MODELLING HYBRID RULE/FRAME-BASED EXPERT SYSTEMS USING COLOURED PETRI NETS

Simon C. K. SHIU*, James N. K. LIU, Daniel S. YEUNG
Department of Computing, Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
*Email : csckshiu@comp.polyu.edu.hk

ABSTRACT

High level Petri Nets have recently been used for many AI applications, particularly for modelling traditional rule-based expert systems. The major effect is to facilitate the analysis of the knowledge inference during the reasoning process, and to support the system verification which increasingly becomes an integral part of expert system development. Nevertheless, there is not much attention being put on systems other than the traditional ones. In this paper, we described an approach to model hybrid (rule- and frame-based) expert systems using Coloured Petri Nets and the concept of controlled state tokens. The analysis of the proposed model is by constructing and examining the reachability tree spanned by the knowledge inference. Such methodology has an implication for supporting the verification process in hybrid systems.

Keywords : Artificial Intelligence, Expert System Modelling, Verification, Coloured Petri Nets

INTRODUCTION

Expert Systems (ES) have reached the stage where they are implemented and used in a wide variety of organizations and industries, a selection of operational expert systems in US, Europe, Canada and the Far East can be found in [Lie91, Zar91, SC91 and LYM^{+}91]. There is increasing need for expert systems validation and verification (V&V) because erroneous advice may lead to invaluable economic loss and even fatal loss of life in some domain applications. Traditionally, attention has been concentrated in using verification techniques to tackle rule-based systems [SSS82, NPL^{+}85, Bea90, Eve91, LD91]. 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 today's expert systems are developed [OO93, Dur94]. 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. Although this approach benefits from the advantages of both representation techniques, it complicates the V&V of the expert systems. In this paper, we describe an approach based on Coloured Petri Nets(CPN) proposed by [Jen92] plus the concepts in State Controlled Petri Nets (SCPN) proposed by [Liu91] to model hybrid expert systems. It is through the analysis of the CPN that behavioural properties of the hybrid expert system can be verified and modified, thus enhancing the quality and reliability of the developed expert system.

Traditionally, there are a few approaches in modelling expert systems, such as [Cha91]'s Normal Form approach, [CS87, Van91]'s Decision Table Method, [Lan90]'s Incidence Matrix Method, [deK86]'s Truth Maintenance Systems, [CCS90, SC88, PS91]'s Generic Rule Systems. One of the major criticism of the above techniques is that none or very little consideration is given to allow for the dynamic checking of the knowledge inference. On the other hand, CPN, can support a formal description for modelling systems which consists of concurrent and synchronous activities. Besides, they also have a graphical representation and a well-defined semantics allowing for dynmaic analysis of the modelled system. In this paper, a contribution is made to the modelling of hybrid rule/frame-based expert systems. We will introduce an approach based on Coloured Petri Nets plus the concept of state tokens for the representation of knowledge inference in a HES. We will examine the transition sequences and check against the properties of the network in CPN for HES modelling. The paper is organized in five main sections. The first section gives the introduction, the second section describes a practical example of a hybrid expert system, and the third section gives the definitions of CPN and illustrates how the HES example can be modelled by CPN. Section four discusses the methods for analyzing the CPN and the implications of our approach to support formal verification of the systems. The last session gives the conclusion and discussion.

A HYBRID EXPERT SYSTEM EXAMPLE

We adopt a simplified version of a personnel selection expert system which has currently been used in Hong Kong [Hue93] for illustrating the HES modelling by our proposed CPN methodology. The aim of this system is 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 1 and details of relevant frames in the hierarchical structure are given below.

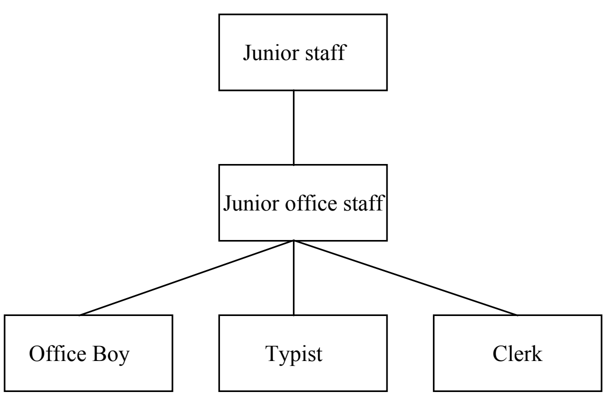

Figure 1 : The Frame Hierarchy

A Junior Staff Frame :

Slot Name	Value	Type	Demons
Job Title	Junior Staff	String	
Office Hours	09:00 17:00	Time	
Qualification Requirement	Five HKCEE	String	
Salary Pay Scale	1 to 10	String	
Father Frame	Staff		
Son Frame	Junior Office Staff		

A Junior Office Staff Frame :

Slot Name	Value	Type	Demons
*Job Title	Junior Office Staff	String	
*Office Hours	09:00 17:00	Time	
*Qualification Requirement	Five HKCEE	String	
*Salary Pay Scale	1 to 10	String	
Department	General Sec.	String	
Annual leave	21 days	Integer	
Father Frame	Junior Staff		
Son Frame	Clerk, Typist and Office boy.		

* denotes slots inherited from parent frame

A Clerk Frame :

Slot Name	Value	Type	Demons
*Job Title	Clerk	String	
Clerk Name		String	
Address and Telephone		String	

HKID#		String	IF process HKID# THEN Privilege is Local ELSE Privilege is Overseas.
Privilege		Local/ Overseas	
Sex		M/F	
*Office Hours	09:00 17:00	Time	
*Qualification Requirement	Five HKCEE	String	
*Department	General Sec.	String	
*Salary Pay Scale	1 to 10	String	
Present Salary Point		Integer	Present Salary Point must between 1 to 10 inclusive.
Years of Service		Integer	
*Annual leave	21 days	Integer	
Leave taken		Integer	
Leave balance		Integer	Leave balance = Annual leave - Leave taken
Knowledge of Work		G/M/L (Good, Medium, Low)	
Acceptance of Responsibility		G/M/L	
Organization of Work		G/M/L	
Initiative		G/M/L	
Relations with Colleagues		G/M/L	
Relations with Public		G/M/L	
Expression on Paper		G/M/L	
Oral Expression		G/M/L	
Supervisory Skills		G/M/L	
Leading Skills		G/M/L	
Performance		G/M/L	
Experience		G/M/L	
Ability		G/M/L	
Quality of Services		G/M/L	
Seniority		G/M/L	
Promotion		Yes /Wait /Reject	
Father Frame	Junior Office Staff		
Son Frame	NIL		

* denotes slots inherited from parent frame

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 the frame. In the above example, the three frames are Class frames. Each individual clerk's information is captured 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 and demons in the slots with a * are inherited from the parent frame, the data value of Privilege and Leave balance are updated by firing the demons in HKID# and Leave balance. 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 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 [Tay84]). Typical rules are as follows:

Rule 1 : IF Quality of Services is Good
AND Seniority is Good
THEN Promotion is Yes.

Rule 2 : IF Quality of Services is Good
AND Seniority is Medium
THEN Promotion is Wait.

Rule 3 : IF Privilege is Local
AND Years of Service is more than 5 years
AND Present Salary Point is 10
THEN Seniority is Good.

Rule 4 : IF Performance is Good
AND Experience is Good
AND Ability is Good
THEN Quality of Service is Good.

In order to model the above HES, we have identified the need to provide an enriched modelling language which must be able to represent the hierarchical framed data structure, the individual frame instance, the various data types of each slot in a frame instance, the demons attached to individual slots, the separate rule sets which the truth of predicates are determined by firing the rules using the value in the slots. Details are as follows.

MODELING THE HYBRID EXPERT SYSTEMS USING COLOURED PETRI NETS

Definition of CPN

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 : $P \cap T = P \cap A = T \cap A = \varnothing$.
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. It is assumed that each colour set has at least one element.

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 kind (ie 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, ie a multi-set over C(p).

Illustrative Example

The Personnel Selection Expert System can be modelled by CPN as follows :

First, each framed data structure is represented by a compound colour set, and each frame instance is represented by a token in that set. For instance, there are fifteen sets of non-empty types or colour sets being used in the illustrated example, ie Σ=AA,BB,...OO:

Color AA = string; (all text strings)
Color BB = with Local | Overseas; (colours explicitly specified)
Color CC = with Male | Female;
Color DD = time; (date)
Color EE = integer with 0..10;(between 0&10)
Color FF = integer;
Color GG = with Good | Medium | Low;
Color HH = with Yes | Wait | Reject;
Color II = list AA with 4; (a list of four strings)
Color JJ = list AA with 3;
Color KK = list FF with 5;
Color LL = list GG with 15;
Color MM = with Clerk | Typist | Office Boy;
Color NN = product II * BB * CC * DD * JJ * KK * LL * HH; (all tuples (i,b,c,d,j,k,l,h) where $i \in II$, $b \in BB$,....$h \in HH$)
Color OO = with Yes | No; (for state token, if the value is Yes, it denotes that the predicate is true, else if the value is No, the negation of the predicate is true.)

Var i:II; var b:BB; var c:CC; var d:DD; var j:JJ; var k:KK; var l:LL; var h:HH; var clerk : NN; (var denotes variable declaration which introduces one or more variables. Here we have one variable, clerk, which is with colour NN. We may use var clerk1, clerk2, clerk3 : NN for declaring three different clerks for example.)

Secondly, places in the CPN are taken to correspond to predicates of the production rules which are pre-defined earlier by the user, and the transitions in the CPN correspond to the implications of the rules. The transition operations are represented by the arc expression functions. (eg. Rule 1 can be represented in CPN as shown in Figures 2a & 2b:)

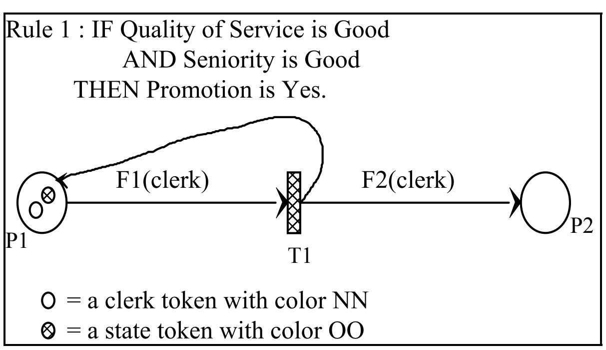

Figure 2a : Rule 1 (before firing) with an input token Clerk and a state token in P1 with 'Yes'

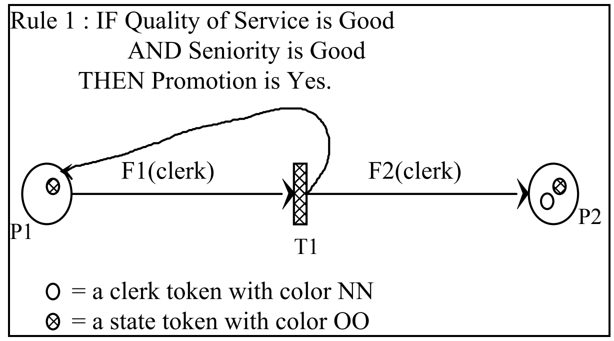

Figure 2b : Rule 1 (after firing) with an output token Clerk and a state token in P2 with 'Yes'

P1 and P2 are Places with colour type NN, T1 is a Transition which is enabled iff the input arc expression F1(clerk) is evaluated to be true. If F1(clerk) is true then T1 is fired, it implies that Rule 1 is executed. The token clerk will be removed from P1 and a new token clerk will be created in P2 with new data values determined by the output arc expression F2(clerk). A state token with 'Yes' value will also be created in P2 for further inference (if any). In order to preserve the state of the predicate in P1, a state token marked with 'Yes' is created in P1 via the self-loop. In the example F1(clerk) is an arc expression function which evaluates if the data value of variable l[14][1] and l[15] are equal to "Good", and if true, it will assign the value "Yes" to variable h. After a sequence of firings has been exhaustly executed, we can examine the attributes in the token clerk for consideration for promotion to senior clerk.

Similarly, a demon can also be represented by a Places/Transition tuple. (eg. the demon attached to slot HKID# is given in Figure 3 prior to its execution)

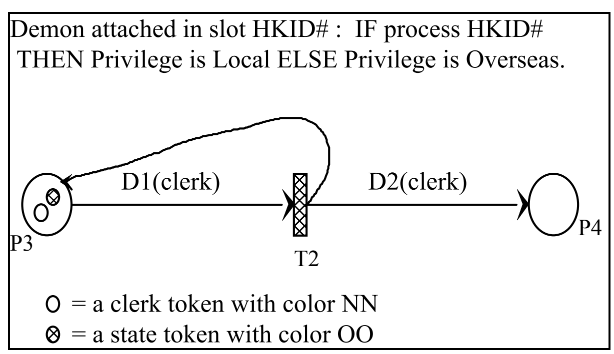

Figure 3 : Demon attached to slot HKID#

ANALYSIS OF COLOURED PETRI NETS

[1] l[14] means the 14th element in the list variable l:LL with color Good | Medium | Low.

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 possible 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. In our example, by constructing and examining the reachability tree using one particular clerk token (equivalent to an initial marking of the CPN because different clerks have different values in their attribute slots), we can detect any irregularities of the predicate places. Obviously such a tree/graph may become very large even for a small CPN. Therefore, for simplicity reason, without taking any transition conditions or transition operations into consideration, we can concentrate our analysis by enabling a specific transition and then check the reachability set for any irregularities of the associated predicate places. In a hybrid rule/frame based expert system, irregularities may come from the rule sets or the frames or both. Typically, there may be redundant rules or frames, dead end rules, missing rules or frames, misplaced slots and frames, duplication of frames, subsumptions of rules, auxiliary rules or frames, circular rule sets, contradiction rules, inconsistent rules. The checking of the above irregularities 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.

CONCLUSION AND DISCUSSION

In this paper, we have described an approach to model hybrid rule/frame based expert system using Coloured Petri Nets and the concept in State Controlled Petri Nets. The frame data structures are represented using colour tokens. The data value of each colour token may be of an arbitrarily complex type, thus enabling various classes of frames be represented. The rules and demons are represented by a Place/Transition tuple with a self-loop in order to preserve the state of the predicate. The analysis of the CPN is by constructing and examining the reachability tree to detect irregularities of the predicate places.

In a pure frame-based expert system, reasoning is by comparing descriptions of incoming facts with the frames in the knowledge base, and retrieving the class frame that best matches the situation. The main inference mechanism or strategy for applying general information to specific instances is inheritance. This reasoning mechanism is rather limited in practical situations. In a pure rule-based expert system, reasoning is by firing a sequence of rules using incoming facts. Although this method is simple and useful, complex domain knowledge could not be represented. Our approach is useful to model systems that combined rule/frame based representation techniques.

Future work will include formalizing our approach and developing of algorithms to detect irregularities in the HES. We would also like to investigate further the capability of the methodology to handle fuzzy systems and the complexity involved against the traditional approaches.

REFERENCES

[Bea90] 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.

[CCS90] Chang, C.L., Combs, J.B. and Stachowitz, R.A. A Report on the Expert Systems Validation Associate (EVA). *Experts Systems with Applications,* Vol. 1, No. 3, pp219-230, 1990.

[Cha91] Charles, E. Checking Knowledge Bases for Inconsistencies and other Anomalies. In *Workshop Notes from the Ninth National Conference in Artificial Intelligence,* AAAI-91, Knowledge-Based Systems Verification, Validation and Testing, 17 July, Anaheim CA. 1991.

[CS87] Cragen, B.J. and Steudel, H.J. A Decision Table Based Processor for Checking Completeness and Consistency in Rule-Based Expert Systems. *International Journal of Man-Machine Studies,* Vol 26, pp633-648, 1987.

[deK86] de Kleer, J. An Assumption-Based TMS. *Artificial Intelligence* 28, pp127-162, 1986.

[Dur94] Durkin, J. *Expert Systems: Design and Development,* Macmillan Publishing Company, pp12-23, pp711-771, 1994.

[Eve91] 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.

[Hue93] 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.

[Jen92] Jensen, K. *Coloured Petri Nets : Basic Concepts, Analysis Methods and Practical Use, Volume 1*, Springer-Verlag, 1992.

[Lan90] Landauer, C. Correctness Principles for Rule-Based Expert Systems. *Expert Systems with Applications,* Vol 1, No. 3, pp291-317, 1990.

[LYM+91] Lee, J.K., Yeung, D.S., Mizoguchi R. and Narasimhalu D. *Operational Expert System Applications in the Far East*, Published by Pergamon Press, 1991.

[Lie91] Liebowitz, J. *Operational Expert System Applications in the United States*, Published by Pergamon Press, 1991.

[LD91] Liu, N.K. and Dillon, T.S. An Approach Towards the Verification of Expert Systems Using Numerical Petri Nets. *International Journal of Intelligent Systems,* 6, pp. 255-276, 1991.

[Liu91] Liu, N.K. *Formal Description and Verification of Expert Systems.* Ph.D. dissertation, Department of Computer Science and Computer Engineering, School of Mathematical and Information Sciences, La Trobe University, Bundoora, Victoria, Australia, 1991.

[NPL+85] 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.

[OO93] O'Keefe, R.E. and O'Leary, D.E. Expert System verification and validation: a survey and tutorial, *Artificial Intelligence Review* 7, pp3-42, 1993.

[PS91] Preece, A.D. and Shinghal, R. DARC: A Procedure for Verifying Rule-Based Systems. In *Expert Systems World Congress Proceedings,* Liebowitz, J. (ed) Vol. 2, Pergamon Press, pp971-979, 1991.

[SC88] Stachowitz, R.A. and Chang, C.L. Verification and Validation of Expert Systems. *Tutorial note at AAAI-88.*

[SC91] Suen, C.Y. and Chinghal, R. *Operational Expert System Applications in Canada*, Published by Pergamon Press, 1991.

[SSS82] Suwa, M., Scott, A.C. and Shortliffe, E.H. An approach to verifying completeness and consistency in a rule-based expert system. *AI Magazine*, pp16-21, 1982.

[Tay84] Taylor, R.N. *Behavioural Decision Making*, Scott Foresman & Company, 1984.

[Van91] Vanthienen, J. Knowledge Acquisition and Validation Using a Decision Table Engineering Workbench. In *Expert Systems World Congress Proceedings,* Liebowitz, J. (ed), Pergamon Press, Vol 3, pp1861-1868, 1991.

[Zar91] Zarri, G.P. *Operational Expert System Applications in Europe*, Published by Pergamon Press, 1991.