A Knowledge-Based Systems Building Tool

Un outil pour la construction de systèmes à base de connaissance

Ein Shell-Programm für die Konstruktion von Expertensystemen

Silvia BRUNO
Civil Eng.
Univ. di Roma "La Sapienza"
Rome, Italy

Carlo GAVARINI
Prof. of Structural Mechanics
Univ. di Roma "La Sapienza"
Rome, Italy

Antonio PADULA
Researcher
Univ. di Roma "La Sapienza"
Rome, Italy

Federico VITTORI
Civil Eng.
Univ. di Roma "La Sapienza"
Rome, Italy

SUMMARY

The increasing interest in Artificial Intelligence as a powerful aid in solving civil enginee-ring problems has suggested the realisation of a domain-independent tool for knowledge-based systems construction. The paper describes a fuzzy inference engine, which has been developed in order to build knowledge bases and to perform evaluations. Know-ledge acquisition issues and approximate reasoning techniques are also illustrated and discussed.

RÉSUMÉ

L'intérêt croissant pour l'intelligence artificielle comme aide puissante dans la résolution des problèmes de génie civil, a suggéré le développement d'un outil bien adapté à la construction de systèmes à base de connaissances, indépendamment du domaine d'ap-plication. L'article décrit la réalisation d'un moteur inférentiel flou, qui peut être utilisé soit pour la création de bases de connaissance, soit pour obtenir des évaluations. Les pro-blèmes les plus importants dans l'acquisition des connaissances et les techniques de raisonnement flou sont aussi illustrés et discutés.

ZUSAMMENFASSUNG

Das wachsende Interesse an der Verwendung künstlicher Intelligenz bei der Lösung von Problemen im Bauwesen hat die Realisierung eines vom Wissensgebiet unabhängigen Shell-Programms für die Konstruktion von auf wissensbasierten Systemen nahegelegt. Für diese Shell wurde eine sogenannte fuzzy-Inference Maschine realisiert. Fragen der Wissensakquirierung und der verwendeten Techniken werden aufgezeigt.

1. INTRODUCTION

Nowadays engineering problems are seen in terms of decision, management and prediction; solutions are seen in terms of faster access to more information and of increased aid in analyzing, understanding and utilizing the information that is available, and in coping with the information that is not. Aiming at modelling knowledge, the methodologies and concepts of Artificial Intelligence, as embodied today in the field of knowledge-based expert system, can potentially provide tools for dealing with these two elements, large amounts of information coupled with large amounts of uncertainty, which constitute **complexity**, the ground of many major engineering problems.

As pointed out in [1], a pressing need to improve the capability to acquire, assimilate, and codify knowledge that currently exists only in the form of personal engineering experience, judgements, and heuristics has arisen. Knowledge acquisition has thus become a crucial area for insuring continued progress in the development and application of expert systems.

A prevailing enthusiastic view is perhaps based on the increasing number of successful expert systems. However, problems in the development, maintenance, and enhancement of expert systems may severely restrict their integration into operational settings. There is no doubt about the fact that many of these problems involve faulty knowledge-base development methodologies. In fact, the developmental area most often cited as the "bottleneck" in expert system development is **knowledge acquisition**, i.e. the process of extracting and translating expert-level knowledge into rules that become the heart of an expert system.

The tool that is presented in this paper is intended to aid the domain expert in introducing his expertise, in form of rules, in a computer system, thus preventing the above mentioned problems from arising.

2. KNOWLEDGE ACQUISITION

Knowledge acquisition is concurrently referred to as the most important aspect of expert system development and the most problematic. It alternately has been tagged "knowledge extraction", "knowledge elicitation", and "knowledge acquisition". It refers to the "transfer and transformation of problem-solving expertise" from a knowledge source into rules to implement in a computer program.

The major knowledge acquisition stages are:

• **conceptualization:** it involves specifying how the primary concepts and key relationships among the concepts in the domain are depicted and related by domain experts;

• **formalization:** it requires mapping the recognized concepts, subtasks, relations, into formal representation mechanisms;

• **implementation:** it involves carrying formalized knowledge into an executable computer program connected with an inference engine. The primary goal of this stage is to develop a *prototype system*, which allows developers to test out design and representation mechanism decisions using only a small subset of the complete knowledge-base rules;

• **testing:** it requires that the prototype system be evaluated as to the efficacy of the system's formalization. To enable appropriate testing, developers must investigate and select an appropriate test scenario or problem set. Once the chosen scenario has been applied to test the system, results from testing are used to revise the prototype. Common revisions may include reformulating initial concepts, refining knowledge representation schemes and interrelationships, etc.

3. A KNOWLEDGE-BASED SYSTEMS BUILDING TOOL (KSBT)

AI techniques, supported by appropriate mathemathical frameworks, has successfully been applied to solving **knowledge-intensive problems,** (i.e., complex problems for which neither algorithms nor data are explicitely known) and, among them, structural engineering problems, notably in the field of seismic engineering. Uncertainties and ambiguities involved in structural performances have usually been treated by means of probability theory. However, as complexity often arises in engineering problems, and most decisions are made with a shortage of numerical evidence and depend on informed opinion, some uncertainty which are not random in nature may play important roles in the vulnerability and risk assessment of structures. Those uncertainties might be called *subjective uncertainties*, since they can be evaluated only by an engineer's experience and judgement.

Aiming to fulfil the potential offered by AI methodologies, a shell has been developed in order to provide domain experts with the possibility to build expert systems, directly within a friendly environment. The utilization of this tool for knowledge construction is completely domain-independent. As professional judgements are often expressed as verbal statements (e.g., "the structure is *moderately damaged*" or "the quality of the masonry is *poor*"), with an intrinsic vagueness or *fuzziness* which eludes the usual ordinary set representation, both traditional reasoning methodologies and *fuzzy logic* can appropriately be adopted in the representation of the domain knowledge and in the implementation of the inferential engine. *Approximate reasoning* in particular has been recognized to offer the proper mathematical support to dealing with such descriptive words or phrases. While traditional reasoning is mainly based on the manipulation of *symbols* representing arbitrary objects in the domain and on matching techniques (*symbolic elaboration of information*), approximate reasoning can deal with the *meaning* of propositions, thus being characterized by the ability to perform a *semantic elaboration of information.*

A brief overview of the basic concepts related to the latter methodology is provided in the following section.

3.1 Fuzzy logic and approximate reasoning

A *fuzzy logic* FL, that is, a logic based on *fuzzy set theory*, may be viewed, in part, as a fuzzy extension of a nonfuzzy multi-valued logic, i.e. a logic whose truth values are represented by real numbers in the interval [0, 1] (usually the *standard Lukasiewicz logic* L_1), which constitutes a *base* logic for FL. Truth values in FL are fuzzy subsets of the unit interval with linguistic labels such as *true, false, not true, very true, quite true, more or less false*, etc. The truth value set of FL is assumed to be generated by a context-free grammar, with a semantic rule assigning each linguistic term a *meaning* represented by a fuzzy subset of [0, 1].

One of the appealing features of fuzzy logic is its ability to deal with approximate causal inferences. According to Zadeh, *approximate* or, equivalently, *fuzzy reasoning* can be informally defined as a process by which a possibly imprecise conclusion is deduced from a collection of imprecise premises. More specifically, given an inference scheme "IF *A* THEN *B*" involving fuzzy propositions expressed in natural language, it is possible from a proposition *A'* that *matches only approximately A*, to deduce a proposition *B'* *approximately similar* to *B*, through a logical interpolation called *generalized modus ponens.* Such an inference is impossible in ordinary logical systems. The definition of a *possibility distribution* provides a natural basis for the representation of the meaning of fuzzy propositions, allowing its numerical computation and quantitative treatment (*quantitative fuzzy semantics*). Retranslation of possibility distributions in natural language can be accomplished by *linguistic approximation* procedures. Systematization of the use of words or sentences in a natural language for the purpose of an approximate characterization of the values of variables and their interrelations is accomplished by the concept of a *linguistic variable* (see [13]).

3.2 Use of the generalized *modus ponens*

The concept of a generalized *modus ponens* provides the basis for approximate deductions, allowing subjective judgements, once assigned a meaning and translated into linguistic values through a linguistic approximation procedure, to be treated as linguistic variables within a formal model of fuzzy inference.

Approximate inferences are often of the form:

knowledge: IF x is A THEN y is B
fact: x is A'

---

approximate conclusion: y is B'

$(A, A' \in U; B, B' \in V; U, V$ universes of discourse).

The *fuzzy conditional proposition* "IF x is A THEN y is B" represents a certain relation between A and B. A *translating rule* translates a fuzzy conditional proposition into a fuzzy relation in $U \times V$.

The inference mechanism that has been realized is based on implication coupled with Zadeh's compositional rule of inference (*max-min composition*). Use of approximate reasoning with fuzzy logic has involved the determination of an appropriate rule for implication among those commonly occurring in literature. The axiomatic approach proposed in [2] has been taken into account.

Of the most common implication relations, the so-called *arithmetic rule*, given as

$$R_a(A, B) = (\overline{A} \times V) \oplus (U \times B)$$

$$= \int_{U \times V} (1 \wedge (1 - \mu_A(u) + \mu_B(v)))/(u,v)$$

$(u \in U; v \in V)$.

is the only one meeting certain desirable prerequisites, postulated in order to assure an intuitive understanding of the nature of fuzzy deductions . This rule has thus been widely accepted, as it has appeared to satisfy intuition in many applications.

It is noted that the arithmetic rule is based on the implication rule of Lukasiewicz logic, i.e.:

$$a \rightarrow b = 1 \wedge (1 - a + b), \quad a, b \in [0, 1].$$

Inferences of the form:

knowledge: IF x_1 is A_1 AND x_2 is A_2 AND ... AND x_n is A_n THEN y is B
fact: x_1 is A_1' AND x_2 is A_2' AND ... AND x_n is A_n'

---

approximate conclusion: y is B'

$(A_i, A_i' \in U_i, i = 1, 2, ..., n; B, B' \in V)$,

have been translated into a fuzzy relation in $U_1 \times U_2 \times ... \times U_n \times V$ by an *extended arithmetic rule*, defined as

$$R_a(A_1, A_2, ..., A_n; B) = (A_1 \cap A_2 \cap ... \cap A_n \times V) \oplus (U_1 \times U_2 \times ... \times U_n \times B)$$

$$= \int_{U_1 \times U_2 \times \ldots \times U_n \times V} (1 \wedge (1 - (\mu_{A_1}(u_1) \wedge \mu_{A_2}(u_2) \wedge \ldots \wedge \mu_{A_n}(u_n)) + \mu_B(v)))/(u_1, u_2, \ldots, u_n, v)$$

$(u_i \in U_i, i = 1, 2, \ldots, n; v \in V)$.

It can be shown (see for example [12]) that *the consequence B' is given as the union of the consequences B_i' of ordinary fuzzy reasoning* such that

knowledge: IF x is A_i THEN y is B
fact: x is A_i'

---

approximate conclusion: y is B_i' $\qquad (A_i' \circ R_a(A_i; B))$

Approximate reasoning gives therefore the possibility to cover in a satisfactory way a given domain of knowledge, by means of a relatively small number of rules (fuzzy in nature, and consequently with overlapping regions of applicability).

3.3 Description of KSBT

3.3.1 The inference engine

The shell whose implementation is in progress is intended to the construction of *rule-based fuzzy systems*, i.e. systems directly encoding structured knowledge in the numerical framework introduced in the previous part of this section. Such systems map input fuzzy sets A to output fuzzy sets B. They stores separate fuzzy rules and in parallel fires each of them to some degree for each input (Fig. 1). Outputs B_m' are first obtained as consequences of each of the fired rules, and suitable decisional criteria are then adopted in order to determine the result of the inference process, which is finally assigned a meaning through linguistic approximation.
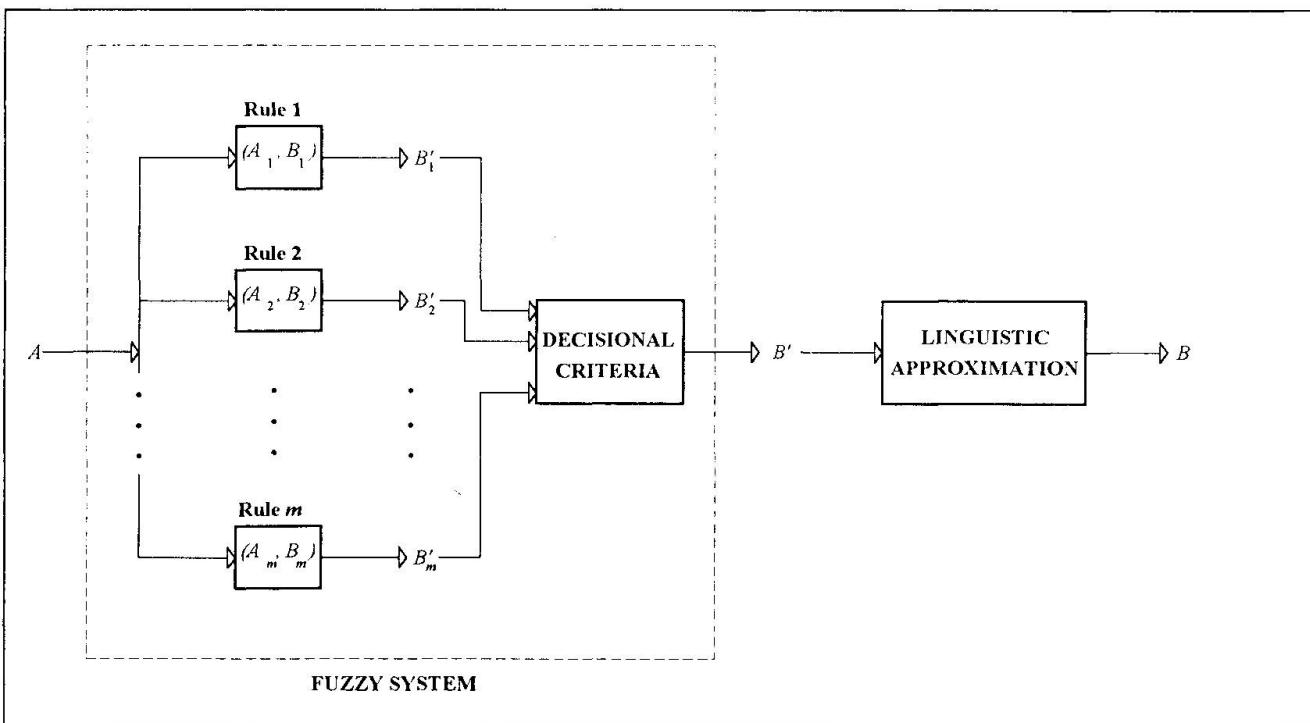

Fig. 1 Fuzzy-system architecture.
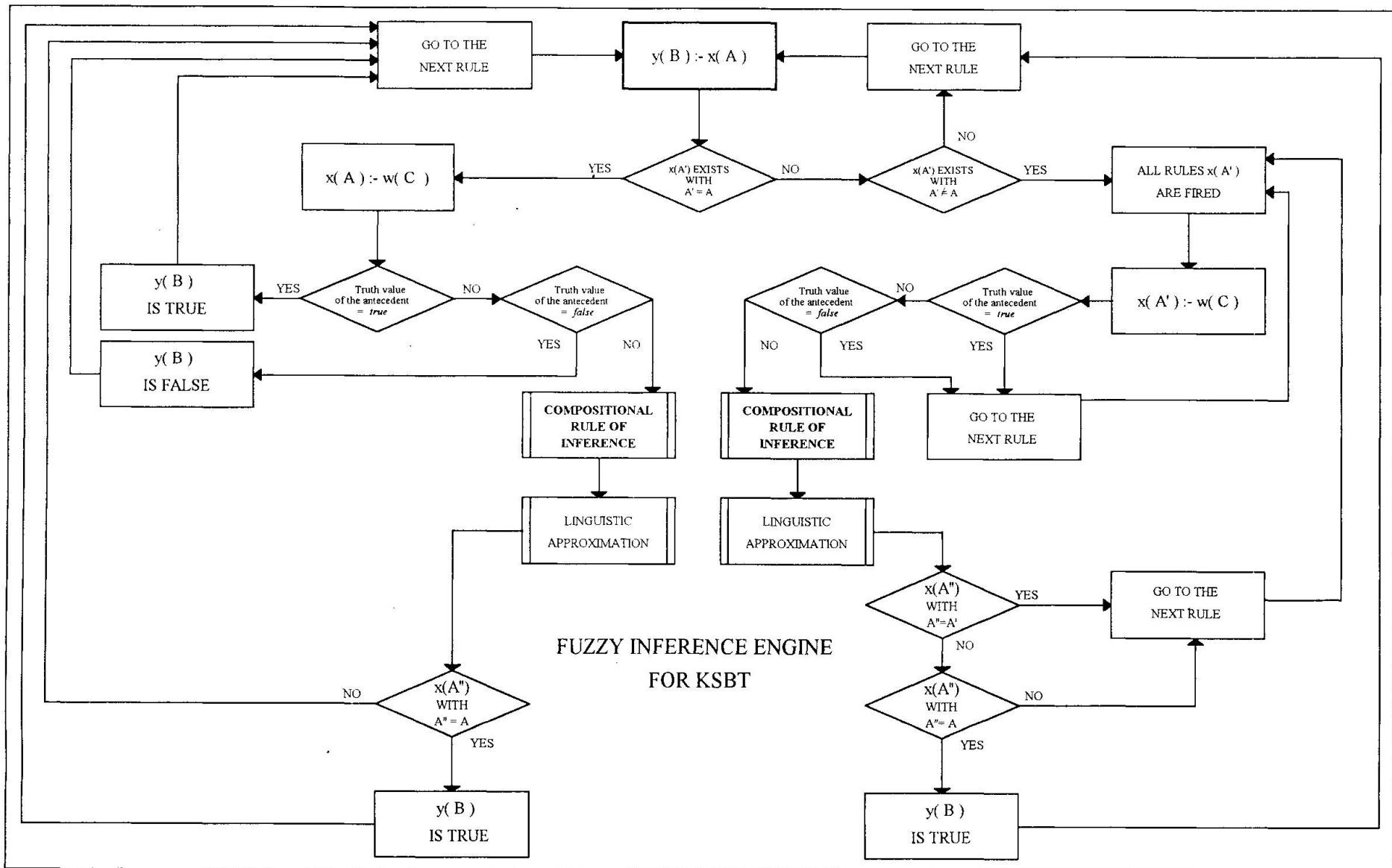
Fig. 2 Fuzzy inferential mechanism
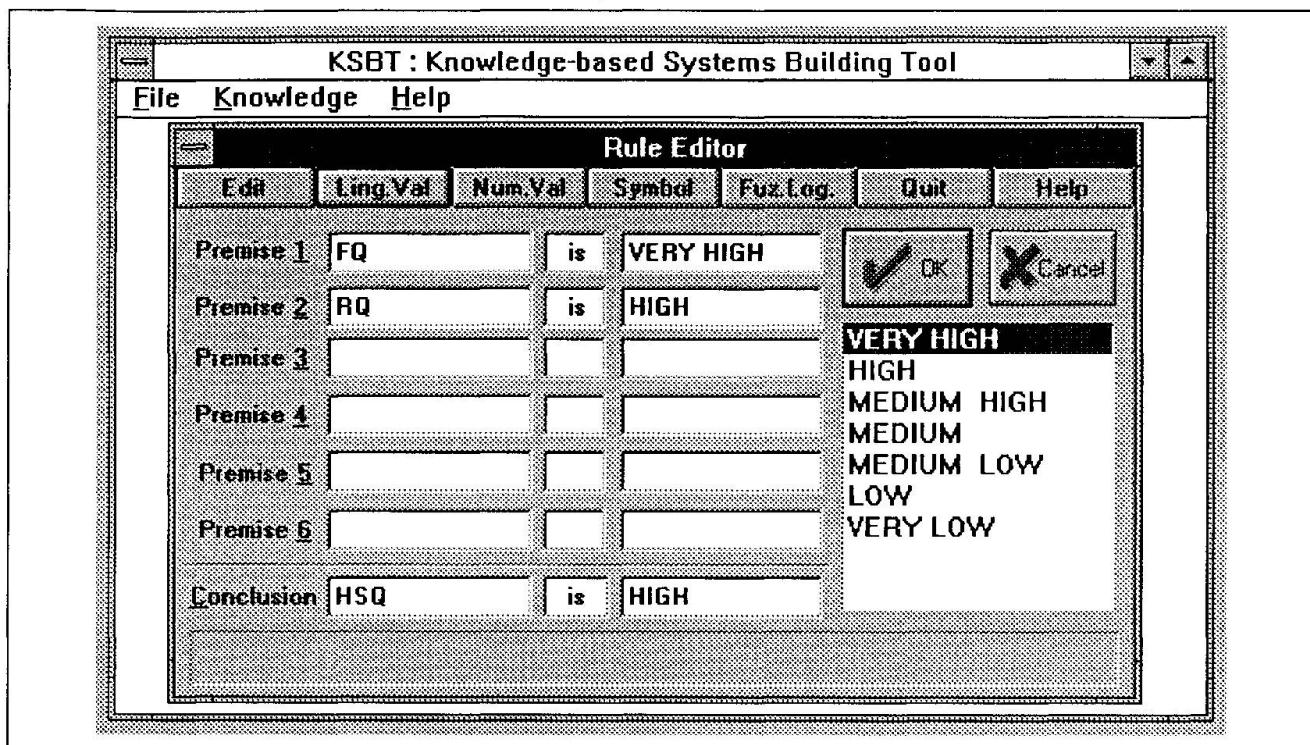
Fig.3 KSBT Rule Editor
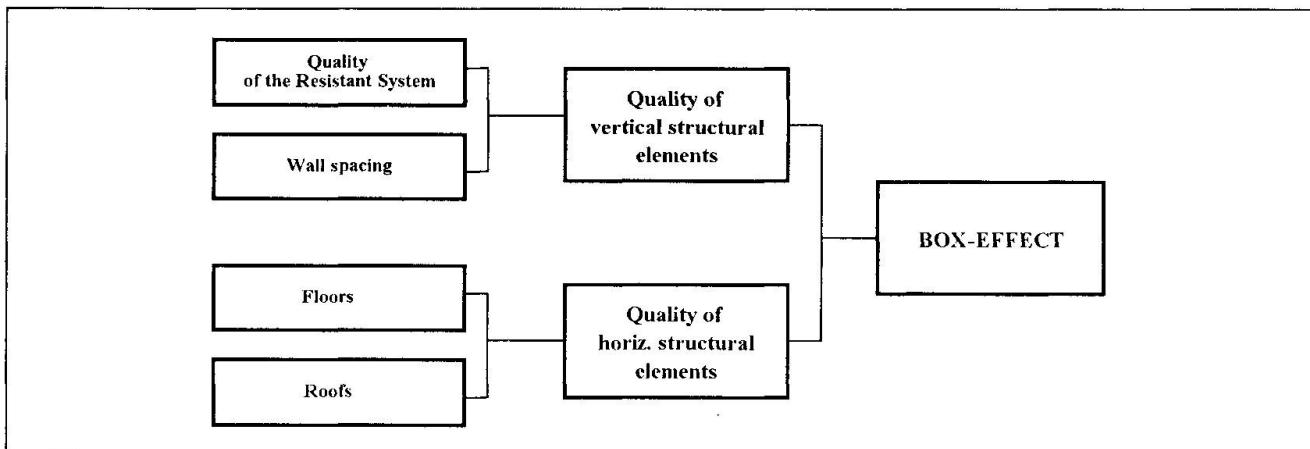
Fig.4 *Box-effect* evaluation

RQ FQ	VL	L	ML	M	MH	H	VH
VL	VL						
L	VL			L		ML	
ML	L		ML			M	
M				M	M	MH	
MH			M		MH		
H		MH		H			
VH						VH	VH

Fig.5 Linguistic matrix for HSQ assessment

Performing evaluations is accomplished by means of a *fuzzy inference engine*. The backward chaining mechanism is represented in Fig. 2. As it can be seen, symbol matching is used in case of coincidence of facts with antecedents of the rules, while the compositional rule of inference is activated in case of approximate matching between facts and antecedents. The compositional rule of inference module can also be utilized for building fuzzy knowledge bases.

3.3.2 Knowledge bases development

An environment for **writing, testing** and **using knowledge-based expert systems** has been developed, in order to assist domain experts in conveying their knowledge on a specified domain in form of rules. The following activities are supported:

- antecedent and conclusion variables identification;
- linguistic or numerical values specification;
- fuzzy or non-fuzzy rules definition and modification;
- fuzzy rule bases assisted development;
- use of knowledge bases for performing simulations and evaluations.

Both fuzzy and non-fuzzy rules are simply edited in a *MS-Windows* environment; furthermore, *fuzzy rules can be defined resorting to a tool which allows fuzzy knowledge bases construction assisted by fuzzy logic*.

Rule editing is performed through the *MS-Windows* dialog box shown in Fig. 3. Multiple antecedents composed of up to six fuzzy or non-fuzzy propositions linked by the logic connective AND can be specified. Addition and modification of fuzzy rules is at present accomplished with the use of a built-in vocabulary of terms in natural language.

The tool for supporting fuzzy knowledge base construction is intended to help the user in minimizing the number of fuzzy rules to define. It actually evidentiates the region of applicability of the rules themselves, thus preventing addition of redundant rules. Whenever the user decides to open the assisted session, the possibility distributions corresponding to the specified linguistic values are manipulated through the Lukasiewicz rule and the *max-min* composition. All possible valid inferences are subsequently obtained by activating the rules introduced in the knowledge base with input linguistic values varying within the whole predefined term set. The approximate conclusions (which, though mathematically correct being obtained by fuzzy calculus, may not represent properly the portion of the domain under consideration) are submitted to users' acceptance. Furthermore, a control module checks if new rules or results of valid inferences preserve the internal consistency of the knowledge base.

4. APPLICATIONS

Shell validation is being carried out in developing an expert system for seismic vulnerability assessment for masonry buildings. The process of fuzzy knowledge base construction can then be illustrated referring to the evaluation of the so-called "box effect", which might be identified as a sub-task of the main problem of vulnerability assessment. The box effect is defined as an estimation of the capability of a masonry building to behave as an effective earthquake-resistant structural system, with sufficiently rigid floors well connected to the vertical walls so as to prevent unresisted out-of-plane bending of the walls themselves. Variables involved in the box effect evaluation can be combined as shown in Fig. 4.

Representing this portion of knowledge of the domain of interest can be achieved following the steps listed below:

a. Let "IF the quality of floors is *very high* AND the quality of roofs is *high* THEN the quality of horizontal structures is *very high*" be the first rule to introduce in a new knowledge base. The user first formalizes it by associating identification symbols to the antecedent and consequent variables (a possible choice might be FQ, RQ, and HSQ, respectively), and selecting from a drop-down menu the linguistic values assumed by the variables within the rule itself.

b. At that stage, once the edited rule has been accepted, the user can either immediately define a second rule, or select the appropriate option for opening the fuzzy logic assisted work session. In this case, the compositional rule of inference module is activated, and the accepted rule is fired with those combinations of linguistic values for the antecedent variables leading to significant inferences. The current content of the knowledge base and the resulting inferences are displayed in two separate windows, so that the user can easily verify the correct behaviour of the system under construction when activated with possible different inputs.

c. The resulting inferences can either be accepted or rejected by the user. Resuming the example, the system proposes "IF FQ is *very high* AND RQ is *very high* THEN HSQ is *high*" as a first possible inference, obviously unacceptable for the problem at hand. The user refuses the suggestion, and is then asked to enter a new rule, with the same linguistic values for the antecedent variables FQ and RQ as in the rejected inference, and an appropriate linguistic value for the conclusion variable HSQ. The fuzzy value *unknown* has to be inserted, in case of ignorance of the value assumed by the conclusion variable, given the specified values of the antecedent variables. The new rule is then defined as follows: "IF FQ is *very high* AND RQ is *very high* THEN HSQ is *very high*"

d. The system goes back to step **b**, now considering a knowledge base made up of two rules.

e. The following possible inference is submitted to user's acceptance: "IF FQ is *very high* AND RQ is *medium high* THEN HSQ is *high*". As judged satisfactory by the user, this inference can be listed in the appropriate box.

f. The system goes back again to step **b**, still considering a knowledge base composed of two rules.

The session ends as soon as the domain of the problem at hand has sufficiently been covered by the rules in the knowledge base. A possible (not necessarily the only satisfactory one) final set of rules for HSQ asessment is represented in Fig. 5 by a 7 × 7 linguistic matrix. For what has been pointed out in the previous sections, this pattern ensures that an output (*unknown* included) can be associated to each possible pair of linguistic values for the input variables. For example, as seen above, appropriate rules are fired by the compositional rule of inference module in order to obtain the acceptable value *high* for HSQ from the fuzzy input values *very high* and *medium high* for FQ and RQ, respectively.

Writing rules with argument others than FQ and RQ for the antecedent variables, and HSQ for the consequent variable requires restarting from step **a**.

5. CONCLUSION

The main purpose of the tool presented in this paper is to provide domain experts, which usually neither are knowledge engineers nor have software developers capabilities, a friendly support for developing expert systems. Both fuzzy and non-fuzzy knowledge can be formalized and utilized, by means of an inference engine performing symbolic or semantic elaboration of information, according to the nature of the current input.

Implementation is being carried out by an appropriate utilization of C++ and *PDC-Prolog* in a *MS-Windows* environment. Further development of the tool for fuzzy rule bases construction will lead to the possibility for expert users to modify existing membership functions, to add new ones, to

adjust the inference mechanism by introducing suitable fuzzy logic operators. The implementation of an explanatory module in form of hypertext system is in progress.

Although the present stage of realization of the shell mainly focuses on the management of fuzzy information, extensions of its usability should be provided by integrating different mathematical frameworks for dealing with uncertainty, in an attempt to reproduce more efficiently the characteristics of real world situations.

REFERENCES

1. ARCISZEWSKI, T., ROSSMAN, L.A. (eds.), *Knowledge Acquisition in Civil Engineering*, ASCE, 1992

2. BALDWIN, J. F., Axiomatic Approach to Implication for Approximate Reasoning with Fuzzy Logic, *Fuzzy Sets and Systems*, **3**, 193 - 219, 1980.

3. BOOSE, J., GAINES, B. (eds.), *Knowledge Acquisition Tools for Expert Systems*, Academic Press, London, 1988.

4. BRUNO, S., C. GAVARINI, A. PADULA, Vulnerability and Damage Assessment of Structures Before and After an Earthquake: Use of Expert Systems and Fuzzy Set Theory, 1st Workshop of the European Group of Structural Engineering Applications of Artificial Intelligence (EG-SEA-AI), Lausanne, March 21-22 1994.

5. BRUNO, S., C. GAVARINI, A. PADULA, Seismic Evaluation of Buildings: a Global AI Approach, 2nd Franco-Italian Symposium of Earthquake Engineering: *Strengthening and Repair of Structures in Seismic Areas*, Nice, October 17-19, 1994.

6. DUBOIS, D., H. PRADE, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, 1980.

7. FENVES, S. J., Expert systems: expectations versus reality, IABSE Colloquium: *Expert Systems in Civil Engineering*, Bergamo, 1989.

8. GAVARINI, C., Towards the Systematic Use of Expert Systems in Seismic Risk Reduction, Proceedings of the 10th WCEE, Madrid, July 1992.

9. KLIR, G. J., T.A. FOLGER, *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall, Englewood Cliffs, N. J., 1988.

10. KOSTEM, C. N., M. L. MAHER, *Expert Systems in Civil Engineering*, ASCE, 1986.

11. MCGRAW, K.L., K. HARBISON-BRIGGS, *Knowledge acquisition: principles and guidelines*, Prentice-hall, Englewood Cliffs, N. J., 1989.

12. MIZUMOTO, M. B., Multifold Fuzzy Reasoning a Interpolative Reasoning. In: *Fuzzy Sets, Neural Networks, and Soft Computing*, edited by R. R. Yager and L. A. Zadeh, ITP, VNR, NY, pp. 188-193, 1994.

13. YAGER, R. R., S. OVCHINNIKOV, R.M. TONG, H.T. NGUYEN (eds.), *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh*, John Wiley & Sons, New York, 1987.