Extended abstracts / *Résumés longs*

THIRD INTERNATIONAL CONFERENCE
TROISIEME CONFERENCE INTERNATIONALE

IPMU

INFORMATION PROCESSING AND MANAGEMENT OF UNCERTAINTY IN KNOWLEDGE-BASED SYSTEMS

TRAITEMENT D'INFORMATION ET GESTION D'INCERTITUDES DANS LES SYSTEMES A BASE DE CONNAISSANCES

Paris - France
July 2 - 6, 1990 / *2 - 6 Juillet 1990*

sponsored by / *patronnée par*
MINISTERE DE LA RECHERCHE ET DE LA TECHNOLOGIE
and / *et*

afcet — Association Française pour l'Intelligence Artificielle — IFSA — IEEE — WOSC

with the support of / *avec le soutien de*

Bull — CAP SESA — cnet — DIRECTION DES RECHERCHES, ETUDES ET TECHNIQUES — INSERM — EDF Electricité de France — MATRA — THOMSON

Edité par l'Association des ingénieurs de techniques avancées

THE ROLE OF UNCERTAINTY AS A CONTROL FEATURE

R. LOPEZ DE MANTARAS, C. SIERRA

CEAB-CSIC
17300 Blanes - Girona

1- INTRODUCTION

Most AI research on reasoning under uncertainty is concerned with global normative methods to propagate and combine certainty values and there is a controversial debate about which methods are most appropiate and why. Disagreement between the proponents of the different methods (Bayesian's, Dempster-Shafer, Fuzzy logicians) is about the meaning of uncertainty and having a formalism that produces rational conclusions with no claim to mimic human uncerainty management methods. In restricted domains where the uncertainty involved permits, for example, a direct interpretation in the probability theory sense and where the main decision of the expert system involves the computation of such uncertainty if apropiate data are available, then probability calculus would be the best independently of how humans solve problems in the same situation. However, the most interesting aspect of Expert Systems research is to gain some insights of human problem solving strategies by trying to emulate them in programs. Although human problem solvers are almost always unceratin about the possible solution, they very often achieve their goals despite uncertainty by using methods to manage uncertainty that are particularized to the type of problem solving that they are performing at a given time. In fact, we believe that managing uncertainty consists in selecting actions that simultaneously achieve solutions and reduce their uncertainty. Since actions are selected not only for their domain effects but also for their effect on uncertainty, problem solving under uncertainty is more constrained than problem solving under total certainty as was also noticed by Cohen [2]. This view leads to consider uncertainty mainly as a control feature because it helps to constrain the focus of attention (i.e. which part of the problem to work next) and action selection(i.e. how to work on it). The problem solving strategies are implemented in the problem solver's control part. The knowledge engineers translate into control strategies the human problem solving strategies. Problem solving strategies are a fundamental component of expertise and have to be acquired by first implementing a set of task-level primitiveswith which experts can describe their strategies. Uncertainty is, in our opinion, a task-level primitive that is used at the implementation-level to descriminate alternativecontrol decisions. Furthermore, when large expert systems emulate human problem solving strategies, the organization of their complex knowledge bases makes the propagation and combination of uncertainty a local, context dependent, process. In our opinion, such large domain expert systems draw their problem solving capabilities more from the power of their organizational and problem solving structures than from the particular uncertainty management formalism they use (different formalisms can be adjusted to give similar answers).

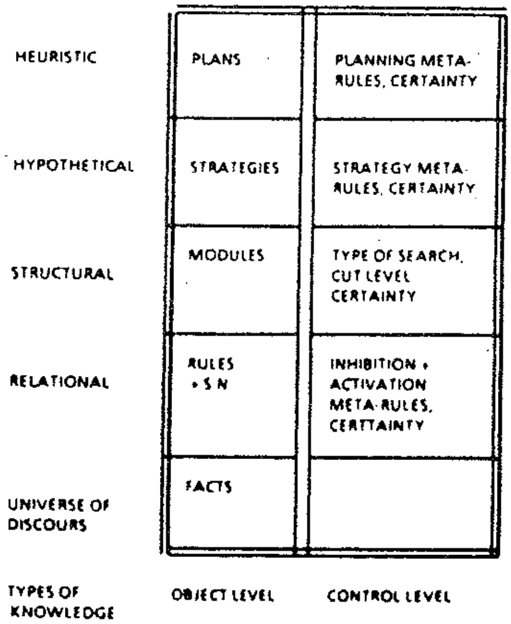
This paper discusses these ideas in the framework of MILORD [3, 4], an expert system building tool whose architecture uses uncertainty as a control feature and performs local combination and propagation of uncertainty.

2- MILORD: AN ARCHITECTURE FOR UNCERTAINTY HANDLING IN CLASSIFICATION PROBLEM SOLVING

MILORD is a classification oriented expert system building tool whose architecture attempts to mirror relevant human organizational and problem solving strategies, and to match the different types of knowledge to their representation.

2.1- Types of knowledge

In MILORD, domain and control knowledge are organized in five different levels. Each one is supported by the next lower level and modelizes a different type of knowledge. This modularity allows to modify any level independently of the remaining levels. The next figure shows this multilevel architecture:



The Universe of discours level contains the domain-dependent facts described by its type (boolean, fuzzy, numerical, subsets), the rules where the facts appear as premise or as conclusion (this is obtained by a pre-compilation process), their value (initially NIL), the associated question to the user (when the fact is non deducible), etc. At this level there is no need of control features.

The Relational level contains rules as well as a semantic network representing respectively empirical associations and static relations between facts. At this level there are two types of control: explicit (by means of meta-rules inhibiting or activating rules or sets of rules) and implicit (select first the rule with highest certainty, then if more than one is left select the most specific). The explicit control is applied first and is followed by the implicit criteria. At this level we can see that the uncertainty already plays a role as a control feature.

The Structural level contains the basic inferential objects of MILORD called modules. A module is formed by a goal, rules for achiveing that goal, metarules and at the control level we have explicit declarations concerning the type of search, the certainty cut level and the type of logic to be used locally to combine and propagate uncertainty.

The uncertainty representation is a qualitative scale that can be locally defined for each module like for example:
IMPOSSIBLE, ALMOST-IMPOSSIBLE, SLIGHTLY-POSSIBLE, QUITE-POSSIBLE, POSSIBLE, VERY-POSSIBLE, ALMOST-SURE, SURE

"Truth tables" are then locally defined for the AND, Or and "→" operations on the uncertainty scale in each module allowing local combination and propagation of uncertainty. Interfaces between modules are defined in order to communicate between modules and to combine results from different modules (see [1] for details).

In a medical application using MILORD, the knowledge is distributed among diagnostic "specialists" that are in the modules which are hierarchicaly structured. There are three types of modules: data acquisition modules, syndromic modules and diagnostic modules. The uncertainty combination operators are local to each module i.e. are hypothesis-specific (for example for one hypothesis a high degree of certainty about physical evidence for the disease in conjunction with a low certainty for historical evidence may result in high certainty for the disease, while the opposite may be true for another hypothesis).

The Hypothetical level contains the strategic knowledge, that is to say knowledge to guide the solution path towards a goal. Such knowledge is represented by the so called strategy meta-rules at this level.
A strategy is then an ordered set of modules, resulting from the application of a meta-rule, that are evaluated in order to achieve a hypothesized solution.

Example:

If the pneumonia is BACTERIAL with certainty < QUITE-POSSIBLE
and is ATYPICAL with certainty POSSIBLE

Then visit the modules: MYCO, VIRUS, CLAM, TBC NOCARDIA, CRIPTOC, P-CARINII

With certainty: QUITE-POSSIBLE

Where the certainty values of BACTERIAL and ATYPICAL have been deduced previously in the BACT-ATYP syndromic Module.

The Heuristic level contains planning knowledge to reconsider strategies of the hypothetical level. The current strategy is reconsidered as soon as new information (deduced or provided by the user) is available. Therefore, the system always works on the most plausible hypothesis. When more than one strategy is plausible, the planning meta-rules at the heuristic level are needed.

Example:

If certainty (x) $>$ certainty (y) and goals (x) $\cap$ goals (y) $\neq \emptyset$
Then Ockham (x, y)
where
Ockham (x, y) = goals (x) $\cap$ goals (y),
termination-test,
goals (x) $\cup$ goals (y) /goals (x) $\cap$ goals (y)

This means that the modules (goals) in the intersection of the strategies should be evaluated first, then a termination test is performed in order to check wether the solution is acceptable and if not the evaluation proceeds with the rest of modules of the two strategies. The termination test of the above meta-rule example is performed by examination of a set of termination meta-rules. An example of termination meta-rule is:

If P-CARINII and NO (TBC is possible) and NO (CRIPTOC possible)
Then STOP

other planning meta-rules are:

If certainty (x) $>$ certainty (y)
and goals (x) $\cap$ goals (y) $\neq \emptyset$
then Priority (x, y)

If certainty (x) $>$ certainty (y) and no-compatibles (x, y)
then First (x, y)

If certainty (x) $>$ certainty (y)
and goals (x) $\subseteq$ goals (y)
then First (x, y)

Example:

If AIDS
Then visit: P-CAR, TBC, CITOMEGA,...
with certainty VERY-POSSIBLE

If Alcoholic
Then visit: PNEUM, ENTEROBACT, LEGIONELLA, ...
with certainty POSSIBLE.

then if the patient has both AIDS and Alcoholism as risk factors, the strategy depending on AIDS, will be selected first according to the first of the three planning meta-rules above.

3- CONCLUSION

The interesting aspect of building Expert Systems is to learn something about human problem solving strategies by trying to reproduce them in programs. Human problem solver s are uncertain in many situations and do not use a simple normative method to handle uncertainty. Instead they take advantage of a good organization in the problem solving task to obtain good solutions using qualitative approximations. This suggests to consider uncertainty as playing an important role at the control level by guiding the problem solving strategies. In order to illustrate these points, we have described MILORD, an architecture that extensively exploits uncertainty as a control feature and uses local context dependent combinationand propagation uncertainty operators.

4- REFERENCES

- AGUSTÍ, J.; SIERRA, C.; SANNELLA, D. (1989) Adding generic modules to flat rule-based languages: a low cost approach. In: Methodologies for Intelligent Systems, 4 (Z. Ras, ed.) Elsevier Science Pub., pp. 43-51.

- COHEN, P.R.; GREENBERG, M.; DELISIO, J. (1987) MU:A Development Enviroment Prospective Reasoning System. COINS Technical Report 87-46.

- GODO, L.; LÓPEZ DE MÁNTARAS, R.; SIERRA, C.; VERDAGUER, A. (1987) Managing Linguistically Expressed Uncertainty in MILORD: Application to Medical Diagnosis. Artificial Intelligence Communications, Vol. 1, 1, pp. 14-31.

- GODO, L.; LÓPEZ DE MÁNTARAS, R.; SIERRA, C.; VERDAGUER, A. (1989) MILORD: The Architecture and Management of Linguistically Expresses Uncertainty . Int. Journal of Intelligent Systems. Vol. 4, n° 4, pp. 471-501.