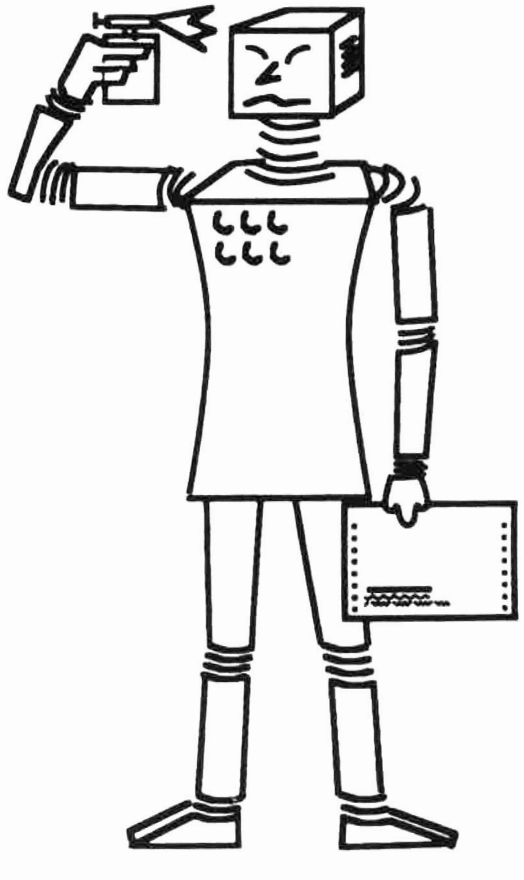
Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern

SEKI - REPORT



Ma~~s~~chine Learning and Knowledge Acquisition in a Computational Architecture for Fault Diagnosis in Engineering Systems

K.D. Althoff

SEKI Report SR-92-15 (SFB)

Machine Learning and Knowledge Acquisition in a Computational Architecture for Fault Diagnosis in Engineering Systems*

Klaus-Dieter Althoff
University of Kaiserslautern, Department of Computer Science
P.O.Box 3049, D-W-6750 Kaiserslautern, FRG
Research Group on Artificial Intelligence and Expert Systems
(Prof. Dr. Michael M. Richter)
email: althoff@informatik.uni-kl.de

Abstract

We present a computational architecture in a domain of extraordinary economical importance: fault diagnosis in engineering systems. We describe the underlying domain requirements leading to this architecture with a special focus on the included learning tasks. In the sense of KADS, the presented architecture represents an operational design model for technical diagnosis within which machine learning techniques are used to fill the model with concrete knowledge. We show how the underlying computational architecture constrains the involved learning processes.

1 Introduction

Machine Learning and Knowledge Acquisition are AI subfields which are of fundamental importance for expert system development. From the expert system point of view, machine learning techniques are a subset of the necessary knowledge acquisition techniques. From a machine learning point of view, concrete knowledge acquisition tasks define special requirements on learning subtasks. In other words: knowledge acquisition tasks represent a "real world context" within which machine learning methods can play various roles.

In production, the quality of the products as well as the availability and reliability of the technical equipment is of great importance. To avoid underutilization, the effort due to repair and maintenance must be kept small. Tool machines like CNC machining centers (CNC = computerized numerical control) play an important role within production because of their multi-purpose usability. They are complex systems consisting of mechanical, hydraulic, electronic, electric as well as pneumatic components which are in changing functional dependency with each other. Since machining centers are very expensive, because of their productive power and the high precision requirements they have to meet, their stoppage would cause a serious loss.

We present an expert system architecture (technical diagnosis shell) which has successfully handled the problem of fault diagnosis of CNC 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; Richter, 1992b). The expert system architecture is embedded in a complete knowledge acquisition workbench (the MOLTKE workbench; cf. Althoff, Maurer & Rehbold, 1990; Althoff, Nökel et al., 1988; Althoff, 1992a; de la Ossa, 1991; Maurer, 1992; Nökel, 1991) which makes use of the three main knowledge sources of the domain: a functional model of the machine, the mental model (compiled knowledge; abstract empirical knowledge) of the respective expert as well as episodic knowledge (cases; concrete empirical knowledge). Given an operational design model in the sense of KADS (cf. Breuker & Wielinga, 1989; Wielinga, Schreiber & Breuker, 1991), machine learning techniques are used to "fill this model" with concrete knowledge (cf. Althoff, Maurer et al., 1991). We show how requirements, which arise from a complex real world domain, determine the functionality and architecture of learning procedures. In other words, we give an intuition of what is the meaning of "machine learning for the fault diagnosis of engineering systems". The applied learning strategies are learning by analogy (the GENRULE system; cf. Althoff, 1992a+b), case-based reasoning (the PATDEX system; cf. Althoff & Wess, 1991; Richter & Wess, 1991; Richter, 1992a), learning by forgetting (GENRULE and PATDEX; cf. Althoff, 1992a), and knowledge compilation (the MAKE system; cf. Rehbold, 1991; Althoff, Maurer & Rehbold, 1990; Althoff, 1992a).

*also: Proceedings of the ML92 Workshop on Computational Architectures for Supporting Maschine Learning & Knowledge Acquisition

While GENRULE's task is the extraction of abstract diagnostic knowledge from cases, MAKE extracts such knowledge from a functional model of the machining center. One of our underlying assumptions is that knowledge being directly contributed by the respective expert or being generated by MAKE is (in general) more certain than episodic knowledge. Therefore, cases are mainly used for the learning of strategic knowledge (GENRULE) as well as for exception handling. The latter is performed by the PATDEX case-based reasoning system which, additionally, offers a learning-apprentice-like support.

Since going into the very details is beyond the scope of this paper, we restrict ourselves to a breadth-oriented overview. We additionally motivate this kind of presentation within the discussion part in stating that the MOLTKE learning component significantly differs from comparable ones because of the underlying constraints.

In the following chapter, we introduce the basic notions of of our domain of interest. We then summarize important domain requirements leading to the MOLTKE architecture with a special focus on the learning aspects. Here we have more general constraints for our machine learning component. These are complemented by more concrete ones in the succeeding chapter. Finally, within the discussion part we show that, in fact, the mentioned learning constraints lead to a functionality and architecture of the MOLTKE learning component which is different from those known from literature.

2 Terminology

Our area of interest is fault diagnosis and we need to introduce the basic notions. We assume a fixed num- ber N of symptoms $S_1, ..., S_N$. With each symptom S_i a range R_i is associated; in principle, symptoms are nothing more than attributes. Typically, Ri 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 $Sit = (a_{i1}, ..., a_{ij}, ..., a_{ik})$ such that $1 \leq k \leq N, 1 \leq i_j \leq N$ for all $j \in \{1, 2, ..., N\}, a_{ij} \in R_{ij}, i_{j1} = i_{j2} \iff j_1 = j_2$. 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 *informationgraph*. Its nodes are labelled with situations and an edge goes from Sit1 to Sit2 if Sit2 has at least one more component than Sit1 and there is a test t available which can provide the value necessary to extend Sit1 to Sit2; in this case t is the label of the edge.

A *diagnosis* (or fault description) is a formula of a subset of the first order predicate calculus using constants and relations over the ranges Ri; for our purposes here it is sufficient to consider this subset as equivalent to the propositional calculus. 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. For the diagnostic process of even complex machines this assumption is usually satisfied (in medical diagnostics this sometimes might 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. With equal right one can say, however, that the real problem is to find an optimal way to complete incomplete situations sufficiently enough such that a diagnosis with a high degree of certainty can be established. This task has been attacked less successfully in the literature.

Important elements of MOLTKE's representation language are cases and rules. A *diagnosticcase* is the "protocol of the real classification behavior" of an expert. It is represented as a list of symptom-value-pairs (the problem description, namely a situation Sit) completed by an empirically justified solution (diagnosis D). Thus, from a simplifying perspective which is sufficient for our purposes here, a diagnostic case C has the syntactic form $C = (Sit, D)$. Compared to diagnostic cases, *strategiccases* differ only the included solution, namely a test T which determines which symptom should be ascertained next. Therefore, a strategic case S has the syntactic form $S = (Sit, T)$. Like diagnostic cases, *diagnosticpaths* are defined as the "protocol of a session with the underlying diagnostic system". From a pure syntactic perspective paths and cases are identical, i.e. paths can be considered as "special diagnostic cases". The problem description of a path are the list of symptom values given as input to the diagnosic system, the solution is the stated diagnosis.

A *diagnosticrule* associates a situation with a diagnosis, an *orderingrule* associates a situation with a test, whereas a *determinationrule* associates a situation with an assignment of a concrete value to a symptom. In case of a *total* determination rule, the represented relation is a causal one. *Partial* determination rules have *determinationfactors* attached which are an approximation of the conditional probability that the assigned symptom value holds if the given situation holds. The relation between determination factors and conditional probabilities is similar to the relation between certainty factors and ordinary probabilities, where the underlying probability distribution is not known, too.

3 Domain Requirements

We now present some important requirements from the domain of technical diagnosis which constrain the architecture of our learning component from a more general perspective.

Acquisition and representation of concrete empirical knowledge as well as modeling of the "diagnostic learning behavior" of the respective experts

- this leads to the definition of diagnostic cases as introduced above, to the development of a case-based reasoning system for diagnostic problem solving in engineering domains (PATDEX) as well as to the development of an incremental inductive learning system with case-based hypotheses generation (partial determination and ordering rules) justified by the use of determination factors (GENRULE)

Automation of the knowledge acquisition process as far as possible

- this leads to the development of a knowledge compilation system which generates a partial knowledge base (diagnostic and total determination rules) from the construction plans for the electronic/electric and hydraulic components (MAKE)

Understandability of the implemented diagnostic problem solving behavior

- this leads to a focus on diagnostic strategic knowledge and to the decomposition of diagnostic reasoning into "classification plus test selection". Therefore, the manual improvement of MAKE-generated partial knowledge bases is possible as well as the learning of strategic knowledge. This results in case-based reasoning being the best available learning paradigm for the integration of problem solving and learning. Another important result is that the TDIDT strategies are not appropriate with respect to the diagnostic strategies they could represent.

Representation and processing of a huge amount of knowledge

- this leads to the construction of the information graph and the further decomposition into different types of knowledge (rule types). This is reflected by the (partially introduced) MOLTKE representation language. Thus, one (natural) constraint of the learning component is the learning of elements of this representation language. Therefore, simple inductive rule learning strategies are not applicable (for additional drawbacks of inductive learning approaches cf. Manago et al., 1992).

4 Constraints on Learning

Within this chapter we present some selected more concrete constraints underlying the MOLTKE learning component. It is organized as a breadth-oriented overview which, as we believe, already makes some interesting statements.

By the use of diagnostic cases, abstract empirical knowledge (diagnostic, determination, and ordering rules) as well as concrete empirical knowledge (diagnostic case memory) have to be constructed. Inferences requiring the interaction with the user (e.g. being carried out within a diagnostic session) must be performed efficiently.

Using sufficiently efficient learning procedures, the user could be easily involved in the evaluation process of the generated learning hypotheses. Here the interlocking of the diagnostic and the learning process is essential which, additionally, allows the "interactive acquisition" of further concrete empirical knowledge (diagnostic cases).

In connection with interactive knowledge acquisition, it is important that the user "develops a feeling" of which cases are known by the system and which are not. Automated knowledge acquisition strategies always have to consider this. Thus, the system must be "aware" of all cases contributed by the expert. Nevertheless, an important aspect of human learning behavior is to be able to forget certain experiences in course of time. Therefore, it is reasonable to restrict the amount of representable empirical knowledge using, e.g., explicit strategies of forgetting, filtering of redundant and/or incorrect cases as well as the definition of "capacity thresholds" for empirical knowledge. This alltogether leads to trade-off which cannot be solved in general. In this connection the underlying assumptions of the knowledge acquisition community, on the one hand, and the machine learning community, on the other hand, appear to be different. Within knowledge acquisition, the motivation of the expert is of central importance. This leads to an orientation towards an "understandable learning apprentice". Within machine learning, theoretical aspects are dominating. This leads to interesting learning strategies (cf., e.g., "non-conservative" learning: Emde, 1991; "inconsistent" learning: Lange & Wiehagen, 1991), and it then heavily depends on the involved experts as well as the complexity of the problem if these strategies are applicable (in the sense of accepted).

We can differentiate between three main views on the MOLTKE learning component resulting in three different groups of constraints: the simulation view, the technical view, and the pragmatic view.

The Simulation View: Modeling the learning behavior of experienced experts:

- learning for a special purpose, namely to improve the diagnostic capabilities of the system. Thus, we have a strong interlocking of the diagnostic and the learning process
- representation and efficient processing of abstract and concrete empirical knowledge
- generating abstract empirical knowledge from concrete empirical knowledge
- representation and efficient processing of technical (engineering) background knowledge
- modeling of the "shortcut-oriented diagnostic problem solving behavior" (ascertaining as few symptoms as possible) of the respective experts
- modeling of forgetting strategies

The Technical View: Knowledge acquisition support as far as possible:

- learning of elements of the MOLTKE representation language(s)
- simple, understandable modeling and processing of uncertain knowledge
- direct interpretation of concrete empirical knowledge
- if a correct diagnostic case is known, then it must be used for classification purposes if the same situation occurs again

The Pragmatic View: Consideration of utility aspects concerning usable resources:

- effective reduction of the needed knowledge acquisition effort
- improved maintainability of the knowledge base
- ease of adaptation of the knowledge base to simple modifications of the machining center
- ease of transfer of available knowledge bases for diagnosing similar technical systems
- efficient learning procedures to supplement interactive knowledge acquisition mechanisms
- classification knowledge acquired manually or generated by MAKE is considered to be more secure than episodic knowledge
- strategic knowledge acquired manually or extracted from episodic knowledge is considered to be more understandable than strategic knowledge which has been extracted from the functional model of the machine (using MAKE)

5 The Learning Component

Since we have stated (some) domain requirements as well as architectural constraints on the learning component, we now, for clarification purposes, want to give a summarizing description of this component.

To represent the given three main 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 the diagnostic representation language is identified with that describing the expert-entered knowledge. Therefore, only two compilation processes are necessary: MAKE compiles the functional model of the machine into diagnosis rules and total determination rules, whereas GENRULE compiles diagnostic cases into partial determination and ordering rules. Thus, the interpretation of functional causal knowledge is very efficient. Additionally, this functional knowledge can be improved by the use of manually acquired knowledge as well as cases. For learning-apprentice-like support and the handling of exceptional situations the cases can also be interpreted via case-based reasoning, i.e. the diagnostic (heuristic) problem solver is supplemented by a case-based one.

5.1 The Diagnosis Shell

The basic aspect here is the diagnostic description language which uses concepts that are easily understandable by the expert (e.g. diagnosis, symptom, test etc.). In addition, the diagnostic process can be decomposed via the introduction of the information graph as well as the definition of different types of rules. A diagnostic system based on the shell gets symptom values (in arbitrary order and number (with respect to the defined ranges R_i)) as input and gives a final or an intermediate diagnosis as output.

5.2 Make

The MAKE system is well suited for the modeling of electronic/electric and hydraulic parts of even complex technical devices. MAKE uses a component-oriented, hierarchical model with a qualitative and static description of the device's behavior (in the sense of qualitative reasoning). The model bases on knowledge about structure, behavior, and function of the technical device. The MAKE-generated partial knowledge base is correct with respect to the description of the device and complete with respect to the specified funtionality of the device. Thus, MAKE gets as input a library of component classes (e.g. relais, valve etc.), concrete component instances (relais21, valve2 etc.), the connectivity of the devices subcomponents, and the intended overall behavior of the technical devicse (function, functionality). From this the above mentioned deep functional model is constructed and compiled into a knowledge base for the diagnosis shell.

5.3 GenRule

The GENRULE system is able to improve the knowledge base by the use of diagnostic cases and paths. GENRULE realizes an incremental inductive learning strategy which generates its (learning) hypotheses (partial determination rules, ordering rules) based on a memory of diagnostic cases and paths. Within the memory symptom values and diagnoses are used for an efficient indexing. All diagnostic paths are automatically generated from the given knowledge base. The rules are indirectly generated via the integration of cases or paths into the memory. The determination factors are efficiently computed and updated by the use of the memory. In addition, the memory functions as a dependency network for the generated rules which automatically establishes or retracts rules of which the determination factor changes to be above or below a given threshold. Thus, the GENRULE gets as input a knowledge base and a case memory. From this, it generates an improved knowledge base as well as an updated memory.

5.4 Patdex

PATDEX realizes a learning-apprentice-like case-based diagnosis system which can cooperate with the problem solver of the diagnosis shell as well as working in a stand-alone manner. PATDEX consists of two case-based reasoning subcomponents, one for classification purposes using diagnostic cases and one for the selection of tests which is based on strategic cases (strategic cases are automatically generated from diagnostic cases). The inference engine bases on the similarity of diagnostic and stragetic cases, respectively. The similarity is computed based on many different similarity measures which base on two different function schemata, one for diagnosis cases and one for strategic cases. The computation process is very efficient because of a dynamic decomposition of the case base as well as a case dependency network for both subcomponents. A diagnostic similarity measure can be automatically adapted to the expert's behavior using connectionist techniques (competitive learning). For the strategic similarity measure the adaptation process is guided by an A*-like procedure which estimates the average costs for ascertaining symptoms. PATDEX can identify pathologic symptom values by the use of causal background knowledge. It uses default values for symptoms as well as partial and total determination rules to improve its similarity judgements. PATDEX is able to handle incomplete, redundant, and/or incorrect case descriptions. PATDEX gets symptom values in an arbitrary number and order (with respect to the defined value ranges) as input and gives a final diagnosis as output (or the message that no diagnosis can be derived based on the known cases). As a "side-effect" new diagnostic and strategic cases are generated and the underlying similarity measures updated.

6 Discussion

In fact, the presented learning constraints lead to a learning component which is different from those known from literature (for a detailed comparison cf. Althoff, 1992a).

A basic difference between GENRULE and other knowledge refinement systems like, e.g., KRUST (cf. Craw & Sleeman, 1990) or INDE+ (cf. Aben & van Someren, 1990) is that these try to improve the classification ability whereas GENRULE focuses on the strategic knowledge. In this sense, it is comparable to the approach of Gruber (1989), but there no learning strategies are used. BOLERO (cf. Lopez & Plaza, 1991) also uses a case-memory approach for the learning of strategic knowledge, but GENRULE does it in a much more specific way because of its combination with the PATDEX and the MAKE system. In addition, GENRULE does not need an explicit training phase.

PATDEX uses a very efficient procedure for the updating of its similarity values. It makes use of the whole MOLTKE knowledge base including the functional knowledge generated by MAKE. CREEK (cf. Aamodt, 1991) only uses causal knowledge. Because of its adaptive capabilities (competitive learning for the adaptation of its similarity measure), PATDEX can be easily applied to other diagnostic problems. This is not easily done by the PROTOS system (cf. Bareiss, 1989) where many parameters and the whole relational language have to be adapted.

MAKE uses a static, hierarchical, component-oriented qualitative model of the structure and behavior of the machining center to generate a partial diagnostic knowledge base "in one compiling step". Since MAKE uses examples for the intended behavior (function) of the machine, this complete compilation becomes possible. This is, e.g., one characteristic difference between the MOLTKE approach and others like, e.g., Friedrich, Gottlob, and Nejdl (1990) (interpreting approach, restriction to decision trees), CONCLAVE (cf. van de Velde, 1988; 1989) (interpreting approach, only use of causal knowledge), or ACES (cf. Pazzani, 1990) (no automatic construction of the knowledge base).

Acknowledgements

Thanks go to Prof. Michael M. Richter for his encouraging supervision of my doctoral dissertation as well as to our research group in Kaiserslautern for the excellent collaboration. This research has been partially supported by the Deutsche Forschungsgemeinschaft, Sonderforschungsbereich 314 "Artificial Intelligence - Knowledge-Based Systems", projects X6 and X9.

References

Aamodt, A. (1991). A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning. Ph.D. Thesis, University of Trondheim

Aben, M. & van Someren, M. W. (1990). Heuristic Refinement of Logic Programs. Proc. ECAI-90, 7-12

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, University of Kaiserslautern (to appear)

Althoff, K.-D. (1992b). Learning of Shortcut-Oriented Diagnostic Problem Solving (in German: Lernen von abkürzungsorientiertem diagnostischen Problemlösen). In: K. Reiss, M. Reiss & H. Spandl (eds.), Maschinelles Lernen - Modellieren von Lernen mit Maschinen, Springer Verlag (to appear)

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-89

Althoff, K.-D., Maurer, F. & Rehbold, R. (1990). Multiple Knowledge Acquisition Strategies in MOLTKE. Proc. EKAW-90, 21-40

Althoff, K.-D., Maurer, F., Traphöner, R. & Wess, S. (1991). The Learning Component of the MOLTKE3 Workbench for Technical Diagnosis (in German: Die Lernkomponente der MOLTKE3-Werkbank zur Diagnose technischer Systeme). In: Morik (1991), 58-64

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. (1991a). Case-Based Knowledge Acquisition, Learning, and Problem Solving in Diagnostic Real World Tasks. Proc. EKAW-91

Althoff, K.-D. & Wess, S. (1991b). Case-Based Reasoning and Expert System Development. In: F. Schmalhofer, G. Strube & T. Wetter (eds.), Contemporary Knowledge Engineering and Cognition, Springer Verlag (to appear)

Bareiss, R. (1989). Exemplar-Based Knowledge Acquisition. London: Academic Press

Breuker, J. A. & Wielinga, B. J. (1989). Models of Expertise in Knowledge Acquisition. In: G. Guida & C. Tasso (eds.), Topics in Expert System Design, Amsterdam: North-Holland, 265-297

Craw, S. & Sleeman, D. (1990). Automating the Refinement of Knowledge-Based Systems. Proc. ECAI- 90, 167-172

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

Emde, W. (1991). Modeling, Knowledge Revision, and Knowledge Representation in Machine Learning (in German: Modellbildung, Wissensrevision und Wissensrepräsentation im Maschinellen Lernen). Springer Verlag

Friedrich, G., Gottlob, G. & Nejdl, W. (1990). Generating Efficient Diagnostic Procedures from Model-Based Knowledge Using Logic Programming Techniques. Computers and Mathematics with Application

Gruber, T. R. (1989). The Acquisition of Strategic Knowledge. Academic Press

Lange, S. & Wiehagen, R. (1991). Polynomial-time Inference of Arbitrary Pattern Languages. New Generation Computing, 8, 361-370

Lopez, B. & Plaza, E. (1991). BOLERO: Case-Based Learning of Strategic Knowledge. Proc. EWSL-91

Manago, M. et al. (1992). Acquiring Descriptive Knowledge for Classification and Identification. In: Wetter, Althoff et al. (1992), 392-405

Maurer, F. (1992). Knowledge Base Maintenance and Consistency Checking in MOLTKE/HyDi. In: Wetter, Althoff et al. (1992), 337-352

Morik, K. (ed.)(1989). Knowledge Representation and Organization in Machine Learning, Springer Verlag

Morik, K. (ed.)(1991). Special Volume on Machine Learning (in German). KI, 5, No. 1, FBO-Verlag

Nökel, K. (1991). Temporally Distributed Symptoms in Technical Diagnosis. Springer Verlag

Pazzani, M. J. (1990). Learning Fault Diagnosis Heuristics from Device Descriptions. In: Y. Kodratoff & R. Michalski (eds.), Machine Learning: An Artificial Intelligence Approach - Vol. III. San Mateo: Morgan Kaufmann, 214-234

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, University of Kaiserslautern

Richter, M. M. (1992a). Classification and Learning of Similarity Measures. In: Proc. of the 16th annual meeting of the German society for classification

Richter, M. M. (ed.) (1992b). MOLTKE - Methods for Fault Diagnosis in Engineering Systems (in German: MOLTKE - Methoden zur Fehlerdiagnose in technischen Systemen). (forthcoming)

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

van de Velde, W. (1988). Learning through Progressive Refinement. Proc. EWSL-88

van de Velde, W. (1989). (Re)Presentation Issues in Second Generation Expert Systems. In: Morik (1989)

van Someren, M. W., Zheng, L. L., Post, W. (1990). Cases, Models or Compiled Knowledge; a Comparative Analysis and Proposed Integration. Proc. EKAW-90, 339-355

Wetter, Th., Althoff, K.-D., Boose, J., Gaines, B. R., Linster, M. & Schmalhofer, F. (eds.) (1992). Current Developments in Knowledge Acquisition - EKAW'92. Springer Verlag

Wielinga, B. J., Schreiber, G. & Breuker, J. A. (1991). KADS: A Modelling Approach to Knowledge Engineering. Document KADS-II/T1.1/PP/ UvA/008/ 1.0, University of Amsterdam (also submitted to Knowledge Acquisition)