Diagnosis Systems in Medicine With Reusable Knowledge Components

Maria Taboada, *University of Santiago de Compostela*
Julio Des and José Mira, *National University for Distance Education, Spain*
Roque Marín, *University of Murcia*

This article shows how we developed a diagnosis-aid system by reusing and adapting generic knowledge components for diagnosing eye emergencies.

Constructioning knowledge systems is viewed as modeling activity for developing structured knowledge and reasoning models. To ensure well-formed models, the use of some knowledge engineering methodology is crucial. Additionally, reusing models can significantly reduce the time and costs of building a new application. Reusing knowledge components across different applications and domains can help acquire expert knowledge and accurately describe the reasoning process. In fact, current knowledge engineering research has taken major initiatives in the development of knowledge systems by reusing generic components, such as ontologies[1] or problem-solving methods.[2] Examples of knowledge engineering approaches based on reusing problem-solving methods include the methodology CommonKADS,[3] the formal framework UPML,[4] and the general-purpose framework Protége-II.[5]

Dieter Fensel and Enrico Motta[6] proposed characterizing method development and selection as a process consisting of navigating through a 3D space composed of methods, tasks, and domain assumptions. The navigation through the space is made by means of adapters. To facilitate this navigation, we used one- or two-dimensional adapters. During the modeling of our diagnosis system, we distinguished two types of task–problem-solving method adapters: *task-based adapters*, which define new subtasks such as input data check-up or filters, and *renaming adapters*, which directly connect I/O roles. We have also used one and two-dimensional refiners to configure the internal method specification and its requirements.

By combining several current approaches for reuse of problem-solving methods, you can model the diagnosis task. Following these approaches, we can characterize a problem by task commitments, problem-solving paradigms, and domain knowledge assumptions.[5,6] We base the steps for developing a reusable knowledge model by selecting and configuring generic problem-solving methods. In our approach, method configuration is an assumption-driven and different method combination-based activity.

The methodological approach

A problem-solving method (PSM) specifies a domain-independent reasoning pattern, characterized by a set of actions that must be carried out to solve a problem, a control structure over these actions, and a set of knowledge roles that specify the needed static and dynamic knowledge.

In CommonKADS, a method (called *task-method*) is defined by a set of inferences that set up the actions to carry out, an inference structure that shows the data flow among these inferences, and a set of knowledge roles. Additionally, CommonKADS provides a library of task-methods, and developing a new application consists of selecting and adaptating methods from this library. However, we can view a task-method as a task's PSM instantiation, so the method includes some

1094-7167/01/$10.00 © 2001 IEEE

specifications about problem commitments. The method-adapting process for a new problem consists of modifying some templates stored in the library; therefore, the reusing of methods is not immediate.

For another approach, we must describe a method in a higher abstraction level, without including specifications about problem commitments. This second option's advantage is its higher reusability potential. However, this description is too abstract and difficult to understand. In fact, different authors use different versions of some generic methods, even with different numbers of input and output roles. In spite of this disadvantage, we considered using the generic PSM's description separately from problem commitments. This approach facilitates reuse methods, whenever methods are unambiguously defined and mechanisms (such as adapters[6]) facilitate the connection between methods and problem types.

In this way, the Protege-II approach distinguishes between tasks and PSMs. The key steps in developing a knowledge system comprise task analysis, method selection, and method configuration. We view task analysis as a modeling activity[5] that identifies a problem and obtains I/O relationships and the available knowledge. Thus, an *application task* is a set of assumptions about the problem space.[7] Additionally, some approaches, such as UPML,[4] distinguish between precondition, postcondition, and domain-level assumptions.

Method selection is too complex because the most appropriate method choice depends on many factors, including task assumptions, solution quality, computational complexity, and so on. This step is frequently performed manually, but some projects have proposed to do this automatically.

You can view *method configuration* as an *assumption-driven activity*[7]—that is, refining an application task (described by a set of assumptions) taking into account the method's functional specification and the available domain theory. You could also consider method configuration as a process of refining generic problem-solving methods using the ontological commitments instead of refining tasks.[6]

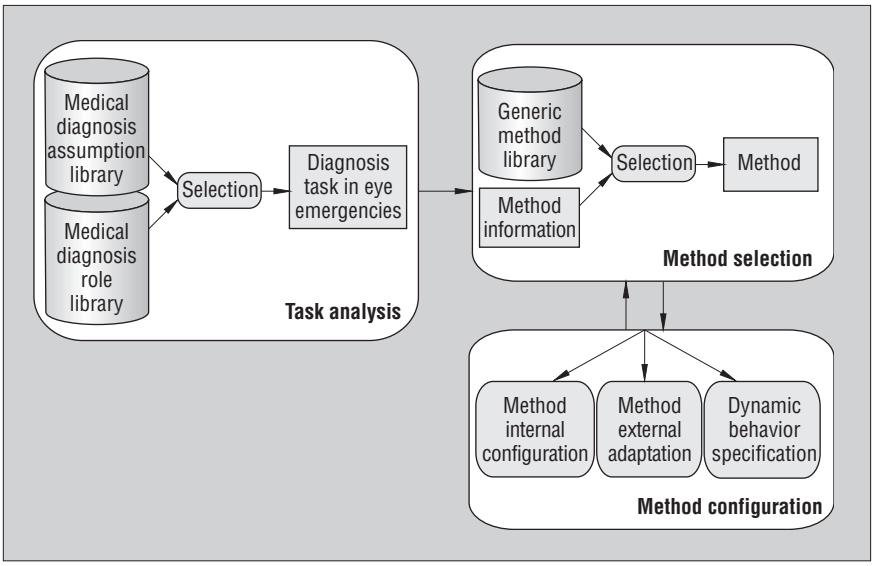

Figure 1. Steps to develop eye emergency diagnosis by reusing problem-solving methods.

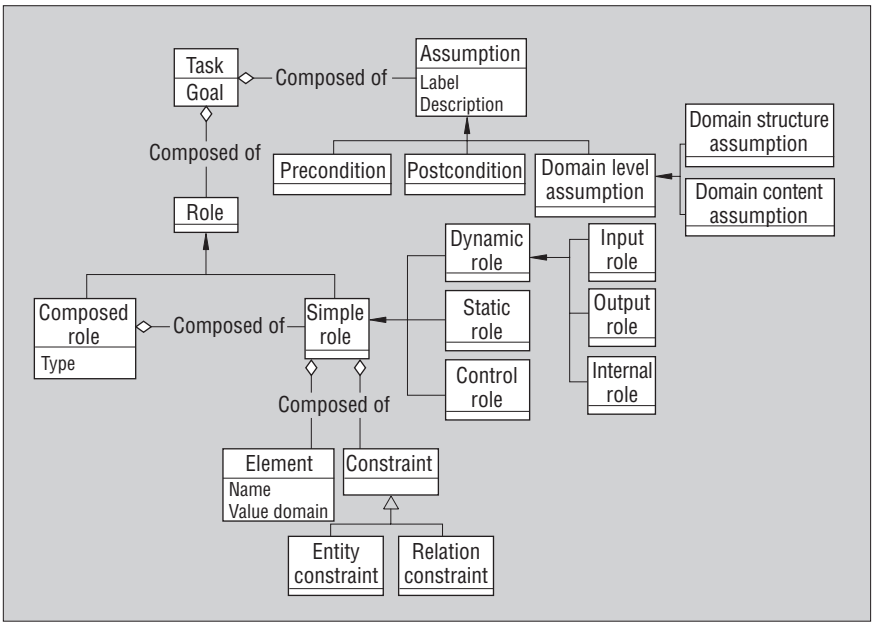

Figure 2. Assumption and role specification.

Building a library

The AI community has extensively studied the diagnosis task and proposed many methods and approaches for both technical and medical domains for reusing knowledge components in diagnosis. Several PSMs were revised. We highlight Richard Benjamins' library of PSMs for technical diagnosis[8] and John and Susan Josephson's work on medical abductive diagnosis. [9] From these works, we built an assumptions library and a library of medical diagnosis roles. We carry out *diagnosis task modeling* by selecting a subset of assumptions and roles from these libraries (see Figure 1). In this way, eye emergency diagnosis has been modeled by a descriptive specification, including a goal, a set of assumptions, and a set of roles (see Figure 2). A label and an explanatory text define each assumption and three types we have distinguished as *preconditions*, which impose requirements about input data, *postconditions,* which deal with output data and the reasoning process, and *domain-level*, which concerns the domain model. Additionally, we classified the latter under *domain structure assumptions* and *domain content assumptions*.

Roles and assumptions

To describe diagnosis of eye emergencies, we selected the following roles and assumptions. Later, during method selection and configuration, we added more assumptions and roles.

Input dynamic roles

set of observations is composed of simple roles named observations, which we define as a three element tuple of the form <*o, f, v*>, with the following constraints: *o* is a kind of entity, f is-a-slot-of *o*, and *v* is an allowable value of *f*.

Output dynamic roles

diagnosis is a set of simple roles, described by one element <*d*> that is a kind of *candidate*. additional data are described by a three elements tuple of the form <*t,o, f*> with the following constraints: *t* and *o* is a kind of entity, *f* is a slot of *o*.

Internal dynamic roles

set of abstracted data is composed of simple roles named abstracted data, which we define as a three element tuple of the form <*o, f, v*> with the following constraints: *o* is a kind of entity, *f* is a slot of *o*, and *v* is an allowable value of *f*.

Static roles

set of candidates is a set of simple roles named candidates, which are described by one element <*c*> that is a kind of entity.

diagnostic associations exist between one candidate and many observations, and between one candidate and many abstracted data.

data associations exist between many observations and many higher abstracted data.

hierarchical and non-hierarchical candidate associations also exist.

Preconditions

User control over introducing data. Clinical experts often want to have control during the introduction of patient information. Therefore, the user should have options available to introduce values for all features of all observables.

Postconditions

Positive coverage. The solution must explain the set of abnormal observations and the set of normal observations must be consistent with the solution.

Multiple fault assumption. This states that an arbitrary number of diseases can be responsible for the appearance of a set of patient findings.

Composed or alternative solutions. The first states that you can have possible solutions together, and the second one states that the possible solutions are alternatives. In our approach, you must compose initial candidates (or their refinements) together, and two candidates derived from an hierarchical refinement are alternative solutions.

Ranked solutions by scoring. You can solve the solution choice by ranking the whole of possible candidates following some scoring method.

Combining data-driven reasoning and solution-driven reasoning. Reduce the set of possible hypotheses from the information that comes in during the reasoning process and obtain the set of additional data from the possible set of hypotheses.

Sensitivity-based selection. Select additional data by taking into account the sensitivity of each finding both for each hypothesis and for the joint hypotheses. If a hypothesis is true, the most probable findings to be present will be the most sensitivity findings. Therefore, if the highest sensitive findings are not present, you can rule out the associated hypotheses.

Reasoning with unknown *and* not valuable *values.* Do not rule out a hypothesis if there is some unknown finding with a higher sensitivity. Additional data include unknown findings to collect them. You cannot rule out a hypothesis whereas there is some nonvaluable higher sensitivity finding.

Independence of hypotheses. An individual hypothesis explains a set of observations regardless of what other hypotheses can explain.

Domain structure assumptions

Simple causal model. Two single levels of detail model causal relations—that is, causal connections between each candidate and a set of abstracted data and causal connections between each candidate and a set of complications (which are candidates).

Solution refinement hierarchies. The solution space is structured according to a refinement hierarchy.

Hierarchical and nonhierarchical data associations. The input data are structured according to hierarchical and nonhierarchical associations.

Domain content assumptions

Complete association set. The associations must form a complete set to generate all relevant hypotheses, and an instantiation of some causal association must exist for each hypothesis.

Observation types. You can classify the set of observations into normal and abnormal observations.

Selecting and configuring generic problem-solving methods

We describe the generic method by an external and an internal specification. The first one describes a method as a black box in terms of its externally visible behavior. It includes specifications about the dynamic and static input data requirements and the provided output data. However, the internal specification describes how the method operates. It describes generic methods in terms of their internal structure (a set of subtasks and internal roles) and the interaction among parts of its internal and external structure (that is, a function structure representing the data and knowledge flow among subtasks). Following this description, method configuration comprises three steps—configuration of the internal specification to achieve the task postconditions; adaptation of the external specification to the task roles, preconditions, and domain-level assumptions; and specification of the dynamic behavior (see Figure 1).

These steps, together with the method selection step, are carried out by an iterative cycle, where each cycle corresponds to the selection and configuration of methods for each subtask. For space reasons, we are omitting the specification of the method dynamic behavior.

In clinical domains, many diagnostic approaches are built using generate and discriminate. This classic method is based on the generate-and-test strategy applied by medical experts during the differential diagnosis process. Internally, the method generates a list of possible hypotheses and then further discriminates among the generated hypotheses, usually by taking into account additional data. We need knowledge about associations between data and candidates to generate and discriminate hypotheses. Also, in many cases in clinical domains, the method should return additional data to be gathered for the set of hypotheses to be more efficiently discriminated. This method satisfactorily carries out the combining solution-driven and data-driven reasoning assumption.

Adaptation of the method external specification

To adapt the external specification to the tasks preconditions, we have considered the following information about method computational complexity and efficiency:

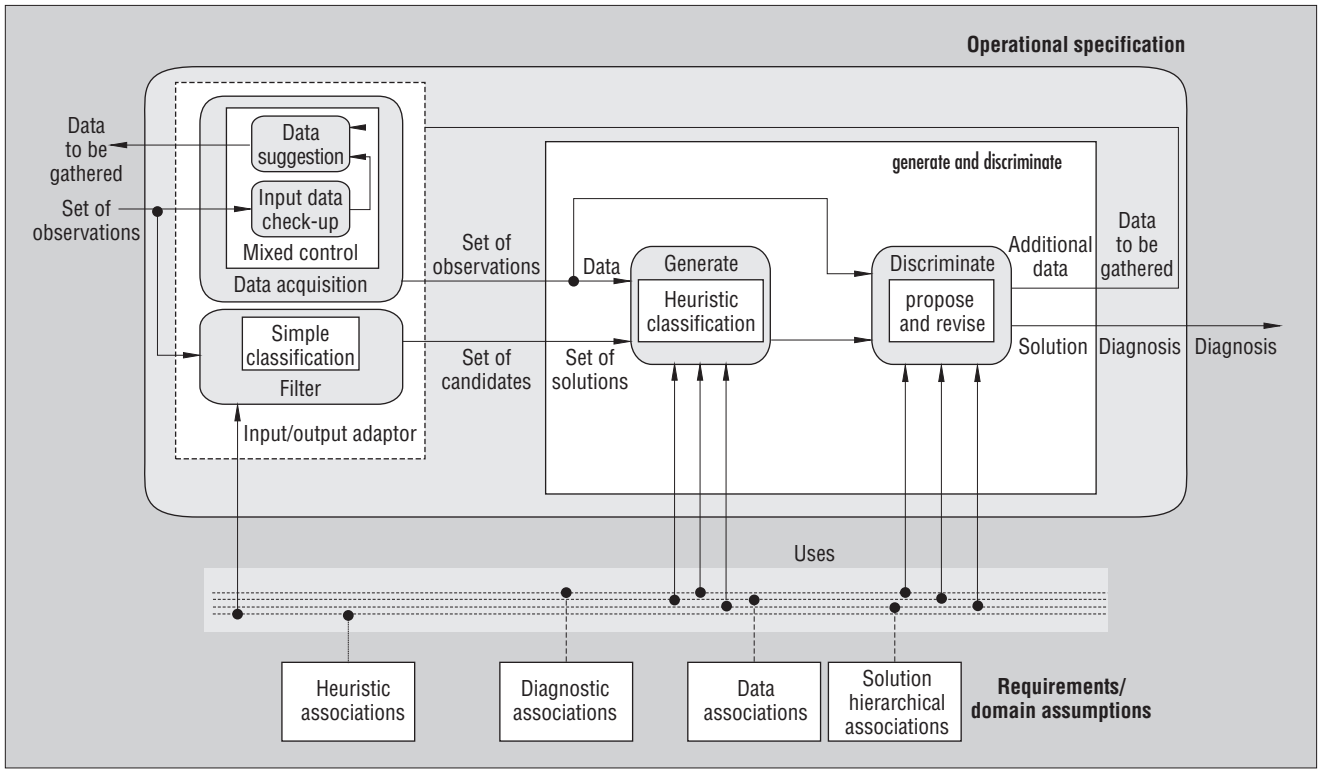

Figure 3. Initial configuration of the generate and discriminate method for the diagnosis task in eye emergencies. You can solve the subtask generate using the heuristic classification method and the subtask discriminate by using the propose and revise method.

- The method's internal configuration will be easier if, during each new clinical session, one feature of an observable can only have one value. The adaptation of the method to the task's precondition will be carried out by a subtask called **input data check-up**. To avoid the execution of the method with data containing more than one value for the same feature of an observable, the subtask **input data check-up** checks the list of initial observations and notifies the user when he or she tries to introduce a new value for a feature with a known value already in due session, values out of range, or values incompatible with other feature's values.
- When the diagnostic model is too extensive, the diagnosis task will likely become more computationally tractable if you reduce the initial candidate space using heuristic knowledge. So, the related subtask will consist of filtering the set of all possible candidates and obtaining a smaller subset, which has been called **set of candidates**. You can solve this subtask using the **simple classification method**.
- The tasks operational specification will be simpler if the control over introducing data is distributed between the user and the task.

We have modeled the method's adaptation to the diagnosis task by an I/O bridge containing two subtasks, **data acquisition** and **filter** (see Figure 3). An ad-hoc, mixed-initiative control method solved the data acquisition and a simple classification method solved the filter. In Figure 3, the three assumptions about domain structure defined in the diagnosis task enrich the method.

Configuration of the heuristic classification method

William Clancey's heuristic classification method[10] solves a task by using three subtasks named **abstract**, **heuristic match**, and **refinement**, by which the available data are abstracted, matched to possible solutions, and then refined (see Figure 4).

Data abstraction consists of inferring new data from existing data using data associations as domain assumptions. In medicine, several types of domain associations are distinguished:

- *Definitional abstractions* between a concept property and expressions about properties of one or several concepts,
- *Qualitative abstraction* between qualitative concepts and quantitative values of concepts and
- *Generalization* and *Specialization* higher level data and lower level data in a hierarchical structure. Additionally, we considered that abnormal or unexpected observation detection can be viewed as a data abstraction task.

In CommonKADS, data abstraction is viewed as an inference, which is solved by a forward chaining. As the latter is a process defined in the design model and not in the expertise model, we considered data abstraction a subtask. A primitive abstraction method solves this subtask. It is defined by an inference structure, which includes two basic inferences, select and match (see Figure 5).

The **heuristic match** subtask obtains the initial set of solutions by comparing each abstracted datum with the set of data explained by each candidate (as shown in Figure 5). The subtask result is a set of individual solutions, each one explaining some observation. Each hypothesis of this set is verified or rejected during the discrimination step.

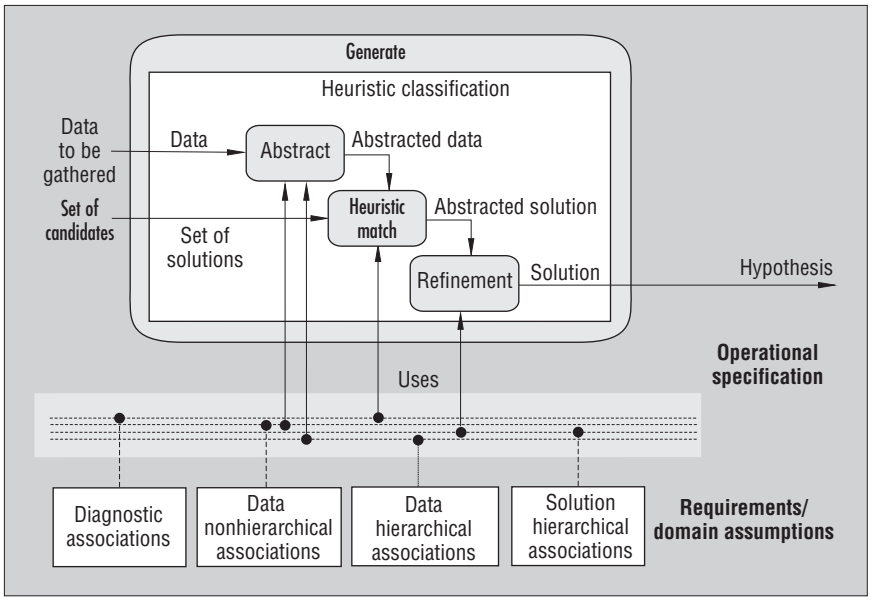

Figure 4. Configuration of the heuristic classification method for the generate subtask.

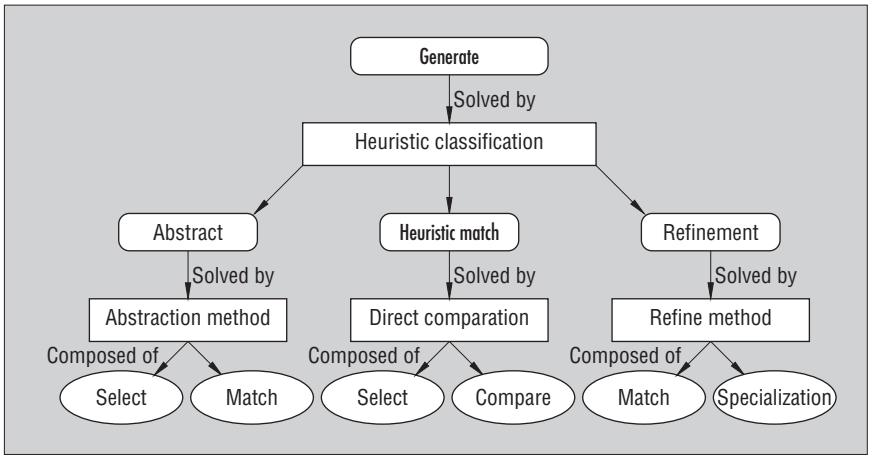

Figure 5. Breakdown of the generate subtask.

Hypothesis *refinement* consists of generating more specialized hypothesis from solution hierarchical associations. It is solved by matching a set of conditions associated to the solution node in the hierarchical structure, and by specializing the solution if some refinement condition is true.

In Figure 4, we can see the connection between task and method I/O roles, which is carried out by renaming roles. In addition, the method has been enriched with two domain structure assumptions about the type of data associations (hierarchical and nonhierarchical).

The propose and revise method solves the discriminate subtask by proposing a plan to gather additional data and revise the set of hypotheses using new data that the user introduced. Two different methods, as a function of the domain data types solve the propose subtask. That is, this subtask has been enriched with an assumption about data types—symptoms and signs. A ranking method is applied to propose a basic plan for gathering symptoms, that is, for carrying out the *Anamnesis* (the patient's case history)—and the propose-verify-modify method obtains a basic plan about the set of tests to achieve. Following the first method, each symptom, which can be caused by a scored hypothesis, taking into account the symptom sensitivity both for individual and joint hypotheses. This method assumes the *sensitivity-based selection* assumption—that is, additional data obtained on the basis of corroborating the set of hypotheses, and not by ruling them out.

On the other hand, the propose-verify-modify method obtains a test plan. To configure the method, it is enriched with two assumptions about the type of tests—physical examination and complementary. The physical examination tests consist of a set of ordered exploration steps, where a set of signs are collected in each step. You carry out each exploration step by using a set of different instrumental techniques and applying a set of different pharmacological actions. As a result, an exploration step can cause an adverse effect in the presence of some disease. In this case, the exploration step is incompatible to a hypothesis and must be replaced by a different step intended to gather the same signs. The propose-verify-modify method proposes a basic plan by selecting the most relevant signs for each hypothesis, merging all the signs into one set, selecting a basic exploration phase for each sign, and ordering the set of phases. It then verifies the suitability of each phase, taking into account the set of diagnosis hypotheses. If a phase is not suitable, it must be substituted for a different phase or ruled out from the plan if the new proposal has already been replaced within the same cycle.

The revise subtask, from the propose-and-revise method, verifies whether or not the set of individual hypothesis explains the set of new additional data. You can carry out this subtask by a basic abductive method, because the *Multiple Fault Assumption* is achieved by this method, whereas the *Independence of hypotheses* are achieved by this method. A basic abductive method decomposes a task in four subtasks:

- propose a set of individual hypotheses,
- verify each hypothesis from the complete set of data,
- refute the set of lower suspect hypotheses, and
- assemble individual hypotheses by applying a basic covering method.

You can solve the proposal of a set of individual hypotheses by directly adding the new hypotheses generated in the generate task (as a consequence of the new additional data), to the set of hypotheses resulting from previous data.

The hierarchical classification method solves the verification of each hypothesis and, in turn, a ranking method solves the establish subtask. This ranking method consists of matching each individual observable that can be explained by each hypothesis against the observed datum.

The Authors

María Taboada is an associate professor of computer science and artificial intelligence at the Department of Electronics and Computation, Univer. de Santiago de Compostela. Her research interests include knowledge engineering and knowledge-system development in medicine. Contact her at the Dept. of Electronics and Computation, Universidad de Santiago de Compostela, Edificio Monte de la Condesa, Campus Sur, 15706 Santiago de Compostela, Spain; chus@dec.usc.es; http://aiff.usc.es/~keam.

Julio Des is an ophthalmologist at the Hospital de Monforte at Lugo and a PhD student in the Department of Artificial Intelligence, National University for Distance Education. His research interests include artificial intelligence applied to medicine and telemedicine. He is an assessor for the refereed journal, *Archivos de la Sociedad Española de Oftalmología.* Contact him at the Hospital Comarcal, Servicio de Oftalmología, Monforte de Lemos 27400, Lugo, Spain; jjdes@ctv.es; http://aiff.usc.es/~keam.

José Mira is a professor of computer science and artificial intelligence and head of the Department of Artificial Intelligence, National University for Distance Education. His current research interests include AI fundamentals from the perspective of a knowledge-modeling discipline similar to electronics engineering, neural modeling of biological structures (and application of these models to the design of more realistic artificial neural nets), and integration of symbolic and connectionist problem solving methods in the design of knowledge environments for Web-oriented medical applications. He is the general chairman of the biennial interdisciplinary meetings on the interplay between neuroscience and computation for International Work Conference on Natural and Artificial Neural Networks. Contact him at the Dept. Artificial Intelligence, Faculty of Science, UNED, Senda del Rey, 28040 Madrid, Spain; jmira@dia.uned.es; www.ia.uned.es/personal/jmira.

Roque Marín is a professor at the Department of Informatics, Artificial Intelligence and Electronics, University of Murcia. His research interests include temporal reasoning, fuzzy systems, knowledge systems, and knowledge engineering methodologies applied to medicine and agriculture. Contact him at the Dept. of Informatics, Artificial Intelligence and Electronics, University of Murcia, Campus de Espinardo, 30100 Espinardo (Murcia), Spain; roque@dif.um.es; http://intelect.dif.um/~aike/roque/index.htm.

Inconsistent, *explained*, *unexplained*, *missing*, and not valuable *values* result from this matching. The ranking method also computes the degree of matching of each hypothesis and the set of data, taking into account the results of previous matching. This computation is based on the *positive coverage* and the *reasoning with unknown* and not valuable *values* postcondition. It then sorts the set of individual hypothesis into two subsets, higher suspect hypotheses and lower suspect hypotheses.

You assemble an individual hypothesis by applying a basic covering method, based on the *independence of hypotheses* assumption. The causal covering method solves the finding of contributors; that is, for each datum, a subset of hypotheses is obtained by containing the hypotheses for which the datum has been labeled as explained or not valuable. The transformation of the individual hypothesis into a hypothesis set is carried out by the basic set covering—that is, a hypothesis covers all contributor sets.

The key steps involved in developing our diagnosis system have been based on the Protége-II approach.[5] In particular, the method configuration step has been considered as an *assumption-driven activity*.[7] Nevertheless, it is not an assumption refinement-driven activity but a different method-combination-based activity. We used generic methods usual in knowledge engineering such as heuristic classification, propose-critique-modify or generate and test, and methods specific to concrete tasks, such as ranking methods or abstraction methods. A prototype of the system was implemented using the knowledge engineering tool Kappa-Pc. This version facilitated the acquisition of new knowledge and the evaluation of the reasoning model. However, as an isolated version, it holds up the updating process. The system is being migrated to a Web environment by using the medical knowledge acquisition tool Keam, which is being developed by our research group. ■

References

1. B. Chandrasekaran, J.R. Josephson, and R. Benjamins, "What Are Ontologies, and Why Do We Need Them?," *IEEE Intelligent Systems*, vol. 14, no. 1, Jan. 1999, pp. 20–26.
2. R. Benjamins and D. Fensel, "Editorial: Problem-Solving Methods," *Int'l J. Human-Computer Studies*, vol. 49, no. 4, Oct. 1998, pp. 305–313.
3. A. Schreiber et al., *Engineering and Managing Knowledge: The CommonKADS methodology*, The MIT Press, 1999.
4. D. Fensel et al., "UPML: A Framework For Knowledge System Reuse," *Proc. 17th Int'l Joint Conf. Artificial Intelligence (IJCAI 99)*, Morgan Kaufmann, San Francisco, 1999.
5. H. Eriksson et al., "Task Modeling With Reusable Problem-Solving Methods," *Artificial Intelligence*, vol. 79, no. 2, Jan. 1995, pp. 293–326.
6. D. Fensel and E. Motta, "Structured Development of Problem-Solving Methods," to be published in *IEEE Trans. Knowledge and Data Eng.*, 2001.
7. B.J.Wielinga, J.M. Akkermans, and A. Schreiber, "A Competence Theory Approach To Problem-Solving Method Construction," *Int'l J. Human-Computer Studies*," vol. 49, no. 4, Oct., 1998, pp. 315–338.
8. R. Benjamins, "Problem-Solving Methods for Diagnosis and Their Role in Knowledge Acquisition," *Int'l J. of Expert Systems: Research and Applications*, vol. 2, no. 8, 1995, pp. 93–120.
9. J. Josephson and S. Josephson, *Abductive Inference*, Cambridge University Press, New York, 1994.
10. W.J. Clancey, "Heuristic Classification," *Artificial Intelligence*, vol. 27, no. 3, Dec. 1985, pp. 289–350.