VOLUME 6 NUMBER 3 JULY–SEPTEMBER 1994

JOURNAL OF EXPERIMENTAL & THEORETICAL ARTIFICIAL INTELLIGENCE

TAYLOR & FRANCIS, LONDON AND WASHINGTON, DC

JOURNAL OF EXPERIMENTAL & THEORETICAL ARTIFICIAL INTELLIGENCE 6(3) 1994

ISSN 0952–813X(1994)

JETAI

JOURNAL OF EXPERIMENTAL & THEORETICAL ARTIFICIAL INTELLIGENCE

Volume 6 Number 3
July–September 1994

Editor Eric Dietrich

Wolff, J. G. (1982) Language acquisition, data compression and generalization. *Language and Communication*, 2(1): 57–89.
Wolff, J. G. (1988) Learning syntax and meanings through opitimization and distributional analysis. In Y. Levy, I. M. Schlessinger and M. D. S. Braine (eds), *Categories and Processes in Language Acquistion* (New York: Lawrence Erlbaum), pp. 179–215.
Wolff, J. G. (1991) *Towards a Theory of Cognition and Computing* (Chichester: Ellis Horwood).

J. EXPT. THEOR. ARTIF. INTELL. 6(1994)303–321.

Local multi-valued logics in modular expert systems

J. AGUSTÍ, F. ESTEVA, P. GARCIA, L. GODO,
R. LOPEZ DE MANTARAS and C. SIERRA

Institut d'Investigació en Intel.ligència Artificial (IIIA),
Centre d'Estudis Avançats de Blanes (CSIC), 17300 BLANES,
Girona, Spain
tel. +34 72 33 61 01
email: {agusti,esteva,pere,godo,mantaras,sierra}@ceab.es

Abstract. In this paper we describe an approach to the problem of dealing with uncertainty by means of finite multi-valued logics in modular expert systems, and the results obtained. The modularity of the systems allows us to address two main characteristics of human problem-solving: the adaptation of general knowledge to particular problems and the dependency of the management of uncertainty on the different subtasks being implemented in the modules of the system, i.e. different modules can have different local multiple-valued logics as part of their local deductive mechanisms. Although the results obtained are general, we use, throughout the paper, examples of a medical expert system that has been designed using a modular language called MILORD-II, that implements them showing the practical interest of the theoretical concepts involved.

Keywords: uncertain reasoning, multi-valued logics, modular expert systems, modular languages

Received 27 October 1992; revised 23 March 1993

1. Introduction

This paper describes an approach to dealing with uncertainty in modular reasoning systems. Modularity allows the association of different local multi-valued logics for uncertainty management to different modules modelling different subtasks of the whole problem-solving task being tackled. The paper starts by describing the local logics approach to uncertainty management in modular systems and raises the question of the communication problem between different local logics when using different modules to solve a problem.

Section 3 describes the requirements that have to be met to ensure the correctness of such communication in terms of consistency preservation.

Address for correspondence: Ramon Lopez de Mantaras, IIIA, Centre d'Estudis Avançats de Blanes (CSIC), 17300 Blanes, Girona, Spain.

0952-813X/94 $10.00

Sections 4 and 5 describe the concrete realization of the general approach to the modular language MILORD-II, that we have implemented, although we want to make clear that the features of this language can be commonly encountered in other languages; therefore the most important contributions of this paper are the theoretical results concerning the local multi-valued logics and not the particular implementation. However, the implementation is useful because it allows one to exemplify the theory, and also shows its applicability.

Section 6 describes the role that uncertainty can play as a control feature in modular systems, and finally we give some conclusions.

2. Local logics: uncertainty management and communication in modular systems

The use of modularization techniques in expert system design (Agustí *et al.* 1989) is due to the need to adequate the general and spread knowledge in a knowledge base (KB)to specific subtasks. Specific subtasks generally make use only of a subset of the whole KB. For instance, the suspicion of a bacterial disease will rule out all knowledge referring to viral diseases, or a patient in coma will render useless all the knowledge units that need patient's answers. Moreover, this adaptation determines the universe of discourse, and this is made by means of selecting certain units of the KB which should be of a variable granularity depending on the problem. However, in any case, the level of granularity will never be as fine as reducing the universe of discourse to an elementary KB object (a rule) but a set of them. These considerations lead to the definition of structured KBs.

As an example in MILORD-II (Agustí *et al.* 1992) the basic units of KB are modules. These modules may be hierarchically organized, and consist of an encapsulated set of import, export, rule, meta-rule, inference system and submodules declarations. The declarations of submodules do not differ from the declaration of modules. These declarations of submodules inside a module are what structures the hierarchy, which reflects the information dependencies among modules. The meaning of the primitive components of a module is:

Import: the list of non-deducible facts needed in the module to apply the rules. These facts are to be obtained from the user at run time.

Export: the list of facts deduced or imported inside a module that are visible from the rest of the modules that include the module as a submodule.

Rules: deductive units that relate import and export components within a module.

Meta-rules: the meta-logical component of a module.

Inference system: defines (i) the set of linguistic certainty terms used in weighting facts, rules and meta-rules, (ii) a renaming mapping between the sets of linguistic terms expressing uncertainty values in the submodules and the set of terms of their parent module, and (iii) the 'and' connective operator used to combine and propagate the linguistic terms when making inference (see Section 4). A more complete description can be found in Augstí *et al.* (1992). Figure 1 shows an example of module definitions using our implemented language.

We use linguistic terms to express uncertainty because psychological experiments (Kuipers *et al.* 1988) show that human problem-solvers do not use numbers to deal with uncertainty. Furthermore, the way they manage uncertainty is situation-dependent. Modular languages such as MILORD-II, allow us to define, in a natural way, local uncertainty calculi attached to each module (see Figure 1), in


```
Module Global_Gram =
    Begin
        Module D = Respiratory_diagnosis
        Module T = Type_of_infection
        Module R = Radiology_diagnosis
        Module X = Sputum
        Export Pneumococcus, Haemophilus
        Deductive knowledge
            Rules:
            R001 If X/DCGP and D/Bacterial then
                    conclude Pneumococcus is very_possible
            R002 If X/DCGP and D/Bacterial and T/Common_acquired then
                    conclude Pneumococcus is quite_possible
            R003 If X/CBGN and D/BCRO then
                    conclude Haemophilus is very_possible
            R004 If X/CBGN and D/BCRO and R/Previous then
                    conclude Haemophilus is quite_possible
            R005 If X/BGN and D/BCRO then conclude BGN is little_possible
            R006 if R/Cavities then Pneumococcus is possible
            Inference system:
                Truth values = (impossible little_possible slightly_possible
                                possible quite-possible very_possible sure)
                Renaming = ; see figure 2
                Conjunction = T
        End deductive
    End

Module Sputum =
    Begin
        Import Class_sputum
        Export DCGP, CBGN, BGN
        Deductive knowledge
            Rules:
            R001 If Class_Sputum = DCGP then conclude DCGP is
            very_possible
            . . .
            Inference system:
                Truth values = (impossible little_possible slightly_possible
                                possible quite-possible very_possible sure)
                Conjunction = T
        End deductive
    End
```

Figure 1. Example of module and submodule definitions.

such a way that the knowledge adequation process can also be applied to the uncertainty management. The need of having different uncertainty calculi in a KB becomes clearer when expert systems involving several human experts have to be built.

When two or more uncertainty calculi coexist in a same expert system, another question arises: how they communicate each other. Let us consider the following example where two experts cooperate in solving a problem.

A physician diagnosing a pneumonia could ask a radiologist about the results of a radiological analysis. The simplest and more frequent type of communication is to get an 'atomic' answer like

'It is *likely* that the patient has cavities in his left lung.'

Then, to use this information in his own reasoning, the physician must only interpret the linguistic expression *likely* used by the radiologist, and perhaps identify it with another term, for example *acceptable*, used by himself. But the communication could have been richer than that 'atomic' answer, and consist of a more complex piece of information. For instance, the radiologist could have answered:

'If from a clinical point of view you are *very confident* that the patient has a bacterial disease and he is also immunodepressed, then it's *nearly sure* he has cavities in his left lung.'

As in the previous answer, to use the radiologist information, the physician must again 'interpret' it. However, this time the translation cannot be only a matter of the uncertainty terms (*very confident*, *nearly sure*) being used, but also a matter of the way of reasoning, if he wants to make use of this information in other situations (patients) which do not exactly match the one described above.

From the point of view of an expert system shell, the first type of communication only requires an order preserving renaming between the linguistic values of the uncertainty calculi attached to different tasks, while the second type of communication requires that the translation should preserve the consistency between the facts deduced in different modules. Figure 2 shows an example of the first type of communication. We have a module called 'Global'_Gram' whose local logic uses the truth values: *impossible*, *little_possible*, *slightly_possible*, *possible*, *quite_possible*, *very_possible and sure*. This module has a submodule called 'Radiology_diagnosis' whose local logic uses: *false*, *unlikely*, *maybe*, *likely* and *true*. Then, in order to correctly interpret in the module the certainty values of the predicate 'cavities' deduced in the submodule, an order preserving renaming of the certainty values is needed as shown in figure 2 (the justification of this renaming is given in Section 3).

More generally we have that:

(1) Every expert, or subtask, is modelled as a different module with a local specific logic.

(2) The communication between tasks and subtasks is modelled as communication between modules.

Looking carefully at how experts communicate their knowledge, and at their problem-solving procedures, we can find complex communication mechanisms. Sometimes experts cannot reduce their interaction only to the communication of certainty values of predicates. A deeper analysis shows that when communicating, experts in medical diagnosis also need:

(a) *To condition their answers*

Module Radiology_diagnosis =
  Begin
    Module Res = Respiratory_diagnosis
    Import immunodepressed
    Export cavities, immunodepressed
    Deductive knowledge =
      Rules:
      . . .
      R0014 **if** Res/Bacterial **and** immunodepressed **then** cavities **is** likely
      . . .
      Inference system:
        Truth values = (false unlikely maybe likely true)
        Conjunction = T_{Rad}
    End deductive
  end

Module Global_Gram =
  Begin
    Module D = Respiratory_diagnosis
    Module T = Type_of_infection
    Module R = Radiology_diagnosis
    Module X = Sputum
    Export Pneumococcus, Haemophilus
    Deductive knowledge
      Rules:
      R001 . . . R005
      R006 if R/Cavities **then** Pneumococcus **is** possible
      Inference system:
        Truth values = (impossible little_possible slightly_possible possible quite-possible very_possible sure)
        Renaming =
          R/false ⇒ impossible
          R/unlikely ⇒ [impossible, possible]
          R/maybe ⇒ possible
          R/likely ⇒ [possible, sure]
          R/true ⇒ sure
        Conjunction = T_{Gram}
    End deductive
  End

Figure 2. Example of a renaming mapping between two modules.

Suppose that it is not known if a patient is allergic to penicillin. A module deducing the possibility of giving penicillin can answer: *Penicillin is a good treatment from a clinical point of view if there is no allergy to it.* From a logical point of view, the answer could be:

{if no(allergy_penicillin) then penicillin is very_possible}

(b) *To give conclusions that have to be considered with the answer*
If in a culture of sputum *Pneumococcus* have been isolated, then it is strongly suggested that an antibiogram to the patient is made, i.e.:

{*Pneumococcus*_isolation is sure, make_antibiogram is definite}

(c) *To give conditioned conclusions to be considered with the answer*
A treatment with ciprofloxacine is not recommended for breast-feeding women. However, if a woman is on a breast-feeding period then the treatment can be carried out if the woman stops breast-feeding, i.e.:

{Cipro is very_possible, if breast-feeding then stop_breast-feeding is definite}

(d) *To give a more general answer*
Imagine that Gram-positive *coccus* are detected. An answer to the predicate *Pneumococcus* is at that moment too precise and cannot be given, but at least the morphological classification can be answered, i.e.: {*coccus* is definite}

To model such communication protocols we need to extend the ES answering procedure. What we need is to answer a given question with a set of formulas (rules and facts). To do so, the rules considered are those in deductive paths to and from the question. The facts in the answer are those that have been obtained in the application of such rules. The rules in the answer are those which could not be applied because they used unknown information. We only consider the rules in the deduction tree of the question because we assume that when a user asks a question he/she expects an answer containing only the relevant information associated with that question. Such a richer communication scenario is being implemented by means of partial evaluation techniques (Puyol *et al.* 1992).

3. Mappings between local logics

In this section we present a general formal framework to cope with complex communication situations. Within this framework we give requirements so that such communication is consistent in the sense that for every fact F deducible in a module M, its corresponding fact F' through a mapping in another module M' will lead only to other logically consistent facts with respect to F, as we will see below. However, in the rest of the paper we focus our attention on the simpler, but fundamental, case of communication that involves predicates and certainty values.

In general the deductive systems associated to local logics can be formalized as entailment relations as follows:

Let M and M' be two modules and $(L, \vdash)$ and $(L', \vdash')$ their corresponding logics, L and L' standing for the languages and $\vdash$ and $\vdash'$ for entailment relations defined on L and L', respectively. Then, the correspondence between module M and module M' can be formalized by a mapping H: $L \rightarrow L'$ relating their languages. In this general setting we propose that at least one of the following three requirements should be fulfilled by the mapping H in order to assure a consistent communication between modules

RQ.1. *If* $\Gamma \vdash e$, *then* $H(\Gamma) \vdash' H(e)$

where Γ and e denote a set of formulas and a formula of L, respectively.

With this requirement we assure that for every formula, e, deducible from a set of formulas Γ in M, its corresponding formula $H(e)$ in M' by the mapping H will also be deducible in M' from the corresponding formulas of $H(\Gamma)$. In other words, there is no inferential power lost when translating from M to M' through a mapping H satisfying RQ-1. Nevertheless the main drawback of requirement RQ.2 is that it does not forbid deductions from $H(\Gamma)$, in M', formulas that are not translations of any formula deducible from Γ in M. The property means that, in the case of modules representing different experts, an expert E' related to M', using knowledge coming from an expert E related to M, will be able to deduce the same facts as E, but not only those facts. The second requirement is

RQ.2. *If* $H(\Gamma) \vdash' H(e)$, *then* $\Gamma \vdash e$

This is the inverse requirement of RQ.1. So, in this case all deductions in M' involving only translated formulas from M are translations of deductions in M, or equivalently if a fact is not deducible in M, then its corresponding fact in M' will not be deducible from the translated knowledge.

These requirements might be too strong, so we have also studied another possibility (Agustí *et al.* 1991b), to translate knowledge between modules, by means of the so-called 'weak conservative maps', which allow the reasoning in a module M' to be less accurate when dealing with knowledge translated from another module M, but not incorrect, in any case, with respect to the result that would be obtained in M and then translated to M' (i.e. deducing facts in M', using knowledge translated from M, with lower certainty than deducing first in M and then translating their certainty to the logic of M'). Formally this can be expressed by the third and last requirement:

RQ.3. *If* $H(\Gamma) \vdash' e'$, *then there exists* e *such that* $\Gamma \vdash e$ *and* $H(e) \vdash' e'$.

This requirement assures that every formula deducible from $H(\Gamma)$ in M' must be in agreement with what can be deduced from Γ in M. This requirement is slightly different from RQ.2, in the sense that it is not necessary that e' be exactly a translation of a deducible formula e from Γ, but only something deducible from such a translation.

In some logical frameworks for uncertainty management, such as the multi-valued approach explained in the following section, e' can be interpreted as a 'weaker' form of e, i.e. e' can be more uncertain than e.

4. A multi-valued logic approach to uncertainty

The uncertainty management system that we consider has the following characteristics:

(1) The expert defines a set of linguistic terms expressing uncertainty that will be the verbal scale he will use to weight facts and rules.

(2) The set of linguistic terms is supposed to be at least partially ordered according to the amount of uncertainty they express, the Booleans 'true' and 'false' always being their maximum and minimum elements, respectively.

(3) The combination and propagation of uncertainty is performed by operators defined over the set of linguistic terms, basically conjunction, disjunction, negation and detachment operators. A method for the elicitation of these operators from the expert has been proposed in Lopez de Mantaras *et al.* (1990). The main difference of this approach with respect to previous ones is that no underlying

numerical representation of the linguistic terms is required. Linguistic terms are treated as mere labels. As has been already pointed out, the only *a-priori* requirement is that these labels should represent an ordered set of expressions about uncertainty. For each logical connective a set of desirable properties of the corresponding operator is listed. Nevertheless, many of these properties are a finite counterpart of those of the uncertainty calculi based on t-norms and t-conorms, which are in turn the basis of the usual [0,1]-valued systems underlying fuzzy sets theory. The listed properties act as constraints on the set of possible solutions. In this way all operators fulfilling them are generated. Finally, the expert may select the one he thinks fits better his own way of uncertainty management in the current task. This approach has been implemented by formulating it as a constraint satisfaction problem.

These three characteristics make clear that the logics associated to our uncertainty management system are a class of finite multiple-valued logics (Rescher 1969), taking the linguistic terms as truth-values and the operators as the interpretations of the logical connectives. In other words, each linguistic term set together with its set of operators defines a truth-values algebra and thus a corresponding multiple-valued logic. For the particular case of our language, the multiple-valued logics that we use to manage uncertainty are defined by:

An *algebra of truth-values*: a finite algebra $A = \langle A_n, 0, 1, N, T, I\rangle$ such that:

(1) The set of truth-values A_n is a chain represented by

$$\mathbf{0} = a_0 < a_1 < \ldots < a_{n-1} = \mathbf{1}$$

where **0** and **1** are the Booleans *false* and *true*, respectively, and they are the common minimum and maximum elements of any truth-value algebra. Usually the set of truth-values A_n stands for a totally ordered set of linguistic terms that the expert uses to express uncertainty, but the approach would be similar if the terms were only partially ordered.

(2) The negation operator N is a unary operation such that the following properties hold:

$N1$: if $a < b$ then $N(a) > N(b)$
$N2$: $N^2 = \mathrm{Id}$.

The only negation operator satisfying $N1$ and $N2$ is defined by $N(a_i) = a_{n-1-i}$

(3) The 'and' operation T is any binary operation such that the following properties hold:

$T1$: $T(a,b) = T(b,a)$
$T2$: $T(a,T(b,c)) = T(T(a,b),c)$
$T3$: $T(0,a) = 0$
$T4$: $T(1,a) = a$
$T5$: if $a \leq b$ then $T(a,c) \leq T(b,c)$ for all c.

Note that in the unit interval [0, 1] these properties define t-norms functions if we add the condition of continuity.

(4) The implication operator is defined by residuation with respect to T, i.e.

$$I(a,b) = \mathrm{Max}\{c \in A_n \text{ such that } T(a,c) \leq b\}$$

i.e. I is the finite counterpart of a so-called R-implication generated by a t-norm (Trillas and Valverde 1985).

A set of *connectives*: not($\neg$), and ($\wedge$), implication($\rightarrow$)

A set of *sentences*: sentences are pairs of classical-like propositional sentences and subsets of truth-values. The classical-like propositional sentences are built from a set of atomic symbols and the above set of connectives. However, the sentences used in our language are only of the following types:

$$(p, V)$$
$$(p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow q, V)$$

where p, p_1, ..., p_n are literals (an atom or the negation of an atom), q is an atom, and V is a subset of truth-values. For each truth-value algebra A, L_A will stand for the set of sentences with intervals of truth-values belonging to A.

The models: defined by valuations, i.e. mappings ρ from the firsts components of sentences to A_n provided that:

$$\rho(\neg p) = N(\rho(p))$$
$$\rho(p_1 \wedge p_2) = T(\rho(p_1), \rho(p_2))$$
$$\rho(p \rightarrow q) = I(\rho(p), \rho(q))$$

The satisfaction relation between models and sentences is defined by:

$$M_\rho \models (p, V) \text{ if, and only if } \rho(p) \in V,$$

where M_ρ stands for the model defined by a valuation ρ.

For each truth-values algebra A, the corresponding *entailment system* $(L_A, \vdash_A)$ is given by

(1) the following set of axioms:

(A.1) $(p, [0,1])$
(A.2) $(\neg p \rightarrow p, \mathbf{1})$

(2) and the following inference rules, which can be easily checked to be sound with respect to the satisfaction relation

(RI.1) WEAKENING: $(p, V) \vdash (p, V')$, where $V \subseteq V'$
(RI.2) NOT-introduction: $(p, V) \vdash (\neg p, N(V))$
(RI.3) COMPOSITION: $\{(p, V_1), (p, V_2)\} \vdash (p, V_1 \cap V_2)$
(RI.4) MODUS PONENS:
$\{(p_1, V_1), \ldots, (p_n, V_n), (p_1 \wedge \ldots \wedge p_n \rightarrow q, W)\} \vdash (q, MP(T(V_1,\ldots,V_n)\, W))$

In RI.4, the expression $T(V_1, \ldots, V_n)$ stands for the recurrent application of T as $T(V_1, T(V_2, \ldots, T(V_{n-1}, V_n) \ldots)$ and letting $T(V) = V$. Moreover, it should

be noticed that in the expressions of the inference rules, N and T are taken as the pointwise extensions of the negation and conjunction operators of the above truth-value algebra.

Furthermore, the modus ponens operator MP is defined through the pointwise extension of the function:

$$MP(a,b) = \begin{cases} \varnothing & \text{if } a \text{ and } b \text{ are inconsistent} \\ [a, 1] & \text{if } b = 1 \\ T(a,b) & \text{otherwise} \end{cases}$$

which gives the set of solutions of the equation $I(a,y) = b$, where a and $b \in A_n$ are said to be inconsistent if there is no solution to this equation.

Note that these inference rules seem to be the minimal ones needed when working on sentences of the above specified types, very common in rule-based ES. However, instead of the rule RI.4, the inference system used in the next section to analyse correspondences between local logics, adopts the following inference rule:

(RI.4′) MODUS PONENS:

$$\{(p_1, V_1), \ldots, (p_n, V_n), (p_1 \wedge \ldots \wedge p_n \rightarrow q, W)\} \vdash (q, T(V_1, \ldots, V_n, W))$$

i.e. we use a conjunction operator as modus ponens operator.

Notice that, although this is correct, for instance in the common case of upper intervals of truth values, from a logical point of view this inference rule is not sound in general with respect to the above defined semantics. Nevertheless, it is well known that from the cognitive point of view, detachment operators share the same properties as those required of conjunction operators (Bonissone 1987). These arguments, together with self-evident computational reasons, have lead us to implement the inference rule RI.4′ instead of RI.4. Therefore, from now on, given a truth-value algebra A we will denote by $(L_A, \vdash)$ the local logic whose language is L_A, and its entailment relation is the minimal one determined by the axioms $A.1$ and $A.2$, and inference rules RI.1, RI.2, RI.3 and RI.4′.

For this reason, and from the deductive point of view, only the ordered set of truth-values (linguistic terms) and the conjunction operator should be specified in the local logics declaration of a module. In the case of a not-totally-ordered set of truth-values, the negation operator should also be specified.

Although our multiple-valued logic approach allows an expert to represent uncertainty by assigning subsets of truth values to rules and facts, in most cases this expressive power is not necessary from the point of view of experts. However, this expressive power is fully exploited in the definition of mappings between local logics, as will be shown in the next section. The intervals assigned to rules and facts are represented by pairs of linguistic terms. For example [*moderately possible*, *very possible*]. In most cases experts use degenerated intervals, for example [*very possible*, *very possible*] which can be represented by a single truth, i.e. *very possible*. This is the case with the examples used throughout the paper.

5. Relating different multi-valued local logics

As has already been noted, to establish communication between modules, it is necessary to consider which kind of relation between their corresponding uncertainty logics is required. On the other hand, the different uncertainty calculus we assign to each module is an entailment system determined by axioms A.1 and A.2 and by inference rules RI.1, RI.2, RI.3 and RI.4′ (see previous section), which depends only on the particular truth-value algebra associated to each module. In this section we will focus on the analysis of the conditions which have to be imposed on the module renaming functions (in the local logic declarations) in order to satisfy the above mentioned requirements to map different entailment systems.

Remember that a truth-value algebra $A = \langle A_n, 0, 1, N, T, I \rangle$ is composed of an ordered set of linguistic terms expressing uncertainty, together with a negation, a conjunction and an implication operator. However, since the implication operator is not needed explicitly (is implicit in the modus ponens) in the formulation of the above mentioned inference rules, we will consider truth-value algebras only with negation and conjunction operators.

Let $A = \langle A_n, 0, 1, N, T \rangle$ and $A' = \langle A_m, 0, 1, N', T' \rangle$ be two truth-value algebras. Let $(L_A, \vdash)$ and $(L_{A'}, \vdash)$ be their corresponding local logics. We are interested in mapping the entailment system $(L_A, \vdash)$ into the entailment system $(L_{A'}, \vdash)$ by means of module renaming functions between the corresponding linguistic term sets. This means that we will consider only those mappings translating sentences from L_A to $L_{A'}$ that just involve translations of truth-values, i.e. any mapping G: $L_A \rightarrow L_{A'}$ will be defined as $G((e, V)) = (e, g(V))$, where g translates subsets of values of A_n into subsets of values of A_m, i.e. g is a mapping between the power sets $\mathcal{P}(A_n)$ and $\mathcal{P}(A_n)$. However, if the natural condition $g(V) = \cup\{g(v),\ v \in V\}$ is required, then it is sufficient to have mappings g from A_n to $\mathcal{P}(A_m)$. In order to map the Boolean values into themselves, such a mapping g must fulfil $g(0) = \{0\}$ and $g(1) = \{1\}$. Moreover, it is a natural requirement that $g(A_n) \subseteq \mathcal{I}(A_m)$, where $\mathcal{I}(A_m) = \{[a,b] \mid a \leq b$ and $a,b \in A_n\}$, being $[a,b] = \{x \mid x \in A_n, a \leq x \leq b\}$, the set of intervals of A_m, that is, the translation of a value of A_n is not any subset of A_m, but an interval. It is worth noticing that any truth value algebra $\langle A_n, 0, 1, N, T \rangle$ can be extended to an algebra $\langle \mathcal{I}(A_n), 0, 1, N^*, T^* \rangle$ of the same type by defining the following ordering and operations on $\mathcal{I}(A_n)$:

$$\mathcal{I}_1 \leq^* \mathcal{I}_2 \text{ iff}_{\text{DEF}} \text{ for all } a \in \mathcal{I}_1 \text{ and for all } b \in \mathcal{I}_2, \text{ then } a \leq b$$

$$N^*([a,b]) = [N(b), N(a)]$$

$$T^*([a_1,b_1], [a_2,b_2]) = [T(a_1,a_2), T(b_1,b_2)]$$

Notice that $N^*(I)$ and $T^*(I_1,I_2)$ are the minimum intervals containing the pointwise image of N and T. Notice also that N^* is a negation operator fulfilling properties $N1$ and $N2$ (see Section 4) and T^* is almost an 'and' operation because it is not-decreasing and it fulfils properties $T1$ to $T4$. Furthermore, if we identify every element a of A_n with the interval $[a,a]$, then $\langle A_n, 0, 1, N, T \rangle$ is a subalgebra of $\langle \mathcal{I}(A_n), 0, 1, N^*, T^* \rangle$. Finally, it is interesting to remark that $(\mathcal{I}(A_n), \leq^*)$ is only a partial ordered set whose ordering is different from the set inclusion, and that N^* and T^* are univocally defined by N and T. (See Esteva *et al.* 1992 for a detailed study on the use of intervals of truth values for the management of uncertainty and imprecision.)

Let us use a simple example:

Let $A = \{0 < a < b < 1\}$ be a chain of four elements. Then the set of intervals

of A is $\mathcal{I}(A) = \{\emptyset, [0,a], [0,b], [0,1], [a,b], [a,1], [b,1], [0,0], [a,a], [b,b], [1,1]\}$. The order relations on A and $\mathcal{I}(A)$ can be represented by the graphs of Figure 3.

For the rest of this section, given an order preserving mapping h: $A_n \to \mathcal{I}(A_m)$ such that $h(0) = 0$ and $h(1) = 1$, we will denote by H the mapping H: $L_A \to L_{A'}$ defined by $H((e,V)) = (e, h(V))$, using the same notation for h and its extension from $\mathcal{P}(A_n)$ to $\mathcal{P}(A_m)$ defined by $h(V) = \cup\{h(v) \mid v \in V\}$. Observe that if f: $A_m \to A_n$ is an on to order preserving mapping, then it generates an order preserving mapping hf: $A_n \to \mathcal{I}(A_m)$ defined by $hf(a) = f^{-1}(a)$.

Taking into account that deductions are performed by applying a finite sequence of inference rules RI.1, . . ., RI.4', it is easy to observe that an entailment such as

$$\{(p_1, V_1), \ldots, (p_n, V_n)\} \vdash (e, V)$$

holds if, and only if, there exists $G = (G_1, G_2)$ such that $e = G_1(p_1, \ldots, p_n)$ and $V \supseteq G_2(V_1, \ldots, V_n)$, where G is a term representing the transformation by inference rules RI.2, RI.3 and RI.4 on the sentences in the premise. Thus, G_2 involves only intersections and N and T operators. For instance, the deduction

$$\{(p, V_1), (\neg r, V_2), (p \wedge r \to q, V_3), (r, V_4)\} \vdash (q, V_5)$$

has associated with it the following term G:

$$G[(p, V_1), (\neg r, V_2), (p \wedge r \to q, V_3), (r, V_4)]$$
$$= \text{RI.4}[(p, V_1), \text{RI.3}[(r, V_4), \text{RI.2}[(\neg r, V_2)]], (p \wedge r \to q, V_3)]$$

and thus

$$G_1[p, \neg r, p \wedge r \to q, r] = q$$

and

$$G_2[V_1, V_2, V_3, V_4] = T(T(V_1, V_4 \cup N(V_2)), V_3)$$

Using this observation, the following theorems give (necessary and/or sufficient) conditions for a mapping H between two local logics to satisfy the requirements

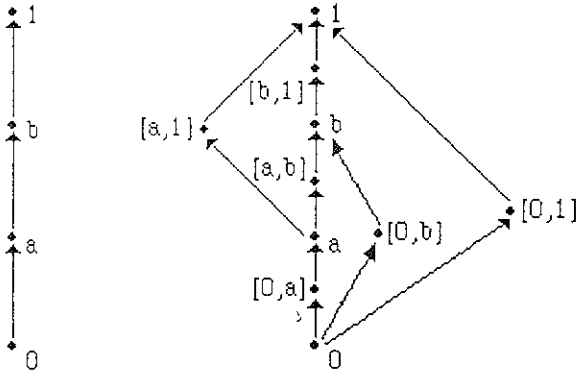

Figure 3. Graphs of A and $\mathcal{I}(A)$.

RQ.1, RQ.2 and RQ.3, proposed in section 3. These requirements were proposed in order to assure that the mapping are in some sense inference preserving. In the following, to simplify notation, we write $\vdash$ and $\vdash'$ instead of $\vdash_A$ and $\vdash_{A'}$.

Lemma: Given a mapping h: $A_n \to \mathcal{I}(A_m)$, if $h(N(V))$ is included in (or, includes) $N'(h(V))$, for every $V \subseteq A_n$, then the equality $h(N(V)) = N'(h(V))$ holds.

Proof. Suppose $h(N(V)) \subseteq N'(h(V))$ (*). Since N and N' are involutions, $N'(h(N(V))) \subseteq h(V)$, and applying (*) $N'(h(N(V))) \supseteq h(N(N(V))) = h(V)$. Therefore, $N'(h(N(V))) = h(V)$, and thus $h(N(V)) = N'(h(V))$ and the lemma is proved.

Theorem 1. Given an order preserving mapping h: $A_n \to \mathcal{I}(A_m)$, the mapping H: $L_A \to L_{A'}$ defined by $H((e,V)) = (e, h(V))$ satisfies the requirement RQ.1 if, and only if, the mapping h fulfils the following conditions:

(1) $h(T(V_1, V_2)) \supseteq T'(h(V_1), h(V_2))$
(2) $h(N(V)) = N'(h(V))$
(3) $h(V_1 \cap V_2) = h(V_1) \cap h(V_2)$

Proof. Suppose first that H satisfies the requirements RQ.1, that is, if $\{(p_1, V_1), \ldots, (p_n, V_n)\} \vdash (e, V)$ then $(p_1, h(V_1)), \ldots, (p_n, h(V_n)) \vdash' (e, h(V))$. Then, in particular we have $\{(p_1, V_1), (p_1 \to p_2, V_2)\} \vdash (p_2, T(V_1, V_2))$, $(p, V) \vdash (\neg p, N(V))$, and $\{(p, V_1), (p, V_2)\} \vdash (p, V_1 \cap V_2)$. Thus, RQ.1 implies that $\{(p_1, h(V_1)), (p_1 \to p_2, h(V_2))\} \vdash' (p_2, h(T(V_1, V_2)))$, $(p, h(V)) \vdash' (\neg p, h(N(V)))$, and $\{(p, h(V_1)), (p, h(V_2))\} \vdash' (p, h(V_1 \cap V_2))$ also hold. By definition of $\vdash'$, these entailment relations hold if and only if $h(T(V_1, V_2)) \supseteq T'(h(V_1), h(V_2))$, $h(N(V)) \supseteq N'(h(V))$, and $h(V_1 \cap V_2) \supseteq h(V_1) \cap h(V_2)$, respectively. Therefore, theorem conditions 1, 2 and 3 are satisfied.

Now, suppose conditions 1, 2 and 3 hold. Let $\Gamma = \{(p_1, V_1), \ldots, (p_n, V_n)\}$ and $E = (e,V)$ be a finite set of sentences and a sentence, respectively, such that $\Gamma \vdash E$. We can assume that there exists $G = (G_1, G_2)$ such that $e = G_1[p_1, \ldots, p_n]$ and $V \supseteq G_2[V_1, \ldots, V_n]$. Since G_2 involves only intersections, T's and N's, using conditions 1, 2 and 3, we have that

$$h(V) \supseteq h(G_2[V_1, \ldots, V_n]) \supseteq G'_2[h(V_1), \ldots, h(V_n)]$$

where G'_2 is the term obtained from G_2 replacing T and N by T' and N', respectively. Therefore, there exists $G' = (G_1, G'_2)$ such that $G_1[p_1, \ldots, p_n] = 3$ and $h(V) \supseteq G'_2[h(V_1), \ldots, h(V_n)]$, that is, $H(\Gamma) \vdash' H(E)$. Thus the theorem is proved.

Corollary 1. If $h(a) \in A_m$ for every $a \in A_n$, then the mapping H satisfies the requirement RQ.1 if, and only if, the mapping h: $A_n \to A_m$ is a monomorphism with respect to the negation and conjunction operators.

Corollary 2. Let $h(a) = f^{-1}(a)$ where f: $A_m \to A_n$ is an exhaustive order-preserving mapping. Then the mapping H satisfies the requirement RQ.1 provided that the mapping f is a morphism with respect to the negation and conjunction operators.

Proof. Since f is a morphism we have that $f(T'(a,b)) = T(f(a), f(b))$ and $f(N'(a)) = N(f(a))$ for every $a,b \in A_m$. This, together with the fact that the inclusion $f^{-1} \circ f \supseteq \mathrm{Id}$. always hold, leads us to the inclusions $T'(a,b) \subseteq f^{-1}(T(f(a), f(b)))$ and $N'(a) \subseteq f^{-1}(N(f(a)))$. Then, in particular we have that $T'(f^{-1}(a), f^{-1}(b)) \subseteq f^{-1}(T(a, b))$ and $N'(f^{-1}(b)) \subseteq f^{-1}(N(b))$. Finally, since $f^{-1}(A \cup B) \supseteq f^{-1}(A) \cup f^{-1}(B)$, we have that for any V_1, V_2 and $V \subseteq A_n$, $T'(f^{-1}(V_1), f^{-1}(V_2)) \subseteq f^{-1}(T(V_1, V_2))$ and $N'(f^{-1}(V)) \subseteq f^{-1}(N(V))$, and thus, by the previous lemma we have also $N'(f^{-1}(V)) = f^{-1}(N(V))$. Therefore conditions 1 and 2 of Theorem 1 are satisfied. Straightforward computation shows that condition 3 is also satisfied, i.e. $f^{-1}(V_1 \cap V_2) = f^{-1}(V_2) \cap f^{-1}(V_2)$.

Theorem 2. The mapping H satisfies the requirement RQ.2 provided that the mapping h fulfils the following conditions:

(1) $h(T(V_1, V_2)) \subseteq T'(h(V_1), h(V_2))$
(2) $h(N(V)) = N'(h(V))$
(3) $h(V_1) \supseteq h(V_2)$ implies $V_1 \supseteq V_2$

Proof. Suppose that conditions 1, 2 and 3 hold. Let $\Gamma = \{(p_1, V_1), \ldots, (p_n, V_n)\}$ and $E = (e,V)$ be a finite set of sentences and a sentence of L_A, respectively, such that $H(\Gamma) \vdash' H(E)$. We can assume that there exists $G' = (G'_1, G'_2)$ such that $e = G'_1(p_1, \ldots, p'n)$ and $h(V) \supseteq G'_2[h(V_1), \ldots, h(V_n)]$. Observe that condition 3 implies $h(V_1 \cap V_2) = h(V_1) \cap h(V_2)$. Therefore, since G'_2 only has operations T' and N' and intersections, using conditions 1, 2 and 3, we have

$$h(V) \supseteq G'_2[h(V_1), \ldots, h(V_n)] \supseteq h(G_2[V_1, \ldots, V_n])$$

where G_2 is the term obtained from G'_2 replacing T' and N' by T and N, respectively. Since $h(V_1) \supseteq h(V_2)$ implies $V_1 \supseteq V_2$, then we have $V \supseteq G_2[V_1, \ldots, V'n]$. Therefore there exists $G = (G'_1, G_2)$ such that $G'_1[p_1, \ldots, p_n] = e$ and $V \supseteq G_2[V_1, \ldots, V'n]$, or equivalently $\Gamma \vdash E$. Thus the theorem is proved.

Corollary 3. If $h(a) \in A_m$ for every $a \in A_n$, then the mapping H satisfies the requirement RQ.2 provided that the mapping h: $A_n \to A_m$ is a monomorphism with respect to the negation and conjunction operators.

In section 3, the requirement RQ.3 stated that if $H(\Gamma) \vdash' E'$, then there should exist a sentence E such that $\Gamma \vdash E$ and $H(E) \vdash' E'$ Interpreting the sentence E' as a weaker form of $H(E)$, that is, E' can only be deduced from $H(E)$ by applying the weakening inference rule RI.1, the following theorem holds.

Theorem 3. The mapping H satisfies the requirement RQ.3 if. and only if, the mapping h fulfils the following conditions:

(1) $h(T(V_1, V_2)) \subseteq T'(h(V_1), h(V_2))$
(2) $h(N(V)) = N'(h(V))$

Proof. Let $\Gamma = \{(p_1, V_1), \ldots, (p_n, V_n)\}$ be a finite set of sentences of L_A and $E' = (e', V')$ a sentence of $L_{A'}$, such that $H(\Gamma) \vdash' E'$. Then we can assume that there exists $G' = (G'_1, G'_2)$ such that $e' = G'_1(p_1, \ldots, p_n)$ and $V' \supseteq G'_2[h(V_1),$

$\ldots, h(V_n)]$. Again, since G'_2 involves only intersections and operations T' and N', using conditions 1 and 2, we have

$$V' \supseteq G'_2[h(V_1), \ldots, h(V_n)]) \supseteq h(G_2[V_1, \ldots, V_n])$$

where G_2 is the term obtained from G'_2, replacing T' and N' by T and N, respectively. Here, notice that $h(V) \cap h(W) \supseteq h(V \cap W)$ always holds. Consider now the sentence $E = (e, G_2[V_1, \ldots, Vn])$ of L_A. Then it is clear that $\Gamma \vdash E$ and $H(E) \vdash' E'$ as RQ.3 requires.

Let us suppose now that RQ.3 is satisfied, that is, if $H(\Gamma) \vdash' E'$ then there exists E such that $\Gamma \vdash E$ and $H(E) \vdash' E'$. Take $\Gamma = \{(p, V)\}$ and $E' = (\neg p, N'(h(V)))$. It is clear that $H(\Gamma) \vdash' E'$. On the other hand, since only RI.1 can be applied to deduce E' from $H(E)$, E must be of the form $E = (\neg p, W)$. Then $H(E) = (\neg p, h(W))$, and since $H(E) \vdash' E'$ it must be the case that $h(W) \subseteq N'(h(V))$. On the other hand, if $\Gamma \vdash E$, then $N(V) \subseteq W$, and so $h(N(V)) \subseteq h(W) \subseteq N'(h(V))$, and condition 2 is satisfied. Now take $\Gamma = \{(p_1, V_1), (p_1 \to p_2, V_2)\}$ and $E' = (p_2, T'(h(V_1), h(V_2)))$. It is clear that $H(\Gamma) \vdash' E'$. Again, since only RI.1 can be applied to deduce E' from $H(E)$, E must be of the form $E = (p_2, V)$. Then $H(E) = (p_2, h(V))$, and since $H(E) \vdash' E'$ it must be the case that $h(V) \subseteq T'(h(V_1), h(V_2))$. But $\Gamma \vdash E$ implies that $V \supseteq T(V_1, V_2)$, and so $T'(h(V_1), h(V_2)) \supseteq h(V) \supseteq h(T(V_1, V_2))$. Therefore condition 1 is satisfied, and the theorem is proved.

Corollary 4. If $h(a) \in A_m$ for every $a = A_n$, then the mapping H satisfies the requirement RQ.3 if, and only if, the mapping h is a morphism with respect to the negation and conjunction operators.

From these results it is clear that if the mapping h is a morphism from A_n to A_m then requirement RQ.3 is satisfied, and if h is a monomorphism then the requirements RQ.2 and RQ.3 are also satisfied.

As an example, let us consider the renaming function from module *Radiology_diagnosis* to module *Global_Gram* given in Figure 2. If a_0, a_1, a_2, a_3 and a_4 stand for *false*, *unlikely*, *may_be*, *likely* and *true*, respectively, and b_0, b_1, b_2, b_3, b_4, b_5, and b_6 stand for *impossible*, *little_possible*, *slightly_possible*, *possible*, *quite_possible*, *very_possible*, and *sure*, respectively, the conjunction operators T and T' are given by the matrices of Figure 4.

Let us consider the truth-value algebras $A = \langle\{a_0, a_1, a_2, a_3, a_4\}, a_0, a_4, N, T\rangle$ and $A' = \langle\{b_0, b_1, b_2, b_3, b_4, b_5, b_6\}, b_0, b_6, N', T'\rangle$ which are the algebras of truth-values of the local logics of modules *Radiology_diagnosis* and *Global_Gram*, respectively. It can be checked that there is no morphism from A to A'. However, the renaming function given in the module *Global_Gram* declaration, that is, the mapping h defined by:

$$h(a_0) = b_0$$
$$h(a_1) = [b_0, b_3]$$
$$h(a_2) = b_3$$
$$h(a_3) = [b_3\, b_6]$$
$$h(a_4) = b_6$$

T	a_0	a_1	a_2	a_3	a_4
a_0	a_0	a_0	a_0	a_0	a_0
a_1	a_0	a_0	a_1	a_1	a_1
a_2	a_0	a_1	a_2	a_2	a_2
a_3	a_0	a_1	a_2	a_2	a_3
a_4	a_0	a_1	a_2	a_3	a_4

T'	b_0	b_1	b_2	b_3	b_4	b_5	b_6
b_0	b_0	b_0	b_0	b_0	b_0	b_0	b_0
b_1	b_0	b_1	b_1	b_1	b_1	b_1	b_1
b_2	b_0	b_1	b_2	b_2	b_2	b_2	b_2
b_3	b_0	b_1	b_2	b_3	b_3	b_3	b_3
b_4	b_0	b_1	b_2	b_3	b_4	b_4	b_4
b_5	b_0	b_1	b_2	b_3	b_4	b_5	b_5
b_6	b_0	b_1	b_2	b_3	b_4	b_5	b_6

Figure 4. Conjunction tables.

fulfils the conditions required in Theorem 3, and thus the requirement RQ.3 for the communication between those modules is satisfied.

6. Uncertainty as a control feature

Finally, let us describe how uncertainty is used in the system as a control feature. The system has some introspection capabilities that allow to control the problem solving process. This control is based on the uncertainty degree that predicates have obtained so far, and it is used to determine the focus of attention, the set of modules to be used (knowledge adaptation) and the problem-solving strategy (Lopez de Mántaras *et al.* 1992).

In MILORD-II, the introspection capabilities are represented by meta-rules that control the applicability of submodules and rules. That is, it determines which rules and submodules are useful for the current case. This general mechanism is used in different ways. We are going now to see the most important ones.

6.1. *Evidence increasing*

The current uncertainty of facts can be used to control the deduction steps in order to increase the evidence of a given hypothesis. So, for example, if we have an alcoholic patient showing a cavity in the chest X-ray and there is low evidence for tuberculosis, then the Ziehl–Nielssen test to determine more clearly whether he has a tuberculosis should not be done. But if he also presents a risk factor for AIDS then we shall increase our evidence for tuberculosis and the test will be suggested. This is expressed as follows:

If tuberculosis > moderately_possible **then conclude** Test-Ziehl–Neelsen

If risk_factor_for_AIDS **then conclude** tuberculosis **is** possible

If Alcoholic **and** Cavities **then** tuberculosis **is** almost_impossible

It should be noticed that the first rule is a rule of the meta-logic component of the language whilst the others are rules at the object-logic level.

6.2. *Strategy focusing*

The uncertainty of facts can determine the set of hypothesis to be followed in the sequel. Example:

If the pneumonia is bacterial with certainty < quite_possible **and**
the pneumonia is atypical with certainty > possible
Then focus on
Mycoplasm, Virus, Chlamydia, Tuberculosis, Nocardia, Cryptococcus, Pneumocystis carinii
with certainty quite-possible

This example means that the modules to be used in order to find a solution to the current case are those indicated in the conclusion of the meta-rule and should be considered in the order specified there.

Strategies have a certainty degree attached to them. This is useful to differentiate the strategies generated by very specific data from those generated by general data. As an example consider the case of a patient with AIDS (a kind of immunodepression). If we know that the patient suffers from AIDS, a more specific strategy (and also more certain) can be generated. But if we know only that the patient has an immunodepression, a less certain general strategy would be generated. Since we may have several candidate strategies simultaneously, combining different strategies is a matter of great importance in the control of the system. This is also achieved by looking at the uncertainty of the strategies, as shows the next example:

If Strategy (X) **and** Strategy (Y) **and** Certainty (X) > Certainty (Y) **and**
Goals (X) ∩ Goals (Y) ≠ ∅
Then Ockham (X,Y)

where Ockham (X,Y) is a combination of the strategies that gives priority to those modules found in the intersection of both strategies (Goals (X) ∩ Goals (Y)), and then resumes with the goals of strategy X and finally those of strategy Y.

6.3. *Knowledge adaptation*

As indicated at the beginning of the paper a KB is a set of knowledge units that have to be adapted to the current case. For example alcoholism is a useful concept when determining a bacterial pneumonia, but it is useless for non-bacterial diseases. Then, a possible use of the uncertainty of the fact bacterianicity is to decide about the use of a given concept in the whole KB, i.e. to adequate the general knowledge to the particular problem, example:

If no bacterial disease
Then do not use alcoholism in finding the solution

6.4. *Solution acceptance*

The degree of uncertainty of a fact can also be used to stop the execution of the system; for example:

If pneumocystis carinii **and** tuberculosis < possible **and** cryptococcus < possible
Then stop

The control tasks we have discussed use uncertainty as a control parameter and are tasks of the meta-logic level. They are represented as a local meta-logic component of each module in what is called the control knowledge component of a module.

7. Conclusions

We have presented a new approach to deal with uncertainty by means of finite multiple-valued logic in modular reasoning systems. The modularity allows one to associate different local uncertainty calculi to modules performing different subtasks. The communication between modules with different uncertainty calculi has been analysed and a solution to the problem of inference-preserving translation of certainty values is given. More complex forms of communication not limited to certainty values as explained in Section 2, raise problems that are presently being studied. This approach has been successfully tested in the design of MILORD-II, a modular language for building expert systems that has been used as a running example throughout this paper.

We have also described the role that uncertainty plays as a meta-logic control feature guiding the problem-solving process.

Furthermore, given the fact that the system consists of a hierarchy of submodules, the meta-logical components act one upon the other, in a pyramidal fashion. This allows us to have as many meta-logic levels as necessary in an application. Further research will be pursued along this line. A richer representation of the logic components in the meta-level will also be investigated and sound semantics, from the logic point of view, will be defined.

Acknowledgements

This research was partially supported by the ESPRIT-III Basic Research Action DRUMS-II (Project 6156), by the CICYT TIC91-0430 project TESEU and by the CICYT TIC92-05769 project ARREL.

References

Agustí, J., Esteva, F., García, P. and Godo, L. (1991a) Formalizing multiple-valued logics as institutions. In B. Bouchon-Meunier, R. R. Yager and L. A. Zadeh (eds) *Uncertainty in Knowledge Bases*. Lecture Notes in Computer Science 521 (Springer-Verlag) pp. 269–278.

Agustí, J., Esteva, F., García, P., Godo, L., López de Mántaras, R., Murgui, L., Puyol, J. and Sierra, C. (1992) Structured local fuzzy logics in Milord. In L. A. Zadeh and J. Kacprzyk (eds) *Fuzzy Logic for the Management of Uncertainty* (New York: John Wiley) pp. 523–551.

Agustí, J., Esteva, F., García, P., Godo, L. and Sierra, C. (1991b) Combining multiple-valued logics in modular expert systems. In Bruce d-Ambrosio *et al.* (eds) *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence* (San Mateo, California: Hogan Kaufman) pp. 17–25.

Agustí, J., Sierra, C. and Sannella, D. (1989) Adding generic modules to flat rule-based languages: a low cost approach. In Zbigniew W. Ras (ed.) *Methodologies for Intelligent Systems* (Amsterdam: North Holland) pp. 43–52.

Bonissone, P. P. (1987) Summarizing and propagating uncertain information by triangular norms. *International Journal of Approximate Reasoning*, **1**(1): 71–101.

Esteva, F., García, P. and Godo, L. (1992) Enriched interval bilattices: an approach to deal with uncertainty and imprecision. Short version in *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems. IPMU'92*. Mallorca, July 1992, pp. 279–283. Full version to be published in *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, **1–2**.

Godo, L. (1990) Contribució a l'estudi de models d'inferència en els sistemes possibilistics. PhD. Thesis, Universidad Politécnica de Catalunya.

Godo, L., López de Mántaras, R., Sierra, C. and Verdaguer, A. (1989) MILORD, the architecture and management of linguistically expressed uncertainty. *International Journal of Intelligent Systems*, **4**: 471–501.

Kuipers, B., Moskowitz, A. J. and Kassirer, J. P. (1988) Critical decisions under uncertainty: representation and structure. *Cognitive Science*, **12**: 177–210.

López de Mántaras, R., Godo, L. and Sanguesa, R. (1990) Connective operators elicitation for linguistic term sets. In *Proceedings of the International Conference on Fuzzy Logic and Neural Networks*. Iizuka, Japan, July 20–24, 1990, pp. 729–733.

López de Mántaras, R., Sierra, C. and Agustí, J. (1992) The representation and use of uncertainty and metaknowledge in MILORD. In R. R. Yager and L. A. Zadeh (eds) *An Introduction to Fuzzy Logic Applications in Intelligent Systems* (Boston: Kluwer), pp. 253–263.

Puyol, J., Godo, L. and Sierra, C. (1992) A specialisation Calculus to improve Expert Systems Communication. In B. Neumann (ed.) *Proceedings of the 10th European Conference on Artificial Intelligence, ECAI'92*. Vienna, Austria, August 1992, pp. 144–148.

Rescher, N. (1969). *Many-valued Logic*. (New York: McGraw-Hill).

Sierra, C. (1989) MILORD: Arquitectura multi-nivell per a sistemes experts en classificació. Ph.D. Thesis, Universitat Politécnica de Catalunya.

Trillas, E. and Valverde, L. (1985) On implication and indistinguishability in the setting of fuzzy logic. In J. Kaprzyk and R. R. Yager (eds) *Management Decision Support Systems using Fuzzy Sets and Possibility Theory*. Interdisciplinary Systems Research, **83**, (Köln: Verlag TÜV Rheinland) pp. 198–212.