

## Research Article

# NEST: A Compositional Approach to Rule-Based and Case-Based Reasoning

Petr Berka<sup>1,2</sup>

<sup>1</sup>Department of Information and Knowledge Engineering, University of Economics, W. Churchill Square 4, 130 67 Prague 3, Czech Republic

<sup>2</sup>Centre of Biomedical Informatics, Institute of Computer Science of the Academy of Sciences, Pod Vodarenskou vezi 2, 182 07 Prague 8, Czech Republic

Correspondence should be addressed to Petr Berka, berka@vse.cz

Received 10 December 2010; Revised 9 April 2011; Accepted 16 May 2011

Academic Editor: Weiru Liu

Copyright © 2011 Petr Berka. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Rule-based reasoning (RBR) and case-based reasoning (CBR) are two complementary alternatives for building knowledge-based “intelligent” decision-support systems. RBR and CBR can be combined in three main ways: RBR first, CBR first, or some interleaving of the two. The NEST system, described in this paper, allows us to invoke both components separately and in arbitrary order. In addition to the traditional network of propositions and compositional rules, NEST also supports binary, nominal, and numeric attributes used for derivation of proposition weights, logical (no uncertainty) and default (no antecedent) rules, context expressions, integrity constraints, and cases. The inference mechanism allows use of both rule-based and case-based reasoning. Uncertainty processing (based on Hájek’s algebraic theory) allows interval weights to be interpreted as a union of hypothetical cases, and a novel set of combination functions inspired by neural networks has been added. The system is implemented in two versions: stand-alone and web-based client server. A user-friendly editor covering all mentioned features is included.

## 1. Introduction

Rule-based reasoning (RBR) and case-based reasoning (CBR) are two complementary alternatives for building knowledge-based “intelligent” decision-support systems. The first approach is closely related to expert systems. *Expert Systems* (ES) are typically defined as computer programs that emulate the decision-making ability of a human expert. The power of an ES is derived from the presence of a *knowledge base* filled with expert knowledge, mostly in symbolic form. In addition, there is a generic problem-solving mechanism used as the *inference engine* [1]. Some other typical features of expert systems include uncertainty processing, dialogue mode of the consultation, and explanation abilities. Beside ES dedicated to specific applications, “empty” expert systems (also called “shells”) have been developed, which can be coupled with an arbitrary knowledge base encoded in an appropriate format. Research in the area of expert systems started in the mid-1970s, classical examples of early systems that influenced other researchers are MYCIN

[2] and PROSPECTOR [3]. The knowledge of an expert is usually represented in the form of IF-THEN rules, which are applied in a deductive way: if the condition of a rule is satisfied, then this rule can be applied to either derive some conclusion or to perform the respective actions. The central point of all these systems was the *compositional* approach to inference, allowing us to compose the contributions of multiple rules (leading to the same conclusion) using a uniform combination function, regardless of their mutual dependencies. This approach was later subjected to criticism by most of the uncertainty-processing community [4], which resulted in ES research grounds dominated by *probabilistic approaches*. However, although probabilistic reasoning is more sound, it is also much harder for casual users to understand than rule-based reasoning. The opportunity for practical applications of probabilistic systems which require direct capture of human expertise (rather than to learn from a large set of cases) is thus significantly limited. Many developers of expert system applications, for example, members of company IT departments, decide to ignore

uncertainty modeling at all, or only adopt scoring schemes, which unfortunately have zero capacity for explanation and thus little potential for sharing.

The so-called knowledge acquisition bottleneck (i.e., the problem of eliciting the domain-specific knowledge from experts in the form of sufficiently general rules) has several workarounds. One of them is to use machine learning techniques to acquire knowledge from data representing situations successfully solved in the past. Another way is to use case-based reasoning, where the knowledge is represented in the form of a “list” of prototype problems and their solutions, so-called cases [5]. A case can have a form of just a list of attribute-value pairs specifying the problem and its solution, or it can have the very structured form of a frame with many slots, meta slots and procedures. *CBR systems* solve new problems by analogy, that is, by matching and adapting cases that have successfully been solved before. This seems to be a more psychologically plausible model of human reasoning than using rules as we do in classical rule-based (expert) systems.

Rule-based reasoning and case-based reasoning can be combined in three main ways: RBR first, CBR first, or some interleaving of the two. The RBR-first strategy is appropriate when the rules are reasonably efficient and accurate to begin with. If the rules are deficient in some way, the CBR first strategy may make more sense. If the rules and cases offer more balanced contributions to the problem solving, then an interleaving strategy may be best [6]. One of the early attempts to combine rule-based and case-based reasoning has been presented at IJCAI 1989. In the system CABARET proposed by Rissland and Skalak, both reasoning methods operate separately. An agenda-based controller then heuristically directs and interleaves the two modes of reasoning [7]. The system proposed by Chi and Kiang combines case-based reasoning to solve the problem with a rule-based explanation mechanism. The explanation mechanism can generalize an old case, if no exact match can be found for the new problem. The cases are represented using frames and the rules have a prolog-like form [8]. In the system described in [9], the CBR results are integrated in the RBR framework by rule refinement process. On the other hand, if no suitable case can be retrieved by the CBR component, RBR is used without integration. The cases here have the form of attribute-value pairs. The hybrid architecture reported in [10] is mainly focused on rule-based reasoning. The cases play a role of exceptions to the general rules. The results produced by case-based reasoning are used to determine whether a rule will fire or whether the solution proposed by a case is valid. Shi and Barnden propose to combine CBR and RBR for diagnosing multiple medical disorder cases. Rules are used to control and refine the CBR process in their approach [11]. In the hybrid CBR system described in [12] CBR and RBR components are invoked separately and the results are then combined using a fuzzy approach. Lee proposes a sequential activation of RBR and CBR, starting with RBR. The proposed system is built for internal audit of a bank [13]. The hybrid system, designed by Yang and Shen for diagnosis of turbomachines again starts with RBR. The user can then decide, whether or

not to continue with CBR. The results of both components are presented in parallel [14]. Kumar et al. describe a hybrid approach in which a CBR system is enhanced by a rule-based component [15]. Cabrera and Edye present an integration method where RBR is superior to CBR. If the RBR component can successfully (without doubt) infer the conclusion, the CBR step need not be performed, otherwise, CBR is used to find the solution. A case is represented as a set of logical attributes describing the presence or absence of the symptoms considered for the diagnosis [16].

We have developed a system that allows us to build both rule-based and case-based applications. Compared to traditional systems, we added more flexibility to knowledge representation (rules, contexts, integrity constraints, and cases) and reasoning. Our system covers the functionality of classic compositional rule-based expert systems, noncompositional (prolog-like) expert systems, and case-based reasoning systems. The paper provides an in-depth view of the features of NEST with respect to knowledge representation, inference mechanism, modes of consultation, and implementation. The system is freely available for download at <http://sorry.vse.cz/~berka/NEST/>.

## 2. Knowledge Representation

NEST uses attributes and propositions, rules, integrity constraints, and contexts for rule-based reasoning and cases for case-based reasoning to express the task-specific (domain) knowledge.

*Propositions* are derived from values of attributes. *Attributes* are used to describe the properties of the consulted case. Four types of attributes can be used in the system: binary, single nominal, multiple nominal, and numeric. According to the type of attribute, the derived propositions correspond to the following:

- (i) values True and False for a binary attribute;
- (ii) each value for a nominal attribute; the difference between single and multiple nominal attributes is apparent only when answering the question about the value of the attribute;
- (iii) fuzzy intervals for a numeric attribute; each interval is defined using four points; fuzzy lower bound (FL), crisp lower bound (CL), crisp upper bound (CU), and fuzzy upper bound (FU). The value of the attribute is transformed into the weight of the proposition “value within interval INT” as follows:

$$w(\text{INT}) = \begin{cases} -1 & \text{iff value} \leq \text{FL}, \\ 2(\text{value} - \text{FL})/(\text{CL} - \text{FL}) - 1 & \\ & \text{iff FL} < \text{value} \leq \text{CL}, \\ 1 & \text{iff CL} < \text{value} < \text{CU}. \\ -2(\text{value} - \text{CU})/(\text{FU} - \text{CU}) + 1 & \\ & \text{iff CU} < \text{value} \leq \text{FU}, \\ -1 & \text{iff FU} \leq \text{value}. \end{cases} \quad (1)$$

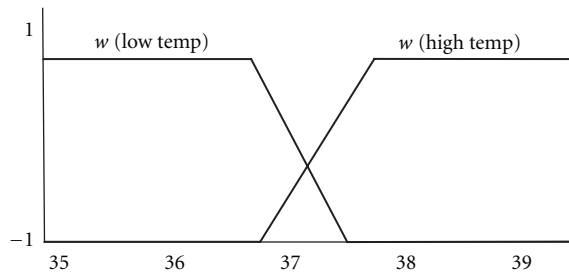


FIGURE 1: Body temperature discretized into two fuzzy intervals.

The values FL, CL, CU, and FU need not to be distinct; this allows creation of rectangular, trapezoidal, and triangular fuzzy intervals (Figure 1).

Rules are defined in the form

$$\text{IF } \textit{condition} \text{ THEN } \textit{conclusion} \text{ AND } \textit{action}, \quad (2)$$

where *condition* is the disjunctive form (disjunction of conjunctions) of literals (propositions or their negations), *conclusion* is a list of literals, and *action* is a list of actions (external programs). The set of rules can be understood as an inference network composed of nodes (propositions) and arcs (rules). Three types of propositions can be distinguished according to their position in the set of rules: questions are propositions only occurring in conditions, goals are propositions only occurring in conclusions, and intermediate propositions can occur both in conditions and conclusions. We distinguish three types of rules.

- (i) *Compositional*. Each literal in the conclusion has a weight that expresses the degree of uncertainty of the conclusion if the condition holds with certainty. The term compositional denotes the fact, that to evaluate the weight of a goal or intermediate proposition, all rules with this proposition in the conclusion are evaluated and combined.
- (ii) *Apriori*. Compositional rules without condition: these rules can be used to assign implicit weights to goals or intermediate propositions.
- (iii) *Logical*. Noncompositional rules without weights: only these rules can infer the conclusion with the weight true or false. One activated rule thus fully evaluates the conclusion; that is why we call these rules noncompositional.

A list of actions (external programs) can be associated with each rule. These programs are executed if the rule is activated.

*Integrity constraints* are used to express relations between literals that must hold after the inference in the set of rules. The syntax of an integrity constraint is

$$\textit{Ant} \implies \textit{Suc}(\textit{imp}), \quad (3)$$

where *Ant* and *Suc* are disjunctive forms of literals and *imp* is importance of the integrity constraint. Unlike in rules, questions can appear in succedent and goals in antecedent. In

this way integrity constraints can express relations between goals or questions. An example of such a constraint can be

$$\textit{Diagnosis}(\textit{TBC}) \implies \neg \textit{Diagnosis}(\textit{healthy}). \quad (4)$$

*Contexts* are disjunctive forms of literals that (having positive weight) determine the applicability of a rule or integrity constraint. Contexts are thus used to group rules or constraints into semantically related chunks. The contexts can speed up the consultation by pruning irrelevant parts of the knowledge base.

*Cases* are in the form of a vector of truth values (weights) assigned to all propositions (attribute-value pairs) describing a consultation (questions) as well as to propositions (attribute-value pairs) that define the possible results of a consultation (goals). Such representation allows the definition of distance and similarity between cases using functions operating on numeric values (in our case values from the interval  $[-1, 1]$ ). It is possible to assign to each attribute a number from the interval  $[0, 1]$  that expresses the importance of the respective attribute. These numbers are taken into account when evaluating the similarity between cases.

The knowledge base is stored as an XML file. There are several efforts to standardize the notation of rules using XML. Let us mention here PMML (<http://www.dmg.org/>) which defines models obtained during machine learning (regression models, decision trees, neural networks, Bayesian classifiers, and association rules), and RuleML (<http://www.ruleml.org/>) which is rule oriented. Unfortunately, neither of these formalisms can be used for our knowledge base because some important parts of the rules are not defined. Therefore we defined our own DTD from scratch; an example of XML syntax of the rule

$$\begin{aligned} &\text{IF } \textit{airways} \text{ AND } \textit{Body temperature (high)} \\ &\text{THEN } \textit{Diagnosis (Flu)}[3.00], \textit{Diagnosis (cold)}[-2.00] \end{aligned} \quad (5)$$

is shown in Figure 2.

### 3. Inference Mechanism

NEST can perform both rule-based reasoning and case-based reasoning. Similar to the approaches described in [14] or [16], both components can be run in parallel.

*3.1. Inference in the Network of Rules.* The inference using rules is usually based on deduction: if the conditions of a rule are true, then the rule is applicable (1) to infer the truth of the conclusion or (2) to perform the respective list of actions. As NEST is a diagnostic expert system, it performs inference of the first type. This is accomplished by processing the uncertainty that is related both to the rules in the knowledge base and the answers given by the user during consultation (see later). There are two standard ways to search for applicable rules, backward chaining, and forward chaining. Backward chaining starts from goals of the consultation and looks for rules that have these goals as their

```

<compositional_rule>
  <id>c7</id>
  <condition>
    <conjunction>
      <literal>
        <id_attribute>airways</id_attribute>
        <negation>0</negation>
      </literal>
      <literal>
        <id_attribute>temperature</id_attribute>
        <id_proposition>high</id_proposition>
        <negation>0</negation>
      </literal>
    </conjunction>
  </condition>
  <conclusions>
    <conclusion>
      <id_attribute>Diagnosis</id_attribute>
      <id_proposition>Flu</id_proposition>
      <negation>0</negation>
      <weight>3</weight>
    </conclusion>
    <conclusion>
      <id_attribute>Diagnosis</id_attribute>
      <id_proposition>Cold</id_proposition>
      <negation>0</negation>
      <weight>-2</weight>
    </conclusion>
  </conclusions>
</compositional_rule>

```

FIGURE 2: XML syntax of a rule.

conclusions; forward chaining starts with known facts and looks for rules that contain these facts in its conditions.

During consultation, NEST uses rules to compute weights of goals from the weights of answers to questions. This is accomplished by (1) selecting relevant rules during the current state of consultation and (2) applying the selected rules to infer the weight of its conclusion.

- (1) The selection of relevant rules can be done using either backward or forward chaining. The actual direction is determined by the user when selecting the consultation mode (see later).
- (2) For rules with weights (compositional and apriori ones), the system combines contributions of rules using a compositional approach as described in next subsection. For rules without weights, the system uses a noncompositional approach based on (crisp) modus ponens (to evaluate the weight of a conclusion, and (crisp) disjunction to evaluate a set of rules with the same conclusion.

The weights are propagated not only towards the actual goal but by using all rules applicable at a given moment [17].

3.2. *Inference Using Cases.* The inference cycle of a CBR system usually consists of four steps [18]:

- (i) retrieve the most similar case(s),
- (ii) reuse the case(s) to attempt to solve the problem,
- (iii) revise the proposed solution if necessary,
- (iv) retain the new solution as a part of a new case.

A new problem is matched against cases in the case base and one or more similar cases are retrieved. A solution suggested by the matching cases is then reused and tested for success. Unless the retrieved case is a close match the solution will probably have to be revised producing a new case that can be retained.

NEST performs this inference cycle in the following way:

- (i) find the most similar cases(s) by evaluating the similarity between the vector representing the weights assigned to answers of all questions for the current

consultation and the vectors representing the weights assigned to answers of all questions for the cases stored in the case base (the retrieve step);

- (ii) compute the weights of the goals for the current consultation from the weights of goals of the retrieved cases (the reuse and revise step). Similar to rule-based inference, the case bases inference in NEST can have two forms. For compositional inference the weights of goals for similar cases are combined (revised) to obtain the resulting weighs of goals, whereas for noncompositional (logical) inference the weights of the most similar case are just reused for the current problem,
- (iii) after finishing the consultation with the CBR component, the user can store the current consultation as a new case (the retain step).

**3.3. Uncertainty Processing in Rule-Based Reasoning.** Uncertainty processing in NEST is based on the algebraic theory of Hajek [19]. This theory generalizes the methods of uncertainty processing used in the early expert systems like MYCIN and PROSPECTOR. Algebraic theory assumes that the knowledge base is created by a set of rules in the form

$$\text{condition} \implies \text{conclusion}(\text{weight}), \quad (6)$$

where *condition* is conjunction of literals, *conclusion* is a single proposition, and *weight* from the interval  $[-1, 1]$  expressing the uncertainty of the rule.

During a consultation, all relevant rules are evaluated by combining their weights with the weights of their conditions. Weights of questions are obtained from the user, weights of all other propositions are computed by the inference mechanism. Four combination functions are defined to process the uncertainty in such a knowledge base:

- (1)  $NEG(w)$ —to compute the weight of the negation of a proposition,
- (2)  $CONJ(w_1, w_2, \dots, w_n)$ —to compute the weight of the conjunction of literals,
- (3)  $CTR(a, w)$ —to compute the contribution of the rule to the weight of the conclusion (this is computed from the weight of the rule  $w$  and the weight of the condition  $a$ ),
- (4)  $GLOB(w'_1, w'_2, \dots, w'_n)$ —to compose the contributions of more rules with the same conclusion.

Algebraic theory defines sets of axioms, the combination functions must fulfill. For example, the axioms for the function  $CTR$  are

$$CTR(1, w) = w$$

$$\text{for } a < 0 \quad CTR(a, w) = 0$$

$$\text{for } a_1 < a_2 \quad CTR(a_1, w) \leq CTR(a_2, w)$$

or the axioms for the function  $GLOB(w'_1, w'_2, \dots, w'_n) = w'_1 \oplus w'_2 \oplus \dots \oplus w'_n$  are

$$\oplus \text{ is commutative and associative,}$$

$$1 \oplus w = w \oplus 1 = 1, \text{ iff } w \in (-1, 1],$$

$$-1 \oplus w = w \oplus -1 = -1, \text{ iff } w \in [-1, 1),$$

$$0 \oplus w = w \oplus 0 = w, \text{ iff } w \in [-1, 1],$$

$$w \oplus -w = 0, \text{ iff } w \in (-1, 1),$$

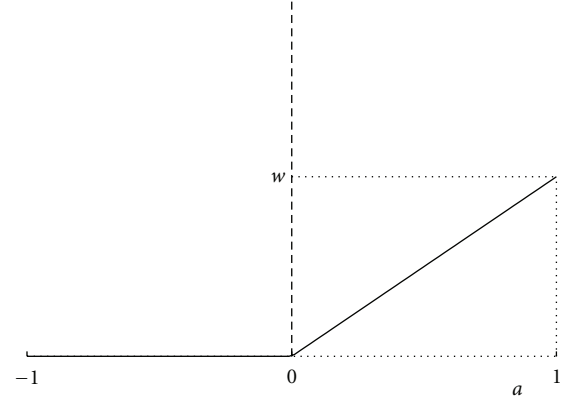


FIGURE 3: Function  $CTR$  for standard inference mechanism.

for any  $w_1, w_2, w_3 \in (-1, 1)$ , if  $w_1 < w_2$ , then  $w_1 \oplus w_3 \leq w_2 \oplus w_3$ .

Different sets of combination functions can thus be implemented. We call these sets “inference mechanisms”. The NEST system uses “standard”, “logical”, and “neural” inference mechanism. These mechanisms differ in the definition of the functions  $CTR$  and  $GLOB$ . The remaining functions are defined in the same way for all three mechanisms:

$$NEG(w) = -w,$$

$$CONJ(w_1, w_2) = \min(w_1, w_2),$$

$$DISJ(w_1, w_2) = \max(w_1, w_2).$$

**3.3.1. Standard Inference Mechanism.** The standard inference mechanism is based on the “classical” approach of MYCIN and PROSPECTOR expert systems. The contribution of a rule is computed MYCIN-like, that is,

$$w' = CTR(a, w) = a \cdot w \quad \text{for } a > 0; \quad (7)$$

this corresponds to the graph shown in Figure 3.

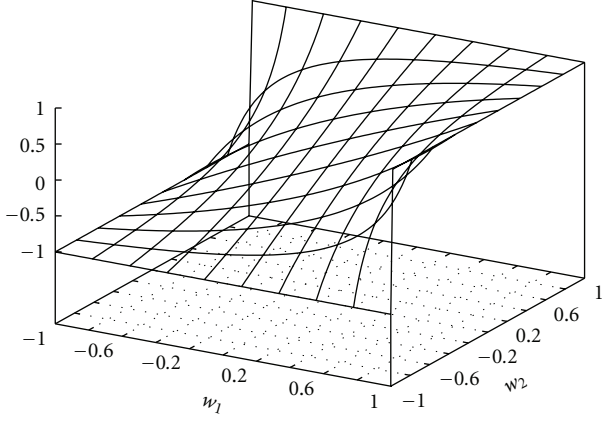
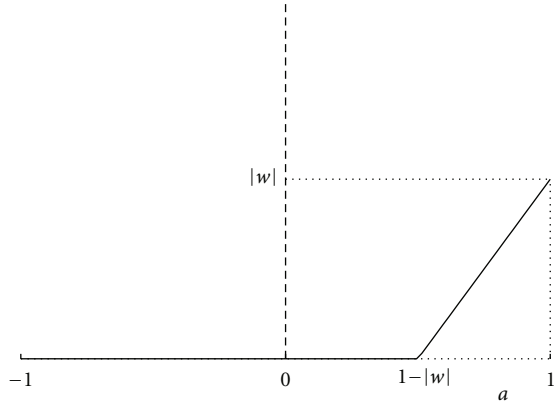
The combination of contributions of rules with the same conclusion is computed PROSPECTOR-like, that is,

$$GLOB(w'_1, w'_2) = \frac{w'_1 + w'_2}{1 + (w'_1 \cdot w'_2)}. \quad (8)$$

The graph of this function is shown at Figure 4. It is important to notice, that using this function:

- (i) the value  $GLOB(w'_1, w'_2, \dots, w'_n)$  equals to 1 (resp., -1) only if one of the weights  $w'_i$  is equal to 1 (resp., -1) and none of the other weights is equal to the opposite extreme value,
- (ii) once the value of  $GLOB(w'_1, w'_2, \dots, w'_n)$  equals to 1 (resp., -1), this value remains as is regardless of other weights that are eventually “added” to  $GLOB(w'_1, w'_2, \dots, w'_n)$ .

**3.3.2. Logical Inference Mechanism.** The logical inference mechanism is based on an application of the completeness theorem for Lukasiewicz many-valued logic. The task of the inference mechanism is to determine the degree in which

FIGURE 4: Function *GLOB* for standard inference mechanism.FIGURE 5: Function *CTR* for logical inference mechanism.

each goal logically follows from the set of rules (understood as a fuzzy axiomatic theory) and the user's answers during consultation [20]. This degree can be obtained by using the modus ponens inference rule:

$$\frac{\alpha, \alpha \Rightarrow \beta}{\beta} \left( \frac{x, y}{\max(0, x + y - 1)} \right). \quad (9)$$

So the contribution of a rule is computed as

$$w' = CTR(a, w) = \text{sign}(w) \cdot \max(0, a + |w| - 1) \quad (10)$$

for  $a > 0$ .

The graph of this function is shown in Figure 5.

To combine contributions of more rules, the logical inference mechanism uses the fuzzy disjunction

$$GLOB(w'_1, \dots, w'_n) = \min \left( 1, \sum_{w'_i > 0} w'_i \right) - \min \left( 1, \sum_{w'_i < 0} |w'_i| \right). \quad (11)$$

The graph of this function is shown in Figure 6. Unlike when using the "standard" version of the function *GLOB*, now

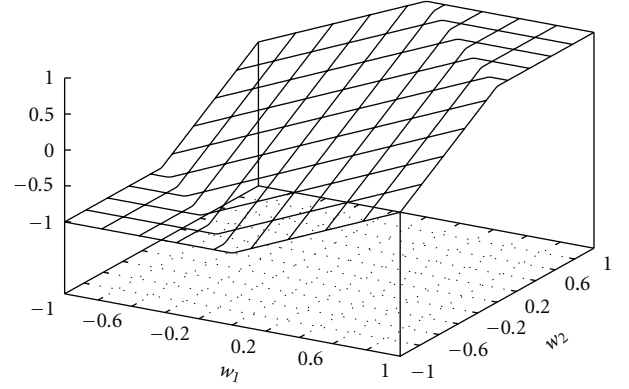
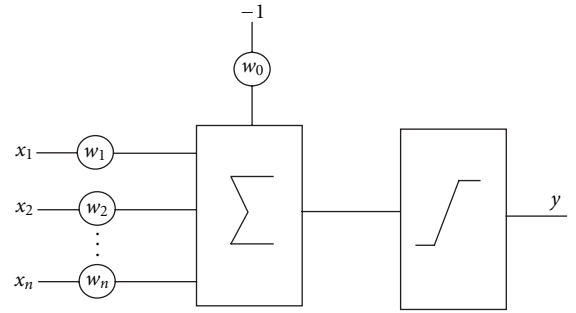
FIGURE 6: Function *GLOB* for logical inference mechanism.

FIGURE 7: Scheme of a linear neuron.

- (i) the value 1 (resp., -1) can be obtained by "summing up" a number of positive (resp., negative) contributions,
- (ii) once the value of  $GLOB(w'_1, w'_2, \dots, w'_n)$  is equal to 1 (resp., -1), a negative (resp., positive) contribution "added" to it will give as a result a non-extreme value.

**3.3.3. Neural Inference Mechanism.** The neural inference mechanism is based on an analogy with computations performed by neural networks [18]. Let us consider a single linear neuron as shown in Figure 7. The active dynamics of this neuron can be described by the following set of rules:

$$\begin{aligned} x_1 &\Rightarrow y(w_1) \\ \neg x_1 &\Rightarrow y(-w_1) \\ &\vdots \\ x_n &\Rightarrow y(w_n) \\ \neg x_n &\Rightarrow y(-w_n) \\ \text{True} &\Rightarrow y(w_0). \end{aligned} \quad (12)$$

To obtain results that correspond to the output of the neuron, the contribution of a rule to the weight of a conclusion is computed as the weighted input of the neuron

$$w'_i = CTR(x_i, w_i) = x_i \cdot w_i \quad \text{for } x_i > 0, \quad (13)$$

and the global effect of all rules with the conclusion  $y$  is computed as the piecewise linear transformation of the sum of weighted inputs

$$GLOB(w'_1, \dots, w'_n) = \min\left(1, \max\left(-1, \sum_{w'_i} w'_i\right)\right). \quad (14)$$

**3.3.4. Intervals of Weights.** Two different notions of a “not known” answer are introduced in NEST. The first notion, “irrelevant”, is expressed by the weight 0; this weight will prevent a rule having either a proposition or its negation in the conditional part from being applied. The second notion, “unknown”, is expressed by the weight interval  $[-1, 1]$ ; this weight interval is interpreted as “any weight”. Uncertainty processing has thus been extended to work with intervals of weights. The idea behind is to take into account all values from the interval in parallel. Due to the monotonicity of the combination functions, this can be done by taking into account the boundaries only in the following way:

$$NEG[w_1, w_2] = [NEG(w_2), NEG(w_1)],$$

$$CONJ([w_1, w_2], [v_1, v_2]) = [CONJ(w_1, v_1), \\ CONJ(w_2, v_2)],$$

$$DISJ([w_1, w_2], [v_1, v_2]) = [DISJ(w_1, v_1), DISJ(w_2, v_2)],$$

$$CTR([a_1, a_2], w) = [CTR(a_1, w), CTR(a_2, w)],$$

$$GLOB([w_1, w_2], [v_1, v_2]) = [GLOB(w_1, v_1), \\ GLOB(w_2, v_2)]. \quad (15)$$

**3.4. Uncertainty Processing in Case-Based Reasoning.** To properly use the CBR component, that is, to evaluate the similarity between cases and to revise the weights of goals, we have to define functions that will compute

- (i) similarity between weights of corresponding proposition (answer to a question) for current consultation and a case stored in the case base,
- (ii) similarity between the current consultation and a case from the case base,
- (iii) weights of goals for the current consultation from the weights of goals of retrieved cases.

All these numbers are from the interval  $[-1, 1]$ .

Analogously to the RBR component, we define different functions for compositional and noncompositional approaches. The main difference between the compositional and the noncompositional approach is that in the noncompositional approach, the weights of goals of the most similar

case are simply reused for the current consultation, while in the compositional approach, the weights of goals of all cases with similarity greater than 0 are revised to obtain the weights for the current consultation.

**3.4.1. Compositional Approach.** The function  $csimv$  that computes the similarity between two weights of corresponding proposition is defined as follows. Let  $v$  be a proposition (answer to a question) that has weight  $w_1$  in consultation  $c_1$  and weight  $w_2$  in consultation  $c_2$ , and let weights  $w_i$  be numbers from the interval  $[-1, 1]$ . Then

$$csimv(v_{c_1}, v_{c_2}) = 1 - |w_1 - w_2|. \quad (16)$$

The function  $csimc$  that computes the similarity between two consultations (cases) is defined as follows. Let  $c_1$  and  $c_2$  be two consultations. Let  $wa$  be the importance of the attribute  $a$  (if no value is given, the default is 1), let  $nva$  be the number of propositions derived from attribute  $a$ , let  $v_{kji}$  denote the  $i$ -th proposition (derived from attribute  $a$ ) in case  $c_j$ . Then

$$csimc(c_1, c_2) = \frac{\sum_{a \text{ is question}} (wa/nva) \sum_{i=1}^{pva} csimv(v_{c_1i}, v_{c_2i})}{\sum_{a \text{ is question}} wa}. \quad (17)$$

The weight  $w_g$  that is assigned to the goal proposition  $g$  of the current consultation  $c$  is computed as

$$w_g = \frac{\sum_i \max(0, csimc(c, c_i)) w_{gi}}{ncon_+}. \quad (18)$$

In the formula above  $c$  is the current consultation,  $c_i$  is a case from the case base,  $w_{gi}$  is the weight of the goal proposition  $g$  in the case  $c_i$ , and  $ncon_+$  is the number of cases  $c_i$  from the case base for which  $csimc(c, c_i) > 0$ . The weight  $w_g$  is thus computed as a weighted average of weights  $w_{gi}$  for all cases for which their similarity with the current consultation is positive.

**3.4.2. Logical Approach.** The function  $lsimv$  that computes the similarity between two weights of corresponding proposition is defined as follows. Let  $v$  be a proposition (answer to a question) that has weight  $w_1$  in consultation  $c_1$  and weight  $w_2$  in consultation  $c_2$ . Let weights  $w_i$  be numbers from the interval  $[-1, 1]$ . Let  $p$  be a given threshold,  $p > 0$ . Then

$$lsimv(c_1, c_2) = 1 \quad \text{for } csimv(c_1, c_2) > p, \\ lsimv(c_1, c_2) = -1 \quad \text{otherwise.} \quad (19)$$

Thus the similarity between the two weights can be only 1 or -1.

The function  $lsimc$  that computes the similarity between two consultations (cases) is defined in a similar way as the function  $csimc$ .

$$lsimc(c_1, c_2) = \frac{\sum_{a \text{ is question}} (wa/nva) \sum_{i=1}^{pva} lsimv(v_{c_1i}, v_{c_2i})}{\sum_{a \text{ is question}} wa}. \quad (20)$$

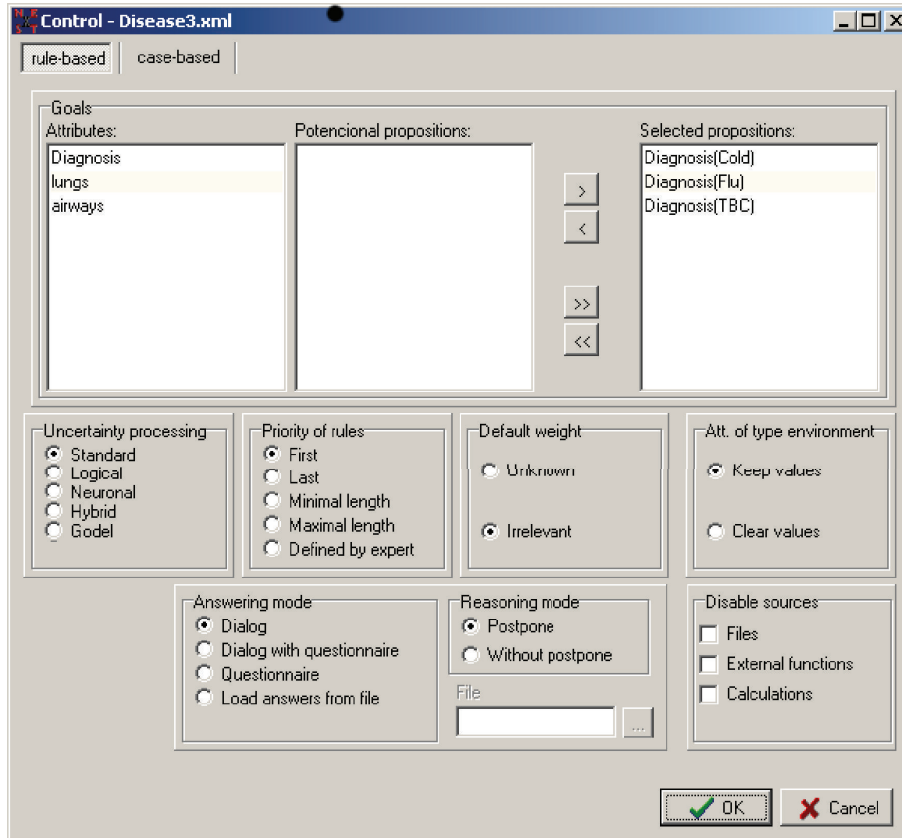


FIGURE 8: Initial settings for rule-based reasoning.

The weight  $w_g$  that is assigned to the goal proposition  $g$  of the current consultation  $c$  is computed as

$$w_g = w_{gi} \quad \text{where } i = \arg \max_j [l_{simc}(c, c_j)] \quad (21)$$

for  $l_{simc}(c, c_i) > 0$ ,

that is the weights of the most similar case are reused as weights for the goals for the current consultation.

**3.5. Evaluating Integrity Constraints.** Integrity constraints are evaluated after completion of the inference (either rule-based or case-based). An integrity constraint is violated, if the weight of *Ant* is greater than the weight of *Suc*. This is evaluated using the function *IMPL*:

$$IMPL(a, s) = \max(0, \min(1, a - s)) \quad \text{for } a > 0. \quad (22)$$

The result of this function (called the degree of violation) is then combined with the importance of the integrity constraint using the combination function *CTR* to compute the contribution of the constraint to the inconsistency of the consultation as  $CTR(IMPL(a, s), imp)$ .

#### 4. Consultation with the System

At this point, both RBR and CBR components can be used only independently. The user starts the work with NEST by

choosing one of these components. Figure 8 shows the initial screen for RBR mode and Figure 9 shows the initial screen for CBR mode. There is a number of parameters that can be set in order to control the reasoning process in both modes. The most important parameter is the answering mode. This parameter controls user interaction with the system.

Four answering modes (modes of consultation) are available for rule-based reasoning:

- (1) dialogue mode: classical question/answer mode in which the system selects the current question using backward chaining;
- (2) dialogue/questionnaire mode: the user can input some volunteer information (using a questionnaire), the system asks questions if needed during further consultation;
- (3) questionnaire mode: after filling in the questionnaire the system directly infers the goals from given answers using forward chaining;
- (4) load answers from a file: previously stored answers can be read into the system and changed using the questionnaire.

In each of these modes, the user answers questions concerning the input attributes. According to the type of attribute, the user gives the weight (for binary attributes), the value and its weight (for single nominal attributes), list of values and



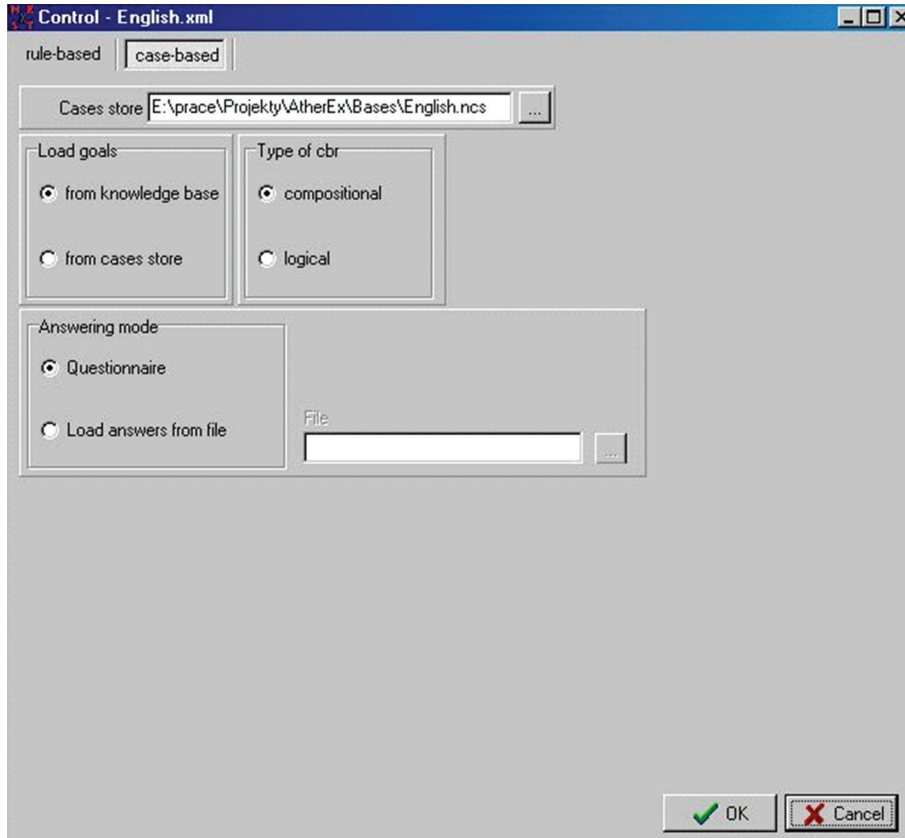


FIGURE 9: Initial settings for case-based reasoning.

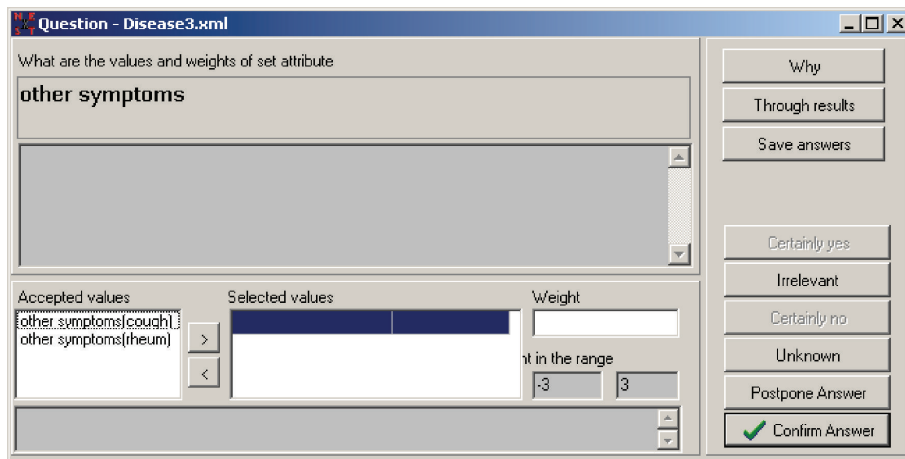


FIGURE 10: Example of a question in the dialogue mode.

their weights (for multiple nominal attributes), or the value (for numeric attributes). Questions not answered during consultation get the default answer “unknown” (the interval  $[-1, 1]$ ) or “irrelevant” (the interval  $[0, 0]$ ). Answers can be postponed and the user can return to them after finishing the consultation. Figure 10 shows the screenshot of a question concerning a multiple nominal attribute as displayed in the dialogue mode.

For CBR, only the questionnaire mode and input from a file are available. Figure 11 illustrates an example of a

questionnaire that shows answers loaded from a file (the “load answers from a file” mode).

The result of the consultation is shown for both RBR and CBR components in the same way, as a list of goals together with their weights (Figure 12). Alternatively, the user can display weights of all propositions, can change answers (for this option the list of answers is shown using a questionnaire), and can save the answers (the corresponding file can be loaded into the other component). NEST can also explain the results of the inference by allowing a user to

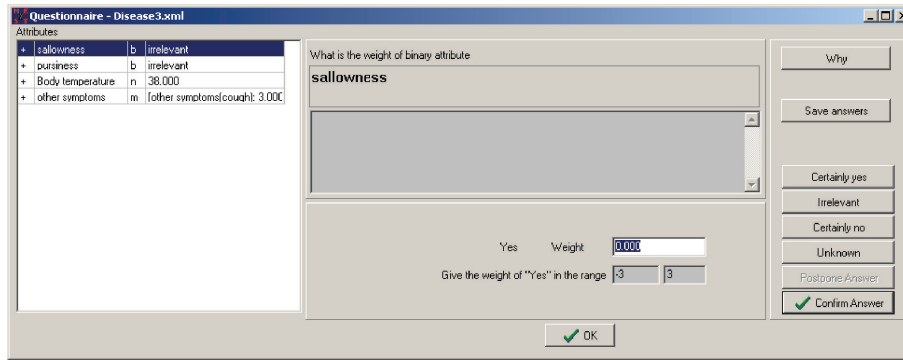


FIGURE 11: A questionnaire showing answers loaded from a file.

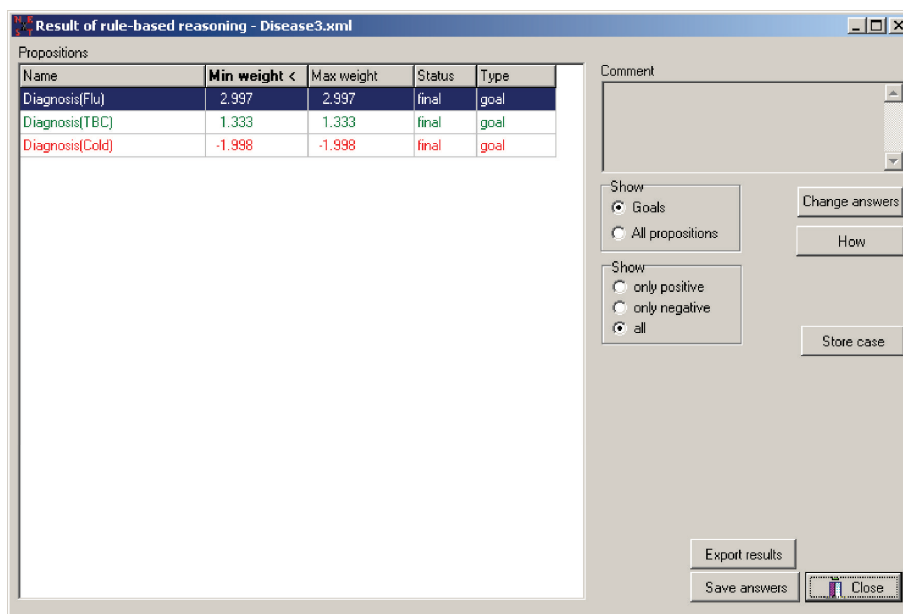


FIGURE 12: A screenshot showing results of a consultation.

inspect the part of the knowledge that was used to infer the goals using the “how” button. The “how” option shows the activated rules for the RBR component (Figure 13) and the used cases for the CBR component (Figure 14).

## 5. Implementation Details

Two versions of NEST have been implemented at this stage, a stand-alone, version and a web-based client-server version. The screenshots shown in the previous section were taken from the stand-alone version. Both versions are implemented in Borland Delphi (version 7.0) for PCs running Windows (Win95 and higher). The stand-alone version is implemented as a single .exe file, and the client-server version is implemented as a web server that can use any web browser as a client.

The main difference between the stand-alone and web-based client-server versions is that the stand-alone version

allows us to set a number of parameters (to fine-tune the inference process) while the client-server version assumes these parameters to be fixed. This makes the stand-alone version suitable for development and testing of the knowledge base while the client-server version is preferable for deployment of a fully developed application. An interesting feature of the web-based version is the possibility to modify the layout of dialogue pages the web browser displays during consultation. The layout of the pages (HTML code) is stored in a separate folder for each knowledge base. This code, which can be completely modified by the knowledge base administrator, contains a kind of “pseudo-tags”; a part which is used by the NEST server to input or output values considered important for the consultation. The knowledge base administrator can thus further hide details of the inference process the user does not need to know.

Figures 15 through 17 show the difference between the layouts of the stand-alone and client-server versions. Figure 15 shows a question as displayed using the dialogue

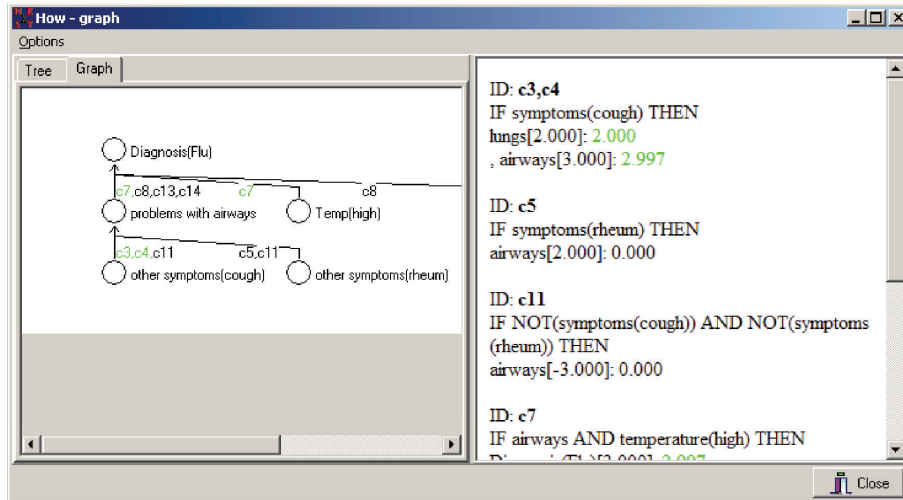


FIGURE 13: Explanation “how” for the RBR component.

mode in the stand-alone version, Figure 16 shows the same question as displayed using standard layout in the client-server version and Figure 17 shows this question as displayed using a layout prepared for a particular application.

To support a knowledge engineer during knowledge base encoding, the knowledge base editor NestEd has also been implemented. This windows-based program lets the user fill in the XML tags with values and also performs some syntax checking. A screenshot of the window that is used to edit rules is shown in Figure 19.

The current implementation is localized into the Czech and English languages. Since all text messages are stored in separate files, providing another language version is just a question of adding a new text file. The current version of the system can be downloaded from <http://sorry.vse.cz/~berka/NEST/>.

## 6. Example Application

The screenshots shown in Figures 15 through 18 are taken from one of our real-life applications, a system developed to assess the risk of atherosclerosis of a person on the basis of his lifestyle, personal, and family history. The goal of the system is to classify a person into one of the four groups w.r.t. level of atherosclerosis risk (the screenshots at Figures 18 and 20 show the difference between the final screen of the stand-alone and tailored client-server version). We have build the knowledge base in two steps. At first, we created the initial set of rules from data using machine learning. The data used in the learning step were collected during an extensive epidemiological study of atherosclerosis primary prevention. The study included data of more than 1400 middle-aged men living in a Prague district (Prague 2). The set of rules obtained during this first step was then refined according to suggestions by a domain expert. When testing the RBR component on 75 testing examples, we obtained an overall accuracy of 0.67, accuracy of 0.50 for the nonrisk group, and an accuracy of 0.88 for other groups (when compared with

the decisions made by the expert). The most severe errors (those causing the relatively low level of accuracy for the nonrisk group) were misclassifications of high-risk patients as nonrisk ones. When analyzing these errors we found that this usually happened for patients whose data were outside of the scope of the data used to build the rule base (the data used to build the rule base were collected from middle-aged men and the wrongly classified patients were either too old or too young). After taking this into account by adding respective cases into the case base and considering CBR prior to RBR (to deal with such “exceptions”), we further improved the classification accuracy. The results obtained by our approach were also compared with the results obtained from so-called CVD risk calculators. These calculators usually ask questions about lifestyle and about results of various examinations and laboratory tests to compute an overall risk (expressed as a weighted sum of used risk factors) that a given person will suffer from cardiovascular disease (CVD) in 10 years. Three popular CVD risk calculators, Risk Assessment Tool [21], PROCAM Risk calculator [22], and Heart Score [23] were involved in this comparison. Table 1 summarizes the classification accuracy of the compared systems (here the column “risk” refers to low-risk, medium-risk, and high-risk groups and the column “no risk” refers to the nonrisk group). The results of NEST, where we used only knowledge about lifestyle, personal, and family history, were comparable with the results of the tested CVD calculators that rely on a number of laboratory tests. Anyway, none of the systems made reliable classifications of nonrisk patients. A detailed description of this application can be found in [24].

## 7. Conclusion

In the design of NEST, we attempted to partially overcome the problem that represented the most severe hindrance to compositional system deployment: limited expressiveness of proposition-rule networks for real-world modeling purposes. Compared to traditional systems, we added more

Number	Similarity	sallowness	pursiness	temperature	symptoms	Diagnosis
New case		1.000	-1.000	38.000	[other sympt]	Irrelevant
1	-0.438	3.000	3.000	36.500	[other sympt]	[Diagnosis(TBC): 2.229][Diagnosis(Col
2	1.000	-3.000	-3.000	38.000	[other sympt]	[Diagnosis(TBC): 0.390][Diagnosis(Col
3	-0.125	-3.000	-1.500	36.500	[other sympt]	[Diagnosis(TBC): 0.390][Diagnosis(Col

FIGURE 14: Explanation “how” for the CBR component.

What is the weight of binary attribute

**Do you have hypertension?**

Hypertension = high level of blood pressure. Hypertension is considered for values greater than 140/90 mm Hg, when suffering from diabetes melitus for values greater than 130/85. Attention: If taking pills to decrease the blood pressure answer "certainly yes" regardless on the current blood preasure values.

Yes      Weight     

Give the weight of "Yes" in the range -1      1

- Why
- Through results
- Save answers
- Certainly yes
- Irrelevant
- Certainly no
- Unknown
- Postpone Answer
- Confirm Answer

FIGURE 15: A binary question displayed in the stand-alone version.

## NEST - New Expert SysTem

[End consultation](#)

### Question

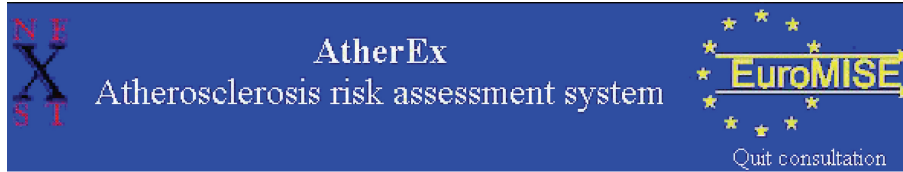
What is the weight of binary attribute

#### Do you have hypertension?

Hypertension = high level of blood pressure. Hypertension is considered for values greater than 140/90 mm Hg, when suffering from diabetes melitus for values greater than 130/85. Attention: If taking pills to decrease the blood preasure answer "certainly yes" regardless on the current blood preasure values.

Weight  Give the weight for "yes" in the range -1,1

FIGURE 16: A binary question displayed in the standard layout of the client-server version.



**Do you have hypertension?**

Hypertension = high level of blood pressure. Hypertension is considered for values greater than 140/90 mm Hg, when suffering from diabetes melitus for values greater than 130/85. Attention: If taking pills to decrease the blood pressure answer "certainly yes" regardless on the current blood pressure values.

FIGURE 17: A binary question displayed in the tailored layout of the client-server version.

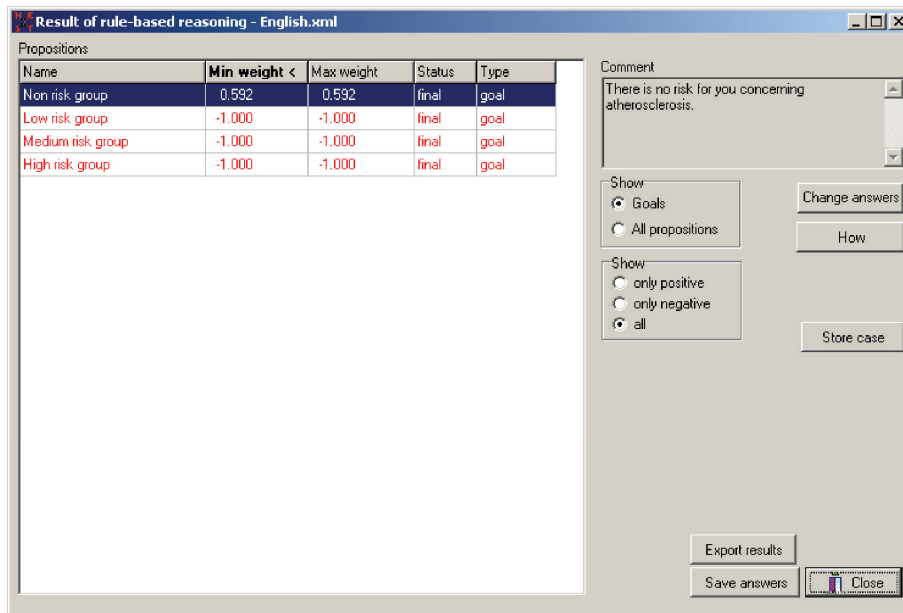


FIGURE 18: Results of consultation displayed in the stand-alone version.

TABLE 1: Classification accuracy.

System	Overall accuracy	Accuracy risk	Accuracy no risk
NIH	0.76	0.84	0.63
ProCam	0.69	0.95	0.52
Heart	0.67	1.00	0.50
NEST RBR	0.67	0.88	0.50
NEST CBR + RBR	0.73	0.89	0.56

flexibility to the logical formulae used in rule condition and conclusion (disjunctive forms) and also introduced integrity

constraints allowing detection of inconsistent weight patterns. To allow flexible incorporation of uncertainty-free portions of a knowledge base (e.g., legal knowledge), distinct logical rules (in the sense of crisp propositional logic) can be specified.

Due to the flexibility of the inference mechanism, NEST covers the functionality of classic compositional rule-based expert systems, noncompositional (prolog-like) expert systems, and case-based reasoning systems. The knowledge base developer can thus deploy the RBR component in different scenarios: pure compositional, pure noncompositional, or a combination of both. In addition, the CBR component can

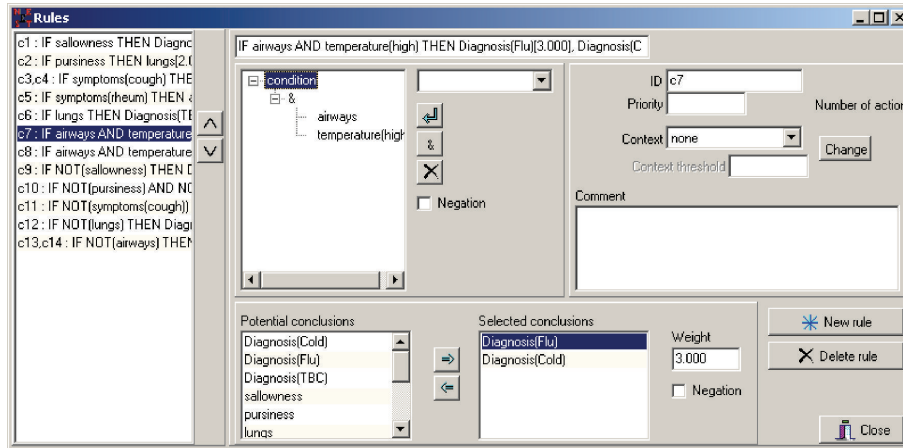
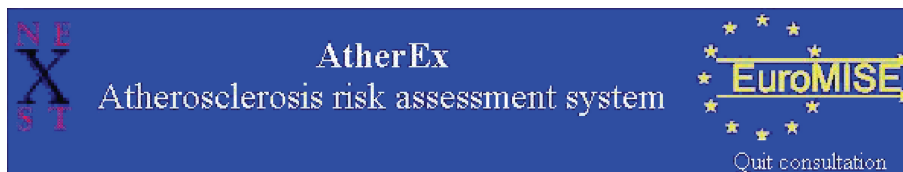


FIGURE 19: Editing rules.



## Results

You belong to the group **Non risk group** Recommendation **There is no risk for you concerning atherosclerosis.**

If you see all resulting groups, you probably answered the questions too uncertain. To get more specific results, please be more concrete.

You can inspect or change your answers, that resulted in this recommendation.

Quit consultation

FIGURE 20: Results of consultation displayed in the tailored layout of the client-server version.

be used in two ways: compositional or noncompositional. By now, the rule-based and the case-based components have to be invoked manually. In our planned deeper integration we intend to implement a scenario in which a consultation will start with case-based reasoning and the rule-based reasoning will be invoked only if the CBR component cannot retrieve sufficiently similar cases. In our opinion this scenario better reflects the human problem-solving process where a person first tries to recall a similar situation solved in the past and tries to apply some general rules only if such a situation does not exist. We experimentally tested a manual version of this scenario on an application from the atherosclerosis risk domain and found out that integrating RBR and CBR reasoning can produce better results than those obtained only from RBR.

## Acknowledgments

The research reported in this paper is supported by Grant MSM6138439910 (from the Ministry of Education of the Czech Republic) and by Grant GACR 201/08/0802 (from the Grant Agency of the Czech Republic).

## References

- [1] E. A. Feigenbaum, "Themes and case studies of knowledge engineering," in *Expert Systems in the Micro Electronic Age*, D. Michie, Ed., Edinburgh University Press, Edinburgh, UK, 1979.
- [2] E. H. Shortliffe, *Computer-Based Medical Consultations: MYCIN*, Elsevier, New York, NY, USA, 1976.

- [3] R. O. Duda and J. E. Gasching, "Model design in the prospector consultant system for mineral exploration," in *Expert Systems in the Micro Electronic Age*, D. Michie, Ed., Edinburgh University Press, Edinburgh, UK, 1979.
- [4] R. E. Neapolitan, *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*, John Wiley & Sons, New York, NY, USA, 1990.
- [5] J. L. Kolodner, "An introduction to case-based reasoning," *Artificial Intelligence Review*, vol. 6, no. 1, pp. 3–34, 1992.
- [6] A. R. Golding and P. S. Rosenbloom, "Improving accuracy by combining rule-based and case-based reasoning," *Artificial Intelligence*, vol. 87, no. 1-2, pp. 215–254, 1996.
- [7] E. L. Rissland and D. B. Skalak, "Combining case-based and rule-based reasoning: a heuristic approach," in *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI '89)*, pp. 524–530, Morgan Kaufmann, 1989.
- [8] R. H. Chi and M. Y. Kiang, "An integrated approach of rule-based and case-based reasoning for decision support," in *Proceedings of the 19th Annual Conference on Computer Science (CSC '91)*, pp. 255–267, 1991.
- [9] S. Montani and R. Bellazzi, "Integrating case based and rule based reasoning in a decision support system: evaluation with simulated patients," in *Proceedings of the AMIA Symposium*, N. Lorenzi, Ed., pp. 887–891, Hanley & Belfors, 1999.
- [10] J. Prentzas and I. Hatzilygeroudis, "Integrating hybrid rule-based with case-based reasoning," in *Proceedings of the 6th European Conference on Advances in Case-Based Reasoning (ECCBR '02)*, S. Craw and A. Preece, Eds., pp. 336–349, Springer, 2002.
- [11] W. Shi and J. A. Barnden, "How to combine CBR and RBR for diagnosing multiple medical disorder cases," in *Proceedings of the 6th International Conference on Case-Based Reasoning (ICCBR '05)*, H. Munoz-Avila and F. Ricci, Eds., vol. 3620 of *Lecture Notes in Computer Science*, pp. 477–491, Springer, 2005.
- [12] W. M. Wang, C. F. Cheung, W. B. Lee, and S. K. Kwok, "Knowledge-based treatment planning for adolescent early intervention of mental healthcare: a hybrid case-based reasoning approach," *Expert Systems*, vol. 24, no. 4, pp. 232–251, 2007.
- [13] G. H. Lee, "Rule-based and case-based reasoning approach for internal audit of bank," *Knowledge-Based Systems*, vol. 21, no. 2, pp. 140–147, 2008.
- [14] M. Yang and Q. Shen, "Reinforcing fuzzy rule-based diagnosis of turbomachines with case-based reasoning," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 12, no. 2, pp. 173–181, 2008.
- [15] K. A. Kumar, Y. Singh, and S. Sanyal, "Hybrid approach using case-based reasoning and rule-based reasoning for domain independent clinical decision support in ICU," *Expert Systems with Applications*, vol. 36, no. 1, pp. 65–71, 2009.
- [16] M. M. Cabrera and E. O. Edye, "Integration of rule based expert systems and case based reasoning in an acute bacterial meningitis clinical decision support system," *International Journal of Computer Science and Information Security*, vol. 7, no. 2, pp. 112–118, 2010.
- [17] P. Berka, V. Laš, and V. Svátek, "NEST: re-engineering the compositional approach to rule-based inference," *Neural Network World*, vol. 14, no. 5, pp. 367–379, 2004.
- [18] A. Aamodt and E. Plaza, "Case-based reasoning: foundational issues, methodological variations, and system approaches," *AI Communications*, vol. 7, no. 1, pp. 39–59, 1994.
- [19] P. Hajek, "Combining functions for certainty degrees in consulting systems," *International Journal of Man-Machine Studies*, vol. 22, no. 1, pp. 59–76, 1985.
- [20] P. Berka, J. Ferjenčík, and J. Ivánek, "Expert system shell SAK based on complete many-valued logic and its application in territorial planning," in *Fuzzy Approach to Reasoning and Decision Making*, V. Novák et al., Ed., pp. 67–74, Academia, Prague and Kluwer Academic Publishers, Dodrecht, The Netherlands, 1992.
- [21] "NCEP APT III system," <http://hp2010.nhlbihin.net/atp/iii/calculator.asp?usertype=prof>.
- [22] "PROCAM Risk Calculator," [http://www.chd-taskforce.de/procam\\_interactive.html](http://www.chd-taskforce.de/procam_interactive.html).
- [23] "Heart Score," <http://www.heartscore.org/Pages/welcome.aspx>.
- [24] P. Berka and M. Tomečková, "Atherosclerosis risk assessment using rule-based approach," in *Advances in Data Management*, Z. Ras and A. Dardzinska, Eds., pp. 333–350, Springer, Berlin, Germany, 2009.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

