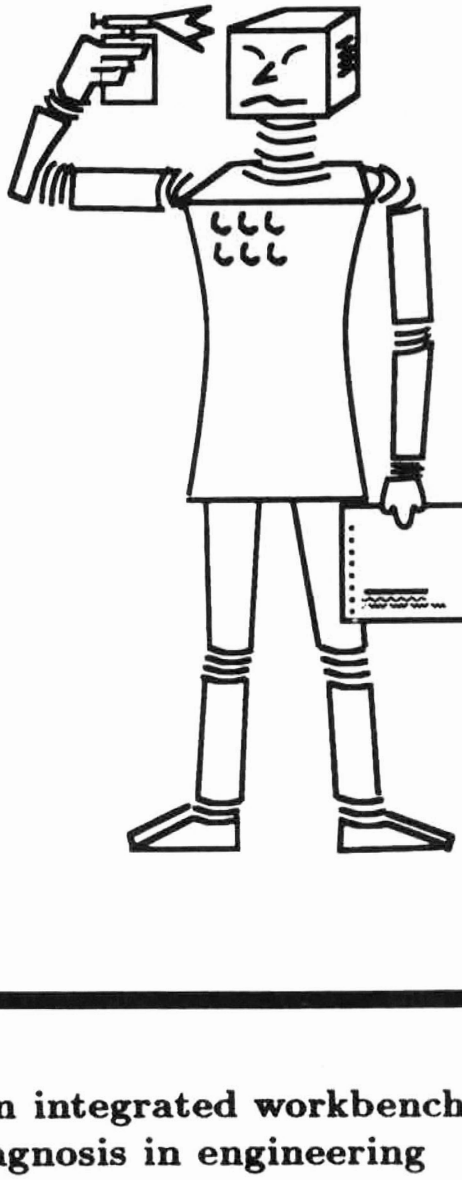
Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern

SEKI - REPORT



MOLTKE - an integrated workbench for fault diagnosis in engineering systems

K.D. Althoff, F. Maurer, S. Wess

SEKI Report SR-92-16 (SFB)

MOLTKE - AN INTEGRATED WORKBENCH FOR FAULT DIAGNOSIS IN ENGINEERING SYSTEMS

Klaus-Dieter Althoff, Frank Maurer,
Stefan Wess
Universität Kaiserslautern
AG Prof. M. M. Richter
Postfach 3049
6750 Kaiserslautern (Germany)

Ralph Traphöner
tecInno
Gesellschaft für innovative Software–Systeme und Anwendungen mbH
Sauerwiesen 2
6750 Kaiserslautern 25 (Germany)

ABSTRACT[1]

We present the MOLTKE workbench for technical diagnosis, a fully implemented knowledge acquisition environment which has been tested on several real world applications (e.g. fault diagnosis of CNC machining centers). MOLTKE combines different techniques of second generation expert systems (qualitative, case-based, inductive reasoning). Thus, it is able, to process knowledge from the three main sources of technical diagnosis tasks: knowledge about the function and behavior of the machine (technical model), abstract diagnostic knowledge from the expert (mental model), as well as episodic knowledge of service technicians (cases).

INTRODUCTION

Fault diagnosis in engineering systems is a domain of extraordinary economical importance. Currently, many approaches and applications exist which try to handle diagnostic tasks, but there are still some unsolved problems. Due to task customization and rapid further developments, only few fully identical machines of a given type exist. This makes it difficult to find somebody who is really an expert for the maintenance of a special machine, especially if the type is relatively new.

Service technicians troubleshooting such a new machine cannot rely on machine-specific heuristics, but have to use general technical knowledge, a detailed description of structure and behavior of the machine, and a general understanding of the functionality of the device to find any faults. In the course of time, technicians learn more about a special machine type through the faults they are confronted with during their work (cases). Such empirical knowledge enables the refinement and application of machine-specific heuristics.

Important problems with existing (successful) diagnostic expert systems are that they are often built in a single-use-manner and need much effort for knowledge acquisition and maintenance. Thus, using existing approaches the overall development cycle is too slow for industrial real world applications. What's missing in current research, is an approach which is both application-oriented (in the sense of being usable for real world applications) and research-oriented (in the sense of integrating state-of-the-art methods). Such a system should be able to speed up the development of diagnostic expert systems to meet the requirements of the above described scenario.

Within this paper, we present an expert system architecture (technical diagnosis shell) which has successfully handled the problem of fault diagnosis of CNC (computerized numerical control) machining centers (as well as similar problems) on the level of fully implemented research prototypes which have been developed in cooperation with a mechanical engineering research institute (cf. Althoff, Faupel et al., 1989). The expert system architecture is embedded in a complete knowledge acqui-

1 also in: S. Hashemi, J. P. Marciano, and G Gouarderes (eds.), Proc. 4th International Conference Artificial Intelligence & Expert Systems Applications (EXPERSYS-92), Paris, 1992

sition workbench (the MOLTKE[2] workbench; cf. Althoff, Maurer & Rehbold, 1990; Althoff, Nökel et al., 1988; Althoff, 1992a+b) which makes use of the three main knowledge sources of the domain.
After the introduction of the basic workbench characteristics, we present its main subcomponents (which deal with these knowledge sources) during the following sections. Finally, we summarize the key features of the presented approach.

WORKBENCH: MODELLING KNOWLEDGE FOR DIAGNOSTIC TASKS

The MOLTKE workbench combines different techniques of second generation expert systems (qualitative, case-based, inductive reasoning). It is able, to process knowledge from the main sources of technical diagnosis tasks: knowledge about the function and behavior of the machine (technical model), abstract diagnostic knowledge from the expert (mental model; heuristic knowledge; abstract empirical knowledge), as well as episodic knowledge of service technicians (cases; concrete empirical knowledge). The workbench consists of four main components: a diagnostic object-oriented shell (MOLTKE Shell), a knowledge compiler which generates a (partial) knowledge base from construction plans of the machine (iMAKE), an inductive learning system for the extraction of strategic knowledge from cases (GenRule), and a case-based reasoning system which cooperates with the shell problem solver (PATDEX). Building a knowledge base with MOLTKE starts with the graphical modelling of subparts of the machine. Then MOLTKE generates a component-oriented, hierarchical model of the technical device and a qualitative, static description of the device´s behavior which is (incrementally) compiled into the knowledge base. This knowledge can be improved by the (incremental) generation of strategy knowledge from cases as well as manual knowledge acquisition techniques. Additionally, the case-based reasoning system offers a learning apprentice-like support.

MOLTKE SHELL: THE BASIC DIAGNOSTIC PROBLEM SOLVER

To represent the given knowledge sources, the MOLTKE workbench uses three different representation languages for describing cases, the functional model of the machine, and the knowledge manually entered by the expert, respectively. In principle, these languages have to be compiled into a representation formalism which is "understandable" by the underlying diagnostic problem solver. For reasons of simplicity, this diagnostic representation language is identified with that of the expert-entered knowledge. Thus, this description language functions as the terminological basis for the whole workbench (cf. Fig. 1).
The MOLTKE shell itself is a rule-based object-oriented expert system shell within which the knowledge about the diagnostic process can decomposed in a "horizontal" as well as in a "vertical" manner. For the latter one, the knowledge is split into *contexts* which correspond to rough, intermediate and final diagnoses and build up a directed acyclic graph, the *context graph*. Within a context, three kinds of knowledge can be differentiated: ordering rules (for choosing a reasonable order of tests), shortcut rules (for deriving unknown symptom values from known ones), and diagnostic rules (for representing the relations between symptoms and all kinds of diagnoses).
Within the shell, there exist additional components for the processing of temporally distributed symptoms (cf. Nökel, 1989; Nökel & Lamberti, 1991), for maintenance and consistency checking of MOLTKE knowledge bases, as well as the adaptation of such knowledge bases to changes in the underlying technical system (cf. Maurer, 1992; De la Ossa, 1991). The integration of hypermedia techniques is currently under development (cf. Traphöner & Maurer, 1992).

---

2 MOdels, Learning, and Temporal Knowledge in Expert systems for Engineering Domains

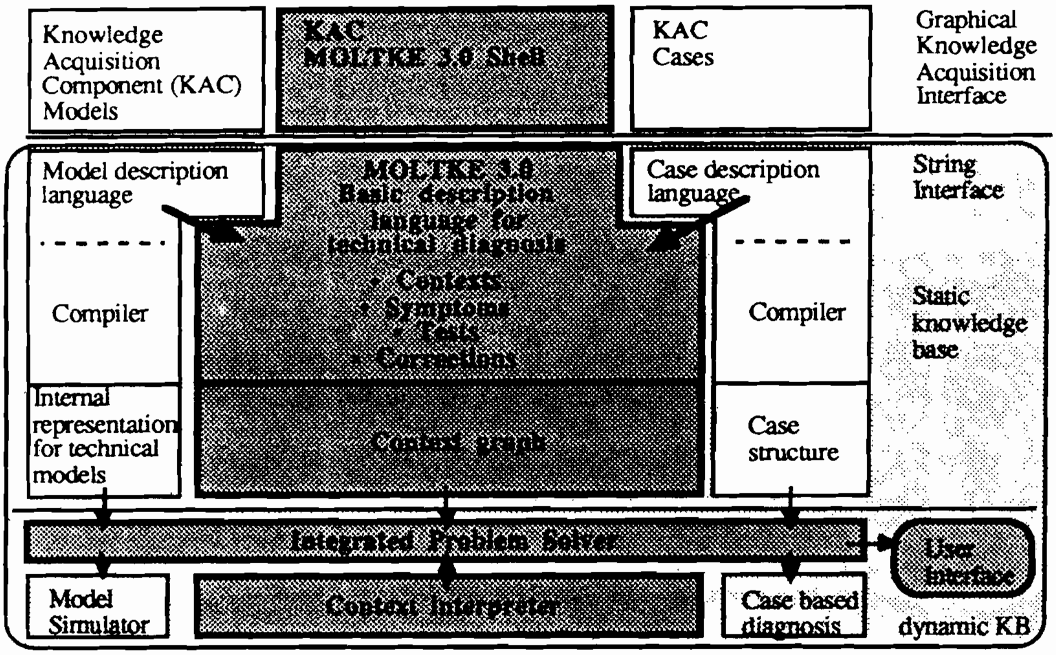


Fig. 1: The MOLTKE workbench

IMAKE: BUILDING THE BASIC EXPERT SYSTEM

iMAKE (cf. Rehbold, 1991) gets construction plans of the machine and uses its built-in knowledge of electrical and mechanical engineering (i.e. its "technical common sense") - e.g. the behavior of typical components - to form a (deep) model of the machine and generate an initial MOLTKE knowledge base. We call such an expert system without any machine specific heuristic knowledge *basic expert system.*
An important point is that the construction plans can be entered by anyone who is able to read them. There is no need for the special knowledge of an expert to do that. The expert is needed only to provide information about the typical atomic parts in the descriptions, e.g. valves, switches, contactors, hydraulic pistons.
The generation of a basic expert system then consists of four major steps:

- An expert builds up a library of the primitive (and may be common complex) components that are used. This task needs an expert since it should be done as precisely and completely as possible for the sake of the resulting expert system. Of course, the library can be used for many different machines. Thus, an expert is not needed for every single machine type or series.
- A person with technical skill - at least he/she shall be able to read, understand and reproduce technical structure descriptions - is needed to enter the connectivity of a given machine, i.e. to build up the structural model. He/she will select parts from the library and build them together to more complex ones finally describing the whole machine. A graphical editor that allows to mimic the construction plans helps to accomplish this step. Note that this task is independent from the availability of a diagnosis expert if the library (cf. step 1) completely covers all components used. With a fully computerized design, development and production system (computer integrated manufacturing: CIM)) it should be possible to get the connectivity information immediately from the design data.
- The designer of the machine can specify the desired functionality in form of input/output behavior.
- Finally, the iMAKE system transforms the model of the machine (put together from the library and the connectivity) with help of the desired input/output behavior into a knowledge base for the MOLTKE shell.

GENRULE: LEARNING OF STRATEGY KNOWLEDGE

In principle, the basic (iMAKE-generated) expert system is able to diagnose given fault situations. What´s still missing is strategic knowledge: how to find the underlying fault(s) as "economical" as possible. Up to now, the generated basic expert system can only apply uninformed (blind) search because strategic knowledge is not part of the respective knowledge source (construction plans). In spite of this, strategic knowledge can be directly entered by the expert using the shell or represented by the use of cases which reflect observed expert behavior.

The GenRule system (cf. Althoff, 1992a) is able to improve the generated basic expert system by the use of diagnostic cases. GenRule realizes an incremental inductive learning strategy which generates its learning hypotheses (shortcut rules, ordering rules) based on a memory of diagnostic cases and the generated knowledge base. Within the memory, symptom values and diagnoses are used for an efficient indexing. The basic idea of GenRule is to compare observed cases with existing diagnostic "paths" of the knowledge base. If a case includes a better strategy than the strategy known by the knowledge base (which results in a certain "path"), additional shortcut and ordering rules are generated. Both kinds of rules are used to represent a better diagnostic strategy, i.e. the classification ability of the basic expert system will not be influenced.

All diagnostic paths are automatically generated from the given knowledge base. The rules are indirectly generated via the integration of cases and/or paths into the case memory. The rules are included into the knowledge base if their evaluation based on a special statistic measure exceeds a given threshold. For these rules, the case memory functions as both a dependency network for efficiently updating the statistic rule justification as well as a means for automatically retracting and/or establishing rules within the knowledge base.

PATDEX: CASE-BASED DIAGNOSIS

Within real world applications, cases are an important knowledge source. While GenRule extracts abstract strategic knowledge from frequently occuring cases, PATDEX concentrates on the processing of the more exceptional cases. Since exceptional situations are "very informative", they must play an important role within the knowledge acquisition process. PATDEX interprets such cases directly using case-based reasoning and realizes an extension of both the workbench´s classification abilities as well as its strategic abilities. Actually, it is an alternative problem solver to the basic shell problem solver. Thus, the underlying case representation language is both an intermediate knowledge representation for further processing via GenRule or by hand as well as a final representation for case-based diagnosis.

The PATDEX system (cf. Althoff & Wess, 1991; Richter & Wess, 1991; Althoff, De la Ossa et al., 1989) consists of two subcomponents for classification and test selection, respectively. Both components rely on a case-based reasoning mechanism which bases on the similarity of the underlying cases. The similarity is interpreted by the use of evaluation functions (similarity measures). A procedure which dynamically partitions the case base enables an efficient computation and updating of the similarity.

In case of the classification subcomponent, the applied similarity measures are completely dynamical. The underlying evaluation function can be adapted to observed expert behavior using a connectionist learning technique (competitive learning). For the test selection component, the adaptation of similarity measures is based on an estimation of the average diagnostic costs using an A*-like procedure.

PATDEX can deal with redundant, incomplete, and incorrect cases and includes the processing of uncertain knowledge via default values for symptoms. All the knowledge being represented within the workbench can be used as background theory (e.g. for the identification of abnormal and/or relevant symptoms as well as the derivation of further symptom values). PATDEX offers a learning-apprentice-like support for the development and the maintenance of the case memory. It can also work as a stand-alone case-based diagnosis system.

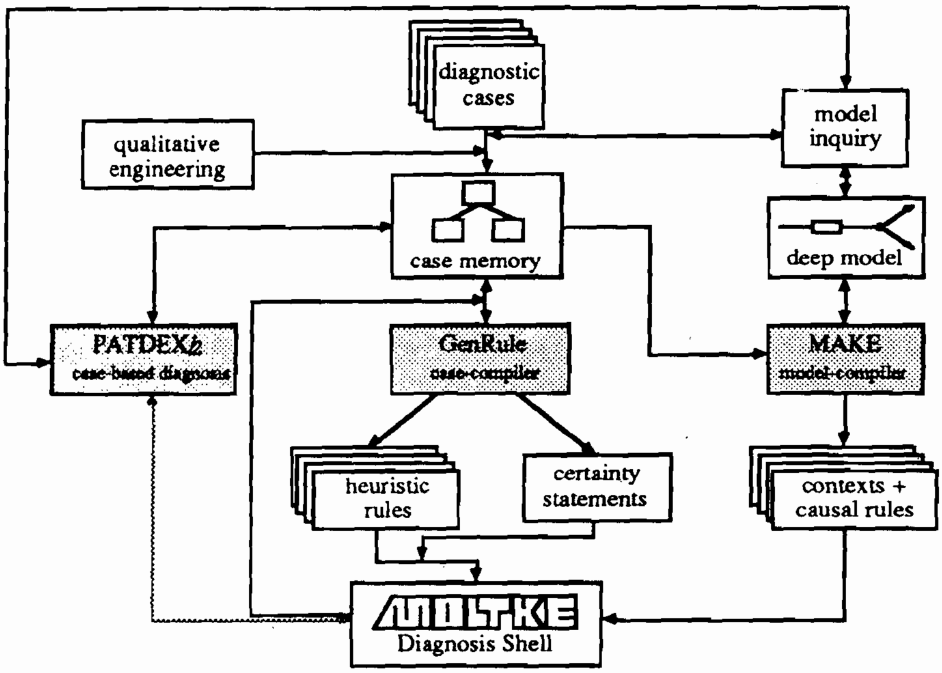


Fig. 2 : The organization of the MOLTKE workbench

CONCLUSION

The MOLTKE workbench is a fully implemented knowledge acquisition environment which is both application-oriented and research-oriented. It has been successfully applied to several real world domains. It integrates different knowledge representation formalisms, learning strategies and reasoning schemes. Thus, MOLTKE enables a flexible knowledge acquisition support which automates as much of the knowledge base development as *reasonable*. Of course, the automated tools underly certain restrictions, e.g.: iMAKE can mainly deal with hydraulic and electric/electronic components of technical devices. Since iMAKE uses a static description of the devices´ behavior, the machine should have a central control unit. The (space and time) complexity of GenRule´s case memory requires a strongly decomposable domain. PATDEX´s adaptive learning mechanism is restricted to the given (built-in) schema of the evaluation function. Normally, *fault diagnosis in engineering systems* describes a class of application domains which meet these requirements.

As an illustration of the dependencies of the above described components, the overall organization of the workbench is shown in Fig. 2.

Using the MOLTKE workbench we developed (e.g.) knowledge bases for fault diagnosis of CNC machining centers, 3D-CNC measuring machines, driving machines in mining, and for the diagnosis of failures in heterogenous computer networks. The MOLTKE system is implemented in Smalltalk80 on Sun-, Apollo- and HP-Unix-workstations. It runs on all machines for which the respective virtual machine for Smalltalk80 was implemented including DECstations, MacII, 80x86-PCs etc.

ACKNOWLEDGEMENT

The work presented herein was partially supported by the Deutsche Forschungsgemeinschaft, Sonderforschungsbereich 314: "Artificial Intelligence - Knowledge-Based Systems", projects X6 and X9. We would like to thank Michael M. Richter, Klaus Nökel, Robert Rehbold, Benedikt Faupel, Alvaro De la Ossa, Sabine Kockskämper, and Mike Stadler for a fruitful cooperation within the MOLTKE project.

REFERENCES

Althoff, K.-D. (1992a). A Case-Based Learning Component as an Integrated Part of the MOLTKE Workbench for Technical Diagnosis (in German: Eine fallbasierte Lernkomponente als integrierter Bestandteil der MOLTKE-Werkbank zur Diagnose technischer Systeme). *Doctoral Dissertation*, Kaiserslautern University (to appear)

Althoff, K.-D. (1992b). Machine Learning and Knowledge Acquisition in a Computational Architecture for Fault Diagnosis in Engineering Systems. Proc. International Conference on Machine Learning, Workshop "Computational Architectures for Supporting Knowledge Acquisition and Machine Learning" (to appear)

Althoff, K.-D., De la Ossa, A., Maurer, F., Stadler, M. & Wess, S. (1989). Adaptive Learning in the Domain of Technical Diagnosis. *Proc. Workshop on Adaptive Learning*, FAW Ulm

Althoff, K.-D., Faupel, B., Kockskämper, S., Traphöner, R. & Wernicke, W. (1989). Knowledge Acquisition in the Domain of CNC Machining Centers: the MOLTKE Approach. *Proc. EKAW 1989*, 180-195

Althoff, K.-D., Maurer, F. & Rehbold, R. (1990). Multiple Knowledge Acquisition Strategies in MOLTKE. *Proc. EKAW 1990*, 21-40

Althoff, K.-D., Nökel, K., Rehbold, R. & Richter, M. M. (1988). A Sophisticated Expert System for the Diagnosis of a CNC Machining Center. *Zeitschrift für Operations Research (ZOR), 32, 251-269*

Althoff, K.-D. & Wess, S. (1991). Case-Based Knowledge Acquisition, Learning, and Problem Solving in Diagnostic Real World Tasks. *Proc. EKAW 1991*, (20 pages)

De la Ossa, A. (1991). Knowledge Adaptation: a Means for Knowledge Acquisition. *Proc. KAW-91*

Maurer, F. (1992). Knowledge Base Maintenance and Consistency Checking in MOLTKE/HyDi. *Proc. EKAW 1992*, 337-352

Traphöner, R. & Maurer, F. (1992). Integrating Hypermedia and Expert System Technology for Technical Diagnosis. *Proc. Expersys-92*

Nökel, K. (1989). Temporal Matching: Recognizing Dynamic Situations from Discrete Measurements. *Proc. IJCAI 1989*

Nökel, K. & Lamberti, H. (1991). Temporally Distributed Symptoms in a Diagnostic Application, *Artificial Intelligence in Engineering*, vol. 6, no. 4, 196-204

Rehbold, R. (1991). Integration of Model-Based Knowledge into Expert Systems for Technical Diagnosis (in German: Integration modellbasierten Wissens in technische Diagnostik-Expertensysteme). *Doctoral Dissertation*, Kaiserslautern University

Richter, M. M. & Wess, S. (1991). Similarity, Uncertainty, and Case-Based Reasoning in PATDEX. *Automated Reasoning - Essays in Honor of Woody Bledsoe*, Kluwer Academic Publishers