Annales Univ. Sci. Budapest., Sect. Comp. 12 (1991) 223-229

A PROTOTYPE FUZZY EXPERT SYSTEM SHELL*

Zenon A. Sosnowski**
Knowledge Systems Laboratory
Institute for Information Technology
National Research Council of Canada
Ottawa, Canada K1A 0R6

Abstract: The paper describes changes made on top of the CLIPS system to implement a fuzzy expert system shell. The modifications made to CLIPS contain the capability of handling fuzzy concepts and reasoning. This extended version allows any mix of fuzzy and normal terms, numeric-comparison logic controls, and uncertainties in the rules and facts. Fuzzy sets and relations deal with fuzziness in approximate reasoning, while fuzzy numbers manipulate the uncertainty.

Keywords: Expert systems, knowledge representation, fuzzy logic, CLIPS

1. INTRODUCTION

Many of today's commercial expert system building tools or shells use different approaches to handle uncertainty in the knowledge or data, but they cannot cope with fuzzy data, which constitute a very significant part of a natural language. Several systems support some fuzzy reasoning but they are purposely built from high-level languages for a specific domain of application. Low availability of these systems on conventional computers and their poor integration with other languages make embedded application difficult. To solve these problems some work has been undertaken on developing an expert system tool written in and fully integrated with the C language for high portability, low cost, and easy integration with external systems. The paper describes a system that is an extension of CLIPS [1] for representing and manipulating fuzzy facts and rules.

* NRCC Publication No. 31799

** On leave from the Department of Computer Science, Technical University of Bialystok, Bialystok, Poland.

2. REPRESENTATION OF FUZZY DATA

Much information resident in the knowledge base of an expert system is imprecise, incomplete, or not totally reliable. The representation of this kind of information will be based on the concept of the fuzzy set theory [2]. Two basic inexact concepts, fuzziness and uncertainty, which are distinct from each other in the system, are used.

2.1. Fuzziness

Fuzziness occurs when the boundary of a piece of information is not clear-cut. Example:

```
(age young)
(defrule one (Speed_error big)
⇒ (assert (Throttle_change small)) }
```

where *young, big, and small* are fuzzy terms.

The representation of fuzziness is based on the concept of a linguistic variable [3]. Any fact or LHS pattern is actually a data proposition of the form: (<object> <value>). If <value> contains fuzzy terms, then <object> is called a fuzzy object and is considered to be a name of a linguistic variable [3]. All fuzzy objects must be defined before use with the aid of the deffuzzy statement. The definition must contain a name of a linguistic variable (a fuzzy object), a list of all the possible primary terms (defined as fuzzy sets) it can take, and a list of all modifiers.
Example:

```
(deffuzzy Speed_error ; lv – linguistic variable
      ( ; primary term definitions
       (big     0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1)
       (small   1 0.9 0.8 0.7 0.6 0.5 0.4 0.3.0 2 0.1 0) )
      ( ; modifier definitions
       (more_or_less sqrt) ) )
```

The syntax and semantics associated with the linguistic variable are described in [4].

2.2. Uncertainty

Uncertainty occurs when one is not absolutely certain about a piece of information. The degree of certainty is represented by a crisp or fuzzy number [5].

Example:

```
(3 is small) # close_to 1
(defrule two (declare (CF 0.9))
    (?n is small)
    ⇒ (assert (=(+ ?n 1) is small)) )
```

where *close to* 1 and 0.9 are certainty factors expressed, respectively, by fuzzy and crisp numbers.

Uncertainty and fuzziness may occur simultaneously.

If the certainty factor of a fact or a rule is missing, it is assumed that it is equal to 1.

3. INFERENCE TECHNIQUES

Rule evaluation depends on whether or not antecedents or consequences of the rule contain fuzzy objects. Three types of rule can be distinguished: CRISP_, FUZZY_CRISP, and FUZZY_FUZZY. If the antecedent of the rule does not contain a fuzzy object, then the type of rule is CRISP_ regardless of whether or not a consequence contains a fuzzy object. If only the antecedent contains a fuzzy object, then the type of rule is FUZZY_CRISP. If both antecedent and consequence contain fuzzy objects, then the type of rule is FUZZY_FUZZY.

3.1. Simple rules

In this section the rule of the form "*if* A *then* C" where A is an antecedent and C is a consequence will be considered. Both antecedent and consequence can have only a single pattern. The method of handling rules which contain many patterns in the antecedent or consequence will be described later in this paper.

The inference mechanism realizes a generalized modus ponens rule. It may be described as follows.

$$\frac{A',\ if\ A\ then\ C}{C'}$$

Let CF_r , CF_f, and CF_c be uncertainties of the rule, fact, and conclusion, respectively.

If the type of rule is CRISP_, then A′ must be equal to A in order to apply this rule. In that case the conclusion C′ is equal to C, and

$$CF_c = CF_r * CF_f \tag{1}$$

where * denotes fuzzy numbers multiplication [5].

If the type of rule is FUZZY_CRISP, then A′ must have the same fuzzy object as A in order to apply this rule. Values of a fuzzy object in A and A′ represented by fuzzy sets F_a and F'_a do not have to be equal. In that case the conclusion C' is equal to C, and

$$CF_c = CF_r * CF_f * S$$

where S is a measure of similarity [6] between fuzzy sets F_a and F'_a.

If the type of rule is FUZZY_FUZZY, then it was shown in [3] that the antecedent and consequence of such a rule are connected by the fuzzy relation. The algorithms for forming this relation can be found in [7]. The calculation of conclusion is based upon the compositional rule of inference [3]. The certainty factor of the conclusion is calculated according to (1).

3.2. Composed rules

In CLIPS, the consequent part of the rule can contain only multiple patterns with the implicit and conjunction between them. They are treated as multiple rules with a single consequence. Therefore, only the problem of multiple patterns in the antecedent with a single pattern in consequence needs to be considered. If the consequent pattern does not contain a fuzzy object, no special treatment is needed. However, if the consequent pattern contains a fuzzy object, the fuzzy value of this object is calculated using the following basic algorithm [8].

If logical **and** is used, one has "*if* A_1 **and** A_2 *then* C" and A'_1, A'_2 are fuzzy facts which match the antecedent of this rule. The fuzzy set describing the value of the fuzzy object in the conclusion is calculated according to the formula

$$F'_c = F'_{c_1} \cup F'_{c_2}$$

where $\cup$ denotes the union of two fuzzy sets, and F'_{c_1} is a result of fuzzy inference for the fact A'_1 and the simple rule "*if* A_1 *then* C" whereas F'_{c_2} is a result of fuzzy inference for the fact A'_2 and the simple rule "*if* A_2 *then* C".

The certainty factor of the conclusion is calculated according to MYCIN's model

$$CF_c = \min(CF'_{f_1}, CF'_{f_2}) * CF_r$$

where min denotes the minimum of the two fuzzy numbers.

If logical **or** is used, the calculation is the same except that the intersection of two fuzzy sets is taken instead of union, and the maximum of the fuzzy numbers is taken rather then minimum.

The above algorithm can be applied repeatedly to handle any combination of antecedent patterns.

3.3. Global contribution

In the case where the fact containing a fuzzy object is asserted as a result of performing the RHS action of a rule, this fact is treated as giving contributing evidence towards the conclusion containing the same fuzzy object. In the current version of the system, the new value of the fuzzy object is calculated in accordance with the formula

$$F_g = F_f \cup F'_c$$

where F_g is the new value of the fuzzy object, F_f is the existing value of the fuzzy object, and F'_c is the value of the fuzzy object to be asserted.

The uncertainties are also aggregated to form an overall uncertainty using the following MYCIN-like formula

$$CF_g = CF_f + CF_c - (CF_f * CF_c)$$

where CF_g is the combined uncertainty, CF_f is the uncertainty of the existing fact, CF_c is the uncertainty of the asserted fact; $+, -$ and $*$ are the operations on fuzzy numbers.

4. CONCLUDING REMARKS

The run-time user interface uses the textual interface of CLIPS which can display the facts base, rules, and the current agenda. The Macintosh version of the interface is visualized in Figure 1.

The current version of the system does not exhibit the full power of a fuzzy expert system shell. Future work will involve implementation of some other methods of fuzzy reasoning and extension of the representation of fuzzy sets. This will move the system closer to a fuzzy meta-expert system shell. Another extension being contemplated is a window based interface, which will make the system easier to use.

REFERENCES

[1] Artificial Intelligence Section, Lindon B.Johnson Space Center. *CLIPS Reference Manual, Version 4.3.*, May 1989.

[2] L.A.Zadeh, Fuzzy Sets. *Information and Control*, 8(1965), 338-383.

[3] L.A.Zadeh, *The Concept of a Linguistic Variable and its Application to Approximate Reasoning.* (New York, 1973)

[4] Z.A.Sosnowski, A linguistic variable in FLISP programming language. The Second Joint IFSA-EC and EURO-WG Workshop 'Progress in Fuzzy Sets in Europe', April 6-8, 1988. Vienna, 71-74.

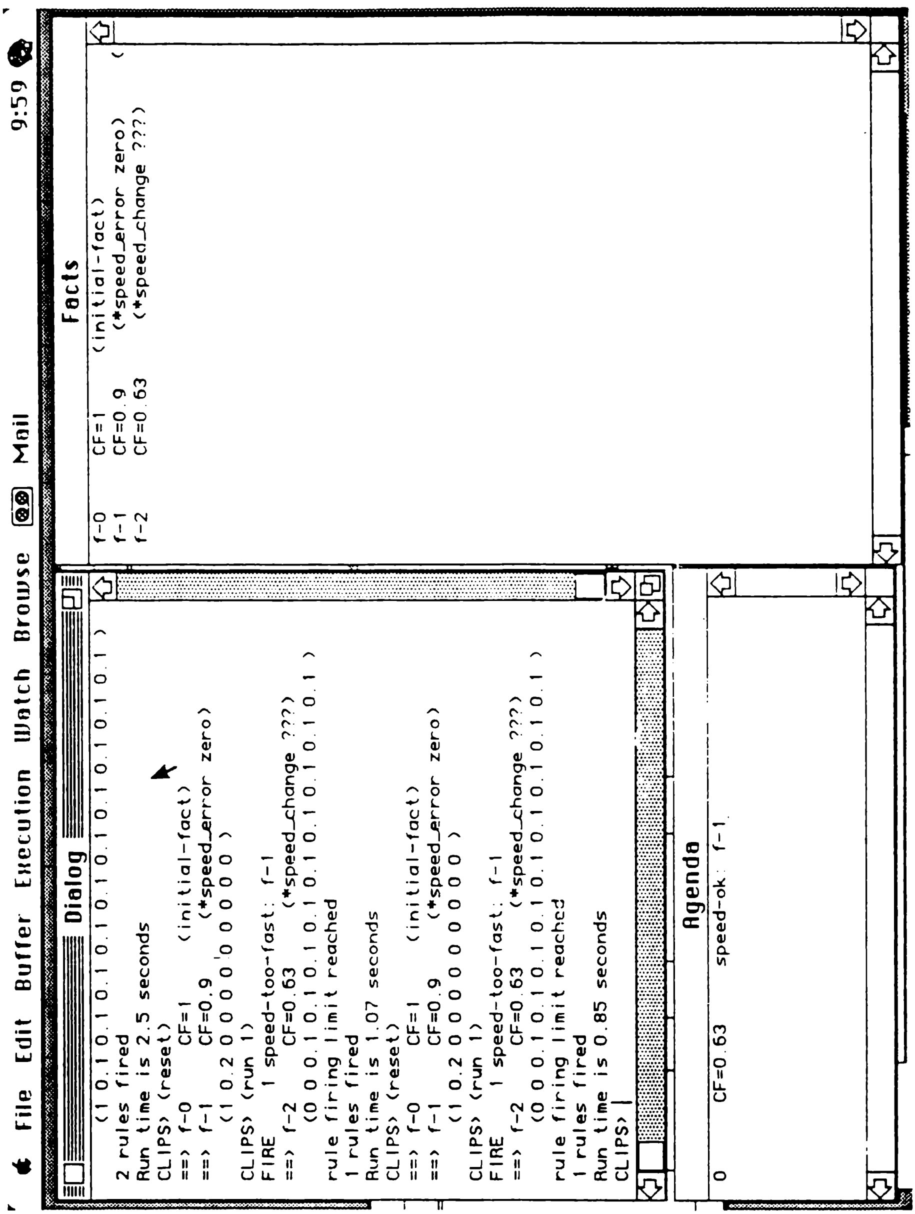

Fig.1. The Macintosh version of the interface.

[5] A.Kaufman and M.M.Gupta, *Introduction to Fuzzy Arithmetic. Theory and Applications.* (Van Nostrand Reinhold, 1985.)

[6] M.Cayrol, H.Farency and H.Prade, Fuzzy pattern matching. *Kibernetes*, **11**(1982), 103-106.

[7] M.Mizumoto, S.Fukami and K.Tanaka, Some methods of fuzzy reasoning. In: M.M.Gupta, R.K.Ragade and R.R.Yager (eds.), *Advances in Fuzzy Set Theory and Applications*, (North-Holland, Amsterdam, 1979), 117-139.

[8] T.Whalen and B.Schott, Issues in fuzzy production systems. *Int. J. Man-Machine Studies*, **19**(1983), 57-71.