ELSEVIER Artificial Intelligence in Medicine 21 (2001) 153–162

Artificial Intelligence in Medicine

www.elsevier.com/locate/artmed

Renoir, *Pneumon-IA* and *Terap-IA*: three medical applications based on fuzzy logic

Lluís Godo*, Ramon López de Mántaras, Josep Puyol-Gruart, Carles Sierra

Artificial Intelligence Research Institute (IIIA), Spanish Scientific Research Council (CSIC), Campus UAB, 08193 Bellaterra, Spain

Received 15 February 2000; received in revised form 26 June 2000; accepted 1 August 2000

Abstract

The research at the IIIA has produced over more than a decade two versions of a tool for developing knowledge-based systems: *Milord* and Milord II. This tool has been mainly used for the development of medical applications. In this paper we summarize the Milord II approximate reasoning approach based on fuzzy sets, and three medical applications: rheumatology diagnosis (*Renoir*), pneumonia diagnosis (*Pneumon-IA*) and pneumonia treatment (*Terap-IA*). © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Approximate reasoning; Fuzzy logic; Medical applications; Knowledge-based systems

1. Introduction

In this paper we describe the programming environment Milord II, from the point of view of its fuzzy set based approximate reasoning capabilities. Milord II's reasoning mechanisms are an extension of those provided by its predecessor Milord [1]. In [2–4] the interested reader can find a complete description of the language and its logical semantics. To illustrate how useful these mechanisms have proved to be we briefly describe three large medical applications. *Renoir* is an expert system aimed to aid the nonspecialist physician to diagnose rheumatic diseases. *Renoir* knowledge base spans over the 37 major diagnoses in collagen diseases and inflammatory arthropathies and 15 diagnostic variants. *Pneumon-IA* and *Terap-IA* are expert systems devoted to the diagnosis and treatment of community-acquired pneumonia. *Pneumon-IA* covers over 22 etiological agents, and

* Corresponding author. URL: http://www.iiia.csic.es.
E-mail addresses: godo@iiia.csic.es (L. Godo), mantaras@iiia.csic.es (R.L. de Mántaras), puyol@iiia.csic.es (J. Puyol-Gruart), sierra@iiia.csic.es (C. Sierra).

PII: S0933-3657(00)00080-4

Terap-IA suggests treatments based on over 35 antibiotics. Community-acquired pneumonia are frequent infections, especially for people with chronic diseases and old people. It is one of the most common causes of mortality related to infectious diseases (the first in USA).

2. Milord II

Milord II is a modular language for knowledge-based systems. The structural construct of Milord II is the *module*. A program consists of a set of modules that can recursively contain other modules, then forming a hierarchy.

The approximate reasoning capability of Milord II is based on attaching to each module a particular logic from a family of finitely-valued fuzzy logics. Each logic is defined by an algebra of truth-values. This allows to assign a degree of truth within a module to each one of its propositions. These graded assignments are used to model the inherent incompleteness of data and knowledge.

The many-valued logic attached to a module is completely determined by fixing (1) an ordered set of linguistic terms representing truth degrees and (2) a *conjunction* operator defined over the linguistic terms. Hence, the programmer may generate different multiple-valued logics by simply varying these components.

In *Renoir*, for instance, several modules share these linguistic terms: *impossible*, *almost impossible*, *slightly possible*, *moderately possible*, *possible*, *quite possible*, *very possible*, *almost definite* and *definite*, where *impossible* stands for the boolean *false* and *definite* for the boolean *true*. An example of conjunction is $T(a_i, a_j) = \min(a_i, a_j)$.

2.1. Propositions and variables

Propositions and variables are the simplest knowledge representation units in Milord II. They are structures that represent the concepts dealt within a module. Their declaration is made by binding an atomic name (identifier) with a set of attributes. The attributes may be a long name, the type, relations with other propositions or variables, and so on. For instance, relations like "membership to a group of antibiotics" or "smaller spectrum" are intensively used when modelling antibiotics in *Terap-IA*.

The type is the only attribute that is mandatory in these declarations and determines the set of allowed values a proposition or a variable can take, apart from the special value *unknown*.

There are three types of propositions, *boolean*, *fuzzy* and *many-valued*; and three types of variables, *numerical*, *linguistic* and *set*. *Boolean* propositions represent concepts which can only be evaluated as either *false* or *true*, and the value of a *numerical* variable is a real number. The rest of this section concentrates on the other types which are more interesting with respect to the focus of this paper.

2.1.1. Fuzzy propositions

In some cases we need to deal with vague concepts. For instance, in *Terap-IA* we are interested in the degree of truth of *presence of creatinine* in order to support the diagnosis

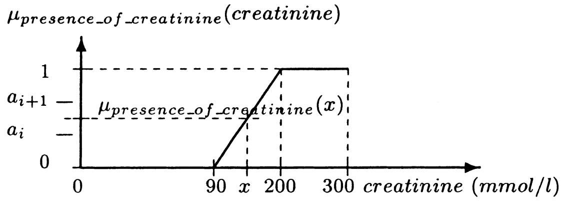

Fig. 1. Fuzzy set representing the concept ‘creatinine’.

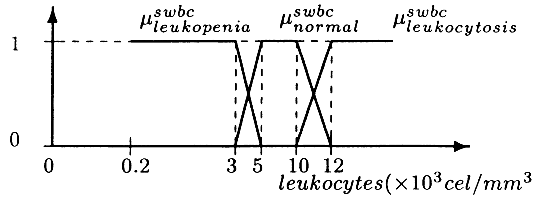

Fig. 2. Linguistic variable representing the concept ‘swbc’.

of a renal failure, instead of a concrete numerical value of the variable *creatinine* (see the example in Section 2.2). Vagueness of concepts can be quantified by the degree of membership of a numerical value to a fuzzy set, so, the *presence of creatinine* is modelled by giving a fuzzy set (see Fig. 1) that takes as argument the numerical value in millimolar per liter of the amount of *creatinine*.

The value of a fuzzy proposition is obtained by the application of its associated fuzzy membership function to the value of a numerical variable. The final answer is the narrowest interval of linguistic terms[1] containing that number.

2.1.2. Linguistic variables

Their values are a user-defined finite set of linguistic values. Similarly to the case of fuzzy propositions we declare a linguistic variable by associating to every linguistic value, a fuzzy set with respect to a numerical variable.

In Fig. 2 we can see a representation of the concept *status of white blood cells* (swbc for short) by means of three fuzzy sets (linguistic values), *leukopenia*, *normal*, and *leukocytosis*. Given a numerical value for *leukocytes* the system can calculate the truth degree of a predicate “is” used to compare variables and linguistic values — the value for *swbc is leukopenia*, *swbc is normal* and *swbc is leukocytosis* — by applying the corresponding fuzzy sets.

2.1.3. Many-valued propositions

The concepts represented as many-valued propositions are those whose truth may be graded. That is the case of most deduced concepts in Milord II (see Section 2.2), for instance, *severity of illness of the patient* or *resistance of pneumococci to penicillin*.

[1] For this purpose we consider the set of linguistic terms to be uniformly distributed in the interval $[0, 1]$.

2.1.4. Set variables

They are fuzzy sets over finite domains. An example is the set variable *allergic reactions*. It is a set whose domain is the possible allergic reactions of the patient. Giving value to a set variable means associating to each element of the domain a truth degree. If a patient has only a clear allergy to penicillin, the set variable *allergic reactions* is bound to a set with value *true* for penicillin and *false* for the other allergic reactions of the domain.

Milord II contains comparison predicates over set variables and permits operations over them. We can compare different sets by their non-empty intersection, inclusion, or equality degree. For instance, we can know which is the degree of the presence of *penicillin* and *macrolides* in the set *allergies* by applying the intersection degree relation between the set variable *allergic reactions* and the crisp set with elements *penicillin* and *macrolides*.

2.2. Rules

In Milord II, a rule is composed of a premise (a conjunction of conditions), a conclusion, and a truth-value. In the case of conditions containing variables, the language provides with a set of predefined predicates that when applied to the conditions, produce as result intervals of truth values. Premises of rules are conjunctions of elemental conditions either in affirmative or in negative form. Conditions can be just propositions. For instance, using the fuzzy proposition *presence of creatinine* defined above, we can build the rule *If presence of creatinine then renal failure is definite*,[2] and with many-valued propositions a rule could be *If penicillin can be administrated to a patient and his situation is severe then it is almost impossible to administrate ampicillin*.

In Milord II rules we can use numerical expressions composed by numbers, numerical variables and arithmetic operations. For instance, *If respiratory frequency is* ≥ 30 *breaths per minute then we can conclude that tachypnea is definite*. We can also build set expressions that evaluate as degrees of inclusion, intersection and equality between fuzzy sets. For instance, *If the patient has allergic reaction to penicillin then he is also allergic to cephalosporins and cabapenems*. In this case *allergic reaction to penicillin* means that the set variable *allergies* has a non-empty intersection with the crisp set *{penicillin}*.

For an example of use of a linguistic variable, take the previously defined linguistic variable *state of white blood cells* to define: *if state of white blood cells is leukopenia then there is analytical evidence of severity of pneumonia*.

Conclusions of rules are simpler than conditions. Conclusions may appear either in affirmative or in negative forms. Only many-valued propositions and set variables can be used as conclusions in rules. In fact we have seen examples of conclusions in the examples above. For instance the conclusion *tachypnea* is about a many-valued proposition and, *the patient has allergic reaction to cephalosporins and cabapenems* is about a set

[2] We use here an English version of the real syntax of rules. The interested reader is referred to [2] for a complete syntax description of Milord II.

variable. Deduction in these logics is performed by a "modus ponens"-like inference rule [3].

3. Renoir

Diagnosing rheumatic diseases is usually difficult for the nonspecialist physician, partly due to lack of skills through the pregraduate studies, and partly because of the absence of pathognomonic findings in many rheumatic diseases. If we look at most classification criteria tables we see that diagnosis (which is in itself basically a classification task) is usually performed combining a bizarre group of findings. For the same reasons, the nosology of the rheumatic diseases is not clear-cut. In this sense, we can say that rheumatology is a *fuzzy domain of knowledge* because the limits or boundaries among the diseases are not always well defined and the information available has high levels of uncertainty associated with it. Clear examples of these statements are collagen diseases and vasculitis.

The medical knowledge included in *Renoir* [5] comes from three main sources: (1) widely available rheumatologic concepts from books, manuals, and journals; (2) the more recent and internationally accepted classification criteria tables and trees for those diseases where such criteria are available; and (3) rules-of-thumb extracted from the clinical experience.

3.1. Materials and methods

Renoir's KB spans along 37 main inflammatory arthropathies and collagen diseases plus 15 diagnostic subtypes of these. The number of representation elements are: 978 propositions and variables, 1058 rules, 34 modules, and 220 meta-rules to control the applicability of the rules, to prune irrelevant facts, and to generate strategies, that is, problem solving methods. The main groups of propositions and variables in *Renoir* are: anamnesis, exploration, hematology, biochemistry, serology, urine, microbiology, skin, spine, articular, synovial-liquid, diagnosis, and therapy.

Although our goal was not to diagnose non-inflammatory diseases, we decided to implement a small group of rules for osteoarthritis, fibromyalgia, and non-inflammatory myopathies (Steinert disease, trichinosis). It should be noticed that *Renoir* has not been primarily designed to deal with multiple rheumatic diagnoses in the same patient.

Rules directly or indirectly related to the same diagnosis are grouped in 34 modules. Which can be classified into the following five general classes:

1. A data-gathering module.
2. Four intermediate hypothesis-generating (IHG) modules intended to develop intermediate hypothesis such as *chronic polyarthritis* or *serositis*. These intermediate hypothesis are used to set the problem in more concrete search spaces or *contexts*.
3. Twenty-three disease-specific modules to refine the reasoning process. Their goals are single disease entities such as *rheumatoid arthritis* or *polymyositis*.

4. Five general purpose modules to perform a qualitative abstraction from laboratory and radiologic data.
5. One last module with meta-rules to generate problem solving strategies.

3.2. Results

Renoir has been validated in a multicentric trial that has been described in detail in [6]. We have performed a validation of *Renoir* based on 32 patients from unselected hospitalisation reports from the *Hospital General de Castelló* in Spain. In this test, *Renoir* reached an overall 75% diagnostic accuracy when comparing the system's diagnoses with those in patient charts and reports. A formal, double blind, multicentric validation of 81 cases from five hospitals in several communities of Spain has been also done. In this validation a cluster analysis of the results was made to compare the way *Renoir* and 12 physicians with diverse experience in rheumatology performed diagnostic tasks. *Renoir* clustered near the top experts' cluster, so we can say that its performance is indeed very good.

4. Pneumon-IA

The aim of *Pneumon-IA* is to assess the etiology of community-acquired pneumonia from clinical, radiological, and laboratory data obtained at the onset of the disease. *Pneumon-IA* considers 22 possible etiological agents: 15 bacterial pneumonia, four viral pneumonia, two fungal pneumonia, and a pneumonia caused by parasites (*Pneumocystis carinii*).

Although of obvious importance for its treatment, etiological diagnoses of pneumonia imply great uncertainty, since etiology is seldom confirmed and therefore it is difficult to establish a gold standard to compare this knowledge-based system with human experts.

4.1. Materials and methods

Pneumon-IA's KB is mainly implemented through rules. It comprises 487 propositions and variables, 659 rules, 92 meta-rules and 25 modules. Each etiological agent is represented by a module. Each diagnosis is qualified with one of eight labels of possibility.

Validation was performed using data from medical records of 76 patients with confirmed clinical diagnoses of pneumonia. The etiological diagnosis provided by *Pneumon-IA* were compared to those established by five specialists unrelated to the development of the knowledge-based system. For each etiological possibility, both *Pneumon-IA* and the experts provided a causal possibility, expressed by means of linguistic labels. Linguistic labels were then converted to numeric values. In the majority of cases, an etiological diagnosis was unavailable to be used as a gold standard. To overcome this limitation, distances between arrays of etiological possibilities given by specialists and by *Pneumon-IA* were considered as an agreement measure between diagnosis. Cluster analysis based on those distances was used to rank *Pneumon-IA* in relation to experts. See [7] for details.

4.2. Results

The results obtained applying our approach to *Pneumon-IA* validation show that differences between etiological diagnosis made by the knowledge-based system and those made by some specialists were smaller that differences between some specialists themselves. Specialists with highest proficiency scores gave the closest diagnosis. The best specialist was, moreover, the one with the least *omitted* etiologies (that is, an etiology mentioned by every specialist, including *Pneumon-IA*, except one) and least *singular etiologies* (those mentioned only by one specialist). *Pneumon-IA* was much closer to the ''best'' than to the ''worst'' specialist. It supplied the highest number of singular etiologies, largely because it takes viral etiologies into consideration more often. The success of *Pneumon-IA* in cases of confirmed etiology was similar to that of clinical specialists. Etiological diagnoses emmited by the knowledge-based system agreed with the best known specialist in our area. See [7,8] for details.

5. Terap-IA

There are many microorganisms causing pneumonia. Nowadays, with the available diagnostic methodology, it is still very difficult to determine which of the microorganisms is the infecting agent in a particular pneumonia case. The research focused to determine which are the microorganisms causing pneumonia only succeeds in 50% of the cases.

Despite the uncertainty of the diagnosis, a treatment has to be speedily administrated to avoid a negative evolution of the severity of the illness or in some cases the death. Besides the uncertainty on the diagnosis, data about the patient is in many cases also uncertain and incomplete.

In *Terap-IA* we made two main assumptions:

1. *Existence of a previous diagnosis:* A pneumonia is normally caused by only one microorganism, but symptoms and signs are not specific enough to determine which one. Diagnosis usually gives evidence for two or three microorganisms possibly causing pneumonia. We assume that such diagnosis already exists. It can be obtained from *Pneumon-IA* (see Section 4).
2. *Independence of treatments:* We can independently find the best treatment for every microorganism appearing in a diagnosis. Moreover, these treatments can be combined to give a treatment covering all the possible causes. By ''covering'' we mean that a treatment is specific for a particular microorganism.

5.1. Materials and methods

The concepts managed in *Terap-IA* are those related to the pharmacological knowledge about pneumonia treatments and those related to the clinical condition of the patient. The goal of the system is to find the best combination of antibiotics to treat a patient with pneumonia (see [9] for details).

We consider an *antibiotic treatment* for a microorganism as a set of *antibiotics*, normally one and occasionally two. An *antibiotic combination* is the result of covering more than one microorganism. It is a set of one (some antibiotics cover more than one microorganism), two, or exceptionally three antibiotics.

The implementation was made by programming about one hundred Milord II modules. Next we give an idea of their commonalities by explaining groups of modules.

5.1.1. Pharmacological modules

These modules contain the knowledge about antibiotics in the domain of *Terap-IA*. We represent each antibiotic as a concept for which we declare the *pharmacological group* it belongs to; which is the *administration route* of that antibiotic (oral or parenteral); the possible *interactions* with other drugs administrated to the patient and which are the antibiotics with the same *activity*.

5.1.2. Data acquisition modules

These modules gather the patient data that the expert considered to be relevant for a correct treatment determination.

Many concepts in the domain are vague. In some cases quantitative data are qualitatively abstracted by means of fuzzy sets to facilitate the task of the expert and the user. For instance, it is easier for the expert to reason about the concept *the state of white blood cells is normal* than *the number of leukocytes is 7000 cells per cubic millimeter*. This qualitative abstraction allows the expert to say, for example, that there is *penicillin-resistance of pneumococci* when the *state of white blood cells is leukopenia*.

5.1.3. Sieve modules

These modules modify the current antibiotic treatment independently of the microorganism we want to treat, by eliminating antibiotics belonging to concrete groups of the list of possible treatments for a patient. These modules use the pharmacological and acquisition data modules to determine which are the groups of antibiotics that is possible to administrate to a given patient. The modules of this group are: *pregnancy*; *allergies*; *renal failure*; and *genetic conditions*.

5.1.4. Microorganism modules

There are twenty-two groups of microorganism modules, one group for each microorganism treated by *Terap-IA*. They deduce which antibiotics to use to treat a microorganism, giving a truth-value for each antibiotic. The truth-value of an antibiotic is obtained taking into account the truth-value of its pharmacological group (obtained by the sieve modules) and data about the patient. Given a *diagnosis* (one, two or three possible microorganisms) the system only considers the modules corresponding to the microorganisms in the diagnosis.

5.1.5. Combination modules

These modules combine the results of the microorganism modules. The results of these modules are weighted antibiotic combinations. These combinations take into account some restrictions. Examples of restrictions are: combinations never contain antibiotics belonging to the same group, or combinations never contain antibiotics that have the same sensibility.

5.1.6. Sieve combinations modules

The combinations of antibiotics are ''sieved'' taking into account other restrictions such as specificity or cost.

The treatment generation process gives a ranking of combinations from which the clinician chooses according to his/her preferences.

5.2. Results

The expert has developed a partial validation of the system contrasting real world cases of antibiotic treatment of pneumonia with the answers of *Terap-IA*. The treatment for each previous diagnosis has been verified separately, producing hopeful results. The ongoing final validation is carried out by providing fifty real world cases of pneumonia to four human experts and *Terap-IA*, and comparing the results.

6. Future directions

We are working on the use of Milord II in a multi-agent environment as the deliberative component of agents in the framework of the SMASH project. It is very promising to model an hospital as a set of specialised autonomous agents communicating among them. Besides the interaction among human agents — physicians, nurses and administrators — artificial agents can interact among them and also with human agents facilitating some tasks. Milord II would be adequate as the deliberative engine of agents performing intelligent tasks.

7. Resources

The last version — available for research and educational purposes — and more information on Milord II can be found at http://www.iiia.csic.es/~milord.

Acknowledgements

We acknowledge the Spanish Comisión Interministerial de Ciencia y Tecnología (CICYT) for its continued support through projects: ACRE (CAYCIT 836/86), SPES (880J382), ARREL (TIC92-0579-C02-01) and SMASH (TIC96-1038-C04-01). This research has also been partially supported by the Esprit Basic Research Action number 3085 (DRUMS) and by the European Community project MUM (Copernicus 10053). The definition of this language has profited from ideas coming from many people, among which we would like to thank Francesc Esteva, Jaume Agustí and Don Sannella. We also thank the experts that have developed applications based on Milord II, Albert Verdaguer, Miquel Belmonte, Marta Domingo, Pilar Barrufet, Lluís Murgui and Ferran Sanz.

References

[1] Godo L, López de Màntaras R, Sierra C, Verdaguer A. Milord: the architecture and management of linguistically expressed uncertainty. Int J Intell Syst 1989;4:471–501.

[2] Puyol-Gruart J, Sierra C. Milord II: a language description. Mathware Soft Comput 1997;4(3):299–338 .

[3] Puyol-Gruart J, Godo L, Sierra C. Specialisation calculus and communication. Int J Approximate Reasoning (IJAR) 1998;18(1/2):107–30.

[4] Puyol-Gruart J, Godo L, Sierra C. Handling fuzzy information in Milord II. In: Proceedings of the 9th INCOM'98 of the Symposium of the International Federation of Authomatic Control in Manufacturing, vol. 3, 1998. p. 85–90.

[5] Belmonte M, Sierra C, López de Mántaras R. Renoir: An expert system using fuzzy logic for rheumatology diagnosis. Int J Intell Syst 1994;9(11):985–1000.

[6] Hernández C, Sancho JJ, Belmonte MA, Patak A, Sierra C, Sanz F, Validation of the medical expert system renoir. In: Proceedings of the World Conference on Expert Systems, Lisbon (Portugal), 1994.

[7] Verdaguer A, Patak A, Sancho JJ, Sierra C, Sanz F. Validation of the medical expert system pneumonia. Comput Biomed Res 1992;25(6):511–26.

[8] Meseguer P, Verdaguer A. Expert system validation through knowledge base refinement. Int J Intell Syst 1996;11(7):429–62.

[9] Barrufet P, Puyol-Gruart J, Sierra C. A knowledge-based system for pneumonia treatment. In: Proceedings of the EIS'98 of the International ICSC Symposium on Engineering of Intelligent Systems, vol. 1, 1998. p. 176–82.