_I, respectively.

Step3. The transitions $I_{R_1}^{AProD}$, $I_{R_2}^{AProD}$, and $I_{R_3}^{AProD}$ are concurrently fired, then the tokens are distributed to the place $I_{P_{11}}^{APro}$, $I_{P_{12}}^{APro}$, $I_{P_{13}}^{APro}$, $I_{P_{21}}^{APro}$, $I_{P_{22}}^{APro}$, $I_{P_{23}}^{APro}$, $I_{P_{31}}^{APro}$, $I_{P_{32}}^{APro}$, and $I_{P_{33}}^{APro}$, respectively.

Step4. The transition set I_{R}^{AProMF} is fired, then the tokens are transferred to the place $I_{P_{11}}^{AProMD}$, $I_{P_{12}}^{AProMD}$, $I_{P_{13}}^{AProMD}$, $I_{P_{21}}^{AProMD}$, $I_{P_{22}}^{AProMD}$, $I_{P_{23}}^{AProMD}$, $I_{P_{31}}^{AProMD}$, $I_{P_{32}}^{AProMD}$, and $I_{P_{33}}^{AProMD}$, respectively. The MD of each place is calculated by its trapezoidal MF described in Section 3.1. In this example, the calculated MDs are:

$$\begin{aligned} I_{P_{11}}^{AProMD} &= \mu_{Low}(TCPUU)=0 \\ I_{P_{12}}^{AProMD} &= \mu_{Medium}(TCPUU)=0.35 \\ I_{P_{13}}^{AProMD} &= \mu_{High}(TCPUU)=0.65 \\ I_{P_{21}}^{AProMD} &= \mu_{Low}(PCPUU)=0.35 \\ I_{P_{22}}^{AProMD} &= \mu_{Medium}(PCPUU)=0.65 \\ I_{P_{23}}^{AProMD} &= \mu_{High}(PCPUU)=0 \\ I_{P_{31}}^{AProMD} &= \mu_{Low}(TMU)=0 \\ I_{P_{32}}^{AProMD} &= \mu_{Medium}(TMU)=0.75 \\ I_{P_{33}}^{AProMD} &= \mu_{High}(TMU)=0.25 \end{aligned}$$

Step5. The transition set R^{RFS} and the transition $R^{MAX()}$ are fired in order, and then the firing strength of each activated rule and the winning rule is calculated by the *MIN* and *MAX* composition operator respectively. It yields

$$\begin{aligned} FR_1 &: MIN(0.35, 0.35, 0)=0 \\ FR_2 &: MIN(0.65, 0.35, 0.75)=0.35 \\ FR_3 &: MIN(0.65, 0.65, 0.25)=0.25 \\ MAX(FR_1, FR_2, FR_3) & \\ &= MAX(0, 0.35, 0.25)=0.35 \end{aligned}$$

Step6. According to the result of *MAX* composition operation in the above Step5, the final winning rule is:

FR₂: IF “TCPUU is High” AND “PCPUU is Low” AND “TMU is Medium” THEN “RT is Medium”.

In case that the antecedents of rules are evaluated to be the same level, the max-min inference cannot determine the winning rule. Hence, other defuzzification methods are considered to use. The center of gravity method (COG) is the most popular method (Van Broekhoven and De Baets, 2004; Negnevitsky, 2002; Yager, 1992), which is used to solve this problem.

Step7. Assume the Level_II of the reasoning processor has the following rules:

FR₄: IF “HT is Medium” AND “RT is Medium” AND “EHL is Normal” THEN “DLDM is Slightly High”.

FR₅: IF “HT is Short” AND “RT is Medium” AND “EHL is Normal” THEN “DLDM is Normal”.

FR₆: IF “HT is Long” AND “RT is Medium” AND “EHL is High” THEN “DLDM is High”.

Then, $HT = 0.35$ and $EHL = 0.72$ are imported to the places ${}^{II}P_1^{AProIO}$ and ${}^{II}P_2^{AProIO}$, respectively. The defuzzification of RT is calculated as $RT = 0.598$ by the centroid of the aggregate output MF in the Level_I. This RT value is then imported to the antecedent propositions in the Level_II.

Step8. The transition set ${}^{II}R^{AProD}$ is fired to distribute the tokens to the place set ${}^{II}P^{APro}$.

Step9. As the transition set ${}^{II}R^{AProMF}$ is fired, the tokens are then transferred to the place set ${}^{II}P^{AProMD}$. The MDs of antecedent propositions in the Level_II are calculated by the trapezoidal MF and listed as below.

$$\begin{aligned} {}^{II}P_{11}^{AProMD} &= \mu_{Short}(HT) = 0.25 \\ {}^{II}P_{12}^{AProMD} &= \mu_{Medium}(HT) = 0.75 \\ {}^{II}P_{13}^{AProMD} &= \mu_{Long}(HT) = 0 \\ {}^{II}P_{21}^{AProMD} &= \mu_{Low}(EHL) = 0 \\ {}^{II}P_{22}^{AProMD} &= \mu_{Medium}(EHL) = 0.4 \\ {}^{II}P_{23}^{AProMD} &= \mu_{High}(EHL) = 0.6 \\ {}^IP_{23}^{CProMD} &= \mu_{Medium}(RT) = 1 \end{aligned}$$

Step10. By using the *MIN* and *MAX* composition operators, the firing strength of each activated rule and the winning rule is calculated respectively as

$$\begin{aligned} FR_4: MIN(0.75, 1, 0.4) &= 0.4 \\ FR_5: MIN(0.25, 1, 0.4) &= 0.25 \\ FR_6: MIN(0, 1, 0.6) &= 0 \\ MAX(FR_4, FR_5, FR_6) &= MAX(0.4, 0.25, 0) = 0.4 \end{aligned}$$

Step11. According to the result of *MAX* composition operation in the above Step10, the final winning rule is:

FR₄: IF “HT is Medium” AND “RT is Medium” AND “EHL is Normal” THEN “DLDM is Slightly High”.

Following the steps of the FRVPNs Reasoning Algorithm in Section 5, the final winning rule in the Level_I is FR₂, which indicates that the “RT is Medium”, and in the Level_II the final winning rule is FR₄, which indicates the “DLDM is Slightly High”. The Mamdani fuzzy method of the MATLAB tools is also used to compare the inference results under the same conditions (same inputs, same linguistic values, same ranges). As shown in Figure 7, rules FR₁, FR₂, and FR₃ are aggregated and defuzzified to have a crisp value of $RT = 0.592$ by calculating the centroid, which indicates the rule FR₂ is the winner. In reference to the consequent proposition of FR₂, “RT is Medium” is thus inferred for Level_I. In Figure 8, in the Level_II, the crisp value of $DLDM = 0.416$ is calculated, which indicates the rule FR₄ is the winner. In reference to the consequent proposition of FR₄, “DLDM is Slightly High” is thus inferred for the monitored AP and the OS. In comparison with the inferred results, both methods have the same reasoning outcomes. This example and other four examples by using the two methods are included and listed in Table 2, which shows that the inference outcomes of all the examples are the same.

7. Conclusions

In this article, the fuzzy set theory and the fuzzy production rule method are used to establish the fuzzy rules for the failure events of the AP. The proposed FRVPNs is designed based on the FRDT in association with the PN technique, and constructed with the rule-checking process

as well as the verification and modification module. The computational complexity (Chen et al., 1990; Gao et al., 2003; Kungas, 2005) is defined by $O(ptl)$ and determined by the number of p , t and l , where p and t are the numbers of places and transitions, respectively, and l is the maximum number of transitions in the longest place-transition path. The FRVPNs reasoning algorithm apparently has higher computational complexity than a general FRDT, because it uses more places and transitions in the rule-checking process and in the verification and modification module. Even though the complexity, higher accuracy could be obtained, and problems of redundancy, conflict, circularity, and incompleteness would be reduced. Comparing the FRVPNs with the fuzzy logic toolbox of MATLAB by the simulation results of several sets of arbitrarily selected examples, both methods conclude the same outcomes of fuzzy reasoning. It verifies that the damage level of the failure event for the monitored AP and the OS can be successfully reasoned by the proposed FRVPNs. To sum up, the proposed FRVPNs model is capable of reasoning and determining the failure event efficiently and accurately, and it is thus suitable to be included in the diagnosis mechanism of the EDDM. Regarding the verification and modification algorithm, it is addressed in another article.

Acknowledgements

This research is supported by NSC89-TPC-7-033-009 & NSC88-2212-E-033-018.

Acronyms

CPU	central processing unit
DLDM	damage level decision-making
EDDM	error detection and diagnosis mechanism
EHL	error hazard level
FRDT	fuzzy rule decision tree
FRVPNs	Fuzzy Reasoning and Verification Petri Nets
HT	handling time
MD	membership degree
MF	membership function
PCPUU	process CPU usage
PNs	Petri nets
ProMF	membership function of proposition

RFS	firing strength of rule
RT	response time
TCPUU	total CPU usage
TMU	total memory usage
WRFS	winning-rule firing strength
Notation	
i	i^{th} input object of an antecedent proposition, $i = 1, 2, 3, \dots$
j	j^{th} fuzzy set of input object of an antecedent proposition, $j = 1, 2, 3, \dots$
p, t, l, s, h, k, m	a serial positive integer; p, t, l, s, h, k , and $m = 1, 2, 3, \dots$
N	a set of non-negative integers
P_k	k^{th} place in the PNs
R_m	m^{th} transition in the PNs
FR_k	k^{th} fuzzy rule
PlkAProIO	Input object of antecedent proposition
RlkAProD	distribution of input object of antecedent proposition
PlijAPro	antecedent proposition of rule
RlijAProMF	calculation of MD of antecedent proposition by MF
${}^lWF_{ij}$	weight function
PlijAProMD	MD of antecedent proposition
RIkRFS	calculation of firing strength of a rule by <i>MIN</i> composition operation

PlkRFS	Firing strength of rule
R^{MAX()}	MAX composition operation
PlkWRFs	place of winning-rule's firing strength
RlkFSC	Comparison of firing strength
PlkWFR	winning firing rule
Pl[-k]RFS	a set of "firing strength of rules" except the firing strength of the winning rule
l_{pC Pro}O	linguistic variable of output object of a consequent proposition
RlkFDR	final decision of winning rule
l_{pDMI}	decision-making identification of winning firing rule
PlkCProM	marking of consequent proposition
RlkDMI	identification of consequent proposition
PlkCPro	result of consequent proposition of a rule
RlkCProMD	calculation of MD of consequent proposition by MF
PlkCProMD	MD of consequent proposition
RkFDM	final decision-making computation
p^{FDM}	final decision-making result

References

- Cardoso, J.; Camargo, H. Fuzziness in Petri Nets. Physica-Verlag; New York: 1999. p. 4-24.
- Chen SM. Weighted Fuzzy Reasoning Using Weighted Fuzzy Petri Nets. IEEE Transactions on Knowledge and Data Engineering 2002;14(2):386–397.
- Chen SM, Ke JS, Chang JF. Knowledge Representation Using Fuzzy Petri Nets. IEEE Transaction on Knowledge and Data Engineering 1990;2(3):311–319.
- Delgado MR, Von Zuben F, Gomide F. Multi-objective decision making: towards improvement of accuracy, interpretability and design autonomy in hierarchical genetic fuzzy systems. Proceedings of the 2002 IEEE International Conference on Fuzzy Systems 2002;2:1222–1227.

- Fay A. A Fuzzy Petri Net approach to decision-making in case of railway track closures. IFSA World Congress and 20th NAFIPS International Conference 2001;5:2858–2863.
- Gao M, Zhou MC, Huang X, Wu Z. Fuzzy Reasoning Petri Nets. IEEE Transactions on System, Man, and Cybernetics, Part A 2003;33(3):314–324.
- Girault, C.; Valk, R., editors. Petri Nets for Systems Engineering—A Guide to Modeling, Verification, and Applications. Springer-Verlag; Berlin: 2003.
- Inprise Corporation. Borland C++ Builder6 Developer's Guide. Borland Software Co.; U.S.A.: 2002.
- Joo MG, Lee JS. A class of hierarchical fuzzy systems with constraints on the fuzzy rules. IEEE Transactions on Fuzzy Systems 2005;13(2):194–203.
- Koriem SM. A Fuzzy Petri Net Tool for Modeling and Verification of Knowledge-Based Systems. Computer Journal 2000;43(3):206–223.
- Kungas P. Petri Net Reachability Checking Is Polynomial with Optimal Abstraction Hierarchies. Symposium on Abstraction, Reformulation and Approximation 2005;3607:149–164.
- Looney CG. Fuzzy Petri Nets for Rule-Based Decision-Making. IEEE Transactions on System, Man, and Cybernetics 1998;18(1):178–183.
- Lu WB, Ting Y, Chen CH, Chen YC. The Application of Knowledge Representation and Knowledge Acquisition to Fault Diagnosis. The 31st International Conference on Computers & Industrial Engineering 2003:69–74.
- Negnevitsky, M. Artificial Intelligent – A Guide to Intelligent Systems. Addison-Wesley; New York: 2002.
- Peterson, JL. Petri Net Theory and the Modeling of Systems. Prentice-Hall; New Jersey: 1981.
- Qu W, Shirai K. Belief learning in certainty factor model and its application to text categorization. Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing and Fourth Pacific-Rim Conference on Multimedia 2003;12:1192–1196.
- Richter, J. Programming Applications for Microsoft Windows. 4. Microsoft Press; Washington: 1999.
- Scarpelli H, Gomide F, Yager RR. A reasoning algorithm for high-level fuzzy Petri nets. IEEE Transactions on Fuzzy Systems 1996;4(3):282–294.
- Ting Y, Lu WB, Chen CH, Wang GK. Design the Reasoning Process of Failure Detection and Diagnosis Mechanism (EDDM) by Fuzzy Rule-Based Method. The 2004 IEEE International Conference on Systems, Man, and Cybernetics 2004:2344–2349.
- Ting Y, Shan FM, Lu WB, Chen CH. Implementation and Evaluation of Failsafe Computer-Controlled Systems. International Journal of Computers and Industrial Engineering 2002;42(2–4):401–415.
- Tsang ECC, Yeung DS, Lee JWT. Learning capability in fuzzy Petri nets. The 1999 IEEE International Conference on Systems, Man, and Cybernetics 1999;3:355–360.
- Van Broekhoven E, De Baets B. A Comparison of Three Methods for Computing the Center of Gravity Defuzzification. 2004 IEEE International Conference on Fuzzy Systems 2004:25–29.
- Wang LX. Analysis and design of hierarchical fuzzy systems. IEEE Transactions on Fuzzy Systems 1999;7(5):617–624.
- Yager RR. Fuzzy sets and approximate reasoning in decision and control. IEEE International Conference on Fuzzy Systems 1992:415–428.
- Yang SJH, Chu WC, Lee J, Huang WT. A fuzzy Petri nets based mechanism for fuzzy rules reasoning. The 21st Annual International, Computer Software and Applications Conference 1997:438–443.
- Yang SJH, Tsai JJP, Chen CC. Fuzzy Rule Base Systems Verification Using High-Level Petri Nets. IEEE Transactions on Knowledge and Data Engineering 2003;15(2):457–473.
- Zhou, MC.; Venkatesh, K. Modeling Simulation and Control of Flexible Manufacturing Systems – A Petri Net Approach. World Scientific Pub.Co.; Singapore: 1998.

Fig. 1.

Fig. 2.
The FRVPNs Model of the Level_I

Fig. 3.
The FRVPNs Model of the Level_II

Fig. 4.
Reasoning the Sub-Object in the Level_I

Fig. 5.
Reasoning the Object in the Level_II

Fig. 6.
(a) Fuzzy Rules in the Level_I and Level_I II (b) FRVPNs Model (c) Marking Dynamic Reasoning Behavior

Fig. 7.
Final decision of the Level_I

Fig. 8.
Final decision of the Level_II

Table 1
Linguistic values and ranges of fuzzy sets for FRVPNs
TCPUU, PCPU, TMU

Linguistic values	Notation	Numerical range (normalized)
Low	L	0.0–0.4
Medium	M	0.2–0.8
High	H	0.6–1.0
RT		
Linguistic values	Notation	Numerical range (normalized)
Fast	F	0.0–0.4
Medium	M	0.2–0.8
Slow	S	0.6–1.0
HT		
Linguistic values	Notation	Numerical range (normalized)
Short	S	0.0–0.4
Medium	M	0.2–0.8
Long	L	0.6–1.0
EHL		
Linguistic values	Notation	Numerical range (normalized)
Low	L	0.0–0.4
Normal	N	0.2–0.8
High	H	0.6–1.0
DLDM		
Linguistic values	Notation	Numerical range (normalized)
Normal	N	0.0–0.4
Slightly High	SH	0.2–0.8
High	H	0.6–1.0

Table 2

Simulation Examples of the FRVPNs

Level I														Level II				Matlab RT	Matlab DLDM
Rule No.		TCPUU	PCPUU	TMU	RT	MIN	HT	EHL	DLDM	MIN									
Example_1	1	M	L	L	F	0	S	N	N	0.25									
	2	H	L	M	M	0.35	M	N	SH	0.4									
	3	H	M	H	S	0.25	L	H	H	0									
Measurement values		0.73	0.33	0.65		MAX 0.35	0.35	0.72		MAX 0.4				M 0.592	SH 0.416				
Example_2	1	M	L	L	F	0.2	S	L	N	0									
	2	H	L	M	M	0.35	M	N	SH	0.3									
	3	H	M	H	S	0	L	L	N	0									
Measurement values		0.69	0.33	0.36		MAX 0.35	0.26	0.71		MAX 0.3				M 0.432	SH 0.582				
Example_3	1	L	M	L	F	0.35	S	N	N	0									
	2	H	L	L	F	0	M	L	N	0.2									
	3	M	L	M	M	0.4	M	H	SH	0.25									
Measurement values		0.3	0.27	0.28		MAX 0.4	0.36	0.65		MAX 0.25				M 0.391	SH 0.465				
Example_4	1	M	L	H	S	0	S	N	N	0									
	2	M	M	L	M	0	M	N	SH	0									
	3	H	H	M	S	0.6	L	H	H	0.7									
Measurement values		0.72	0.75	0.5		MAX 0.6	0.74	0.86		MAX 0.7				S 0.83	H 0.835				
Example_5	1	M	L	L	M	0.25	S	N	N	0									
	2	M	L	M	M	0.25	L	L	N	0									
	3	H	M	M	S	0.6	L	H	H	0.75									
Measurement values		0.75	0.34	0.32		MAX 0.6	0.78	0.75		MAX 0.75				S 0.71	H 0.838				

A. Note: RT and DLDM are the crisp values calculated by the Mamdani fuzzy method of MATLAB tools.

B. The boldface words are the winning rules.