ELSEVIER
Artificial Intelligence in Medicine 12 (1998) 25–41
Artificial Intelligence in Medicine

Case-based reasoning for medical decision support tasks: The Inreca approach

Klaus-Dieter Althoff [a,*], Ralph Bergmann [a], Stefan Wess [a], Michel Manago [b], Eric Auriol [b], Oleg I. Larichev [c], Alexander Bolotov [c], Yurii I. Zhuravlev [d], Serge I. Gurov [d]

[a] *Centre for Learning Systems and Applications, Dept. of Computer Science, University of Kaiserslautern, P.O. Box 3049, D-67653 Kaiserslautern, Germany*
[b] *AcknoSoft, 58a, rue du Dessous-des-Berges, F-75013 Paris, France*
[c] *Institute for Systems Analysis, Russian Academy of Sciences, 9 pr. 60-let Octiabrya, 117312 Moscow, Russia*
[d] *Russian Academy of Sciences, Computing Center, ul. Vavilova, d. 40, 117967 Moscow, Russia*

Received 31 August 1996; received in revised form 30 September 1996; accepted 15 June 1997

Abstract

We describe an approach for developing knowledge-based medical decision support systems based on the new technology of case-based reasoning. This work is based on the results of the Inreca European project and preliminary results from the Inreca+ project which mainly deals with medical applications. One goal was to start from case-based reasoning technology for technical diagnosis and 'scale-up' to more general non-technical decision support tasks as typically given in medical domains. Inreca technology has been used to build an initial decision support system at the Russian Toxicology Information and Advisory Center in Moscow for diagnosing poison cases caused by psychotropes. © 1998 Elsevier Science B.V.

Keywords: Case-based reasoning; Induction; Medical decision support; Toxicology domain

* Corresponding author. Tel.: +49 631 2053362; fax: +49 631 2053357; e-mail: {althoff,bergmann}@informatik.uni-kl.de

0933-3657/98/$19.00
PII S0933-3657(97)00038-9

1. Introduction

In this article we describe an approach for developing knowledge-based medical decision support systems based on the new technology of *case-based reasoning* (CBR). CBR is an approach for solving problems based on solutions of similar past cases [1,19]. A *case* consists at least of a problem description (e.g. symptoms) and a solution (e.g. a diagnosis or a therapy). Cases are stored in a database of cases called a *case base*. To solve an actual problem a notion of *similarity* between problems is used to *retrieve* similar cases from the case base. The solutions of these similar cases are then used as starting points for solving the actual problem.

While in technical domains CBR systems for decision support and diagnostic problem solving are already in daily use in many companies [4], this is not so true for medical decision support tasks. Well known examples from the literature include using CBR for expertise relocation in developing countries [27], using CBR for intelligent retrieval of radiology images [22], integrating various reasoning capacities of physicians like logical, deductive, uncertain and analogical reasoning [24], trading case-specific and model-based knowledge in bone healing [35], utilizing CBR for trend prognosing for medical problems [33], automated image interpretation of myocardial perfusion scintigrams [18], and using CBR for selecting of an antibiotics therapy [34]. The GS.52 system [16] is an example that is in practical use as a real-life application; it uses CBR to address the domain of dismorphic syndromes.

The work reported here is based on the results of the Inreca ('Induction and Reasoning from Cases') Esprit project of the European Union[1] and preliminary results from the Inreca+ ('Integrating Induction and Case-Based Reasoning for Diagnostic Problems with Focus on Medical Domains') project[2] as well as on results on CBR methodology development from the project WiMo (Knowledge Acquisition and Modeling for Case-Based Learning, funded by Stiftung Rheinland-Pfalz für Innovation). All three projects aim at developing information technology to build systems to solve diagnosis and identification problems by using past history, where Inreca+ especially focuses on medical problems.

In this article we address the topic of developing case-based decision support systems for diagnosing intoxications caused by administering drugs. The main goals are to reduce the time required to come to a decision, particularly, in an emergency case, to compensate for lack of experience of young medical staff, and to distribute available experience to different sites.

Such a system would have potential use in the Russian Toxicology Information and Advisory Center in Moscow, in many Russian hospital ambulances, and a

[1] The partners of the Inreca-project are AcknoSoft (France; prime contractor), tecInno (Germany), Irish Medical Systems (Ireland), and the University of Kaiserslautern (Germany).

[2] The partners of the Inreca+ project are AcknoSoft (France; prime contractor), University of Kaiserslautern, Institute of Mathematics (Moldova), Reliable Software (Belarus), All-Russian Institute for Scientific and Technical Information (Russia), and Russian Academy of Sciences (Russia).

number of toxicology centers that still need to be built at central places in Russia. In addition, a case-based expert system could also be useful for urgent ambulance consultations in many European countries (a.o.). It is known that every year Russia has more intoxication cases than any other country in Europe. Therefore it is reasonable to use valuable experience of the best Russian toxicologists.

2. Requirements for a medical decision support systems

We will first describe a concrete subject area from the field of medical decision support applications. This domain will be used for illustrating the requirements derived afterwards.

2.1. A decision support system for diagnosing poison cases caused by psychotropes

As a main subject domain we present a special problem of medical diagnosis: decision support for therapy selection in case of intoxication with drugs, in particular with psychotropes. This requires a fast diagnosis of a practical situation and a choice of therapy. The reason for selecting this clinical discipline is that here the 'information problem' comes to the forefront, where the major difficulty in treatment of patients with acute poisoning is usually poison identification in the patient's organism and the circumstances of exposure (dose, routes of administration, time of injection, etc.). The goal is to create a decision support system which is useful for the following purposes:

- To be used by an ambulance physician in the cases of an intoxication by medicines.
- To be used by physicians whose specialty lies outside the domain of toxicology[3]. Having received the clinical symptoms of an intoxicated patient as input, the system identifies the substance taken and provides a necessary course of actions.

There appears also to be an additional use even for domain experts in two directions. First a decision support system can confirm the experts' decision while they work in the call-center (one expert usually processes 10–30 poison cases per day, where 3–5 of these cases are complex decisions[4]). Second the respective expert can extend his knowledge in the parts of the toxicology domain he is not familiar with (even the best expert cannot know the whole domain).

As part of the Inreca + project, a concrete CBR application was built based on actual case data on acute poisoning collected by the Toxicology Information and Advisory Center of the Russian Federation Ministry of Health and Medical Industry. Table 1 shows the different eight types of medicines that are considered in this application.

[3] The information problem is always present because physicians simply cannot learn all the types of existing poisons.

[4] The most difficult problems correspond to a combination of different types of poison and also for the background of alcohol.

Table 2 shows a list of 86 parameters that have been identified as important for this diagnosis task. We do not include in the clinical information list any type of laboratory data analysis because, as a rule, when a toxicology laboratory is available for a physician, and there is enough time to carry out the analysis, it provides the correct diagnosis and the computer assistant is not needed.

2.2. Requirements

In the following we describe a number of requirements that arise for: (i) (medical) decision support problems; and (ii) a decision support system in the toxicology domain.

2.2.1. Short response time

One of the most important requirements for a (critical) medical decision support system is the response time. The system must be able to present a diagnosis/therapy based on the observed symptoms within less than a minute. Only a fast problem solution enables the physician to immediately start a therapy which might be crucial for saving the patient's life. This is particularly true for toxicological cases. The initial therapeutic action must be as adequate as possible which requires a quick and accurate prediction of the toxic cause without waiting for the toxicological analysis [23].

2.2.2. Justifiability of results

A result (e.g. a selected therapy) presented by the system must be justified by the system in a way that a physician can validate the outcome and judge its accuracy. The justification presented by the system should be in a form the physician is familiar with in order to facilitate a speedy decision.

2.2.3. Dealing with incomplete information

Incomplete information is a basic characteristic of medical domains. Often the value of certain attributes cannot be acquired because the information is simply not available at the time of diagnosis or the required tests would take too long, are too risky, or are too expensive. This is also particularly true in the toxicology domain.

Table 1
Types of drugs considered

1	Ethanol
2	Barbiturates
3	Methanol
4	Amynotryptelene
5	Malathion
6	Acetic acid
7	Parathion
8	Dichloroethane

Table 2
Attributes considered in the toxicology domain

No.	Attribute	No.	Attribute
2	Age	279	Pain in epigastrium
3	Sex	282	Sickness
8	Beds (days)	287	Vomiting
11	Outcome	291	Coffee-ground vomit
179	Respiration rate	293	Toxic gastroenteritis
230	Systolic BP	343	Clear consciousness
231	Diastolic AP	344	Impaired consciousness
232	Pulse AP	345	Mental confusion
238	Pulse on admission	346	Loss of consciousness
239	Tachycardia	347	General weakness
1366	Type of poison	348	Lethargy
13	Death	349	Retardeness
46	Satisfactory state	350	Flabbiness
47	Medium severity state	351	Excitement
48	Severe state	352	Psychic excitement
60	Pale dermal cover	353	Motor excitement
62	Dermal hyperemia	355	Inadequate consciousness
63	Dermal cyanosis	356	Adynamia
64	Cyanosis of nasolabial triangle	358	Sopor
65	Dermal acrocyanosis	360	Coma
74	Cold ski	361	Deep coma
75	Sweating	362	Initial coma
106	Hyperemia of oral cavity	366	Coma duration
113	Edema of oral cavity	373	Blurred speech
146	Thorax muscle rigidity	374	Poorly responsive
149	Spontaneous myofibrillations	375	Nonresponsive
151	Hand tremor	376	Disoriented in space
153	Asthenia	377	Disoriented in time
165	Reflexes present	378	Noncritical
168	Disturbed respiration	379	Headache
169	Disturbed aspiration-obturation	380	Dizziness
173	Rigid respiration	400	Medium-size pupils
189	Rale	401	Meiosis
191	Dry rale	403	Medriasis
200	Bronchorrhea	415	Live photoreaction of pupils
205	Pulmonaty hperhydration	416	Zero photoreaction of pupils
217	Bronchopneumonia	420	Preserved pain reaction
237	Collaps (AP < 90 mmHg)	421	Reduced pain reaction
252	Decompressed shock	422	Zero pain reaction
256	Mouth odour	423	Cough reflex preserved
257	Alcohol odour	426	Drunken man behaviour
267	Pain during esophagus palpation	1034	Torpor
268	Pain during swallowing	1061	Salivation

2.2.4. Dealing with measured values and conceptual terms

In ‘medical domains’, the attributes that describe a patient usually contain values that are the result of some measurement (e.g. the systolic blood pressure) and values that describe some conceptual terms (e.g. 'impaired consciousness'). While measured values are usually represented as numeric values (e.g. 120 mmHg), conceptual terms are usually binary or n-ary features.

3. The solution: the Inreca approach

We now briefly describe an approach to building decision support systems which reason from cases. The employed application areas demonstrate the requirements mentioned above.

Generally accepted characteristics of CBR systems already meet some of the above mentioned requirements. CBR systems are in principal able to deal with incomplete information (e.g. unknown attribute values), make use of vague relationships by means of similarity measures, as well as to allow numeric and symbolic attributes. However, CBR approaches that fulfill all these requirements together tend to lose the ability of efficient retrieval as the case base grows. The reason for this is that case-based systems typically *interpret* the specific knowledge contained in all the cases at run time, i.e. during the consultation of the system. The more cases present, the more computational effort must be spent on their interpretation. This is a major problem when applying CBR to real-life applications and makes it difficult to achieve the requirement 'short response time'. Naturally, this is a very critical aspect with respect to user acceptance.

The Inreca approach presents a successful solution to this important problem. The Inreca system allows *compilation* of the specific knowledge contained in the cases into more general rules that can be efficiently evaluated thus reducing the system consultation time. This approach can be viewed as an integration [9] between classical CBR and inductive machine learning approaches [26].

3.1. Cases and similarity measures

In a case-based decision support system, a case describes a past situation in which a particular decision was taken. In medical domains, a case contains a description of the symptoms observed during examination of a patient as well as the diagnosis or the treatment that was identified, e.g. by a physician. The diagnosis that was identified in a particular case is also turned into an attribute (called *target attribute*).

When a new problem must be solved, e.g. a patient with an unknown intoxication must be diagnosed, some of the symptoms must be checked and noted as a new problem case. The CBR process then proceeds by searching for the most similar known cases from the case base. For this purpose, the similarity between two cases (the problem case and the respective case in the case base) must be defined through the similarity of the attributes used in the case representation (except for the target attribute). A very common approach also used in Inreca is to define similarity through a *similarity measure* $SIM(X,Y)$, which is a function that assigns a pair of cases X,Y a real-valued number out of the range [0. .1]. A high value represents a high similarity between X and Y.

3.2. The Inreca-Tree: a data structure for indexing cases

The efficiency of case retrieval is of high importance and a serious problem if a case base has reached a considerable size. The core idea behind our approach is a new indexing structure we call *Inreca-Tree*. This indexing structure is based on the concept of the k-d tree [14], which is a multi-dimensional binary search tree. The Inreca-Tree is an n-ary tree in which the branches represent constraints for certain attributes of the cases. Since we need to handle ordered and unordered value ranges as well as unknown attribute values, we introduce different types of branches.

Every node within the tree represents a subset of the case base and the root node represents the whole case base. Every inner node partitions the represented case set into disjoint subsets. The leaves of an Inreca-Tree (we call them *buckets* as in a k-d tree) contain all cases that fulfill all constraints that occur in the path from the root of the tree to the respective leaf. Fig. 1 shows an example of an Inreca-Tree for the toxicology domain. The top of the tree shows a branch node for the attribute ‘Pulse’, which holds values from the ordered type ‘integer’. This node partitions the

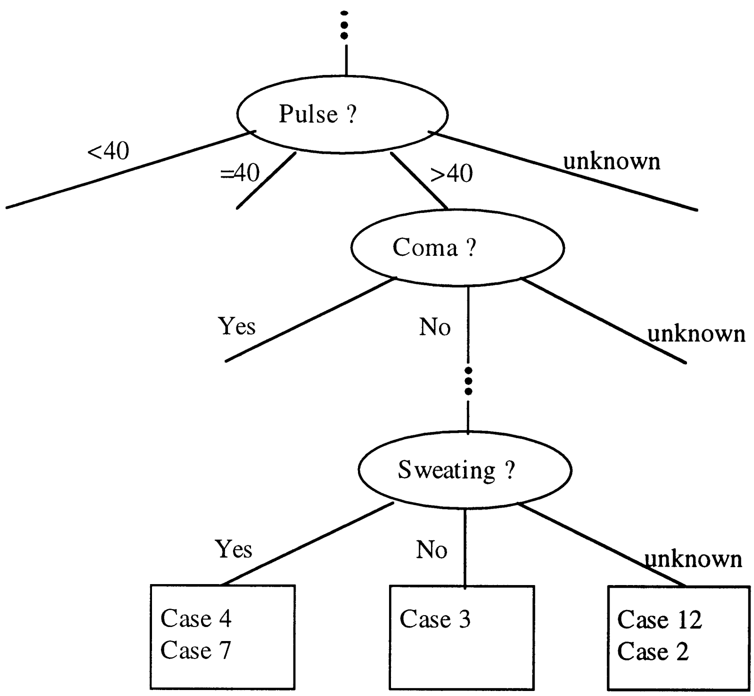

Fig. 1. Example of an Inreca-Tree for the toxicology domain.

set of available cases into three subsets in which the patient's pulse is less than 40, equal to 40, higher than 40, or unknown. The next node partitions the set of cases with a pulse higher than 40 into three subsets, depending on the 'Coma' attribute. At the leaf nodes of the tree, some buckets are displayed which contain the respective cases.

The Inreca-Tree is built prior to the first consultation of the system. It is assumed that all available cases are already stored in the case base (CB) and are accessible. The basic recursive procedure for building an Inreca-Tree is quite simple and is described in detail in [38].

3.3. Retrieval with the Inreca-Tree

The Inreca-Tree can be used for efficiently retrieving the most similar case(s) for a given new problem case. The search is done via a recursive tree search procedure according to the global similarity measure $SIM(X,Y)$. During the search, the two test procedures '*Ball-Overlap-Bounds*' (BOB) and '*Ball-Within-Bounds* (BWB) are used to focus on the relevant search region. These procedures are extensions of equivalent procedures known from k-d trees [14].

While the search is performed, a priority list is maintained which contains the k most similar cases known so far, together with their similarity to the problem case. This list is updated when new cases are visited. The recursive procedure (beginning with the root node) runs as follows:

- If the current node is an inner node, the procedure is first iterated on one of the child nodes. The procedure follows the branch whose constraint is fulfilled by the value of the respective attribute contained in the problem case.
- If the current node is a leaf node, the priority list is updated according to the similarity of the cases belonging to the bucket with the problem case. Then the BWB test checks whether it is guaranteed that all k-nearest neighbors have been found. If this is the case, the search is terminated. If this is not the case, the search backtracks to a parent node and considers whether to investigate an additional portion of the tree.
- If the current node is an inner node that is reached through backtracking from a child node, a test is executed to examine whether it is necessary to inspect one of the other child nodes. This is performed by the BOB test. If this test is false, the partition of the other child nodes cannot contain any k-nearest neighbors with respect to the query. Therefore, no further examination takes place and the search backtracks to the parent of the current node. If this test is true, the procedure is iterated on these child nodes, i.e. it continues the search in the respective sub-tree.

The BOB and BWB test procedures are relatively simple geometrical tests in which a m-dimensional ball is drawn around the current query. The radius of this ball is determined by the similarity of the least similar case which is currently in the priority list (kth most similar case). Every case that is outside this ball is less similar than the currently known k most similar cases and consequently need not be visited. In order to recognize whether a node is 'of interest' (it may contain some candidates), the geometrical bounds of the node are used to define a test point that is most similar to the current query, but still lies within the geometrical bounds of the current node.

In the best case, when no backtracking is required, retrieval with the Inreca-Tree leads to a running time that is proportional to the depth of the tree; the best retrieval effort is $O(\log(n))$, with n being the number of cases in the case base. However, in the worst case in which backtracking is required for every node, the whole tree must be investigated, leading to a retrieval effort of $O(n)$. Experiments conducted in several non-medical domains have clearly shown that in the average case important reductions in the retrieval speed compared to a linear retrieval approach (complexity $O(n)$) can be achieved [37].

3.4. Compiling inductive knowledge into the similarity assessment

We now describe an approach that allows speedier retrieval as the number of cases increases drastically. The idea behind this mechanism is to move from an approach in which the knowledge contained in the cases is interpreted to an approach that compiles some of the knowledge contained in the cases into rules that further improve the retrieval efficiency. The generation of general rules out of cases (often called examples) has been intensively studied in the field of inductive machine learning [26]. We focus on a particular approach from this field, namely the top-down induction of decision trees (TDIDT: [30,31]).

A decision tree is an n-ary tree whose inner nodes are labeled with an attribute. The links that lead from a node to its child nodes are labeled with a value that belongs to the value range of the attribute the node is labeled with. Each leaf node of a decision tree is labeled with a *decision class*, e.g. the diagnosis referring to the kind of intoxication. Obviously, a decision tree is very similar to the previously described Inreca-Tree. The main difference in this data structure is that the leaf node of the Inreca-Tree contains a set of cases (bucket), while a leaf node of a decision tree contains a decision class[5]. However, the main difference between the Inreca-Tree and a decision tree lies in the way it is used during problem solving. A decision tree is only traversed once from the root node to a leaf node. At every node, the link is followed which matches the respective attribute value in the problem case. BOB and BWB tests are not used. The decision class noted at the leaf node is then returned as result of the decision tree consultation. However, problem solving with a decision tree can be compared with a retrieval using an Inreca-Tree in which no backtracking occurs.

The efficiency of decision tree consultation is one of its main advantages. However, decision tree consultation has also several drawbacks which we expect to lead to major problems [9] particularly in medical decision support tasks. One problem stems from the fact that a decision tree consultation completely ignores the concrete cases from which the decision tree was built. Information contained in the cases but not contained in the branches of the tree is not used for decision making. This is a major disadvantage if only a small number of cases are currently available.

[5] On a more technical level, their are some more differences, e.g. in the kind of branches that are allowed in a decision tree. However, these differenes are not important for the further discussion.

The Inreca-Tree can be used to realize an intermediate approach between TDIDT and CBR. We begin with pure CBR and move 'more to the side of induction' as more cases arise. This approach is called *seamless integration* between CBR and induction [3,2]. With this procedure, we want to avoid the problem of increased retrieval time as more and more cases are added to the case base.

The general idea behind realizing this shift from CBR to induction is to include more and more nodes in the Inreca-Tree for which backtracking is not required. As an example consider the Inreca-Tree shown in Fig. 1. In this tree, backtracking may be eliminated in the 'yes-branch' of the 'Coma' node. If coma is observed in the current problem case, then all cases not describing comatose patients can be ignored, because some special treatment is always needed in this situation. All cases about patients who do not have coma can therefore be ignored.

We have developed an approach that allows compilation of general knowledge, extracted from the decision tree, into the similarity measure [37,2]. The more cases available, the more reliable the general knowledge extracted from the induced decision tree, because the inductive hypothesis is based on a larger set of known examples. However, we strongly propose that even when the case base grows more and more such induced knowledge is carefully validated by an expert before it is used.

4. The evaluation of Inreca technology for the toxicology application

For legal and ethical reasons a medical system should not be introduced into clinical practice before it has been properly evaluated [36]. Evaluation should cover all stages of the development process of a decision support system. An important point is that evaluation is a process to be continued after the introduction of a decision support system into practice, as is also done in post-marketing surveillance studies of drugs.

We describe in detail how the Inreca system, and especially the integration described in Section 3, meets the requirements for medical decision support systems as presented in Section 2. Then we present a plan for introducing case-based decision support systems in medical environments. Finally, some first experimental results on the evaluation on two initial CBR prototype systems are summarized.

4.1. Meeting the domain and task requirements

We now discuss how the Inreca approach fulfills the requirements of decision support tasks in medical domains.

4.1.1. Short response times

Efficient retrieval of cases was one of the major motivations for the development of the Inreca-Tree and particularly the seamless integration described in Section 3.4. First of all, the indexing of a large case base by an Inreca-Tree (see Section 3.2) already allows a very efficient case retrieval. Experiments in several non-medical domains have shown a significant speedup compared with other approaches to retrieval [38,37]. Further important improvements can be achieved through the compilation of inductively learned general knowledge into the similarity measure. Even if this improvement of efficiency is not of great importance for small case bases, large case bases, which are used in real-life applications [25], require such an efficient indexing approach to fulfill the tight constraints on acceptable response time.

4.1.2. Justifiability of results

The traditional justification of a diagnosis achieved with a CBR approach is to present the complete information contained in the most similar case. The physician who uses the system can then validate the similarity between the new case and the retrieved case. With the Inreca approach an alternative type of justification is also possible. If the Inreca-Tree is viewed not as an indexing tree but as a decision tree, the user can validate the decision path followed by the system. Physicians that prefer 'thinking in rules' will consider this information as very valuable [20].

4.1.3. Dealing with incomplete information

One major advantage of CBR is its ability to cope with incomplete information, i.e. unknown attribute values. The Inreca-Tree explicitly considers situations in which some attribute values in the problem case or even in a case stored in the case base are unknown.

4.1.4. Dealing with measured values and conceptual terms

The Inreca approach can handle numeric and symbolic attributes. While numeric attributes are usually required for handling measured values, ordered or unordered symbolic attributes are required for expressing conceptual terms. The Inreca-Tree can use both types of attributes for efficient indexing.

4.2. Developing case-based medical decision support systems

Developing CBR applications generally requires defining the area of competence (e.g. cardiology, rheumatology, toxicology, etc.), the purpose of the system (call center support, ambulance support, education, etc.), and its intended users (students, less experienced physicians, expert physicians, etc.). The next step is to select an appropriate CBR shell. A CBR shell realizes the basic mechanisms for case representation, similarity assessment, and retrieval [4]. It also provides interfaces to other software components (e.g. databases) and to the user. For further discussion we assume that a CBR shell based on Inreca technology is considered, such as Kate tools (AcknoSoft, France) or CBR-Works (tecInno, Germany).

Based on the above decisions, a number of complicated, interacting development tasks must be carried out:

- Defining an appropriate case representation, i.e. selecting relevant diagnostic signs to be used as attributes, determining an appropriate value range for each attribute including the definition of the respective decision classes.

- Defining the similarity assessment, i.e. the local similarity measures and the attribute weights.
- Collecting cases.

Due to the difficulty of these development tasks, an incremental development strategy is currently considered to be most successful [10,5]. The underlying model is comparable with the 'spiral model' well-known in software engineering [11]. During the development process a sequence of incrementally improved CBR prototype systems is generated. Each prototype system must be validated in order to define the steps required to further improve the system. For the development of medical decision support systems we propose to differentiate at least the following system development phases.

During the initial system building phase, a first and simple case representation and similarity assessment is defined and an initial set of cases is collected. This development phase requires only limited involvement of an expert physician. The resulting initial CBR prototype system must then be analyzed according to its classification behavior. Based on this analysis the initial CBR prototype system will be revised with respect to case representation and similarity assessment.

In the next phase, the case base will be further extended and a novice physician will validate the system using a number of collected test cases. This validation step will lead to further revisions or improvements of the system. This process will be iterated with a number of novice physicians, experienced physicians and domain experts. While novice physicians can test the system using mainly standard cases, more experienced physicians can also use more complicated and unusual cases for testing.

If the last validation step has been successful, then a pilot CBR system can be installed and used by an expert physician (e.g. in a toxicology call center). An expert is required here because they are able to interpret each case in the case base and, by this, can decide whether a suggestion of the CBR system is appropriate. If the system has been successfully used for some time, it is possible to extend the group of possible users to less experienced physicians and finally, to students for educational purposes.

4.3. Evaluating two initial CBR prototype systems: first experimental results

Up to now two initial CBR prototype systems have been built and evaluated as part of the Inreca+ project. In the following sections we present some of the first experimental results. Kate-CBR was used as CBR shell for building these initial systems. We used the standard similarity assessment provided by this shell without any domain-specific optimizations.

For the purpose of evaluation we also built decision support systems based on alternative methods, namely

- Kate-Induction, which is a commercial inductive tool based on the decision tree algorithm,
- two classification systems based on the Bayes (BC) and the linear discriminant function approach (LDFC), and

Table 3
Comparison of classification discrepancy between LDFC, BC, Kate-Induction and Kate-CBR

Training sample (no. cases)	LDFC discrepancy (%)	BC discrepancy (%)	Kate-Induction discrepancy (%)	Kate-CBR discrepancy (%)
4	—	—	35.0	15.6
16	40.6	27.0	20.0	4.7
24	26.6	18.0	17.0	4.7
32	18.5	17.4	14.0	3.1

- a specialized classification approach, called algebraic approach (AS), which consists of an optimized combination of four different classifiers as described by Zhuravlev [39].

4.3.1. Initial CBR system for the cardiology domain

We consider the following medical classification problem from the cardiology domain, where some experiments already carried out in the scope of the work by Bolotov and Larichev [12] could be reused: the differential diagnostic of pulmonary thromboembolism (PTE) and myocardial infarction (MI). Experts gave the following set of symptoms: past history, breathing, skin color, arterial blood pressure, ECG, and lung radiography. There are three decision classes (diagnoses): I for PTE, II for MI and III for PTE in conjunction with MI.

An initial case base of 64 cases was developed. In Table 3 the precision of the classification is given depending on the respective sample size. The discrepancy is evaluated for each method (the average numbers are given for five random samples).

CBR appears to be much better for this particular domain than induction and than LDFC and BC. Even with a small size of the training sample, CBR leads to an acceptable classification accuracy.

4.3.2. Initial CBR system for the toxicology domain

We developed an initial CBR system for the toxicology domain, in particular for the task of poison recognition during acute poisoning. A case data set of 459 cases, based on the eight types of drugs shown in Table 1, was acquired. In several runs, the Kate-CBR, Kate-Induction, and the AS algorithm were used on the same data sets. Table 4 presents the results of this experiment.

Table 4
Comparison of classification accuracy between AS, Kate-Induction and Kate-CBR in the toxicology domain

Algorithm	Range of classification accuracy for different training samples (%)
AS algorithm	92.0–96.0
Kate-Induction	78.5–86.6
Kate-CBR	89.3–93.8

It appears that the CBR approach leads to a very high classification accuracy which is only slightly worse than the accuracy of the AS algorithm, which is highly optimized for the toxicology domain. Thus, this result is particularly notable when considering the low development effort required for applying Kate-CBR or Kate-Induction compared to the effort for developing the AS algorithm. Based on these preliminary experimental results we are quite optimistic that a systematic approach to CBR system development will lead to valuable case-based toxicological decision support system. Further developments along the lines discussed in Section 4.2 will, of course, be necessary.

5. Discussion

A detailed comparison of Inreca with other CBR/CBR-related approaches and approaches to the integration of CBR and induction is provided by Althoff [2].

A success story for a knowledge-based medical decision support system in the toxicology field is provided by Darmoni et al. [13] who report on the SETH approach at Rouen University. The domain was chosen because drug poisoning is a frequent problem there. The aim of SETH is to give end-users specific advice concerning treatment and monitoring of drug poisoning. It simulates expert reasoning, taking into account for each toxicological task delay, sign, and dose. It is in daily use by hospital residents as telephone response support since April 1992. It is also used as an educational tool for drug poisoning.

Malek et al. [23] describe an interesting combination of CBR and neural nets for solving toxic comas diagnostic problems. The authors compared their approach with approaches from k-nearest neighbors and decision trees. We believe that approaches that cannot guarantee to find the most similar case(s) available, because of using heuristic generalization and/or retrieval techniques as it is described in this paper, are not appropriate for such critical decision support tasks.

Puppe et al. [29] compared four different techniques for building medical decision support systems on acute abdominal pain cases. They stated that the diagnostic performance of a knowledge-based system depends more on the amount and quality of knowledge exploited than on the problem solving method chosen. If some piece of knowledge or data, essential for making a certain decision, is missing from the knowledge base or case description, no problem solving method can be expected to produce satisfactory results. They concluded that the building of medical decision support systems is still an art rather than a routine task of software engineering.

Goos and Schewe [17] describe a successful CBR application to clinical rheumatology. They compared the performance of their CBR approach against an expert system based on general knowledge only, as described by Gappa et al. [15]. Goos and Schewe [17] report that the CBR approach required only one third of the development effort of the earlier (general) knowledge-based system. However, the results were worse than those of the (general) knowledge-based system because the used case base was too small to allow all combinations of diagnoses occurring in real-life situations to be found.

One possible conclusion here could be that the CBR approach appears to have some advantages concerning system development if compared with other knowledge-based methods [7]. However, describing the similarity assessment mechanism adequately is a detailed knowledge engineering task that possibly ‘consumes’ part of the effort ‘saved’ by using previous cases.

For medical experts CBR is not more natural than other reasoning methods such as rule-based reasoning. In addition, it is obvious that medical experts do not reason from cases only [20,28,21]. So, why do we need CBR for medical decision support systems? We believe that there are the following reasons.

CBR explicitly represents, memorizes, and reasons about cases, which are very important entities in medical contexts (partially already available electronically). Thereby, CBR inherently combines problem solving and learning. By this the system development process is supported by automatic learning techniques as well as the update/maintenance process after the initial system has already been constructed. The observation that experts of long standing use a ‘compiled form of knowledge’ [20] could be simulated by a CBR system by using induction as a means for compiling cases into general knowledge (as exemplified in Section 3.4).

CBR also offers a very natural approach to differential diagnostics: physicians readily admit that the crucial point in making a diagnosis involves excluding diseases with very similar symptoms [20]. As a consequence, the actual test selection strategy, used during real problem solving, can focus on such similar cases and discriminate between these cases based on the user’s answers [8,32].

A CBR system can be used by an expert in the field to supplement their knowledge on unusual cases. CBR technology also offers certain degrees of flexibility with respect to broadening the scope of its usage, for instance towards more general information retrieval problems (integration into clinical information systems, similarity based retrieval in databases).

Acknowledgements

We thank John Daly for helpful comments on an earlier version of this article.

References

[1] A. Aamodt, E. Plaza, Relating case-based reasoning: foundational issues, methodological variations and system approaches, AI Commun. 7 (1) (1994) 39–59.

[2] K.-D. Althoff, Evaluating Case-based Reasoning Systems: The Inreca Case Study, Habilitationsschrift, University of Kaiserslautern, 1996.

[3] K.-D. Althoff, A. Aamodt, Relating case-based problem solving and learning methods to task and domain characteristics: towards an analytic framework, AI Commun. 9 (1996) 1–8.

[4] K.-D. Althoff, E. Auriol, R. Barletta, M. Manago, A Review of Industrial Case-based Reasoning Tools, AI Intelligence, Oxford, 1995.

[5] K.-D. Althoff, B. Bartsch-Spörl, Decision support for case-based applications, Wirtschaftsinformatik 38 (1996) 8–16.

[7] K.-D. Althoff, M.M. Richter, W. Wilke, Case-Based Reasoning: A New Technology for Experience Based Construction of Knowledge Systems, technical report, University of Berkeley, 1997.
[8] K.-D. Althoff, S. Wess, Case-based knowledge acquisition, learning and problem solving in diagnostic real world tasks, in: M. Linster, B. Gaines (Eds.), Proc. 5th European Knowledge Acquisition for Knowledge-Based Systems Workshop: EKAW-91, GMD-Studien No. 211, Sankt Augustin, 1992, pp. 48–67.
[9] K.-D. Althoff, S. Wess, R. Bergmann, F. Maurer, M. Manago, E. Auriol, N. Conruyt, R. Traphöner, M. Bräuer, S. Dittrich, Induction and case-based reasoning for classification tasks, in: H.H. Bock, W. Lenski, M.M. Richter (Eds.), Information Systems and Data Analysis, Prospects-Foundations-Applications, Springer-Verlag, Heidelberg, 1994, pp. 3–16.
[10] B. Bartsch-Spörl, How to make CBR systems work in practice, in: H.D. Burkhard, M. Lenz (Eds.), 4th German Workshop on Case-Based Reasoning—System Development and Evaluation—Informatik-Bericht No. 55, Humboldt University Berlin, 1996, pp. 36–42.
[11] B.W. Boehm, A spiral model of software development and enhancement, Computer 21 (5) (1988) 61–72.
[12] A.A. Bolotov, O.I. Larichev, Comparison of pattern recognition methods by precision of approximating the separating hyperplanes, Automation Remote Control 56 (1995) 1004–1010.
[13] S.J. Darmoni, P. Massari, J.-M. Droy, T. Blanc, J. Leroy, Functional evalutation of SETH: an expert system in clinical toxicology, in: P. Barahona, M. Stefanelli, J. Wyatt (Eds.), AI in Medicine—Proc. AIME'95, Springer-Verlag, Heidelberg, 1995, pp. 231–238.
[14] J.H. Friedman, J.L. Bentley, R.A. Finkel, An algorithm for finding best matches in logarithmic expected time, ACM Trans. Math. Software 3 (1977) 209–226.
[15] U. Gappa, F. Puppe, S. Schewe, Graphical knowledge acquisition for medical diagnostic expert systems, Artif. Intell. Med. 5 (1993) 185–211.
[16] L. Gierl, S. Stengel-Rutkowski, Integrating consultation and semiautomatic knowledge acquisition in a prototype-based architecture: experiences with dysmorphic syndromes, Artif. Intell. Med. 6 (1994) 29–49.
[17] K. Goos, S. Schewe, Case-based reasoning in clinical evaluation, in: S. Andreassen, R. Engelbrecht, J. Wyatt (Eds.), AI in Medicine—Proc. AIME'93, IOS Press, Amsterdam, 1993, pp. 445–448.
[18] M. Haddad, D. Mörtl, Porenta G. SCINA: A case-based reasoning system for the interpretation of myocardial perfusion scintigrams, Proc. Comp. Cardiol., 1995.
[19] J.L. Kolodner, Case-based Reasoning, Morgan Kaufmann, San Mateo, 1993.
[20] O.I. Larichev, A study on the internal organization of expert knowledge, Pattern Recognit. Image Anal. 5 (1995) 57–63.
[21] O.I. Larichev, H. Moshkovich, E. Furems, A. Mechitov, V. Morgoev, Knowledge Acquisition for the Construction of Full and Contradiction Free Knowledge Bases, Programma, Groningen, 1991.
[22] R.T. Macura, K.J. Macura, MacRad: Radiology image resource with a case-based retrieval system, in: M. Veloso, A. Aamodt (Eds.), Case-Based Reasoning Research and Development, Springer-Verlag, Heidelberg, 1995, pp. 43–54.
[23] M. Malek, V. Danel, V. Rialle, A hybrid case-based reasoning system applied to toxic comas diagnosis, Technical Report, 1996.
[24] M. Malek, V. Rialle, A case-based reasoning system applied to neuropathy diagnosis, in: M. Keane, J.P. Haton, M. Manago (Eds.), EWCBR-94—Second European Workshop on Case-Based Reasoning, AcknoSoft Press, Paris, 1994, pp. 329–336.
[25] M. Manago, E. Auriol, Integrating induction and case-based reasoning for troubleshooting CFM-56 aircraft engines, in: B. Bartsch-Spörl, D. Janetzko, S. Wess (Eds.), Fallbasiertes Schließen—Grundlagen und Anwendungen, Centre for Learning Systems and Application, University of Kaiserslautern, LSA-Report 95-02, 1995, 73-80.
[26] R.S. Michalski, A theory and methodology of inductive learning, Artif. Intell. 20 (2) (1983) 111–161.
[27] E.T.O. Opiyo, Case-based reasoning for expertise relocation in support or rural health workers in development countries, in: M. Veloso, A. Aamodt (Eds.), Case-Based Reasoning Research and Development, Springer-Verlag, Heidelberg, 1995, pp. 77–87.

[28] B. Puppe, Building a medical knowledge base: Tricks facilitating the simulation of the expert's reasoning, in: S. Andreassen, R. Engelbrecht, J. Wyatt (Eds.), AI in Medicine—Proc. AIME'93, IOS Press, Amsterdam, 1993, pp. 168–171.
[29] B. Puppe, C. Ohmann, K. Goos, F. Puppe, O. Mootz, Evaluating four diagnostic methods with acute abdominal pain cases, Technical Report, University of Würzburg, 1994.
[30] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1986) 81–106.
[31] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan-Kaufmann, San Mateo, 1993.
[32] M.M. Richter, S. Wess, Similarity, uncertainty and case-based reasoning in Patdex, in: R.S. Boyer (Ed.), Automated Reasoning, Kluwer, Dordrecht, 1991, pp. 249–265.
[33] R. Schmidt, B. Heindl, B. Pollwein, L. Gierl, Prognoses of multiparametric time course abstractions in a case-based reasoning system, in: H.D. Burkhard, M. Lenz (Eds.), 4th German Workshop on Case-Based Reasoning—System Development and Evaluation—Informatik-Bericht No. 55, Humboldt University Berlin, 1996, pp. 170–177.
[34] R. Schmidt, B. Pollwein, L. Boscher, G. Schmid, L. Gierl, Der fallbasierte konsiliarius ICONS für die antibiotika-therapie, in: B. Bartsch-Spörl, D. Janetzko, S. Wess (Eds.), 3rd German Workshop on Case-Based Reasoning—Foundations and Applications—Centre for Learning Systems and Applications, University of Kaiserslautern, LSA-95-02, 1995, pp. 54–62.
[35] A. Seitz, A. Uhrmacher, Cases versus model-based knowledge—an application in the area of bone healing, in: H. Burkhard, M. Lenz (Eds.), 4th German Workshop on Case-Based Reasoning—System Development and Evaluation—Informatik-Bericht No. 55, Humboldt University Berlin, 1996, pp. 178–185.
[36] J. Van Bemmel, Criteria for the acceptance of decision support systems by clinicians, in: S. Andreassen, R. Engelbrecht, J. Wyatt (Eds.), AI in Medicine—Proc. AIME '93, IOS Press, Amsterdam, 1993, pp. 7–12.
[37] S. Wess, Fallbasiertes Schließen in wissensbasierten Systemen zur Entscheidungs unterstützung und Diagnostik, Doctoral Dissertation, University of Kaiserslautern, 1995; also: Infix Verlag, Sankt Augustin, Germany.
[38] S. Wess, K.-D. Althoff, G. Derwand, Using k-d trees to improve the retrieval step in case-based reasoning, in: S. Wess, K.-D. Althoff, M.M. Richter (Eds.), Topics in Case-Based Reasoning, Springer-Verlag, Heidelberg, 1994, pp. 167–181.
[39] Y.I. Zhuravlev, On an algebraic approach to the problems of pattern recognition and classification, Probl. Kibern. 33 (1978) 5–58 (in Russian).