Proceedings of the 6th Győr Symposium and 3rd Hungarian-Polish
and 1st Hungarian-Romanian Joint Conference on Computational Intelligence

Using Signatures for Expert System Modeling Concepts definitions

Claudiu Pozna[a,b], Radu-Emil Precup [c], Foldesi Peter[d], Koczy Laszlo[f]

[a]**Department of Informatics, Széchenyi István University, Egyetem tér 1, H-9026 Gyõr, Hungary**

[b]**Department of Automation, Transilvania University of Brasov, Bd. Eroilor 28, RO-500036 Brasov, Romania**

[c]**Department of Automation and Applied Informatics, "Politehnica" University of Timisoara, Bd. V. Parvan 2, RO-300223 Timisoara, Romania**

[d]**Department of Logistic, Széchenyi István University, Egyetem tér 1, H-9026 Gyõr, Hungary**

[f]**Department of Automation, Széchenyi István University, Egyetem tér 1, H-9026 Gyõr, Hungary**

1. Introduction

The expert systems are products of artificial intelligence which proved successful implementations aiming the goal of solving problems which belong to particular domains and use expert level knowledge in those domains. Such successful implementation include the adaptation of emotional experiences of software engineers to the evolutionary design of software systems [6], evolving classification and fuzzy systems [10, 1], human computer interaction based on emotional modeling and physical views [7], distributed and collaborative data stream mining [34], knowledge classification and navigation [14] or adaptive systems [36, 37].

The main contribution of this paper is a new approach to expert system modeling based on signatures. The signatures and their operators are defined in [29], and they are a convenient framework for the symbolic representation of data. The signatures are a generalization of fuzzy signatures [2, 16].

The new modeling approach is formulated as an original three-step algorithm that maps the signatures onto expert systems. The algorithm has two inputs represented by the knowledge base (the rules) and the data base (the facts). The algorithm constructs the signatures which represent expert system models.

Our new expert system modeling approach is important and advantageous with respect to the state-of-the-art because:

- The systematic formulation in terms of an algorithm offers transparency which enables relatively simple modeling.
- The formulation of the algorithm is general and applicable to wide areas of expert systems.
- The algorithm allows the modeling of uncertain expert systems.

This paper is organized as follows: a short overview on signatures and on their operators is presented in the next Section. Section 3 is dedicated to the new modeling approach. An illustrative example is included and the modeling algorithm is expressed. The conclusions are outlined in Section 4.

2. Overview on Signatures and on Operators on Signatures

The concept of signatures and the operators on signatures are defined in [29]. A part of the definitions which enable the expert system modeling is presented in this section. Let $S^{(n)}$ be a set defined recursively as

$$S^{(n)} = \prod_{i=1}^{n} S_i , \tag{1}$$

where

$$S_i = \mathbf{R},\ i = 1...n , \tag{2}$$

or

$$S_i = S^{(m)},\ m \geq 1 , \tag{3}$$

$\mathbf{R}$ is the set of real numbers, and $\prod$ is the Cartesian product.

Definition 1. Let X be a nonempty set. The collection of signatures is defined as the function $A: X \to S^{(n)}$, and the signature of the element $x \in X$ is $A(x) \in S^{(n)}$ given as

$$A(x) = \begin{bmatrix} \ldots \\ a_i \\ \begin{bmatrix} a_{i+1,1} \\ a_{i+1,2} \end{bmatrix} \\ \begin{bmatrix} a_{i+2,1} \\ \begin{bmatrix} a_{i+2,2,1} \\ a_{i+2,2,2} \end{bmatrix} \end{bmatrix} \\ \ldots \end{bmatrix} , \tag{4}$$

and the transposition of the signature $A(x)$ is represented by $A^T(x)$

$$A^T(x) = [\ldots \quad a_i \quad [a_{i+1,1} \quad a_{i+1,2}] \quad [a_{i+2,1} \quad [a_{i+2,2,1} \quad a_{i+2,2,2}]] \quad \ldots]. \tag{5}$$

The following notations are introduced in [29] to simplify the characterization of signatures:

- A signature $A(x)$ with the values $a_1, a_2, \ldots, a_n, a_{i,1}, a_{i,2}, \ldots, a_{i,m}, \ldots a_{j,k,l}, \ldots$, is indicated by $a^{\cdots}$.
- If $\exists x \in X$ and $A^T(x) = [a_1 \quad \ldots \quad a_n]$, then we will use the notation $A(x) = a^{1,\ldots,n}$.
- If $\exists y \in Y$ and $A^T(y) = [a_1 \quad \ldots \quad a_{i-1} \quad [a_{i,1} \quad \ldots \quad a_{i,m}] \quad a_{i+1} \quad \ldots \quad a_n]$, then we will use the notation $A(y) = a^{1,\ldots,[1,\ldots,m]_i,\ldots,n}$. In this case the sets are defined as $S_1 = S_2 = \ldots = S_{i-1} = S_{i+1} = \ldots = S_n = \mathbf{R}$, and their Cartesian product is expressed as $S_i = \prod_{i=1}^{m} \mathbf{R} = \mathbf{R}^m$.
- A signature of type $[\ldots \quad [[a_1]] \quad \ldots]$ is equivalent to the signature $[a_1]$, where $a_1 \in \mathbf{R}$.
- As shown in [29], the signatures can be used in complex data representation. The following operators on signatures have been defined with this regard:
- basic operations: contraction, extension, pruning, and grafting,
- complex operations: addition and multiplication.

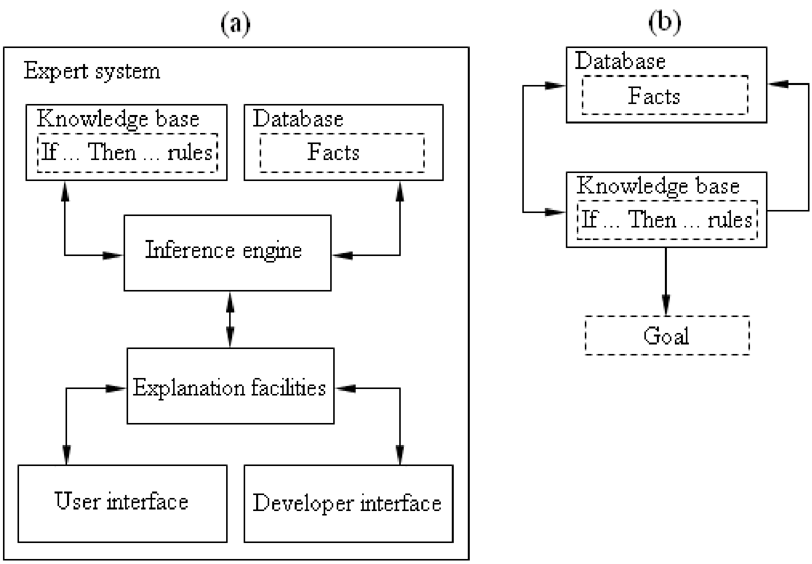

Fig. 1. Structure of rule-based expert system (a), and cycles of inference engine (b).

3. Modeling Approach

The structure of a rule-based expert system is presented in Fig. 1 (a) which points out the following subsystems: the knowledge base which contains the "If...Then..." rules, the database which contains the facts, the inference engine where the goal of the expert system is computed, and the user interface where the user interacts with the expert system. Several internal elements can be added to this structure; they include explanation facilities where the results are explained systematically, and the developer interface where the expert system interacts with the developer. External elements can be included as well like external databases or programs which support the inference engine.

The core of the expert system is the inference engine, where the rules are fired using the known facts. After firing a rule a new fact is inferred; this can fire in turn a new rule. This process is cyclic, and it can be represented by the schema illustrated in Fig. 1 (b). The end of the cycle is obtained when no more rules can be fired and the knowledge on the goal is obtained.

The inference chains can be different because of the observed facts. This means that we can divide the operating processes of expert systems in two steps:

I. First, carry out the backward construction of the signature starting with the goal and replacing the unobserved facts with rules until all rules contains observed facts.

II. Second, apply certain operators to the already constructed signature and compute the goal of the expert system.

The following definition concerns a rule which is a dependency between two types of facts, viz. the antecedents and the consequences.

Definition 8. The modeling of rules by signatures is

Rule $r_1 : If\ A\, Then\ Z$ is equivalent to $r_1 = [A]^T$ and $@(r_1) = [Z]^T$, (24)

or

Rule $r_1 : If\ A \wedge B \wedge C\ Then\ Z$ is equivalent to $r_1 = [A, B, C]^T$ and ${}^f@(r_1) = [Z]^T$, (25)

or

Rule $r_1 : If\ A\, Then\ Z$ $\vee$ Rule $r_2 : If\ B\, Then\ Z$ is equivalent to $r = [A, B]^T$
and ${}^g@(r) = [Z]^T$, (26)

where $\wedge$ stands for the conjunction, $\vee$ stands for the disjunction, and f and g are functions related to the conjunction and to the disjunction, respectively:

$$f : \{0,1\}^n \to \{0,1\},\ f(x_1,...,x_n) = \min_{i=1...n} x_i, \tag{27}$$

$$g : \{0,1\}^n \to \{0,1\},\ g(x_1,...,x_n) = \max_{i=1...n} x_i \cdot \tag{28}$$

The addition of signatures is used in order to replace an antecedent fact with a rule in terms of the following definition.

Definition 9. Let the two rules be

Rule r_1 : *If* $X \wedge Y$ *Then* Z , equivalent to $r_1 = [X, Y]^T$ and ${}^f@(r_1) = [Z]^T$, (29)

Rule r_2 : *If* $V \wedge W$ *Then* X equivalent to $r_2 = [V, W]^T$ and ${}^f@(r_2) = [X]^T$. (30)

The modeling of the replacement of an antecedent fact with these two rules is

$$r = r_1 \oplus_1 r_2 = [[V, W], Y]^T , \tag{31}$$

where r is obtained as an inference of r_1 and r_2 .

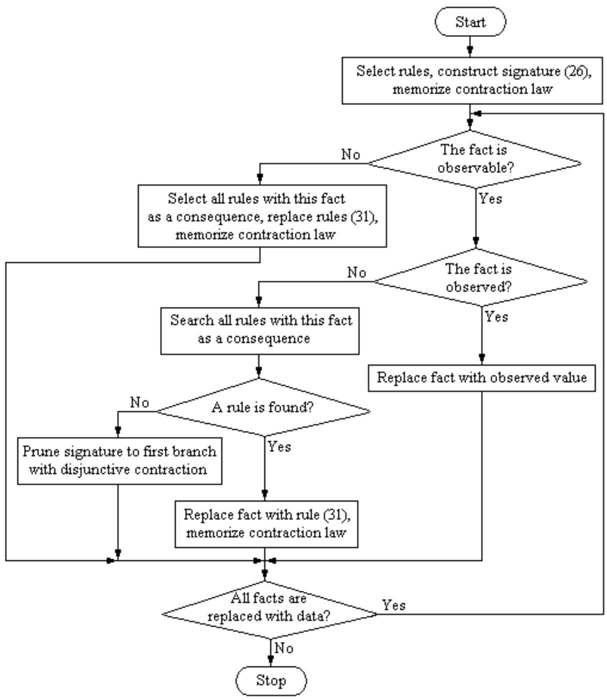

Fig. 2. Flowchart of expert system modeling algorithm.

As suggested in Fig. 1 (a), the algorithm uses two inputs, the knowledge base (the rules) and the database (the facts). The expert system modeling algorithm consists of the following steps:

Step 1. Select from the knowledge base those rules which are related to the expert system goal, use equation (26) to construct the signature, and memorize the contraction law of the signature.

Step 2. Develop the signature by the one-by-one investigation of the facts contained in the signature:

- If the fact is unobservable, select all rules from the knowledge base which refer this fact as a consequence, replace them using equation (31), and memorize the contraction law,
- If the fact is observable, search the database to find out if the fact has actually been observed:
  - If yes, replace it with the observed value,
  - If not, search the database to find the rules which refer this fact as a consequence:
    - If a rule is found, replace the fact with the rule using equation (31), and memorize the contraction law,
    - If a rule is not found, prune the signature from this fact (leaf) to the first branch which supposes a disjunctive contraction.

Step 3. Go to step 2 until all facts of the signature are replaced with data.

The flowchart of the algorithm is presented in Fig. 2. This algorithm can be simplified if the rules which contain unobservable facts are identified. The idea is to compute a priori composed rules. Equation (29) is employed in such cases in order to generate a signature which can be used directly at step 2; equations (40) and (41) illustrate this point. The application of our algorithm is exemplified in the next section by two case studies.

4. Conclusions

This paper has proposed an expert system modeling approach based on the use of signatures. This approach is systematic because the signatures are produced by an original expert system modeling algorithm.

The proposed approach has proved to be effective in accounting for certain observations, and the results have been generalized to uncertain observations

Acknowledgement

The paper is supported by the Romanian Ministry of Education, Research and Innovation through the PNII Idei project 842/2008; by a grant of the Romanian National Authority for Scientific Research, CNCS – UEFISCDI, project number PN-II-ID-PCE-2011-3-0109 and by The presented research studies are sponsored by the TAMOP-4.2.2.C-11/1/KONV-2012-0012: Smarter Transport - IT for co-operative transport system. The Project is supported by the Hungarian Government and co-financed by the European Social Fund.

References

[1] P. Angelov, Fuzzily connected multi-model systems evolving autonomously from data streams, IEEE Transactions on Systems, Man, and Cybernetics – Part B, Cybernetics 41 (4) (2011) 898–910.

[2] Á. Ballagi, L. T. Kóczy, T. D. Gedeon, Robot cooperator without explicit communication by fuzzy signatures and decision trees, in: Proc. Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference (IFSA-EUSFLAT 2009), Lisbon, Portugal, 2009, pp. 1468–1473.

[3] V. Bobál, M. Kubalčík, P. Chalupa, P. Dostál, Self-tuning control of nonlinear servo system: comparison of LQ and predictive approach, in: Proc. 17th Mediterranean Conference on Control and Automation, Thessaloniki, Greece, 2009, pp. 240–245.

[4] B. Brandherm, H. Prendinger, M. Ishizuka, Dynamic Bayesian network based interest estimation for visual attentive presentation agents, in: Proc. 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, 2008, pp. 191–198.

[5] F. J. Cabrerizo, I. J. Pérez, E. Herrera-Viedma, Managing the consensus in group decision making in an unbalanced fuzzy linguistic context with incomplete information, Knowledge-Based Systems 23 (2) (2010) 169–181.

[6] H. Fujita, J. Hakura, M. Kurematu, Intelligent human interface based on mental cloning-based software, Knowledge-Based Systems 22 (3) (2009) 216–234.