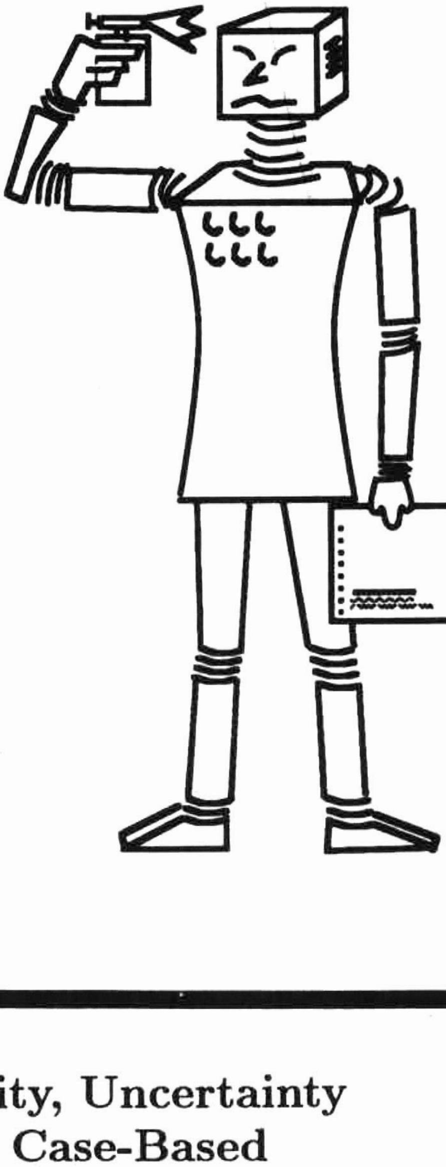
Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern

SEKI - REPORT



Similarity, Uncertainty and Case-Based Reasoning in PADTEX

Michael M. Richter, Stefan Weß

SEKI Report SR-91-01 (SFB)

Similarity, Uncertainty and Case-Based Reasoning in PATDEX

Michael M. Richter, Stefan Wess*
University of Kaiserslautern
Dept. of Computer Science
P.O. Box 3049
D-6750 Kaiserslautern

January 22, 1991

Abstract

PATDEX is an expert system which carries out case-based reasoning for the fault diagnosis of complex machines. It is integrated in the MOLTKE workbench for technical diagnosis, which was developed at the university of Kaiserslautern over the past years, MOLTKE contains other parts as well, in particular a model-based approach; in PATDEX where essentially the heuristic features are located. The use of cases also plays an important role for knowledge acquisition. In this paper we describe PATDEX from a principal point of view and embed its main concepts into a theoretical framework

1 General Considerations

PATDEX[1] is an expert system which carries out case-based reasoning for the fault diagnosis of complex machines. It is integrated in the MOLTKE workbench[2] for technical diagnosis, which was developed at the university of Kaiserslautern over the past years (cf. e.g. [4, 5, 23]), MOLTKE contains other parts as well (cf. e.g. [16]), in particular a model-based approach (cf. [21, 22]); in PATDEX [3] where essentially the heuristic features are located. The use of cases also plays an important role for knowledge acquisition. In this paper we describe PATDEX from a principal point of view and embed its main concepts into a theoretical framework.

This research has a number of mainly indirect connections to the work of Woody Bledsoe. We mention his interest in analogy, his early connectionist work and his influence in merging mathematics and artificial intelligence. For the first author the main point was that Woody Bledsoe brought him in contact with AI at an early stage. More than twenty years ago we started a lively discussion which still goes on and will hopefully last for many more years.

*The work presented herein was partially supported by the Deutsche Forschungsgemeinschaft,SFB 314:" Artificial Intelligence - Knowledge-Based Systems", projects X6 and X9.

[1] PATtern Directed EXpert Systems

[2] MOdels, Learning and Temporal Knowledge in Expert Systems for Technical Domains

1.1 Similarity

Similarity and uncertainty have in common that both can be described by measures of values e.g. in the real interval $[0,1]$. At first glance it seems that here the analogy between these concepts comes to an end; we will, however, discuss some more connections later on.

A similarity measure $sim(x,y)$ can be defined on arbitrary objects of interest as physical objects, situations, problems or formulae; let U be the (finite) universe of these objects.

The basic axioms for sim are:

1. $sim(x,x) = 1$ (reflexivity)
2. $sim(x,y) = sim(y,x)$ (symmetry)

The dual notion is that of a *distance measure* $d(x,y)$ which may attain arbitrary nonnegative values. In the corresponding axioms reflexivity reads as $d(x,x) = 0$. One does not require, however, the triangle inequality and allow $d(x,y) = 0$ for $x \neq -y$ which means that d is neither a metric nor even a pseudo-metric.

One says that d and sim correspond to each other iff there is an order reversing one-one mapping

$$f : range(d) \rightarrow range(sim)$$

such that $f(0) = 1$ and $sim(x,y) = f(d(x,y))$; we denote this by $d \equiv_f sim$.

Popular candidates for f are: $f(z) = 1 - \frac{z}{1+z}$ for unbounded $range(d)$ or$f(z) = 1 - \frac{z}{max}$, if $range(d)$ has a greatest element max.

Usually individual values of sim or d are not so much of interest as certain relations between them. For the use in analogical reasoning the following relations are basic. If d is a distance measure and sim a similarity measure then we define

$$R_d(x,y,u,v) \;:\Longleftrightarrow\; d(x,y) \leq d(u,v) \qquad (1)$$

$$R_{sim}(x,y,u,v) \;:\Longleftrightarrow\; sim(x,y) \geq sim(u,v) \qquad (2)$$

and

$$S_d(x,y,z) \;:\Longleftrightarrow\; R_d(x,y,x,z) \qquad (3)$$

$$S_{sim}(x,y,z) \;:\Longleftrightarrow\; R_{sim}(x,y,x,z) \qquad (4)$$

It is very often easier to determine the relation $S_d(x,y,z)$ than the distance measure d itself and it is also sufficient for many applications.
We say that d and sim are *compatible*, iff

$$R_d(x,y,u,v) \Longleftrightarrow R_{sim}(x,y,u,v) \qquad (5)$$

compatibility is ensured by $d \equiv_f sim$ for some f.

For some set $M \subseteq U$ some $y \in M$ is called *most similar* to x with respect to M iff

$$\forall z \in M : S_d(x,y,z)$$

holds. As usual in topology the measures also define a neighborhood concept. For $\epsilon > 0$ we put

$$V_\epsilon(x) := V_{d,\epsilon}(x) := \{y \mid d(x,y) \leq \epsilon\}$$

and analogously $V_{sim,\epsilon}(x)$ is defined; if d is a metric then these sets are ordinary closed neighborhoods. $S_d(x,y,z)$ expresses the fact that each neighborhood of x which contains z also contains y.

There are two uses of similarity which we have in mind which are related to each other and play a role in PATDEX. In both cases we regard the objects x as incomplete descriptions of other unknown objects x_0 in which we are really interested.

In the first case $sim(x,y)$ is simply interpreted as the degree of certainty that $x_0 = y_0$ holds. For the other extended intention we regard the objects as problems where some problems are associated with solutions. Then $sim(x,y)$ shall express the degree of certainty that the solution for x is also a solution for y (or, more sophisticated, can easily be modified to a solution of y).

We will generally assume that objects are described by attributes which may take on certain values. There is a fixed number of attributes but the objects of the universe U are identified with tuples of attribute values for some set A of attributes where certain attributes may be missing. This gives rise to many similarity measures and we will discuss one in section 1.3.

1.2 Uncertainty

When dealing with incomplete information a certainty measure or factor $\mu(x)$ expresses the degree of certainty that the event x has taken place. Here $\mu(x)$ is w.l.o.g. a real number in $[0,1]$; it may be interpreted in various ways, e.g. as a probability or some kind of evidence. When dealing with problems and solutions uncertainty applies to two kinds of objects. One was mentioned above, namely the uncertainty of two descriptions describing the same problem. In addition we have the uncertainty of having the correct solution. In case-based reasoning (see 1.4 below) both aspects are intimately connected. The problem that in the presence of incomplete information different objects sometimes cannot be distinguished with certainty was treated in the investigations on rough sets and we will introduce the basic notions of this field.

The approach introduced in [17] defines ***rough sets*** as pairs $(U, \approx)$ where $\approx$ is a binary relation on the universe U called the ***indiscernability relation.*** If $x \approx y$ holds then x and y are called indistinguishable. With each subset $M \subseteq U$ two approximations are associated:

$$\begin{aligned} M_l &:= \{x \in U | y \in M \text{ for all } y \text{ with } x \approx y\} \text{ is the lower approximation} \\ M_u &:= \{x \in U | y \in M \text{ for some } y \text{ with } x \approx y\} \text{ is the upper approximation} \end{aligned}$$

The difference $\partial M = M_u / M_l$ is called the ***boundary*** of M. If $x \in M_l$, then x certainly belongs to M, if $x \in M_u$, then x is surely not in M; ∂M is the area of uncertainty. The uncertainty of $M \subseteq U$ can be measured by the correlated accuracy measure

$$\mu_{\approx}(M) := \frac{card(M_l)}{card(M_u)}; \text{ clearly } 0 \leq \mu_{\approx}(M) \leq 1 \text{ holds.}$$

Rough sets occur in various ways. There are essentially two different types which are both connected with diagnostic problems:

- The indiscernability relation is transitive
- The indiscernability relation is not transitive

In the first case $\approx$ defines a partition of U into blocks of indistinguishable elements. The typical example for this arises as above when the objects in U are described by attributes which may take on certain values. Each set A of attributes defines an indiscernability relation $\approx_A$, where $x \approx_A y$ holds iff the values of all attributes in A for x and y are identical.

Because of the finiteness of the universe U the upper and lower approximations of sets as well as the correlated accuracy measure can in principal be computed. In [17] it is shown how this can be expressed in terms of rules and how one can apply it to classification problems; an extension to a decision logic is presented in [18]. For larger numbers these computations become very time-consuming, however.

A prototype for the second case arises from a distance measure d. For each $\epsilon > 0$ there is an indiscernability relation $\approx_{d,\epsilon}$ defined by

$$x \approx_{d,\epsilon} y \text{ iff } y \in V_{d,\epsilon}(x)$$

In case d is a metric $\approx_{d,\epsilon}$ is transitive and the blocks are of the form $V_{d,0}(x)$. The intention is of course, that elements in these blocks are in some sense absolutely indistinguishable, i.e. there is no further information available to separate these elements.

1.3 Diagnosis

Our area of interest is fault diagnosis and we need to introduce the basic notions.

We assume a fixed number N of *symptoms* $S_1, \ldots, S_N$. With each symptom S_i a range R_i is associated; in principal symptoms are nothing than attributes. Typically R_i is either a real interval $[a, b]$ or the boolean domain $0, 1$ or some other finite set. Symptoms may take on values in their range and these values are assumed to be the only source of information.

Values of symptoms are obtained by carrying out a *test*. A test can be an observation, a measurement or simply the answer to a question. In some situations certain tests may not be allowed.

The information at some stage of the diagnostic process is usually incomplete and is expressed in the form of an *information vector* or a *situation*. A situation is a vector $Sit = (a_{i1} \ldots, a_{ik})$, $1 \leq i, j \leq N$ such that each $a_{ij} \in R_{ij}$. The components of Sit are the known symptom values whereas the values of the remaining symptoms are unknown. A situation is *complete*, if every symptom has a value.

Situations are arranged in the information graph. Its nodes are labelled with situations and an edge goes from Sit_1 to Sit_2 if Sit_2 has one more component then Sit_1 and there is a test t_j available which can provide the value necessary to extend Sit_1 to Sit_2; in this case t_j is a label of the edge.

A *diagnosis* (or fault description) is a formula of the first order predicate calculus using constants and relations over the ranges R_i; the precise form of these formulae is not of interest here. To avoid technical difficulties we assume always a single fault. This means that the set of complete situations is partitioned into sets representing these faults; a special set is *"no fault"* and, if wanted, another one is *"unknown fault"*. The applicability of this approach relies on the fact that at least the "interesting" faults can be fully described. In the diagnosis of even complex machines this assumption is usually satisfied; in medical diagnostics this sometimes may be doubtful.

In a diagnostic problem some complete situations have occurred but are only partially known, i.e. one is confronted with some incomplete situation Sit. The task is to determine the diagnosis of the unknown complete situation (at least with some certainty). At first glance this seems to be a pure classification problem. For each set A of symptoms and each fault diagnosis ϕ we can compute the accuracy introduced above as

$$\mu(\phi) = \mu_{\approx}(\{Sit | Sit \text{ a complete situation that satisfies} \phi\})$$

where $\approx = \approx_A$ as introduced in section 1.2. If $\mu(\phi) = 1$, then ϕ is decided by the symptoms in A and the symptoms not in A are redundant. To recognize redundant symptoms some of the computations mentioned in 1.2 are useful; in PATDEX they are complemented using the case-base.

With equal right, one can say however, that the real problem is to find an optimal way to complete incomplete situations sufficiently enough so that a diagnosis with a high degree of certainty can be established. This task has been attacked less successful in the literature.

Hence, when given a situation Sit one may proceed in two ways:

1. Select a test t_k in order to reach a node in the information graph which allows a better diagnosis than in Sit.

2. Present candidates for a diagnosis, possibly accompanied by a certainty factor.

In PATDEX (as well as in the general MOLTKE approach) these two steps play the central role. The second step contains mainly a classification problem. The first step is more complex and can be discussed from different points of views. Above we said that we want an optimal way to complete the information; the term *optimal* is not clearly defined here, however. The main point is that we want an optimization with respect to an unknown target, namely the true diagnosis. We believe that purely information theoretic based approaches like the Top-Down-Inductive-Decision-Trees (ID3, cf.[20]) are insufficient for our purposes so PATDEX chooses another way.

1.4 Case-Based Reasoning

In case-based reasoning (cf. e.g. [14, 13, 25, 8]) one has a base of cases where a case is an ordered pair c = *(problem, solution)*. The cases are stored in a case base. Instead of solving a new problem directly the case base is employed in order to use solutions from earlier problems.

Hence the heart of case-based reasoning is analogical reasoning (cf. e.g. [11, 9, 10, 12, 27]). The various definitions of case-based reasoning differ in the way how the case-base is involved, see e.g. [28]. All descriptions have in common that the case base has a structure which enables the system to find suitable cases that help solving an actual problem. This structure plays an important role and therefore case base reasoning has two ingredients:

- Use analogy to infer solutions from solved problems for an actual problem.
- Add each new case to the case base and update its internal structure.

We will first concentrate on analogy. Given an actual problem p two steps have to be carried out:

1. Select a suitable case $c' = (p', s')$ in the case base.
2. Transfer the solution s' of c' in order to get a solution s of p.

The selection of a suitable case uses a similarity measure. Given p one selects $c' = (p', s')$ such that p' is most similar to p with respect to the case base. In our diagnostic context the problem is a situation and the solution is the diagnosis, hence a case has the syntactic form $c = (Sit, \phi)$. We assume that the diagnosis was correct for the recorded case. There are three basic possibilities for this:

1. Sit did not have sufficiently many symptom values to determine ϕ but one had a good guess.
2. Sit did determine ϕ but contained redundant information, i.e. unnecessary symptom values.
3. Sit did determine ϕ and no smaller situation would do so.

Since one does not know which possibility has occurred one has to take care of all of them[3]. This is a requirement not only for the structure of the case base but also for the similarity measure. A very general type of appropriate similarity measures for situations which are also used in PATDEX is of the form (cf. [3, 2]):

$$sim(Sit_1, Sit_2) = \frac{\alpha\, card(E) + \beta\, card(C) + \eta\, card(U_1) + \gamma\, card(U_2)}{card(E \cup C \cup U_1 \cup U_2)} \qquad (6)$$

where α, β, γ and η are real numbers and

- E := set of attributes with the same values for Sit_1 and Sit_2
- C := set of attributes with different values for Sit_1 and Sit_2
- U_1 := set of attributes which values for Sit_1 but not for Sit_2
- U_2 := set of attributes which values for Sit_2 but not for Sit_1

When a new problem was solved successfully then this gives rise to a new case which is added to the case base. The experience gained in the solution process can, however, also be used to eliminate cases. The situations of the cases are organized in the information graph; in PATDEX this graph will be labelled in a particular way in order to use it for test selection.

[3] Especially we point out here, that the occurrence of incompleteness (1) and redundancy (2) is independent of one another.

2 The PATDEX-System

2.1 Motivation and Overview

PATDEX [4] is a part of the MOLTKE-System which was developed in the past years at the University of Kaiserslautern. The starting point of PATDEX are the above considerations. PATDEX employs the techniques developed in these contexts both in an explicit or implicit way; we will not discuss these details here.

Instead we are interested in some shortcomings of these approaches in real world applications which we will point out next.

- The problem in the use of rough sets based on the relations $\approx_A$ for sets A of symptoms consists mainly in the fact that the diagnosis problem is considered only as a classification problem, in particular the indiscernability relations $\approx_A$ for different A are totally unrelated to each other. If $A \subseteq B$ then $\approx_B$ is finer and therefore more informative than $\approx_A$; this should be reflected in the system. Also, one would like to get hints which B for a given A provides the most suitable new information.
- The difficulty with the similarity measure is that its quality is related to the final success of the whole reasoning procedure; this is an *a posteriori criterion*. A priori it is not clear what the criteria for similarity of objects should be; they do not only depend on the objects themselves but also on the pragmatics of reasoning. In case-based reasoning it is usually clear whether a solution for a given problem is correct but is far from clear what it means that two problems are similar enough that the solution for one problem also works for the other one. An even more serious difficulty arises when the world of problems is continuously changing.

All this suggests that the similarity should not be defined in some fixed way but instead be the result of an adaptive learning process. This will be carried out later on.

2.2 The PATDEX/1 prototype

The first version of PATDEX is PATDEX/1. This prototype contains the basic structures which have been extended later on as described in section 2.3.

In this section we will briefly describe this prototype. As basic techniques, PATDEX/1 applies learning by memory adaptation and analogical reasoning. The system has capabilities to memorize and utilize both its individual experiences and its statistical information. The reasoning process that uses this experience knowledge is combined with another one that focuses on similarities. The overall process of diagnosis is based on the analogical problem solving algorithm (APS) proposed by [11]. The process is started by the user giving some observed symptom values as input to the system.

[4] Actually, there are two systems, PATDEX/1 and PATDEX/2. By PATDEX (or the PATDEX approach) we denote all the information which relates to both systems.

The toplevel algorithm of PATDEX reads as follows:

Input : The actual situation Sit
Output : diagnosis ϕ or failure

1. Find a case in the case base with a situation Sit' most similar to Sit. If there is no case with a situation at least "minimally similar" to Sit then stop with failure.
2. If Sit and Sit' are "sufficiently similar" then accept the diagnosis ϕ of Sit' also for Sit and goto 4).
3. Otherwise select a test in order to obtain an improved situation and goto 1).
4. If the diagnosis is correct then add the case (Sit, ϕ) to the case base and stop with success.
5. If the diagnosis is not correct then cancel temporarly (i.e. for the actual problem) all cases with diagnosis ϕ and goto 3).

Here we need an external teacher who says whether a diagnosis is correct or not. We also have to explain *"minimally similar"* and *"sufficiently similar"*. For this we need a partition of the case base which is given after the introduction of the similarity measure. Finally, we have to describe the selection of the next test.

Therefore PATDEX has two main features, similarity and the experience net. Both make use of the case base but are independent and could work in parallel.

For situations PATDEX uses as a first proposal the similarity measure from equation 6 with parameters $\alpha = 1, \beta = -2, \gamma = \eta = -1/2$. It should be remarked that this measure is normalized to $[-2, 1]$ and reads as

$$sim(Sit_1, Sit_2) = \frac{card(E) - 2card(C) - 1/2(card(U_1) + card(U_2))}{card(E \cup C \cup U_1 \cup U_2)} \qquad (7)$$

This special choice of the parameters is at the moment mainly motivated by experimental results. It has a defensive, pessimistic character. A high negative contribution to the measure is given for conflicting symptom values, i.e. we strongly wish to avoid false diagnoses.

If the value assigned to a given case by the similarity measure exceeds a lower bound ϵ (hypothesis-threshold), this case is said to be qualified for further processing. If the value exceeds an upper bound δ it is even qualified as diagnosis (diagnosis-threshold). Both thresholds are locally defined for each case of the case base. If, for a given case, the similarity value equals 1 this case is said to be proven.

A case becomes disqualified for further use in a particular diagnosis session as soon as all symptoms contained in that case do not hold, given a situation encountered during diagnosis, or if there are no unknown symptom values any more and the specified case does not exceed the diagnosis-threshold. Another reason for disqualification is given if the case the system chooses as its hypothesis is refused by the user. For the use of cases in the top-level algorithm we will define different similarity classes. For this we choose real numbers ϵ and δ such that $0 < \epsilon < \delta < 1$ and define:

Sit_1 and Sit_2 are called

indistinguishable	$sim(Sit_1, Sit_2) = 1$
sufficiently similar	$\delta \leq sim(Sit_1, Sit_2) < 1$
probably similar	$\epsilon \leq sim(Sit_1, Sit_2) < \delta$
at least minimally similar	$0 \leq sim(Sit_1, Sit2) < \epsilon$
not minimally similar	$sim(Sit_1, Sit_2) < 0$

We note that we here have the possibility to make the numbers ϵ and δ dependent on the respective cases.

2.2.1 Test Selection - the Planning of the Diagnostic Process

The analogical problem solving mechanisms of PATDEX/1 are adjusted to the needs of the given domain within the field of technical diagnosis. The basic hypothesis is that the observable similarities concerning the fault behavior of the technical system under consideration normally have similar causes. Therefore the description of the situation of the known case serves as a guideline for the completion of the given partial description of the target case. Thus, analogical transfer for technical diagnosis in PATDEX1 means: eventually completing this partial description using the respective most similar case and the experience graph for the guidance of this process, i.e. the target situation is completed upon suspicion, so to speak, and then evaluated with respect to new ascertained symptoms and to the relation between the similarity value of the target situation and the given thresholds of the actual most similar case. Successes and failures of this process have their effect in an improvement of the underlying thresholds, whereas the typicalness or frequency of cases has its effect in an improvement of the weights in the experience graph (cf. Fig. 3).
With each situation Sit we can associate the set

$$C(Sit) = \{Sit' | Sit' \text{ a complete extension of } Sit\}.$$

When Sit is presented in principal every element of $C(Sit)$ could have occurred. One can take advantage of the fact that not all these elements are equally likely. The experience net will record the information for this purpose.

The information net is an information graph (see 1.3) where the nodes represent situations from the case base only. A directed edge leads from Sit_1 to Sit_2 iff Sit_2 contains one more symptom value than Sit_1. In addition, the edges are labelled with determination factors.

Let C be the case base. The determination factor $\Delta(Sit_2, Sit_1)$ is the estimated conditional probability of Sit_2 given Sit_1, i.e.

$$\Delta(Sit_1, Sit_2) = \frac{card(\{c \in C | c = (Sit, \phi), Sit \text{ extends} Sit_2\})}{card(\{c \in C | c = (Sit, \phi), Sit \text{ extends } Sit_1\})} \tag{8}$$

The quality of the determination factor is a function of the quality of the case base and it is clear that the determination factors should be adapted continuously.

2.2.2 Evaluation of PATDEX/1

PATDEX/1 is a stand-alone protoype which has been completely implemented before the completion of the MOLTKE workbench. It served for modeling the given facts of case- based knowledge processing using the diagnostic problem solving of an expert service technician as a guideline. Important features of this approach are the combination of similarity and experience for the diagnosis of technical systems and the differentiation between classification and test selection. This has to be seen as the fulfillment of a requirement of the underlying real world application. Particularly ***derivational analogy*** [10] can be elegantly applied to the field of technical diagnosis. Compared with a human engineer PATDEX/1 came off very well, in particular with respect to the similarity measure which has been defined in equation 7.

Shortcomings of PATDEX/1 are the difficulty to generalize the similarity measure and the fact that the case-focussing test selection is not necessarily globally optimal. Usually the complexity of handling the experience graph is super-exponential complexity concerning space and time because in the worst case all sequences of symptom values have to be represented. Furthermore, PATDEX/1 takes no advantage of causal or functional background knowledge. This increases the possibilities of faulty diagnoses when too many redundant symptom values are presented or some relevant ones are missing.

2.3 PATDEX/2

PATDEX/2 (cf. [6, 28]) is an integral part of the MOLTKE workbench which allows the utilization of all its qualities (proposed in [26]). Therefore it is possible to switch between case-based reasoning and the interpretation of a MOLTKE knowledge base during problem solving. The use of causal knowledge enables PATDEX/2 to identify pathologic symptom values. Thus, redundant information can be filtered off and cannot be the cause for a false diagnosis any more. By the exploitation of functional background knowledge additional symptom values can be derived from the known ones. In this manner the selection of the respective most similar case is considerably speeded up. The overall case-based reasoning approach which is used by PATDEX/2 is comparable to the memory-based reasoning approach proposed by Stanfill and Waltz in [25].

An important aspect of our PATDEX/2 approach is to view the relevances of certain symptom values for special situations as a part of the empirical knowledge which shall be learned. In PATDEX/2 we combine the case-based reasoning approach for diagnosis with a connectionist approach for learning this empirical knowledge.These relevances $w_{ij} \in [0,1]$ are represented by means of a relevance matrix R= $[wij]$ where the symptoms S_i and diagnoses ϕ_j occur as inscriptions of the rows and columns, respectively. In course of time the weights of the symptoms, i.e. the elements of the relevance matrix, are learned by PATDEX/2. The strategy for learning the entries of the relevance matrix is similar to the competitive learning mechanism proposed in [24]. The matrix R reads as follows:

$$\mathrm{R} = \begin{array}{ccccc} & \phi_1 & \phi_2 & \cdots & \phi_m \\ S_1 & w_{11} & w_{12} & \cdots & w_{1m} \\ S_2 & w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_n & w_{n1} & w_{n2} & \cdots & .w_{nm} \end{array}$$

For the degree of relevance of a certain symptom value it is important whether it is a consequence of the normal functioning of the technical system or of a fault. E.g. `relais 21K3 switched` is of the first kind while `voltage 214 too high` is a pathological symptom value. To identify certain pathological symptom values PATDEX/2 can use the functional background knowledge which is represented in the MOLTKE workbench.

Since PATDEX/1 uses its similarity measure *sim* only for the comparison of two cases, it is not necessary to define relations between symptom values. PATDEX/2 extends this view of similarity by the additional use of local similarity measures $\omega_i(a_{ik}, a_{il})$ which determine the similarity between possible symptom values $a_{ik}, a_{il}, \in R_i$ of a symptom S_i. If one of the symptom values is unknown then the similarity ω_i evaluates to zero.

The introduction of R and ω_i leads to the definition of a new similarity measure which is normalized to $[0,1]$ and matches the definition made in section 1.1:

$$sim(Sit_1, Sit_2) = \frac{\alpha E}{(\alpha E + \beta C + \eta U_1 + \gamma U_2)} \tag{9}$$

The new attribute sets (based on the sets defined in section 1.4) are defined by multiplying the relevance of a specific symptom S_i (represented by the relevance matrix R) with the similarity of the observed symptom value a_{ik} in Sit_1 and the defined symptom value a_{il} in the actual case $c = (Sit_2, \phi_j)$:

$$\begin{aligned}
E &:= \sum_{S_i \in E} w_{ij}\omega_i(a_{ik}, a_{il}) && (10)\\
C &:= \sum_{S_i \in C} w_{ij}(1 - \omega_i(a_{ik}, a_{il})) && (11)\\
U_1 &:= \sum_{S_i \in U_1} v_{ij}(1 - \omega_i(a_{ik}, a_{il}J)) = card(U_1) && (12)\\
U_2 &:= \sum_{S_i \in U_2} w_{ij}(1 - \omega_i(a_{ik}, a_{il})) = \sum_{S_i \in U_2} w_{ij} && (13)
\end{aligned}$$

We point out here that ω_i is zero for symptoms S_i which belong to one of the attribute sets U_1 or U_2 , because the corresponding symptom values are unknown (cf. section 1.4).

Additionally, we restrict the representation of redundant symptoms (i.e. $S_i \in U_1$) to pathological ones. Thus, observed redundant symptom values representing the normal behavior of the underlying technical system cannot decrease the value of *sim* any more. Since PATDEX/2 focusses on the learning of symptom relevances only for the respective diagnosis no entries for redundant symptoms S_i can be created. Here we need an alternative weighting v_{ij}. In PATDEX/2 we define $\forall i, j : v_{ij} = 1$, which is motivated by the above mentioned restriction of U_1.

By the use of these definitions we get a similarity measure *sim* which is depending on the values represented in the relevance matrix. After each erroneous diagnosis the weights of the relevance matrix are changed. Thus, the similarity measure *sim* is the result of an adaptive learning process.

2.3.1 Test Selection

As opposed to other known case-based systems which concentrate on the aspect of classification PATDEX/2 uses case-based mechanisms for classification as well as for test selection. In PATDEX/2 the case-focussing test selection procedure is extended by a case- based one [5].

[5] This subcomponent of PATDEX/2 is a case-based reasoning system of its own where strategy cases are used which can be automatically generated out of the known diagnostic cases. As it is an improvement of the experience graph and, beyond that, the cost estimation procedure can be viewed as a kind of graph interpretation, we maintain the denotation experience graph for PATDEX/2 for reasons of simplicity.

This is globally optimal as compared with the already known (strategy) cases. In PATDEX/2 a fixed limit exists concerning the number of representable strategy cases. This helps to deal with the exponential complexity of the procedure (for the worst case all possible subsets of symptom values have to be represented)[6]. If the limit is reached the more typical cases will displace the less typical ones. PATDEX/2 uses an A*-like cost estimation algorithm for solving the conflict to choose from among several comparably similar startegy cases. If PATDEX/2 cannot find a sufficiently similar case, a case-focusing test selection procedure, such as in PATDEX/1, will be applied.

3 Discussion and Evaluation

Well-known case-based reasoning approaches which can be compared to PATDEX are, among others, the PROTOS ([19, 7, 8]), CASEY [15], CREEK [1] and memory-based reasoning, e.g. MBRTALK [25], approaches. In applying case- and analogy-based reasoning techniques to the problematic nature of test selection and in intergrating connectionist methods for adaptive learning, PATDEX goes beyond the state of the art defined by these (and similar) systems, respectively.

PATDEX is similar to the memory-based reasoning approach but, it combines it with the use of additional background knowledge. PATDEX not only exploits causal background knowledge, as CASEY does, but also functional knowledge. PATDEX is a computational model which is fully implemented, unlike CREEK. Especially PATDEX/1 has been tested since early 1989. Because of its adaptive capabilities PATDEX can be easily used for other domains within the field of technical diagnosis. This is not so easily done for PROTOS as many numerical values and the relational structure needed for the explanation facility have to be transferred to the new domain.

4 Conclusion

PATDEX has its well-defined role within the MOLTKE workbench. Thus, PATDEX is fully integrated in an overall view of knowledge acquisition and integration as well as learning. Having this view in mind PATDEX defines a new state of the art for case-based reasoning systems (with the restriction to the field of technical diagnosis).

In summary, PATDEX closes the gap between the general theoretical basis provided by similarity, uncertainty, rough sets and the associated techniques on the one side and the requirements of real applications on the other side.

5 Acknowledgments

The authors are indebted to Klaus Althoff for various discussions. He also was involved in the integration of PATDEX into the MOLTKE projects (cf. [2]) which in main parts was created by him. We thank S. Nökel and Mike Stadler for correcting an earlier version of this paper.

[6] In practice, only a small subset of the possible strategy cases occur. Thus, in spite of the limitation of the number of strategy cases good test selections can be achieved.

References

[1] Aamodt, A.: A Computational Model of Knowledge-Intensive Learning and Problem Solving, in: Proc. EKAW-90, pp. 1-20

[2] Althoff, K.-D.:Lernen aus Fallbeispielen zur Diagnose technischer Systeme doctoral dissertation, University of Kaiserslautern, 1991 (forthcoming)

[3] Althoff, K.-D., De la Ossa, A., Maurer, F., Stadler, M., Wess, S.: Adaptive Learning in the Domain of Technical Diagnosis, in: Proc. of Workshop on Adaptive Learning, FAW Ulm, July 1989

[4] Althoff, K.-D., Faupel, B., Kockskämper, S., Traphöner, R., Wernicke, W.: Knowledge Acquisition in the Domain of CNC Machining Centers: the MOLTKE Approach, in: Proc. of EKAW-89, pp. 180-195

[5] Althoff, K.-D., Maurer, F., Rehbold, R.: Multiple Knowledge Acquisition Strategies in MOLTKE, in: Proc. EKAW-90, pp. 21-40

[6] Althoff, K.-D., Wess, S.: PATDEX/2: Case-Based Knowledge Acquisition in Moltke, Technical Report, University of Kaiserslautern, 1991 (forthcoming)

[7] Bareiss , R., Branting, K. Porter, B.: The role of explanation in exemplar-based classification and learning, in Proc. Case-Based Reasoning, AAAI, 1988.

[8] Bareiss, R.: Exemplar-Based Knowledge Acquisition, Academic Press Inc, 1989.

[9] Carbonell, J.G.: Learning by analogy: formulating and generalizing plans from experience, in: Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (Eds.): Machine Learning, Tiogo Publishing Co., Palo Alto, 1983

[10] Carbonell, J.G.: Derivational Analogy in Problem Solving and knowledge acquisition, in: Michalski, R.S., Carbonell, J.G. , Mitchell, T.M. (Eds.): Machine Learning, Vol II, M. Kaufmann, 1986

[11] Gick, M. L., Holyoak, K. J.: Analogical Problem Solving, Cognitive Psychology, Vol. 12, 1980, pp. 306-355

[12] Hall, R.: Computational Approaches to Analogical Reasoning: A Comperative Analysis, in: Artificial Intelligence 39, 1989

[13] Hammond,K.:Case-Based Planning, Academic Press Inc, 1989.

[14] Kolodner,J: Maintaining organization in a dynamic long-term memory, Cognitive science, Vol 7, 243-280,1983.

[15] Koton, P.: Reasoning about evidence in causal explanations, Proc. AAAI-88, pp. 256-261

[16] Nökel, K.: Temporal Matching: Recognizing Dynamic Situations from Discrete Measurements, in: Proc. IJCAI-89

[17] Pawlak, Z: Rough Classification. Int. J. of Man-Machine Studies 20 (1984), p. 469-483.

[18] Pawlak,Z.: Decision Logic. Preprint Warszawa 1990.

[19] Porter, B.: PROTOS; an experiment in knowledge acquisition for heuristic classification tasks, in Proc. First Intern. Meeting on Advances in Learning, Les Arcs, France, 1986

[20] Quinlan,J.R.: Induction of Decision Trees. Machine Learning 1 (1986), p.81- 106.

[21] Rehbold, R.: Model-Based knowledge acquisition from Structure Descriptions in a Technical Diagnosis Domain, in: Proc. Avignon-89.

[22] Rehbold, R.: Integration modellbasierten Wissens in technische Diagnostik– Expertensysteme, dissertation, University of Kaiserslautern, 1991 (forthcoming)

[23] Richter, M. M., Pfeifer, T., Althoff, K.-D., Faupel, B., Nökel, K., Rehbold, R.: final report project X6, SFB 314 "AI - Knowledge-Based Systems", Kaiserslautern: 1990

[24] Rumelhart, D.E., Zipser, D.: Feature Discovery by Competitive Learning, Cognitive Science 9, 75-112.,1985.

[25] Stanfill, C., Waltz, D.: The memory-based reasoning paradigm, Proc. DARPA Workshop on Case-Based Reasoning, Morgan Kaufmann, 1988

[26] van Someren, M. W., Zheng, L. L., Post, W.: Cases, Models or Compiled Knowledge; a Comparative Analysis and Proposed Integration, Proc. EKAW-90

[27] Wolstencroft, J: Restructuring, Reminding and Repair: What's missing from Models of analogy, in: AI Communications, Vol2, No. 2, 1989

[28] Wess,S.: PATDEX/2: Ein System zum adaptiven, fallfokussierenden Lernen in technischen Diagnosesituationen. SEKI-Working-Paper, SWP-1-91, Kaiserslautern 1991 (in german).