An Approach Towards the Verification of Fuzzy Hybrid Rule/Frame-based Expert Systems

Simon C.K. SHIU[1], James N.K. LIU[2], Daniel S. YEUNG[3]

Abstract. In this paper, we described an approach to model Fuzzy Hybrid Rule/Frame-based Expert Systems (FHES). The approach is based on Coloured Petri Nets (CPN) and Controlled State Tokens (CST). First, we defined the properties of a FHES and illustrate how the various features are being modelled by CPN. Secondly, we applied our approach to a practical personnel selection system currently being used in Hong Kong. The detection and analysis of the anomalies of proposed model is done by constructing and examining the reachability tree spanned by the knowledge inference. Lastly, our approach can provide formal verification of the correctness, consistency, and completeness of the FHES.

1 Introduction

Traditionally, attention has been concentrated on using verification techniques to tackle rule-based systems [1][3][8][10][14]. However, these techniques exhibit a limited range of applicability. They could not cope with the kind of hybrid expert systems (HES), rule-based plus frame-based, which many of the current expert systems are being developed [2][11][15]. The use of this hybrid approach integrates the power of organizing data objects in a class hierarchy and reasoning about the objects through user pre-defined logical associations. This advantage accounts for many popular expert system developing software, such as ADS, ART, EXSYS EL, KAPPA-PC, KBMS, NEXPERT OBJECT, LEVEL5 OBJECT, PROKAPPA, REMIND, which combine some sort of frame-based representation with a rule-based inference engine. Recently, [9][12][13] have extended their State Controlled Petri Nets (SCPN) model to handle the knowledge inference in hybrid rule/frame-based expert systems. This paper extends the SCPN to cover the modelling of fuzzy knowledge in hybrid expert systems.

The paper has six main sections. Next section describes the knowledge representation and inference of a fuzzy hybrid rule/frame-based expert system (FHES), the third section gives the definitions of Coloured Petri Nets (CPN)[6] and illustrates how FHES can be modelled by CPN. Section four applied our methodology to a practical personnel selection system currently being used in Hong Kong. Section five discusses the analysis of CPN and the last section gives the conclusion and discussion.

1,2,3 Department of computing, Hong Kong Polytechnic University
1 E-mail: *csckshiu@comp.polyu.edu.hk*

ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

2 A Fuzzy Hybrid Expert System

A Fuzzy Hybrid Expert System combines multiple representation paradigms into a single integrated environment. For a Fuzzy Hybrid Rule- and Frame-based integration, it composes of the following key features: Object Classes, Slot Attributes, Inheritance Relations, Demons, Methods, Rules, Fuzzy Measure, and Reasoning Strategies. These features can be analyzed using three conceptual views [4] of an expert system, they are: (1) An Object View which encapsulates a module of knowledge (or a concept). These knowledge modules (concepts) are represented by Object Classes. Inheritance Relations describe how these knowledge modules are related. (2) A Function View which specifies the functional behaviour of the objects within the expert system. These functions are represented using Methods and Demons. (3) A Control View which specifies the sequence of knowledge inference in the expert system. These controls are represented in terms of Rules and Reasoning Strategies.

In practical HES development [12][13], Frames are used to represent domain objects, various kinds of Demons are used to implement procedures attached to specific slots, Inheritance is used to inherit Class properties, methods, and demons among Object Classes, Message Passing is used for interaction among different Objects and Methods are used to perform algorithmic actions or some array manipulation within an Object. Rules are used to describe heuristic problem-solving knowledge, Forward and Backward chains are commonly used to reason using rules. Therefore, in HES, the Frame base can be seen as being used to define the vocabulary for the Rule base, (i.e. the possible values that slots can be defined and so specified, and the literal used to construct rules must conform to the restrictions imposed by what is available from the Class hierarchy). The Frame base is married together with the Rules designed to manipulate it. The specific integration mechanisms of HES are as follows:

- Rules with Message Passing : Rules for sending or receiving messages to and from Objects for testing the Rules' premises.
- Rules with Inheritance : Rules for reading and writing data into slots in a parent Object and through inheritance of this slot's value to its children Objects, trigger other Rules to fire.
- Rules with Demons : Rules for reading and writing data into slots and cause the execution of the associated Demons, which then trigger other rules to fire.
- Rules with Methods : Rules are embedded as part of an Object's Methods. Since Methods are arbitrary pieces of

code attached to an Object, they can access the Rules through function calls.

- Rules with Instances : Rules can be used to create/delete an instance of a specific Object Class.

In addition to the above, Fuzzy Measure can be integrated into the expert systems in the following two ways.

- Rules with Fuzzy Measure : In this integration, fuzzy measure was used to construct a function called fuzzy integral which is connected with some inferential evidence. A vague proposition in terms of an implicit or explicit vague predicate can be represented by equating the possibility distribution of the variable whose value is restricted by this predicate, with the membership function of the fuzzy set. The fuzzy set represents the meaning of the possible vague components in the predicate.

- Inheritance of Fuzzy Measure : In simple inheritance without fuzziness, the Object's default value, data type, cardinality, allowed words and range are all inherited to its subclasses. If fuzziness is allowed, we may inherit an extra characteristic about an attribute of an Object, namely Fuzzy Measure. Therefore, in fuzzy inheritance hierarchy, each slot of an object will be associated with an attribute of fuzzy measure. (i.e. the degree of membership of that attribute belongs to that Object. e.g. John is a student and his English is Good (0.9), here John can be represented by an instance of a Object Class Students, English can be represented by a Slot and Good is the value of the slot and 0.9 is the fuzzy membership measure of the linguistic variable "Good"). In this arrangement, if we have three classes of students, (Students**,** Undergraduate Students and Undergraduate Science Students), we may represent their English standard in the following fuzzy inheritance hierarchy:

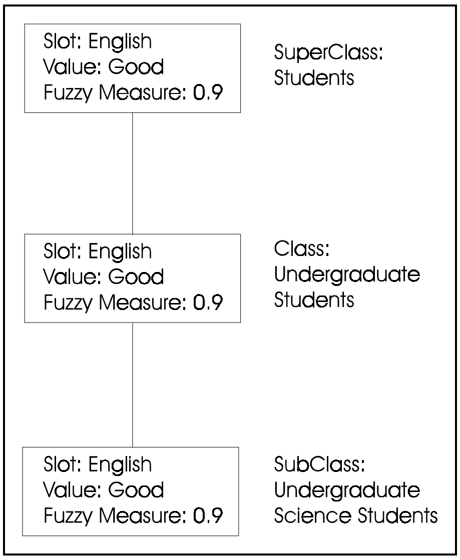

Figure 1 : A simple Class Hierarchy with only one slot: English and Fuzzy Measure: 0.9.

Based on the above concepts of integration, a Fuzzy Hybrid Rule/Frame-based Expert System, therefore, can be formally defined as a tuple FHES = **(C, A, I, In, D, M, Π, Ψ, R, S)** satisfying the requirements below:

C = a finite set of Object Classes, where each object class, **c**, is a Cartesian product of **(A x D x M)**.

A = a finite set of attributes. Each attribute is a tuple of **(a, μ)** where **a** is the value of the attribute and μ is the fuzzy membership measure of the attribute **a**. $\mu \in \Psi$ which is in the interval [0, 1].

I = a specific object element from an object class **c**.

In = an inheritance relation. It is defined from the partially ordered relations in **C**.

D = a finite set of demon functions, **d(a)**. It is defined from A into an expression such that: $\forall \mathbf{a} \in \mathbf{A} : \mathbf{a} \wedge \mathbf{d(a)} \in \mathbf{A}$. (This means the demon functions can only change a slot's value within the same object instance, besides, this demon function: **d(a)** generates only one output from each given input "**a**",).

M= a finite set of methods. It is defined as procedures in **C**.

Π = a finite set of selection threshold τ, in the interval [0, 1].

Ψ = a finite set of fuzzy measure in the interval [0, 1].

R = a finite set of Rules. Each rule is defined as a function from $\bigcup_{i \in C} A_i$ such that $\mathbf{a} \wedge \mathbf{f(a)} \in \bigcup_{i \in C} A_i$.
(This means the literal used to construct Rules must be an element which belongs to the union of all the attribute sets $\mathbf{A_i}$).

S = a finite set of reasoning strategies.

Object class here is defined as having a set of attributes, demons and methods. Each attribute is defined as a simple data type with a possible fuzzy measure associate with it. Each specific object element is called an instance of the Object Class and will have different attribute values and fuzzy measures. Inheritance is defined as a partial order on the set Object Class, it is a relation that is reflexive, antisymmetric and transitive:

- Reflexive : For every Object Class, it inherits the properties from itself.

- Antisymmetric : For every Object Class, if A inherits from B and if B inherits from A, it implies that A is B.

- Transitive : For every Object Class, if A inherits from B and if B inherits from C, it implies that A inherits from C.

The above definition only covers simple inheritance, in the case of multiple inheritance, more elaborate definition is required.

A Demon is defined as a function which is executed when the associated slot value is either updated, or needed. Sometimes, a Demon can also act like a validation trigger which checks the cardinality and/or constraints imposed on a particular slot. The effects of a Demon are always confined locally to the same Object Class.

Methods are procedures attached to some Object Class, that will be executed whenever a message is passed from another Object. This way of executing a method is known as Message Passing.

Rules will interact with the information contained in the slots of the various Object Classes within the HES.

Finally, in HES, there should be a set of reasoning strategies. Some common ones are :

- Backward Chain with Inheritance : Goal directed search with inheritance as one of the means to establish the Rule chains which may pass through several Object Classes.

- Forward Chain with Inheritance : Data directed search with inheritance as one of the means to establish the Rule chains which may pass several Object Classes.

Other important inference strategies include : Pattern Matching, Unification, Resolution and Heuristic Search.

3 Modelling the FHES Using CPN

3.1 Definition of Coloured Petri Net

A Coloured Petri Net can be defined as a tuple CPN = (Σ, P, T, A, N, C, G, E, I) satisfying the requirements below:

Σ = a finite set of non-empty types, called colour sets.
P = a finite set of places.
T = a finite set of transitions.
A = a finite set of arcs such that: $P \cap T = P \cap A = T \cap A = \emptyset$.
N = a node function. It is defined from A into $PxT \cup TxP$.
C = a colour function. It is defined from P into Σ.
G = a guard function. It is defined from T into expressions such that: $\forall t \in T$:[Type(G(t))=B $\wedge$ Type(Var(G(t))) $\subseteq \Sigma$]
E = an arc expression function. It is defined from A into expressions such that : $\forall a \in A$: [Type(E(a))=$C(p(a))_{ms} \wedge$ Type(Var(E(a))) $\subseteq \Sigma$].
I = an initialization function. It is defined from P into closed expressions such that $\forall p \in P$: [Type(I(p))=$C(p)_{ms}$].

The set of colour sets determines the types, operations and functions that can be used in the net inscriptions. The places, transitions and arcs are described by three sets P, T and A which are required to be finite and pairwise disjointed. The node function N maps each arc into a pair where the first element is the source node and the second the destination node. The two nodes have to be of different kinds (i.e. one of the nodes must be a place while the other is a transition). Several arcs may be allowed to link between the same ordered pair of nodes. The colour function C maps each place, p, to a colour set C(p). This means that each token on p must have a token colour that belongs to the type C(p). The guard function G maps each transition, t, to an expression of type Boolean, i.e., a predicate. All variables in G(t) must have types that belong to Σ. A guard is allowed to be a list of Boolean expressions [Bexpr1, Bexpr2..etc]=B. This means that the binding must fulfill each of the Boolean expression in the list. The arc expression function E maps each arc, a, into an expression which must be of type $C(p(a))_{ms}$. This means that each evaluation of the arc expression must yield a multi-set over the colour set that is attached to the corresponding place. The initialization function I maps each place, p, into a closed expression which must be of type $C(p)_{ms}$, i.e. a multi-set (a set which may contain multiple occurrences of the same element) over C(p).

3.2 Fuzzy Hybrid Expert System Model

3.2.1 Object Classes

Each object class's data structure is represented by a compound colour set, and each object instance is represented by a token in that set. For instance, if there are fifteen sets of non-empty types or colour sets being used to represent one object class's data structure, i.e. Σ = AA,BB,....OO; Color AA may be defined as text strings; Color BB may be as Boolean; ...and Color OO may be defined from some already declared coloured sets, e.g. Color OO = Product AA * BB * CC. Each object class instance is declared as a variable of a particular colour set, i.e. var Instance-1 : OO (var denotes variable declaration which introduces one or more variables). Here we have one variable, Instance-1, which is with colour OO. We may use var Instance-1, Instance-2, Instance-3 : OO for declaring three different instances of the same object class with colour OO. In the following sections, we will use three variables, object "a", which is a particular instance of a Super Class A, object "a1", which is a particular instance of Class A. (i.e. "a" IS-A superclass instance while "a1" IS-A class instance) and State "s" which is the state token. State "s" is used to carry the information that identify which object instance shall be fired from which transition. (i.e. var a : OO, a1 : OO and var s : string)

3.2.2 Fuzzy Rules with Inheritance

In CPN, the transition operations are represented by the arc expression functions. By defining the arc expression functions differently, it can help us model different events in the FHES. Therefore, places in the CPN are taken to correspond to two different elements in the FHES. First, places are taken to correspond to predicates of the fuzzy production rules which are pre-defined earlier by the user. Secondly, places are taken to correspond to the Objects class in the FHES's Frame hierarchy. Similarly, transitions in the CPN correspond to two different events in the FHES. First, the transitions correspond to the implications of the fuzzy rules. Secondly, the transitions correspond to the inheritance of the properties, including fuzzy measures, from Classes. The transition operations are represented by the arc expression functions. (e.g. A Fuzzy Rule R can be represented in CPN as shown in Figures 2a, 2b and 2c)

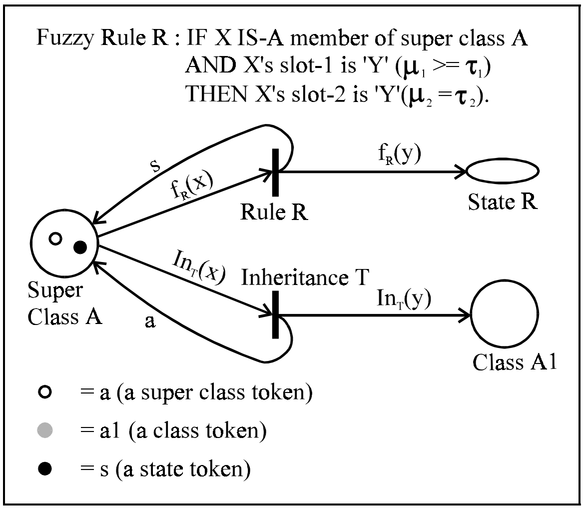

Figure 2a : Fuzzy Rule R with Inheritance (before firing) with an input token "a" & "s" in Super Class A.

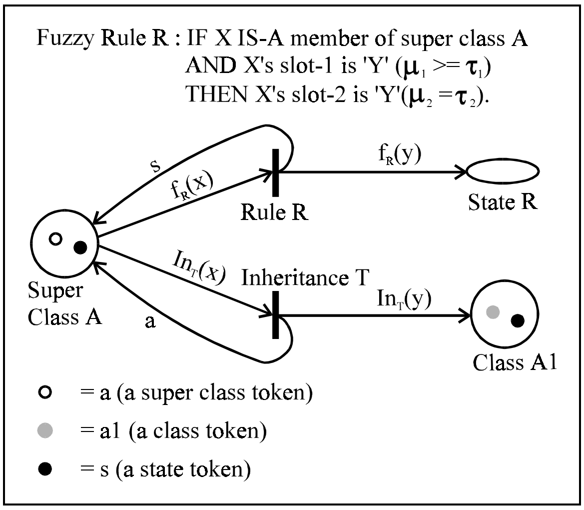

Figure 2b : Fuzzy Rule R with Inheritance (after firing Inheritance T) with an input token "a" & "s" in Super Class A.

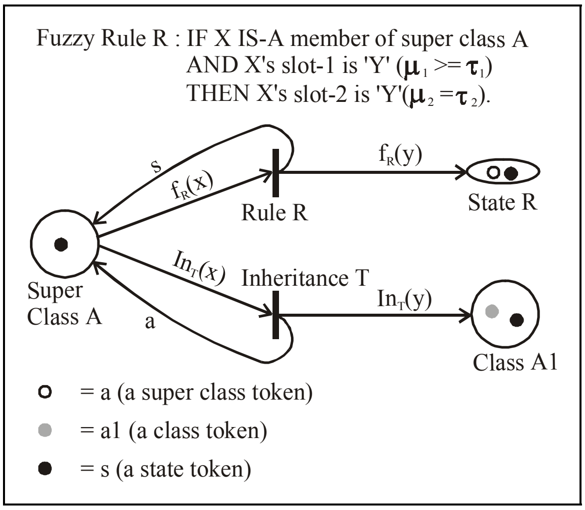

Figure 2c : Fuzzy Rule R with Inheritance (after firing both Rule R and Inheritance T) with output token "a" & "s" in State R and output token "a1" & "s" in Class A1. A state token "s" is also created in Super Class A.

Super Class A is a CPN Place with colour set that was used to represent the data structure of all object instances in Super Class A. Class A1 is a CPN Place with colour set that was used to represent the data structure of all object instances in Class A1. Fuzzy Rule R is a CPN Transition which is enabled iff the input arc expression $f_R(x)$ is evaluated to be true (i.e., the premise X IS-A member of super class A AND X's slot-1 is 'Y' is true with fuzzy measure $\mu_1 >= \tau_1$). If $f_R(x)$ is enabled then Fuzzy Rule R is fired, it implies that Fuzzy Rule R is executed. All tokens will be removed from Super Class A and a new token "a" will be created in State R with new data values determined by the output arc expression $f_R(y)$. (i.e. $f_R(y)$ will assign 'Y' to X's slot-2 with fuzzy measure $\mu_2=\tau_2$). Inheritance T is a CPN Transition which is enabled whenever there is an "a" token in Super Class A, after firing this transition, a token "a1" is created in Class A1 with all the attributes inherited from A. (i.e. a child token is created with the same attributes of its father). These two tokens ("a", "a1") can be used for further inference (if any) in the HES. In this way, we can trace the execution path of these two tokens by examining the information carried by the state tokens created within the CPN network. Moreover, we can also examine the contents of this two tokens to see if any attributes are in conflict with each other. These could serve as an indication of the existence of anomalies within the FHES. (Note that in order to preserve the state of the predicate in Fuzzy Rule R, a state token is created in Super Class A via the self-loop of Fuzzy Rule R and an "a" token is created in Super Class A via the self-loop of inheritance T.)

3.2.3 Fuzzy Rules with Message Passing

Places in the CPN are taken to correspond to predicates of the fuzzy production rules and the transitions in corresponding to the implications of the fuzzy rules. Since the object class instance's data structure is represented by the token of a particular colour set, we can define arc expressions such that they directly read and write data in the token's data slots. This can be illustrated by the following simple example: Pass the message "OK" with fuzzy measure 0.9 to the object Class A's slot promotion.

Colour sets:

Color Classes = with ClassA | Class B;
Color FuzzyMeasure = Real;
Color Value = String;
Color Promotion = product Value * FuzzyMeasure;
Color Objects = product Classes * Promotion;
Color x : Classes;

Arc expression:

IF x=ClassA THEN 1`(ClassA, "OK", 0.9) ELSE empty.

This will serve the purpose of sending or receiving messages (data value) to and from object instance for testing the fuzzy rule's premises.

3.2.4 Fuzzy Rules with Demons

Similarly, a Fuzzy Rule with Demon can also be represented by a Places/Transition tuple, e.g. if a demon is attached with object X's slot-hardworking, whenever the value of slot-hardworking is updated to 'Y' with fuzzy measure $\mu_2 >= \tau_2$ then the demon will execute and assign the value 'Y' with fuzzy measure $\mu_3=\tau_3$ to the slot-promotion. This can be represented by Figure 3a and 3b.

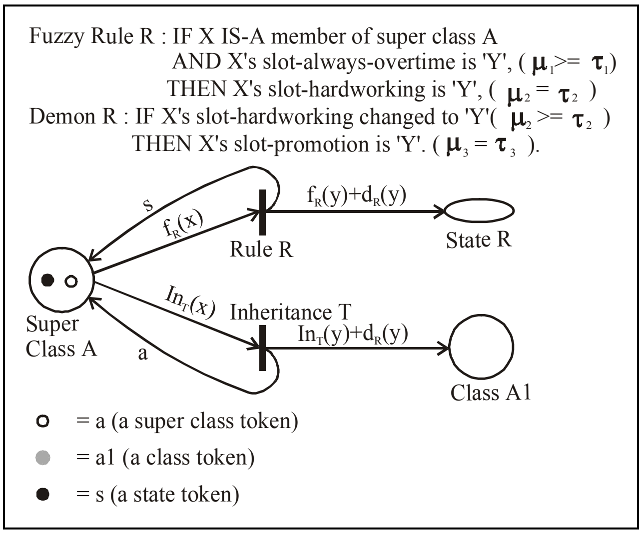

Figure 3a : Fuzzy Rule R with Demon (before firing) with an input token "a" and a state token "s" in Super Class A.

The demon function, $d_R(y)$, is represented as an arc expression. The firing of Fuzzy Rule R will trigger the demon function to execute.

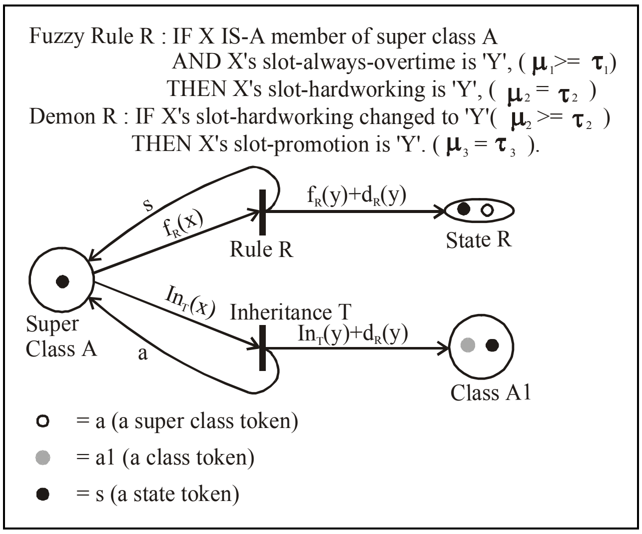

Figure 3b : Fuzzy Rule R with Demon (after firing) with output token "a" & "s" in State R and output token "a1" & "s" in Class A1. A state token "s" is also created in Super Class A.

3.2.5 Fuzzy Rules with Methods

Methods are procedures attached to an Object class, they can be represented by the Functions and Operations declarations in CPN. The function takes a number of arguments and returns a result. The arguments and the result have a type which is a declared colour set, the set of all multi-sets over a declared colour set. A declared function can be used in arc expressions, guards and initialization expressions in the CPN. For example, a typical function which tells whether the argument is even or not might be:

```
fun Even(n:integers)=((n mod 2)=0).
```

Operations can also be used to represent Methods. In both Functions and Operations declarations, different kinds of control structures can be built. e.g. CASE; IF b is true THEN statement 1 ELSE statement 2; WHILE b is true DO; REPEAT statement 3 UNTIL b is true. The Rules with Methods can thus be represented by CPN as follows (Figures 4a-d, the self loops are omitted for clarity reason)

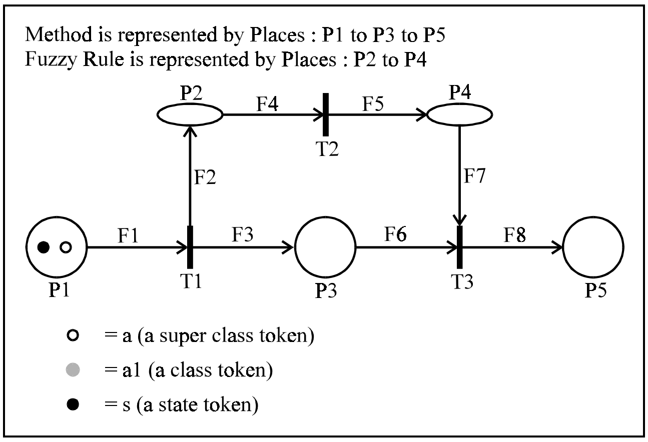

Figure 4a : Fuzzy Rule with Method (before firing) with an input token "a" and a state token "s" in P1.

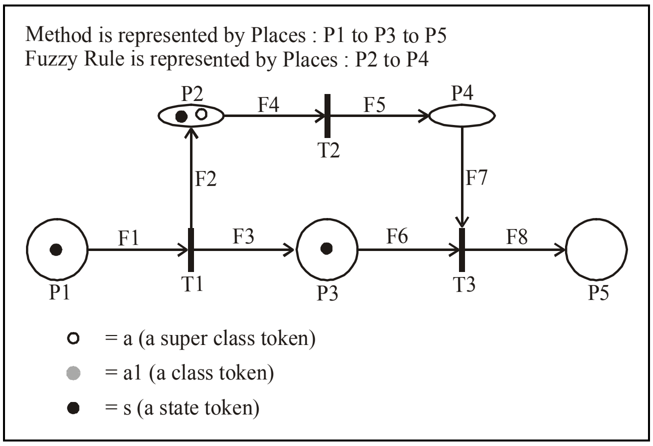

Figure 4b : Fuzzy Rule with Method (Rule is called by the Method). The token "a" was passed to P2 and a state token "s" was created in P1, P2 and P3 respectively.

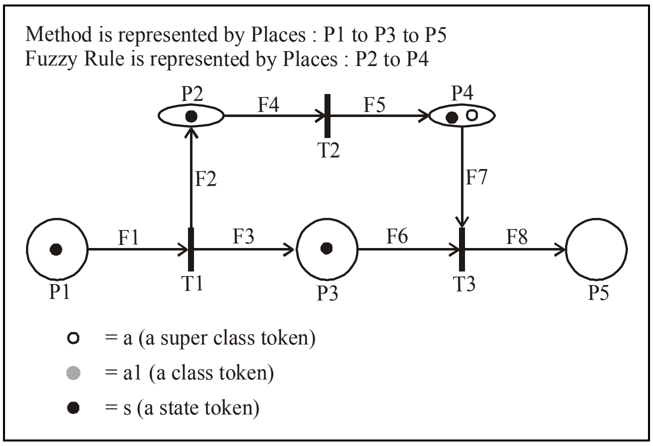

Figure 4c : Fuzzy Rule with Method (After firing). The token "a" is in P4 and a state token "s" in P1, P2 and P3 and P4 respectively.

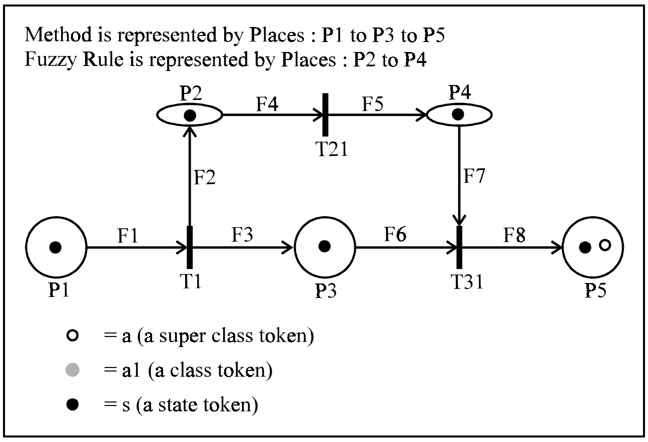

Figure 4d : Fuzzy Rule with Method (Method resumes control). The token "a" was passed to P5. A state token "s" was subsequently created in P1, P2, P3, P4 and P5 respectively.

The modelling of methods is divided into two parts. First the state of the method. (i.e. (1) executed some of the program codes and waiting to pass the control to the Fuzzy Rule, (2) waiting for the Fuzzy Rule to pass back the control, (3) executed all the program codes and waiting to pass the control to other process.) Secondly, the actual program codes of the method itself. (i.e. Represented by the arc expression functions.) In Figure 4a-d, P1 to P3 to P5 represent three states of the Method describe above. P2 to P4 represent the Fuzzy Rule embedded within the Method. The arc expression function F1, a part of the Method, will execute first. The control is then passed to the Fuzzy Rule by F2 which will create the "a" in P2. After firing of the Fuzzy Rule (T2 is enabled and fired), P3 and P4 will allow T3 to be fired. F8, represented the remaining part of the Method, will act on Object A correspondingly. After the execution of this Fuzzy Rule with Method, a state token "s" is deposited in all the Places, P1, P2, P3, P4 and P5 for preservation of the states.

3.2.6 Rules with Instances

This is represented in CPN by the arc expressions because the number of removed/added tokens and the colours of these tokens are determined by the value of the corresponding arc expressions.

Although the integration of a Fuzzy Rule- and Frame-based Expert System can take the advantages of both representation paradigms. The systems are not free from errors and anomalies. In a pure rule-based system, errors and anomalies could include redundancy, dead-end rules, subsumption, duplication, circular rule sets, unsatisfiable conditions, missing rules..etc. In a pure frame-based system, error and anomalies may occur due to the problems of message passing and concurrency, problems of inheritance (including simple, repeated and multiple inheritance) and problems of polymorphism. Instead of covering all the possible errors and anomalies caused by the integration of the above two representation paradigms, we would like to focus ourselves on the additional errors and anomalies attributed to the integration of fuzzy rules with inheritance of object properties. Details will be discussed in the following Section.

4 An Example of FHES: A Personnel Selection System

To illustrate the FHES modelling by our proposed CPN methodology, we adopt a simplified version of a personnel selection expert system currently being used in Hong Kong [5]. This system is used to find out, among all the clerks in the organization, who should be promoted to senior clerk. The organization's employee data structure is represented in a frame-based hierarchy as shown in Figure 5.

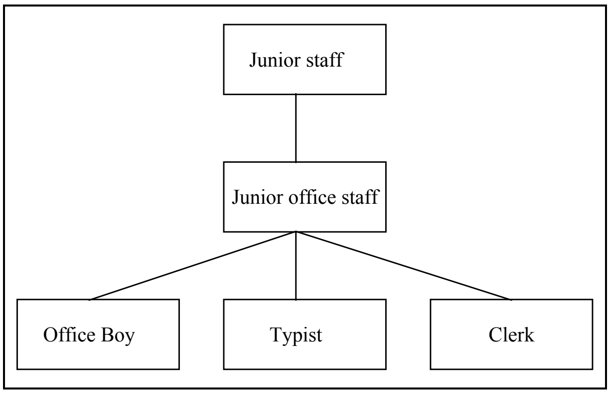

Figure 5 : The Frame Hierarchy

A Clerk frame's slots include: Name, Sex, Address, ID Number, Qualification, Salary, Department, Duties, Privilege, Year of Services, Knowledge of Work, Acceptance of Responsibility, Organization of Work, Initiative, Relations with Colleagues, Relations with Public, Expression on Paper, Oral Expression, Supervisory Skills, Leading Skills, Performance, Experience, Ability, Quality of Services, Seniority and Promotion.

A Clerk frame is similar to a Junior Office Staff frame except that more detailed information about the various types of Clerk duties are included such as Purchasing Clerk, Book Keeping Clerk, Sales Clerk, Inventory Clerk, Customer Services Clerk, Data Entry Clerk...etc. For the purpose of this modelling exercise, we can treat the Class Junior Office Staff as the common job grade in the organization, and the Class Clerk, Office Boy and Typist as specific job categories all belonging to the same job grade. Any new employment regulations and promotion rules that apply to Junior Office Staff grade will be applicable to all Clerks, Office Boys and Typists in the organization.

The major problems of verifying this FHES is due to the fact that some fuzzy rules are applicable to the general class (Super Class : Junior Office Staff) and through inheritance these fuzzy rules are applicable to specific classes as well (Classes : Clerks, Office Boy and Typists). Anomalies exist whenever fuzzy rules specifically applied to a class are in conflict with those rules that are applied to their superclass. Furthermore, these fuzzy rules may be in a subsumbed situation and some of them may be unreachable.

First, we model the above example using our proposed methodology described in previous sections. It is noted that a frame is equivalent to a data structure with various types declarations, (or an object with different attributes). Demons are declared as methods or procedures within some frame. In the above expert system example, the two frames (Junior Office Staff and Clerk) are Class frames. Each individual clerk's information is inferred by the creation of a clerk frame instance. The data value of Clerk Name, Sex, Address...etc are input via the user interface. The data values for slots between Knowledge of Work and Leading Skills inclusively are input by the individual clerk's supervisor at the beginning of the inference process. The data value of Performance, Experience, Ability, Quality of Services and Seniority are being inferred by the execution of the fuzzy rules pre-defined earlier by the personnel manager of the organization. The goal is to find out the data value of the slot Promotion, which can be inferred by forward chaining or backward chaining within the rule sets. (Over 100 rules were constructed for the original expert system based on the Multiple Criteria Decision Model. Detail data structure of a clerk token and some typical rules are given as follows:

A clerk token's colour is :

a. Colors for fixed attributes

Color Name = *string*; (all text strings)
Color Sex = *with* Male | Female; (colors explicitly specified)
Color Address = *string*;
Color IDNumber = *integer*;
Color Qualification = *string*;
Color Salary = *real*;
Color Department = *string*;
Color Duties = *string*;
Color Privilege = *with* Local | Overseas;
Color YrOfService = *integer*;
Color Promotion = *with* Yes | Wait | Reject;

b. Colors for fuzzy attributes

Color GML = *with* Good | Medium | Low;
Color FuzzyM = *real*;
Color GML_FuzzyM = *product* GML*FuzzyM;
Color KnowledgeOfWork_To_Seniority = *list* GML_Fuzzy *with 15*; (a list of 15 color of GML_Fuzzy)

c. Final Color of the clerk token

Color Clerk = *product* Name * Sex * Address * IDNumber * Qualification * Salary * Department * Duties * Privilege * YearOfServices * KnowledgeOfWork_To_Seniority * Promotion; (all tuples (name,sex,.....promotion) where name∈Name, sex ∈ Sex,....promotion ∈ Promotion)

d. Color for state tokens

Color State = *with* Yes | No;
Color StateToken = *product* State * FuzzyMeasure; (for state token, if the value is Yes, it denotes that the predicate is true, with $\mu_{true} >= \tau_{true}$ else if the value is No, the negation of the predicate is true with $\mu_{false} >= \tau_{false}$.)

var statetoken : StateToken; var clerk : Clerk; (var denotes variable declaration which introduces one or more variables. Here we have one variable, clerk, which is with colour Clerk. We may use var clerkJohn, clerkPeter, clerkDavid : Clerk for declaring three different clerks for example.)

Some typical rules are :

Rule 1: IF X is a junior office staff
AND X's quality of service is Good(τ=0.8)
AND X's seniority is High(τ=0.8)
THEN X's promotion is Yes.(μ=0.6)

Rule 2: IF X is a clerk
AND X's quality of service is Good(τ=0.7)
AND X's seniority is High(τ=0.7)
THEN X's promotion is Yes.(μ=0.8)

Rule 3: IF X is a clerk
AND X's quality of service is Good(τ=0.7)
AND X's seniority is High(τ=0.7)
AND X is a local citizen
THEN X's promotion is Yes.(μ=0.8)

Rule 4: IF X is a clerk
AND X's year of service is greater than Five
THEN X's seniority is Not High.(μ=0.5)

Rule 5: IF X is a junior office staff
AND X's year of service is greater than Five
THEN X's seniority is High.(μ=0.7)

Rule 6: IF X is a clerk
AND X's knowledge of wk is Not Good(τ=0.7)
AND X's English is Not Good(τ=0.8)
THEN X needs to attain training course.(μ=0.7)

Rule 7: IF X is a junior office staff
AND X needs to attain training course(τ=0.8)
THEN X's experience is Low.(μ=0.9)

Rule 8: IF X is a clerk
AND X is a junior office staff
THEN X is entitled to 14 days annual leave.

Rule 9: IF X is a office boy
AND X needs to attain training course.(μ=0.7)
THEN X is on Probation.

Rule 10: IF X is a junior office staff
THEN X is required to do typing.

Rule 11: IF X is required to do typing
THEN X is a clerk.

Rule 12: IF X is a clerk
THEN X is a junior office staff.

These rules can be rewritten as :

Rule 1: $A \wedge B(\tau=0.8) \wedge C(\tau=0.8) \Rightarrow X(\mu=0.6)$
Rule 2: $A1 \wedge B(\tau=0.7) \wedge C(\tau=0.7) \Rightarrow X(\mu=0.8)$
Rule 3: $A1 \wedge B(\tau=0.7) \wedge C(\tau=0.7) \wedge D \Rightarrow X(\mu=0.8)$
Rule 4: $A1 \wedge E \Rightarrow \neg C(\mu=0.5)$
Rule 5: $A \wedge E \Rightarrow C(\mu=0.7)$
Rule 6: $A1 \wedge \neg F(\tau=0.7) \wedge \neg G(\tau=0.8) \Rightarrow Y(\mu=0.7)$
Rule 7: $A \wedge Y(\tau=0.8) \Rightarrow H(\mu=0.9)$
Rule 8: $A1 \wedge A \Rightarrow K$
Rule 9: $A2 \wedge Y(\tau=0.7) \Rightarrow Z$
Rule 10: $A \Rightarrow L$
Rule 11: $L \Rightarrow A1$
Rule 12: $A1 \Rightarrow A$

Where the meanings of the literals used in the above rules are as follows:

A = Junior Office Staff
A1 = Clerk
A2 = Office Boy
B = Quality of service is Good
C = Seniority is High
¬C = Seniority is Not High
D = Local citizen
E = Years of service is greater than Five
¬F = Knowledge of work is Not Good
¬G = English is Not Good
H = Experience is Low
K = Entitled to 14 days annual leave
L = Required to do Typing
X = Promotion is Yes
Y = Needs to attain training course
Z = On Probation

The Fuzzy Hybrid Expert System is represented by Coloured Petri Nets shown in Figure 6, according to the methodology proposed in section three. Note that for simplicity, the self-loop associated with each input place is not shown in the net. The rules are labelled R1 to R12. The inheritance relations are represented by T1 to T3. S1 to S7 represents the predicates of these rules.

5 Analysis of Coloured Petri Nets

The major analysis technique, within the context of expert system verification, is the use of reachability tree which represents the reachability set of the CPN (or occurrence graph in Jensen's terminology). The basic idea behind is to construct a tree/graph containing a node for each reachable marking and an arc for each occurring binding element. In expert system verification, it refers to exhaustively exploring all the useful and relevant interactions of predicates within the model. From a given initial state, all possible transitions are generated, leading to a number of new states. This process is repeated for each of the newly generated states until no new states are generated. Obviously such a tree/graph may become very large even for a small CPN. However, research [7] has been taken to allow for a partial examination of a subportion of the reachability graph, therefore reduce the efforts in deriving possible solutions. For simplicity reason, without taking any transition conditions or transition operations into consideration, we concentrate our analysis by enabling a specific transition (i.e. corresponds to some meaningful initial facts) and then check the reachability set for any irregularities of the associated predicate places. The checking of the irregularities and anomalies can be done exhaustively or heuristically by adequately initiation of the sequence of transitions and closely examining the reachability markings. The problems can be located through the trace of the sequence of transitions which may provide alternative or multiple marking effects.

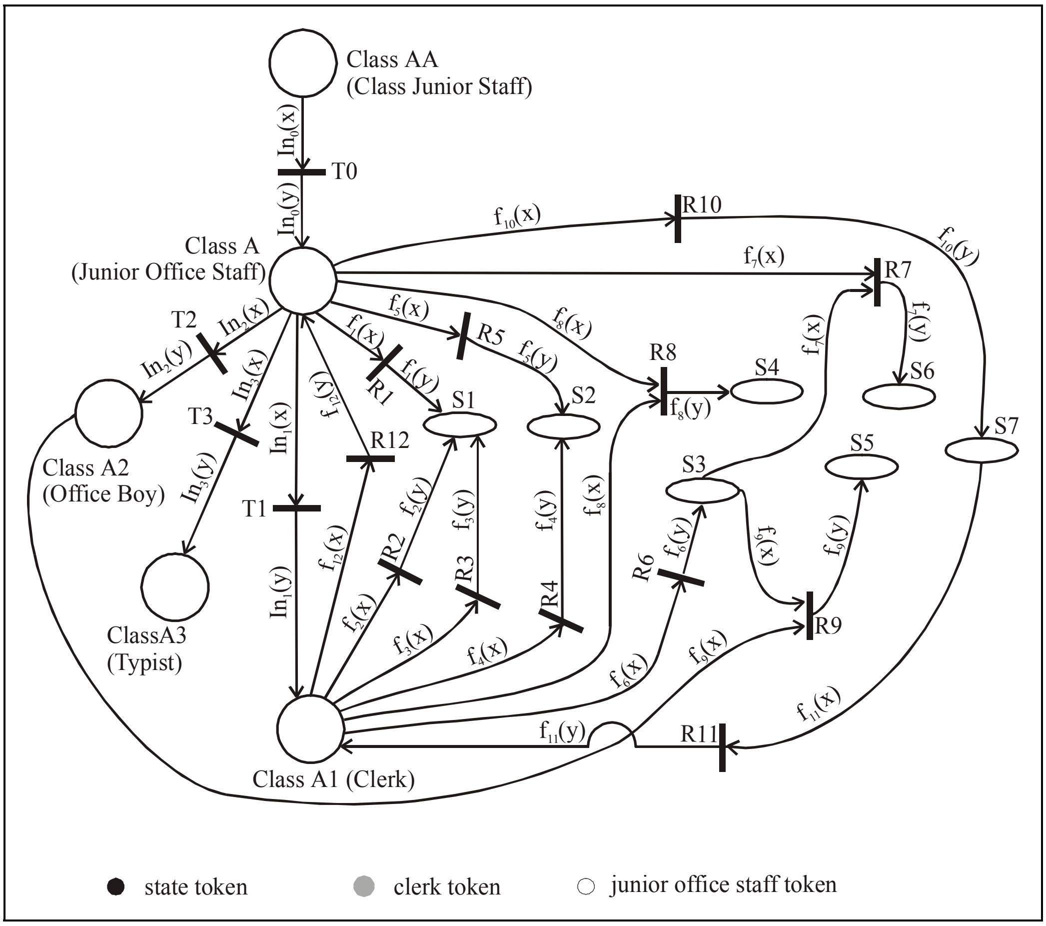

Figure 6 : CPN representation of the given FHES

6 Conclusion and Discussion

In this paper, we have described an approach to model fuzzy hybrid (rule- and frame-based) expert systems using Coloured Petri Nets and the concept of controlled state tokens. We have defined the properties of a FHES and illustrated how to model it by CPN. A practical personnel selection system is used as an example to show how the methodology is being used. The detection and analysis of the anomalies of proposed model is done by constructing and examining the reachability tree spanned by the knowledge inference. Our approach allows for formal verification of the correctness, consistency, and completeness of the fuzzy hybrid knowledge base.

Future work will include formalizing our approach and developing of algorithms and proof to detect irregularities in the FHES.

References

[1] Beauvieux, A. "A General Consistency Checking and Restoring Engine for Knowledge Bases". In Proceedings of the 9th European Conference on Artificial Intelligence, Stockholm, Sweden, 1990.

[2] Durkin, J. Expert Systems: Design and Development, Macmillan Publishing Company, pp.12-23, pp.711-771, 1994.

[3] Evertsz, R. "The Automated Analysis of Rule-based Systems Based on their Procedural Semantics". In Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, 1991.

[4] French, S.W. and Hamilton, D. "A Comprehensive Framework for Knowledge-Base Verification and Validation". In International Journal of Intelligent Systems, Vol. 9, John Wiley & Sons, Inc., pp.809-837, 1994.

[5] Huen, H.S.M. A Prototype Decision Support System for Assessing The Claims for Promotion of Clerical Officers in the Hong Kong Civil Service. M.Sc. Dissertation, Department of Computing, Hong Kong Polytechnic University, 1993.

[6] Jensen, K. Coloured Petri Nets : Basic Concepts, Analysis Methods and Practical Use, Vol. 1, Springer-Verlag, 1992. Vol. 2, Springer-Verlag, 1995.

[7] Li, X., Lai, R. and Dillon, T.S. "A New Decomposition Method to Relieve the State Space Explosion Problem". In Proceedings of the 5th International Conference on Computing and Information, Sudbury, Ontario, Canada, pp.150-154, 1993.

[8] Liu, N.K. "Formal Description Technique for the verification of fuzzy knowledge base redundancy and subsumption". In IEEE Proceedings of the 1st New Zealand International Conference on Artificial Neural Networks and Expert Systems, Dunedin, New Zealand, November, pp142-145, 1993.

[9] Liu, N.K. and Dillon T.S. "Formal Description and Verification of Production Systems", International Journal of Intelligent Systems, Vol. 10, No.4, pp.399-442, 1995.

[10] Nguyen, T.A., Perkins, W.A., Laffey, T.J. and Pecora, D. "Checking an expert system knowledge base for consistency and completeness". In Proceedings of the 9th International Joint Conference on Artificial Intelligence, Los Angeles, CA. 1985.

[11] O'Keefe, R.E. and O'Leary, D.E. "Expert System Verification and Validation: A Survey and Tutorial", Artificial Intelligence Review 7, pp.3-42, 1993.

[12] Shiu, S.C.K., Liu, J.N.K. and Yeung, D.S. "Modelling Hybrid Rule/Frame-based Expert Systems using Coloured Petri Nets", In Proceedings of the 8th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE-95, Melbourne, June 6-8, pp.525-531, 1995.

[13] Shiu, S.C.K., Liu, J.N.K. and Yeung, D.S. "An Approach Towards the Verification of Hybrid Rule/Frame-based Expert Systems using Coloured Petri Nets", In Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics, pp.2257-2262, Vancouver, Canada, October 22-25,1995.

[14] Suwa, M., Scott, A.C. and Shortliffe, E.H. "An Approach to Verifying Completeness and Consistency in a Rule-based Expert System". AI Magazine, pp.16-21, 1982.

[15] Vranes, S. and Stanojevic, M. "Integrating Multiple Paradigms within the Blackboard Framework". IEEE Transactions on Software Engineering, Vol. 21, No. 3, pp.244-262, March 1995.