Case-Based Reasoning and Adaptive Learning in the MOLTKE$_3$ Workbench for Technical Diagnosis[1]

Klaus-Dieter Althoff, Frank Maurer & Stefan Weß

University of Kaiserslautern - Dept. of Computer Science
P.O. Box 3049, D-6750 Kaiserslautern, Germany
Phone: ++49 631/205-3360
e-mail: *althoff*(*maurer* or *wess*)@informatik.uni-kl.de

Abstract

In this paper, we deal with the support of integrated knowledge acquisition workbenches by case-based reasoning techniques. We illustrate that the problem of test selection has to be taken into account for the diagnosis of technical systems. We offer a possible solution which enables both the utilization of all the well-known advantages of case-based reasoning systems and the avoidance of its also known drawbacks. We concentrate on a case-based reasoning system which has its well-defined role within a fully implemented knowledge acquisition workbench for technical diagnosis, namely the processing of temporary and absolute exception cases. It is able to utilize all the workbench's qualities, such as knowledge about the underlying technical system, its analogy-based rule generator, and its heuristic classificator. To meet the requirements which evolve from real world applications, our case-based reasoner deals not only with the classification problem of technical diagnosis, but also with that of test selection. Additionally, it learns the relevances of the symptoms for the respective diagnoses which enable the realization of an adaptive similarity measure. Having all this characteristics in mind our case-based reasoning approach defines a new state of the art for case-based reasoning systems (with the restriction to the field of technical diagnosis).

1 Motivation and Introduction

Knowledge acquisition workbenches which are able to integrate different knowledge sources are a hot research topic. Within this broad research area we want to focus on the support of specialized workbenches for technical diagnosis by the use of case-based reasoning techniques. We illustrate that the problem of test selection has to be taken into account for the diagnosis of technical systems. We have implemented the PATDEX[2] system[3] which learns from diagnostic cases, i.e. protocols of the diagnostic behavior of

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

2 PATtern Directed EXpert Systems

3 Actually, there are two systems, PATDEX$_1$ and PATDEX$_2$. By PATDEX (or the PATDEX

an experienced service technician. It has its well-defined role as a case-based reasoner within the MOLTKE3 workbench[1] (for technical diagnosis), namely the (interactive) processing of temporary and absolute exception cases[2]. It exploits the well-known advantages of case-based reasoning approaches, such as improving the system's transparency with respect to the expert's learning behavior by applying techniques like analogical problem solving and learning by memory adaptation[3]. Additionally, it avoids the well-known drawbacks of index generation and the processing of a large number of cases. Cases which have a corresponding part within the actual knowledge base were, e.g., used by the GenRule component of the workbench to generate heuristic rules for the refinement of the knowledge base (cf. [7], [6]; cf. Fig. 1). Thus, a PATDEX case base is only a supplementation of a MOLTKE knowledge base of the workbench's diagnosis shell.

Since PATDEX is an integral part of the MOLTKE3 workbench it can take advantage of all its causal[4] and functional[5] background knowledge to improve its similarity measuring capabilities (cf. Fig. 1).

In course of time PATDEX adapts its classification and test selection abilities to the cases it knows. Thus, it can learn good classifications as well as adequate test selections. Especially the latter ability represents real research progress within the field of machine learning because the main task being dealt with is that of classification.

Since the user's acceptance depends heavily on the control of the diagnostic process it is of fundamental importance, particularly for real world applications in technical domains. Therefore the MOLTKE project bases on the following view of diagnosis: "Diagnosis = Classification + Test Selection".

The idea behind the PATDEX approach is to describe the underlying knowledge structures on a cognitive level. This level has to be distinguished from a representation- and an implementation-oriented one. Thus a distinction is made between three levels of description which is sufficient for our purposes here. The underlying basic hypotheses of our approach - i.e. that learning by memory adaptation and analogical reasoning are fundamental techniques which human beings use during problem solving - are part of the more comprehensive notion of a "case-based architecture for next generation knowledge acquisition systems". We believe that using these techniques as the system's central mechanisms maximizes the system's transparency in regard to the expert's learning behavior, increases the user's acceptance concerning the system and decisively improves

---

approach) we denote all the information which relates to both systems.

1 MOdels, Learning and Temporal Knowledge in Expert Systems for Technical Diagnosis

2 Exception cases do not have a corresponding part in the actual MOLTKE knowledge base. They are *temporary* if this state will change during the further knowledge acquisition process. *Absolute* exception cases are real exceptions, i.e. exceptions with respect to the knowledge of the expert.

3 Learning by memory adaptation encompasses storing and updating of individual experiences and statistical information.

4 As causal knowledge we use general qualitative technical knowledge which we denote by *Qualitative Engineering* (as opposed to Qualitative Physics which follows the terminology of physicists), e.g.: "relais control valves, valves control the hydraulics etc. "

5 Functional knowledge is deep causal knowledge for the representation of the behavior of the underlying technical system which allows the simulation of it.

the knowledge acquisition support. In this paper we will try to give an intuition of PATDEX which is a first (improvable) approach with respect to the above mentioned case-based architecture. Nevertheless, having all its characteristics in mind PATDEX defines a new state of the art for case-based reasoning systems (with the restriction to the field of technical diagnosis).

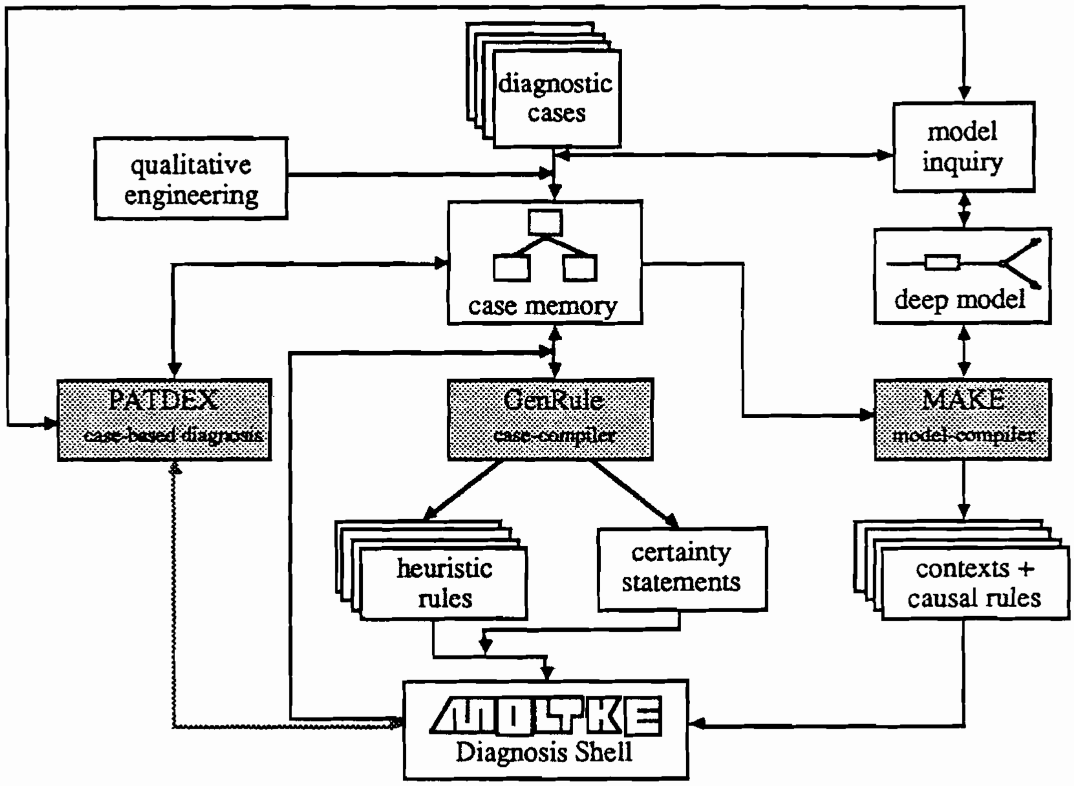

Fig. 1 – The Learning Component in the MOLTKE$_3$ Workbench[1]

In course of time PATDEX adapts its classification and test selection abilities to the cases it knows. Thus, it can learn good classifications as well as adequate test selections. Especially the latter ability represents real research progress within the field of machine learning because the main task being dealt with is that of classification.

Since the user's acceptance depends heavily on the control of the diagnostic process it is of fundamental importance, particularly for real world applications in technical domains. Therefore the MOLTKE project bases on the following view of diagnosis: "Diagnosis = Classification + Test Selection".

The idea behind the PATDEX approach is to describe the underlying knowledge structures on a cognitive level. This level has to be distinguished from a representation- and an implementation-oriented one. Thus a distinction is made between three levels of description which is sufficient for our purposes here. The underlying basic hypotheses

---

1 The learning component is supplemented by the MAKE (Model-Based Automated Knowledge Extractor) system ([20], [21]) which is able to generate a basic MOLTKE knowledge base out of a deep functional model of the technical system under consideration.

of our approach - i.e. that learning by memory adaptation and analogical reasoning are fundamental techniques which human beings use during problem solving - are part of the more comprehensive notion of a "case-based architecture for next generation knowledge acquisition systems". We believe that using these techniques as the system's central mechanisms maximizes the system's transparency in regard to the expert's learning behavior, increases the user's acceptance concerning the system and decisively improves the knowledge acquisition support. In this paper we will try to give an intuition of PATDEX which is a first (improvable) approach with respect to the above mentioned case-based architecture. Nevertheless, having all its characteristics in mind PATDEX defines a new state of the art for case-based reasoning systems (with the restriction to the field of technical diagnosis).

The following chapter provides the necessary terminology for the description of the diagnostic background within the $MOLTKE_3$ workbench. Chapter 3 describes the first version of PATDEX, while Chapter 4 gives an overview of the implemented extensions concerning adaptive learning and background knowledge ($PATDEX_2$). Finally, we compare PATDEX to other case-based reasoning approaches and discuss its underlying representation and mechanisms with respect to competing technical diagnosis techniques.

2 Terminology of the $MOLTKE_3$ Workbench

A *symptom (symptom class)* relates a name to a list of possible values (e.g. Valve --> {open, closed}) whereas *symptom values (symptom instances)* reflect the actual state of a part of the technical device (e.g. Valve 21Y5 --> open). They are ascertained by the use of *tests*. Their actual value may be either *unknown* or an element of the possible value list in the corresponding symptom (class).

The list of all symptom instances is called a *situation*. In the context of predicate calculus the actual situation is the base for the interpretation of a *language of formulas*. It stores the bindings of the variables[1]. We use a three-valued logic with *TRUE*, *FALSE* and *UNKNOWN*.

A *case* describes a situation and the solution for that situation. In particular we distinguish two different solutions: a proved fault (a *diagnostic case*) and a selected test (a *strategy case*).

3 $PATDEX_1$: Case-Based Reasoning for Technical Diagnosis

In this section we will describe $PATDEX_1$, an expert system for fault diagnosis on CNC machining centers which is able to learn while being used. 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 empirical knowledge is combined with

1 Every symptom instance is a variable in the calculus.

another one that focuses on similarities. The overall process of diagnosis is based on the analogical problem solving algorithm (APS) proposed by [15]. The process is started by the user giving some observed symptoms, the (first) actual situation, as input into the system. Our simplified view of the APS-algorithm is now easily described by a loop which contains two main steps:

- retrieve the most suitable case
- ask for a new symptom value

The loop terminates if the case retrieved from the case base reveals to be sufficiently good for diagnosis or if there is no case left which is "good enough" for pursuing the diagnosis. The notion of a "good case" as well as the meaning of "the most suitable case" in this context will be clarified. For this purpose we will first describe the selection of cases by means of a similarity measure. After that, we discuss how the system's experience is used to influence a given selection.

3.1 Similarity as a Criterion for the Evaluation of Cases

After asking a new symptom value, a similarity measure is evaluated for each case in the case base which has not been disqualified before, i.e. of which the value exceeds the dissimilarity-threshold of 0. The similarity measure **SIM** is a function that evaluates the sum of the following weighted parameters:

- the number **U** of symptoms contained in the case of which values are (yet) unknown
- the number **A** of symptoms which are contained in the case as well as in the situation
- the number **D** of symptoms where the value proposed by the case differs from that known in the situation
- the number **N** of symptoms contained in the situation but not in the case

The general definition of **SIM** is as follows:

$$\mathbf{SIM} = \mathbf{SIM}\,(\mathrm{Sit}_{\mathrm{actual}}, \mathrm{Sit}_{\mathrm{case}}) = \frac{\alpha \mathrm{A} + \beta \mathbf{D} + \gamma \mathbf{U} + \delta \mathrm{N}}{\mathbf{A} + \mathbf{D} + \mathbf{U} + \mathbf{N}}$$

The parameters α, β, γ, and δ can be chosen arbitrarily. Our experiments have led us to the following values: $\alpha = 1$, $\beta = -2$, $\gamma = -1/2$, $\delta = -1/2$

SIM is normed to [-2,1]. Its defensive, pessimistic character has been the motivation for it. A high value is set on different symptom values, i.e. it is important to avoid wrong diagnoses.

If the value assigned to a given case by the similarity measure exceeds a lower bound (hypothesis-threshold), this case is said to be qualified for further processing. If the

value exceeds an upper bound it is even qualified as diagnosis (diagnosis-threshold). Both thresholds are locally defined for each case of the case base. The value of the similarity measure cannot exceed 1. 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 the case do not hold, given a situation encountered during diagnosis, or 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.

3.2 The Use of the System's Experience

The system's experience is represented by means of a weighted directed graph, called the *experience graph* (cf. Fig. 2). While the nodes in this graph represent situations, the weights of the directed edges between these nodes represent the conditional probability of one situation (represented by the end node of the specific arc) occurring next in the diagnostic process under the assumption that another situation (represented by the start node of the arc) describes the current one[1]. Each situation is represented by one node at most. The sum of the weights of the edges starting at the same node is exactly one. Every situation containing the same symptoms as a particular case is associated to that case.

When starting up $PATDEX_1$ with an entirely new case base, the first action taken by the system is to build up a graph containing all situations which describe cases, i.e. that contain the same symptoms as an arbitrary case contained in the case base. The resulting graph will also include all nodes representing intersections between the situations mentioned above. Edges contained in the newly built graph will have weights that describe the frequency of situations occurring under the assumption that other situations occurred.

Every time the situation changes (e.g. a new symptom is ascertained) the statistical information represented in the network is used to find out the case which by prior experience is best suited to explain the given situation. This task is accomplished by running a heuristic-driven search through the graph. The result of the search process will be a situation and the case associated to that situation will be the one looked for. Another application of the experience represented in the graph is to find out which symptom values can be determined by a set of known values. This knowledge is used to optimize the order of tests in a given situation.

1 The role of the weights is similar to that of certainty factors for probabilities where the underlying distribution function is not known, too.

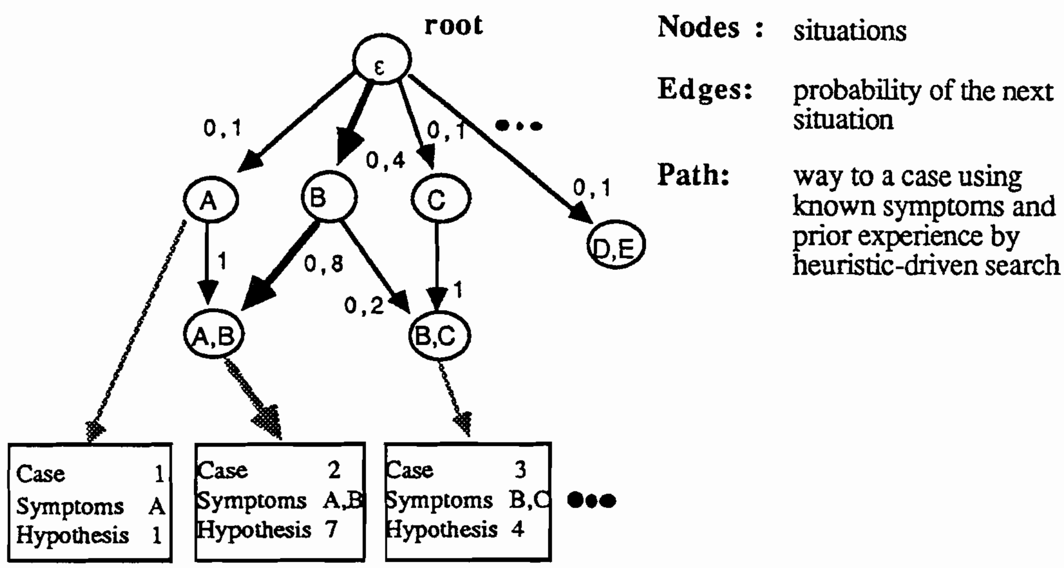

Fig. 2 - Representation of Empirical Knowledge in PATDEX$_1$

3.3 Joining the Results of the System's Experience with the Results Obtained by Measuring Similarity

Given an actual situation, the values computed by the similarity measure induce an ordering on the known cases. If there is no case whose associated similarity value exceeds the associated lower bound, the attempt to recognize the given fault fails and the system stops without finding a diagnosis. However, normally there will be cases for which the similarity measure exceeds their individual lower bound. Given the latter situation, a bonus, depending on the determined symptom values, will be added to the similarity value of the case proposed by system experience. The resulting value can be thought of as a score. Cases whose score now exceeds their associated upper bound are considered "good enough for diagnosis". If no case reveals to be "good enough", the APS-loop continues. Under these circumstances, "the most suitable case" will be chosen from among the cases which have the highest occurring score. This is done by running a conflict-solving heuristic that takes into account the following factors:

- distance between similarity value and upper bound
- costs of finding out the values still needed to complete the case
- consequences of a wrong diagnosis based on the case
- relative frequency of the case
- number of times the case has been chosen as hypothesis

If there are cases which are "good enough for diagnosis", the heuristic described above is used to choose one of those cases, too. The solution contained in the selected case is proposed as the diagnosis which the user may disbelieve, accept or value.

3.4 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 $PATDEX_1$ 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 valuated 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.

3.5 Evaluation of $PATDEX_1$

$PATDEX_1$ is a stand-alone protoype which has been completely implemented before the completion of the $MOLTKE_3$ 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" [13] can be elegantly applied to the field of technical diagnosis. Being confronted with the engineer who was engaged in our project (but not in the development of $PATDEX_1$), $PATDEX_1$ came off very well. In particular this is true for the similarity measure which has been defined in section 3.1.

Shortcomings of $PATDEX_1$ are the difficulty to generalize the similarity measure and the fact that the case-focusing test selection is not necessarily globally optimal. For CNC machining centers the experience graph´s super-exponential complexity concerning space and time (for the worst case all sequences of symptom values have to be represented) enables the processing of exception cases but, $PATDEX_1$ cannot handle the whole diagnostic task all alone. As $PATDEX_1$ takes no advantage of causal or functional background knowledge it could make the wrong diagnosis if too many redundant symptom values are given or some relevant ones are missing.

4 $PATDEX_2$: Adaptive Learning for Technical Diagnosis

$PATDEX_2$ is an integral part of the $MOLTKE_3$ workbench which allows the utilization of all its qualities. 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 abnormal symptom values. Thus, redundant information can be filtered off and no wrong diagnosis has to be made because

of this. By the exploitation of functional background knowledge additional symptom values can be derived out of the known ones. In this manner the selection of the actually most similar case is considerably speeded up.

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. These relevances are represented by means of a *relevance matrix* where the symptoms and diagnoses 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 $PATDEX_2$, or they are entered during an optional training phase by the expert himself. For the degree of relevance of a certain symptom, it is important whether it is a consequence of the normal functioning of the technical system, e.g. "relais 21K3 switched", or not, such as "voltage 214 too high". The latter are associated with a constant high weight.

In $PATDEX_2$ the case-focusing test selection procedure is extended by a case-based one[1]. 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)[2]. 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.

5 Discussion and Evaluation

We give an evaluation of PATDEX and state its relation to other approaches within the area of case-based reasoning (because further approaches are beyond the scope of this paper). Additionally, we describe the state of implementation of PATDEX and all concerned components of the $MOLTKE_3$ workbench.

5.1 Evaluation

PATDEX cannot be evaluated independent of the $MOLTKE_3$ workbench (cf. Fig. 1). It is not expected that it can carry out "arbitrary" learning tasks within a technical diagnosis situation. The attractiveness of the PATDEX approach are its restriction to the processing of (temporary and absolute) exception cases and its supplementation with the case compiler GenRule and the heuristic classification ability of the MOLTKE shell,

1 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 (cf. chapter 2). 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.

2 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.

respectively. In the $MOLTKE_3$ workbench the applied learning strategies have to support the generation of an implementation model for the underlying technical system, based on a design model [11] for technical diagnosis. As compared with the BLIP approach [18] this is a simplifying view because no domain model is learned, but we think that our approach is adequate for the field of technical diagnosis.

From a technical diagnosis point of view PATDEX can be seen as a heuristic-driven associative expert system which is based on partial matching of concrete cases. It uses a combination of a hypothesize-and-test strategy and a differential diagnosis strategy. It employs a Rete-like symptom network for an efficient case handling which is necessary for real world applications. As all cases are categorized based on the similarity measure, an efficient updating of the similarity value of all necessary (i.e. all candidate and touched) cases is possible.

For the considered CNC machine about 5.000 failures (and some more symptoms) can be identified. Though being of exponential complexity in space and time, the experience graph of $PATDEX_1$ is able to handle the exception cases for such a machine because only a few permutations of the symptom orders of the known cases appear in reality. As too many cases would be necessary for achieving a good performance $PATDEX_1$ cannot work as a stand-alone system. Thus, $PATDEX_2$ is an improvement here, because its test selection component can utilize the decomposability of the domain and, additionally, similarities between strategy cases. For a detailed estimation of the complexity of the involved algorithms confer ([9], [2]).

The quintessence of all the applications being realized so far[1] is that the shell can be considered as successful. PATDEX supplements this shell in a natural way. It can be applied to other domains (of technical diagnosis) if they are decomposable in a similar way as the diagnosis of CNC machining centers (which is typical for comparable technical systems).

5.2 State of Realization (Dec. 1990)

$PATDEX_2$ is fully implemented and integrated into the $MOLTKE_3$ workbench ([9], [2]). $PATDEX_1$ as a stand-alone prototype is already available since early 1989 ([5], [3]). The MOLTKE shell [6] and the MAKE system ([20], [21]) are fully implemented, too.

5.3 Classification and Related Work

PATDEX is a learning-apprentice system which learns in a closed loop. Using Wolstencroft's model of analogy [25] PATDEX can be described as follows:

0) Identification
   Being implemented before the completion of the $MOLTKE_3$ workbench $PATDEX_1$ only reasons by analogy and therefore no further reasoning alternatives have to be

[1] Four complex knowledge bases up to now.

considered. This has changed for $PATDEX_2$ which is an integral part of the workbench. For $PATDEX_2$ the MOLTKE shell's heuristic classification is a second reasoning alternative.

1) Retrieval
   $PATDEX_1$ retrieves the most similar case based on the similarity measure and the experience graph. $PATDEX_2$ utilizes its adaptive similarity measure and functional knowledge, included in the shell, to derive additional symptom values for the expansion of the given situation.
2) Elaboration
   Since PATDEX is specialized on technical diagnosis, it uses a simple but sufficient case description language and focuses on the necessary parts of the known case without further reasoning. $PATDEX_2$ is able to learn the most relevant parts here.
3) Mapping
   As the underlying notions for base and target are identical, no mapping has to be done. For an extension of this confer [14].
4) Inference
   Based on the given target situation, all remaining symptoms of the most similar known case, including the diagnosis[1], are hypothesized for the target case. Besides, $PATDEX_2$ can derive additional symptom values based on its case-based test selection procedure.
5) Justification
   PATDEX performs a plausibility check based on its diagnostic reasoning capabilities, the similarity measure, its experience, the causal and functional background knowledge of the shell, and, finally, the user's feedback, if a diagnosis has been chosen.
6) Consolidation/Learning
   The new case (or its correction) is memorized and all necessary thresholds and weights are updated.

Well-known case-based reasoning approaches which can be compared to PATDEX are, among other ones, the PROTOS [10], CASEY [16], CREEK [1] and memory-based reasoning, e.g. MBRtalk, [23] approaches. In applying case- and analogy-based reasoning techniques to the problematic nature of test selection PATDEX goes beyond the state of the art defined by these (and similar) systems.

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 transfered to the new domain.

[1] With respect to the classification task the analogy mechanism of PATDEX is similar to transformational analogy [12].

6 Further Research Work

Within the MOLTKE project research focuses on specialized knowledge-intensive systems in the domain of mechanical engineering which we call Intelligent Engineering Systems. A general overview of this is given in [6], while [19], [20], [4], and [17] stress the aspects of time representation, deep modeling, knowledge acquisition and maintainance, respectively. The MOLTKE shell is the result of a several years' cooperation with a globally acknowledged mechanical engineering institute. It meets all the requirements that have been posted by the institute [22]. An excellent overview over the problematic nature of knowledge integration is given in [24]. The results and integration proposals impressively underline the quality of the MOLTKE3 workbench.

7 Conclusion

PATDEX has been designed for real world applications in the field of technical diagnosis and has its well-defined role within the MOLTKE3 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).

8 Acknowledgments

Thanks go to especially to Prof. Dr. M. M. Richter, and our machine learning research group at Kaiserslautern, for many very engaged discussions which have been an important step for the development of the PATDEX systems as they are described here. We thank R. Rehbold and S. Nökel for correcting an earlier version of this paper. The experience graph has been implemented by M. Stadler.

9 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.: Eine fallbasierte Lernkomponente als integrierter Bestandteil der MOLTKE3-Werkbank zur Diagnose technischer Systeme, doctoral dissertation, University of Kaiserslautern, 1990 (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., Kockskämper, S., Maurer, F., Stadler, M., Weß, S.: Ein System zur fallbasierten Wissensverarbeitung in technischen Diagnosesituationen, in: Proc.

Austrian Conference on AI, 1989, pp. 65-70
[6] Althoff, K.-D., Maurer, F., Rehbold, R.: Multiple Knowledge Acquisition Strategies in MOLTKE, in: Proc. EKAW-90, pp. 21-40
[7] Althoff, K.-D., Maurer, F., Traphöner, R., Weß, S.: Die Lernkomponente der MOLTKE3-Werkbank zur Diagnose technischer Systeme, in: KI, special edition on Machine Learning, Munich: Oldenbourg Verlag, No. 1, 1991
[8] Althoff, K.-D., Traphöner, R.: GenRule: Learning of Shortcut-Oriented Diagnostic Problem Solving in the MOLTKE3 Workbench, Technical Report, University of Kaiserslautern, 1990
[9] Althoff, K.-D., Weß, S.: PATDEX2: Case-Based Knowledge Acquisition in MOLTKE, Technical Report, University of Kaiserslautern, 1990 (forthcoming)
[10] Bareiss, R.: Exemplar-Based Knowledge Acquisition, Academic Press, 1989
[11] Breuker, J., Wielinga, B.: Model-Driven Knowledge Acquisition: Interpretation Models, Memo 87, Deliverable task A1, Esprit Project 1098; 1987
[12] 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
[13] 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
[14] De la Ossa, A.: AKILES: An Approach to Automatic Knowledge Integration in Learning Expert Systems, Technical Report, University of Kaiserslautern, 1990
[15] Gick, M. L., Holyoak, K. J.: Analogical Problem Solving, Cognitive Psychology, Vol. 12, 1980, pp. 306-355
[16] Koton, P.: Reasoning about evidence in causal explanations, Proc. AAAI-88, pp. 256-261
[17] Maurer, F.: Knowledge Base Maintainance and Consistency Checking in MOLTKE, submitted to: IJCAI-91, 1990
[18] Morik, K.: Sloppy Modeling, in: Morik, K. (ed.): Knowledge Representation and Organization in Machine Learning, Springer Verlag, 1989
[19] Nökel, K.: Temporal Matching: Recognizing Dynamic Situations from Discrete Measurements, in: Proc. IJCAI-89
[20] Rehbold, R.: Model-Based knowledge acquisition from Structure Descriptions in a Technical Diagnosis Domain, in: Proc. Avignon-89
[21] Rehbold, R.: Integration modellbasierten Wissens in technische Diagnostik--Expertensysteme, dissertation, University of Kaiserslautern, 1990 (forthcoming)
[22] Richter, M. M., Pfeifer, T., Althoff, K.-D., Faupel, B., Nökel, K., Rehbold, R.: final report project X6, SFB "AI - Knowledge-Based Systems", Kaiserslautern: 1990
[23] Stanfill, C., Waltz, D.: The memory-based reasoning paradigm, Proc. DARPA Workshop on Case-Based Reasoning, Morgan Kaufmann, 1988
[24] van Someren, M. W., Zheng, L. L., Post, W.: Cases, Models or Compiled Knowledge; a Comparative Analysis and Proposed Integration, Proc. EKAW-90
[25] Wolstencroft, J.: Restructuring, Reminding, Repair: What´s missing from Models of Analogy, in: AI Communications, Vol. 2, No. 2, June 1989