

SEKI - REPORT

Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern



Die Lernkomponente der MOLTKE-Werkbank zur Dia- gnose technischer Systeme

K. D. Althoff, F. Maurer, R. Traphöner, S. Weiß
SEKI Working Paper SWP-90-08 (SFB)

**Die Lernkomponente der
MOLTKE-Werkbank zur Dia-
gnose technischer Systeme**

K. D. Althoff, F. Maurer, R. Traphöner, S. Weiß
SEKI Working Paper SWP-90-08 (SFB)

Die Lernkomponente der **MOLTKE₃**-Werkbank zur Diagnose technischer Systeme¹

Klaus-Dieter Althoff, Frank Maurer, Ralph Traphöner & Stefan Weiß
Universität Kaiserslautern
Fachbereich Informatik
Postfach 3049
6750 Kaiserslautern

Zusammenfassung

Im Rahmen des **MOLTKE**-Projektes² wurde am Beispiel einer komplexen, technischen Anwendungsdomäne die **MOLTKE₃-Werkbank** zur Diagnose technischer Systeme entwickelt. In ihr sind verschiedene Expertensystemtechniken der zweiten Generation integriert. Gegenstand dieses Beitrages ist die Lernkomponente der Werkbank, die sowohl die Klassifikations- als auch die Testauswahlfähigkeiten der Diagnoseschale mit Hilfe von Fallbeispielen verbessern kann. Sie besteht aus zwei Komponenten, die auf derselben Fallrepräsentation aufsetzen.

Die Teilkomponente *GenRule*³ lernt dabei "offline" aus Fallbeispielen. Das Ergebnis eines Lernschrittes sind heuristische Regeln für eine **MOLTKE**-Wissensbasis, mit deren Hilfe das Diagnoseverfahren an das "abkürzungsorientierte diagnostische Problemlösen" eines Servicetechnikers angepaßt werden kann.

Die zweite Teilkomponente *PATDEX*⁴ erweitert das Diagnoseverfahren um einen ähnlichkeitsbasierten Ansatz. Hierzu werden die Fallbeispiele mit Hilfe eines fallbasierten Schlußfolgerungsmechanismus direkt interpretiert.

Abstract

Within the **MOLTKE** project a complex, technical domain has been used as a guideline for the development of the **MOLTKE₃ workbench** which integrates different techniques of second generation expert systems. In this paper we concentrate on the workbench's learning mechanism which can improve both the classification and the test selection abilities of the workbench's diagnosis shell using cases. It consists of two subcomponents which both use the same case representation.

The *GenRule*³ system is the "offline" learning part. It generates heuristic rules for a **MOLTKE** knowledge base from cases. Using *GenRule* the diagnostic procedure of the diagnosis shell can be adapted to the "shortcut-oriented diagnostic problem solving" of an (expert) service technician.

The *PATDEX*⁴ system extends the shell's diagnostic procedure with a similarity-based classification mechanism. Therefore the cases are directly interpreted using case-based reasoning.

¹ Die hier dargestellte Arbeit wurde teilweise gefördert durch die Deutsche Forschungsgemeinschaft, Sonderforschungsbereich 314 "Künstliche Intelligenz - Wissensbasierte Systeme", Projekte X6 und X9. Das Projekt X6 wurde in Kooperation mit dem Lehrstuhl für Fertigungsmeßtechnik und Qualitätssicherung (WZL, Prof. Dr.-Ing. T. Pfeifer) der RWTH Aachen durchgeführt.

² **M**odels, **L**earning and **T**emporal **K**nowledge in **E**xpert Systems for Technical Diagnosis

³ **G**enerator of Empirical **MOLTKE** **R**ules

⁴ **PAT**tern **D**irected **EX**pert System

1. Motivation und Einführung

Ein Servicetechniker ist aufgrund seines Erfahrungswissens in der Lage, Diagnosen schneller zu stellen, falsche Diagnosen zu vermeiden, Konsequenzen aus falschen Diagnosen zu ziehen sowie das (erfolgreiche) Diagnoseverfahren bereits erlebter Situationen auf das aktuelle Problem (sinngemäß) zu übertragen. Diese Fähigkeiten beruhen dabei auf seinem intuitivem Verständnis für ähnliche Situationen bzw. seinem Vermögen, verkürzte Lösungswege anwenden zu können.

Ein Beispiel aus der Domäne der CNC-Maschinen soll dies illustrieren: Eine Werkzeugmaschine wird in einem Bereich mit sehr hoher Umgebungstemperatur eingesetzt. Nehmen wir weiter an, daß bei dieser Maschine umgebungsbedingt in 90% der Fälle ein bestimmter Teil der Hydraulik ausfällt. Ein erfahrener Servicetechniker, der diese Maschine kennt, wird im allgemeinen schon nach der Feststellung der ersten Symptome entscheiden, ob der übliche Fehler aufgetreten ist oder nicht. Erkennt er ein Teilmuster der Symptomatik des Hydraulikfehlers, so wird er auf ein Abarbeiten des kompletten Suchweges verzichten.

Ein weiteres Szenario ist die Behandlung von bisher unbekanntem Fehlersituationen: Nach der Installation und den damit verbundenen Testläufen einer CNC-Maschine wurde beim erstmaligen regulären Einsatz ein Fehler festgestellt. Nach einer langwierigen Untersuchung stellte der erneut herbeigerufene Techniker zufällig fest, daß das für den Transport aufgetragene Schmierfett einen Endschalter verklebt hatte. Diese für den Servicetechniker neue Erfahrung versetzt ihn in Zukunft in die Lage, ähnliche Fehlersituationen gezielt zu lösen.

Solches Wissen, das auf konkreten Erfahrungen beruht, steht heutigen Diagnosesystemen meist nicht zur Verfügung. Der Servicetechniker hingegen kennt aus seiner Tätigkeit Fälle für solch spezifisches Fehlverhalten. Schauen wir uns einen konkreten Fall näher an (Abb. 1 und Abb. 2):

Maschinenfehlermeldung	i59	}	die in dieser Reihenfolge erhobenen Symptome
I/O-Status IN36	logisch-1		
I/O-Status OUT7	logisch-1		
Ventil 5Y1	geschaltet		
Ventil 5Y2	nicht geschaltet		
Leitungssystem	in Ordnung		
Aufnahmekegel	nicht verschmutzt		
I/O-Status IN32	logisch-1	}	
 		}	
I/O-Karte	defekt	}	die Diagnose

Abb. 1 – Ein “Fall”⁵ in der MOLTKE-Wissensbasis

⁵ Aus der Wissensbasis extrahierte Fälle heißen *Diagnosepfade*.

Maschinenfehlermeldung	i59	}	die in dieser Reihenfolge erhobenen Symptome
I/O-Status OUT7	logisch-1		
Ventil 5Y1	geschaltet		
Ventil 5Y2	nicht geschaltet		
Leitungssystem	in Ordnung		
Aufnahmekegel	nicht verschmutzt		
I/O-Status IN32	logisch-1		
I/O-Karte	defekt	}	die Diagnose

Abb. 2 – Ein Fall des Servicetechnikers

Wie man leicht feststellt, hat der Techniker das Symptom I/O-Status IN36 mit Wert logisch-1 nicht erhoben. Mit anderen Worten: Er hat es nicht benötigt, um die gleiche Diagnose stellen zu können. Man kann dieses Verhalten auf zweierlei Weise interpretieren:

- Der Servicetechniker betrachtete das Symptom als für die gegebene Situation irrelevant.
- Die Relevanz der übrigen Symptomwerte für die Diagnose oder den fehlenden Symptomwert ist so stark, daß ein Analogieschluß zwischen der vollständigen und der unvollständigen Situation gerechtfertigt ist - also der fehlende Symptomwert durch eine Analogie abgeleitet werden konnte.

In beiden Fällen wäre die fragliche Symptomausprägung durch die restlichen Symptomwerte "partiell determiniert"⁶. Man kann versuchen, den Grad der Determiniertheit durch Lernen aus weiteren Fällen zu verbessern. Allgemein lernt ein Techniker auf eine doppelte Weise:

- Er "lernt" von einem bestimmten früheren Fall, in dem er in einer konkreten Situation eine dortige Lösung sinnvoll auf den aktuellen Fall überträgt.
- Er "lernt" aus der Gesamtheit aller früheren Fälle sein gesamtes Lösungsverhalten zu verbessern.

Die hier gegebenen Beispiele sollen einen ersten Eindruck hinsichtlich der Anforderungen an die Lernkomponente vermitteln. Weitere Anforderungen ergeben sich aus der Integration in die MOLTKE₃-Werkbank, deren grundlegende Terminologie im folgenden Abschnitt beschrieben wird.

2. Die Terminologie der MOLTKE-Werkbank

Im Verlauf des MOLTKE-Projektes entstanden mehrere prototypische Diagnoseanwendungen aus technischen Bereichen (CNC-Bearbeitungszentren, 3D-Meßmaschinen, Tunnelvortriebsmaschinen und Fehlersuche in heterogenen Computernetzen). Basierend auf den dabei gewonnenen Erfahrungen entwickelten wir ein einfaches Modell für technische Diagnoseanwendungen. Wir gehen davon aus, daß sich Diagnose wie folgt beschreiben läßt:

⁶ Partielle Determinationen sind ein Ersatz für bedingte Wahrscheinlichkeiten (mangels genauerer Information). Dabei wird der "Grad der Determiniertheit" durch *Determinationsfaktoren* ausgedrückt, deren Wertebereich jeweils [0,1] ist.

Diagnose = Klassifikation + Testauswahl.

Basierend auf dieser Zweiteilung entstand eine Repräsentationssprache, die die Sprachwelt des Experten widerspiegeln soll. Diese ist im folgenden (abstrakt) beschrieben.

Ein *Symptom* beschreibt einen meßbaren Teil des Zustandes des zu diagnostizierenden Systems. Ein Symptom kann dabei ein konkreter Meßwert sein, z.B. eine gemessene Spannung, oder auch ein daraus abstrahierter Wert, z.B. die Spannung ist zu hoch. Ein aktuell gemessener Wert eines Symptoms wird als *Symptomwert* bezeichnet. Symptomwerte werden mit Hilfe von *Tests* erhoben. Der Vektor, der zu einem Zeitpunkt aktuellen Symptomwerte, wird auch als *Situation* bezeichnet, wobei Symptome auch den Wert "unbekannt" haben können.

Eine *Diagnose* wird durch eine Menge von Symptomwerten charakterisiert. Formal werden Diagnosen über *Diagnoseformeln*⁷ beschrieben. Die Verfeinerungshierarchie der Diagnosen wird mit Hilfe eines Graphen, dem *Kontextgraphen*, modelliert. *Kontexte* repräsentieren dabei Grob-, Zwischen- oder Enddiagnosen. Einem Kontext werden zweierlei Arten von Regeln zugeordnet⁸: *Reihenfolge-* und *Abkürzungsregeln*. Die Bedingungsteile dieser Regeln sind Diagnoseformeln. Als Aktionen werden im Falle einer feuernden Reihenfolgeregel ein Test angestoßen und im Falle einer Abkürzungsregel ein Symptom mit einem Wert belegt. Dabei repräsentieren Abkürzungsregeln Zusammenhänge zwischen Symptomwerten, beispielsweise "Wenn das Licht brennt, dann ist der Schalter geschlossen". Die Strategie des Experten kann durch die Reihenfolgeregeln dargestellt werden. Diese ordnen einer Situation den als nächstes durchzuführenden Test zu.

3. Aufgaben der Lernkomponente

Im Rahmen der MOLTKE₃-Werkbank wird das für die Diagnose relevante Erfahrungswissen eines Servicetechnikers explizit mit Hilfe von Diagnosefällen modelliert. Ein Fall ist dabei ein Protokoll des realen Problemlöseverhaltens eines Experten, bestehend aus einer Situationsbeschreibung und der jeweils gestellten Diagnose⁹.

Ziel der Lernkomponente ist es, das Diagnoseverfahren auf die gleiche Art und Weise zu verbessern, wie dies oben für den Servicetechniker motiviert wurde. Entsprechend werden im Rahmen der MOLTKE₃-Lernkomponente zwei sich ergänzende Ansätze verfolgt, ein fallverallgemeinernder (GenRule) und ein fallfokussierender (PATDEX₂):

- das GenRule-System lernt aus Analogien zwischen neu präsentierten Diagnosefällen und bereits in die Wissensbasis integrierten Diagnosepfaden¹⁰. Diese analogen Schlüsse werden in heuristische Regeln compiliert.
- Das PATDEX₂-System führt fallbasiertes Schließen direkt auf einer Fallbasis durch. Dies ist sowohl zur Behandlung von Ausnahmefällen, als auch zur Wissensakquisitionsunterstützung wichtig (direktes Interpretieren der Diagnosefälle, Akquisition weiterer Diagnosefälle).

Das Hauptunterscheidungsmerkmal der beiden Systeme ist, daß GenRule analoges Schließen "offline" zur Generierung von Regeln einsetzt, während PATDEX₂ es "online" als eigentlichen Problemlösemechanismus verwendet.

⁷ Zugrunde gelegt wird eine Teilmenge einer dreiwertigen Prädikatenlogik 1. Stufe.

⁸ Die später eingeführten "Redundanzregeln" sind lediglich eine "Spielart" der Abkürzungsregeln.

⁹ Siehe auch Abb. 2.

¹⁰ Ein Diagnosepfad ist ein Protokoll eines Diagnoseaufs, basierend auf der aktuellen Wissensbasis des Expertensystems. Ein Pfad besteht aus einer Situation und der gestellten Diagnose, d. h. Diagnosefälle und -pfade sind syntaktisch gleich (siehe auch Abb. 1).

4. Beschreibung der Lernkomponente

Grundlegend für die MOLTKE₃-Werkbank ist die folgende Sicht der Wissensakquisition:

Wissensakquisition = Wissensmodellierung + Wissenserhebung.

Zur Modellierung technischer Diagnoseproblematiken wird dabei das in §2 skizzierte Modell verwendet. Im Sinne von KADS [8] ist somit zum einen ein Designmodell vorhanden, zum anderen existieren Verfahren zur Generierung eines Implementierungsmodells. Im einzelnen sind dies Lernverfahren sowie Verfahren aus dem Bereich des qualitativen Schließens bzw. der Unterstützung manueller Wissensakquisition.

Die eingesetzten Lernverfahren haben also die Funktion, die Generierung eines Implementierungsmodells für ein konkretes technisches System, auf der Basis eines Designmodells für die Diagnose technischer Systeme, möglichst weitgehend zu unterstützen. Dies ist, verglichen mit dem in [14] vorgestellten Ansatz von BLIP, eine vereinfachendere Sichtweise, da das Modell des Sachbereiches nicht gelernt, sondern über das Designmodell bereits vorgegeben wird. Dies erscheint uns für die zugrunde liegende Domäne auch als sinnvoll. Die Lernverfahren werden somit zur Konkretisierung des vorgegebenen Modells eingesetzt, d.h. zur Entwicklung der Klassifikations- und Testauswahlfähigkeiten des Zielsystems. Die beiden Hauptlernstrategien sind dabei *Lernen durch Analogie* (GenRule) und *fallbasiertes Schließen* (PATDEX₂). Beide Teilkomponenten werden nun kurz aus einer funktionalen Sicht heraus beschrieben.

4.1. GenRule

Zur Verbesserung einer gegebenen MOLTKE-Wissensbasis mit Hilfe empirischer Diagnosefälle orientiert sich GenRule an dem "abkürzungsorientierten diagnostischen Problemlösen"¹¹ erfahrener Servicetechniker. Dazu lernt GenRule "Abkürzungen" aus Analogien zwischen bereits in die Wissensbasis integrierten Diagnosepfaden und neu präsentierten Fällen ([6], [1]). Diese Abkürzungen werden in die MOLTKE-Shell als partielle, d.h. heuristische, Abkürzungsregeln integriert. Durch das Anwenden dieser Regeln kann das Diagnoseverfahren beträchtlich abgekürzt und durch die Fokussierung der Diagnosestrategie auf das Feuern der Abkürzungsregeln darüber hinaus auch transparenter gestaltet werden. Zudem können durch Teilkomponenten von GenRule Inkonsistenzen in der Wissensbasis aufgedeckt werden.

4.1.1. Generierung von Abkürzungs-, Redundanz- und Reihenfolgeregeln

Grundlegend für GenRule ist der Vergleich von neu präsentierten Diagnosefällen mit "ähnlichen" Diagnosepfaden. Ein Diagnosepfad ist dabei dem Diagnosefall *ähnlich*, wenn er die gleiche Diagnose hat und zudem alle Symptomwerte des Falles beinhaltet. Ein solcher Diagnosepfad ist dem Diagnosefall umso "ähnlicher", je "kürzer" er ist. Ist der Diagnosefall "echt kürzer" als der Pfad, so wird dies als Abkürzung interpretiert und eine entsprechende Regel generiert. Die linke Seite einer derartigen Regel wird dabei auf der Grundlage der Situation des vorgelegten Diagnosefalles gebildet. Die rechte Seite ist sozusagen der jeweilige "gelernte" Symptomwert des ähnlichsten Diagnosepfades.

Gerechtfertigt wird eine derartige Regel durch die in einem Determinationsfaktor akkumulierte statistische Information. Überschreitet der Determinationsfaktor einen gegebenen Schwellwert (z.B. 0,9), so wird die Regel als Abkürzungsregel in die Shell integriert. Unterschreitet er dagegen einen zweiten Schwellwert (z.B. 0,1), so wird die Regel als *Redundanzregel* inte-

¹¹ Siehe §1 zur Motivation und Einführung.

griert¹². Letzteres ist damit gleichzusetzen, daß das entsprechende Symptom für die Klassifikation in dem jeweiligen Diagnoseteilpfad nicht mehr verwendet wird.

Ist der Diagnosefall "echt länger" als der Diagnosepfad, so wird dies als ein Hinweis auf eine Inkonsistenz in der Wissensbasis interpretiert, falls eine hinreichende statistische Rechtfertigung vorliegt (Schwellwert). Die Konsequenz ist dann die Generierung einer Reihenfolgeregel, die die Erhebung des bzw. der fehlenden Symptome erzwingt.

4.1.2. Organisation der Diagnosefälle

Die Diagnosefälle werden im Fallgedächtnis von GenRule organisiert. Der Aufbau der Speicherstruktur orientiert sich an den Arbeiten von Kolodner [12], stellt aber in seiner konkreten, der Problemstellung angepaßten Realisierung eine Neuentwicklung dar. Grundidee ist die Integration neuen Wissens (hier: Erfahrungswissen in Form von Diagnosefällen) in bereits bekanntes und die dabei erforderliche automatische Reorganisation der Struktur. Folgende Teilaufgaben von GenRule werden durch das Gedächtnis besonders unterstützt:

- die Partitionierung der Fälle gemäß ihrer Diagnose,
- das Auffinden von Diagnosefällen, die bestimmte, vorgegebene Situationen enthalten (zur Berechnung der Determinationsfaktoren),
- Auffinden von Diagnosefällen und -pfaden, die einander ähnlich sind,
- die Aktualisierung der generierten Abkürzungs-, Redundanz- und Reihenfolgeregeln,
- die Aktualisierung der PATDEX₂ zugeteilten Fälle.

4.2. PATDEX₂

PATDEX₂ versucht zu einer gegebenen aktuellen Situation in seiner Fallbasis einen möglichst ähnlichen Fall zu finden. Ist die Ähnlichkeit des ermittelten Falles zur aktuellen Situation ausreichend, so übernimmt es die Diagnose des gefundenen Falles. Andernfalls wird versucht, die Beschreibung der Situation durch die Erhebung eines weiteren Symptoms so zu verbessern, daß anschließend eine Diagnose möglich ist. Zur Lösung eines vorliegenden Diagnoseproblems stellt PATDEX₂ Hypothesen auf und bittet den Benutzer diese zu verifizieren bzw. gegebenenfalls zu korrigieren. Dieses Feedback aus der realen Welt verwendet PATDEX₂ dann zur Verbesserung seines diagnostischen Verhaltens. Die Wiederholung einmal gemachter Fehler ist somit weitgehend ausgeschlossen.

Die Bewertung der Übereinstimmung zwischen der vorliegenden Situation und einem in der Fallbasis gespeicherten Fall erfolgt nach Maßgabe eines Ähnlichkeitsmaßes, welches jedem Fall einen numerischen Wert zuordnet. Die Ordnung über den reellen Zahlen induziert dann eine (totale) Ordnung über der Fallbasis. Zur Berechnung des Ähnlichkeitsmaßes werden übereinstimmende, sich widersprechende sowie fehlende Symptome im betrachteten Fall bzw. in der aktuellen Situation verwendet ([2], [3]). Die einzelnen Symptome werden dabei gemäß ihrer Relevanz für eine bestimmte Diagnose gewichtet. Die Bestimmung der Relevanz eines Merkmals für eine bestimmte Situation ist ein entscheidender Punkt der fallbasierten Wissensverarbeitung. So ist die Feststellung "Glühbirne nicht defekt" für eine mögliche Diagnose "Sicherung defekt" durchaus relevant. Die Feststellung "Innentemperatur = 25 Grad" ist in diesem Zusammenhang aber wahrscheinlich absolut irrelevant. Es sind aber Situationen denkbar, in denen die Bewertung gerade umgekehrt erfolgen muß. Die Relevanz eines Merkmals für eine bestimmte Situation ist somit a priori im allgemeinen nicht feststellbar¹³. In PATDEX₂ verfolgen wir daher den Ansatz, die Relevanz von Merkmalen für bestimmte Situationen als Teil des zu lernenden Erfahrungswissens aufzufassen. Die Gewichtung von Symptomen wird vom System im Laufe der Zeit, oder in einer optionalen Trainingsphase mit

¹² Redundanzregeln sind somit eine speziell interpretierte Klasse von Abkürzungsregeln.

einem Experten, selbst ermittelt. Wir verwenden zur Repräsentation der Relevanz für bestimmte Diagnosen eine *Relevanzmatrix*, bei der Symptome als Zeilenbeschriftungen und Diagnosen als Spaltenbeschriftungen auftreten. Für die Relevanz eines Symptoms ist es in PATDEX₂ von Bedeutung, ob ein Symptom auch Ausdruck eines normalen Systemverhaltens sein kann, z.B. "Relais 21K3 geschaltet", oder ob es einen ausgesprochen pathologischen Charakter, wie z.B. "Spannung 214 zu hoch", besitzt.

5. Kooperation zwischen PATDEX₂, GenRule und der MOLTKE₃-Diagnoseshell

Zum Aufbau der Klassifikationsfähigkeit eines Diagnosesystems ist wichtig, alle (interessanten) Fehler zu identifizieren und in Form von Kontexten im System zu modellieren. Kontexte werden dabei durch Mengen von Symptomwerten, die *Kontextvorbedingungen*, charakterisiert. Durch das Explizieren der Inklusionsbeziehungen der Vorbedingungen wird dann der Kontextgraph aufgebaut. Die Testauswahlfähigkeit wird dadurch realisiert, daß auf der Menge der für einen Kontext interessanten Symptome¹⁴ eine Ordnung definiert wird. Diese Ordnung kann sowohl explizit durch Reihenfolgeregeln angegeben werden als auch implizit über Abkürzungsregeln. Letzteres resultiert dann durch die Konkretisierung einer Metastrategie (z.B. "feuere so viele Abkürzungsregeln wie möglich") in einer anwendbaren Strategie.

GenRule beeinflusst mit allen drei generierten Regelarten die Klassifikationsfähigkeit der Shell. Generierte Reihenfolgeregeln erzwingen dabei, implizit durch die damit verbundene Erweiterung der Vorbedingungen, die Erhebung zusätzlicher Symptome. Abkürzungs- und Redundanzregeln bewirken genau das Gegenteil. Der Benutzer wird allerdings bei jeder Enddiagnose darüber informiert, daß Symptomwerte unsicher abgeleitet bzw. für überflüssig gehalten wurden.

Die Testauswahlfähigkeiten der Shell werden ebenfalls von allen Regelarten beeinflusst. Reihenfolge- und Redundanzregeln haben dabei allerdings lediglich das Hinzufügen bzw. Streichen von Tests zur Folge. Abkürzungsregeln bewirken darüber hinaus die Orientierung der Shell an dem "abkürzungsorientierten diagnostischen Problemlösen" der Servicetechniker durch die Konkretisierung der bereits oben erwähnten Metastrategie.

PATDEX₂ erweitert die Klassifikationsfähigkeit der Shell durch das Einbeziehen von Ähnlichkeiten der aktuellen Situation zu bereits bekannten Fällen. Stehen ausreichend Fälle zur Verfügung, so ist das Testauswahlverfahren von PATDEX₂ eine fallbasierte Alternative zu den bislang beschriebenen Strategien.

5.1. Verbesserung der Wissensbasis

Eine Wissensbasis (WB) kann als eine schwache Hintergrundtheorie angesehen werden, d.h. sie ist redundant, unvollständig und/oder inkonsistent. Die Redundanz äußert sich dabei durch zu viele Diagnosepfade in der WB bzw. durch Pfade, die an sich zu lang sind. Unvollständigkeit der WB dagegen bedeutet, daß Diagnosepfade fehlen bzw. zu kurz sind. Inkonsistenzen in der WB schließlich ergeben sich aus widersprüchlichen Situationen bzw. Diagnosen ihrer Pfade. Es ist nun die Aufgabe der Lernkomponente, die WB mit Hilfe empirischer Diagnosefälle zu verbessern. Für die angesprochenen drei Grundproblematiken bieten sich folgende Lösungsansätze:

¹³ Siehe z.B. [7].

¹⁴ Dies sind alle Symptome, die in den Vorbedingungen möglicher Nachfolgekontexte innerhalb des Graphen vorkommen.

- **Redundanz:**
Diagnosepfade werden durch die Generierung heuristischer Regeln (Abkürzungsregeln, Redundanzregeln) verkürzt. Bei seiner Analyse der WB gibt GenRule eine Warnmeldung aus, sobald es Diagnosepfade findet, die durch das Diagnoseverfahren nicht mehr “verwendet” werden können.
- **Unvollständigkeit:**
Erkennt GenRule, daß Diagnosepfade zu kurz sind, so generiert es Reihenfolgeregeln, die die Erhebung der “fehlenden” Symptome im Rahmen des Diagnoseverfahrens erzwingen. Diagnosefälle, für die GenRule keinen “ähnlichen” Diagnosepfad findet, werden an die PATDEX₂-Teilkomponente weitergereicht, die die Shell um ähnlichkeitsbasierte Klassifikation auf der Grundlage der weitergereichten Fälle erweitert.
- **Inkonsistenz:**
GenRule berücksichtigt explizit die Tatsache, daß zu einer Diagnose häufig mehrere Fehler korrespondieren¹⁵. PATDEX₂ kann widersprüchliche Fälle verarbeiten und nach Ähnlichkeitskriterien anordnen. Lösungen auf Basis derartiger Informationen haben dann eher Hinweischarakter.

Da reale empirische Falldaten nicht fehlerfrei sind, ergibt sich eine zusätzliche Aufgabe für die Lernkomponente, nämlich die vorgelegten Fälle möglichst weitgehend zu filtern. Die beiden grundsätzlichen Hilfsmittel, die die Lernkomponente hierzu verwendet, sind die häufigkeitsorientierte Rechtfertigung der generierten Regeln und die probabilistische Klassifikation auf der Basis von Ähnlichkeiten. So ignoriert GenRule z.B. redundante Diagnosefälle, solange diese nicht “zu häufig” auftreten. Inkonsistente Fälle werden von PATDEX₂ verarbeitet (interaktives Testen der Fallbasis), bis sie nach Auflösung der Inkonsistenzen in die WB Eingang finden. Zum Erkennen unvollständiger Fälle verwendet GenRule eine Kombination unterschiedlicher Determinationsfaktoren¹⁶. Grundsätzlich erscheint hier allerdings, wie auch in den vorher erwähnten Problematiken, eine Einbeziehung allgemeinen technischen Hintergrundwissens sinnvoll.

5.2. Überblick über die Lernkomponente innerhalb der Werkbank

Die Grundlage der Kooperation zwischen PATDEX₂ und GenRule bilden eine gemeinsame Fallrepräsentation sowie ein gemeinsames Fallgedächtnis. Während GenRule immer auf der Fallgesamtheit operiert¹⁷, verwendet PATDEX₂ lediglich die Fälle, für die entweder kein ähnlicher Diagnosepfad existiert oder noch kein Regelgenerierungsverfahren durchgeführt wurde.

Die in der WB vorhandenen Diagnosepfade können automatisch durch eine Teilkomponente von GenRule, den *KnowledgeBaseExpander* (KBE), ermittelt werden. Durch ein inkrementelles Verfahren ist es leicht möglich, Änderungen in der WB bei den extrahierten Diagnosepfaden zu berücksichtigen. Darüber hinaus wird mit Hilfe des Fallgedächtnisses überwacht, ob derartige Änderungen Konsequenzen für die an PATDEX₂ weitergereichten Fälle haben, d.h. ob zusätzlich Fälle weitergereicht bzw. wieder zurückgenommen werden müssen.

Einen Überblick über die Lernkomponente bietet Abb. 3.

¹⁵ Dadurch können “systematische Inkonsistenzen” behandelt werden.

¹⁶ Diese werden durch unterschiedliche Fallgrundgesamtheiten gebildet. Siehe [1].

¹⁷ Dies ist zur Berechnung der Determinationsfaktoren wichtig.

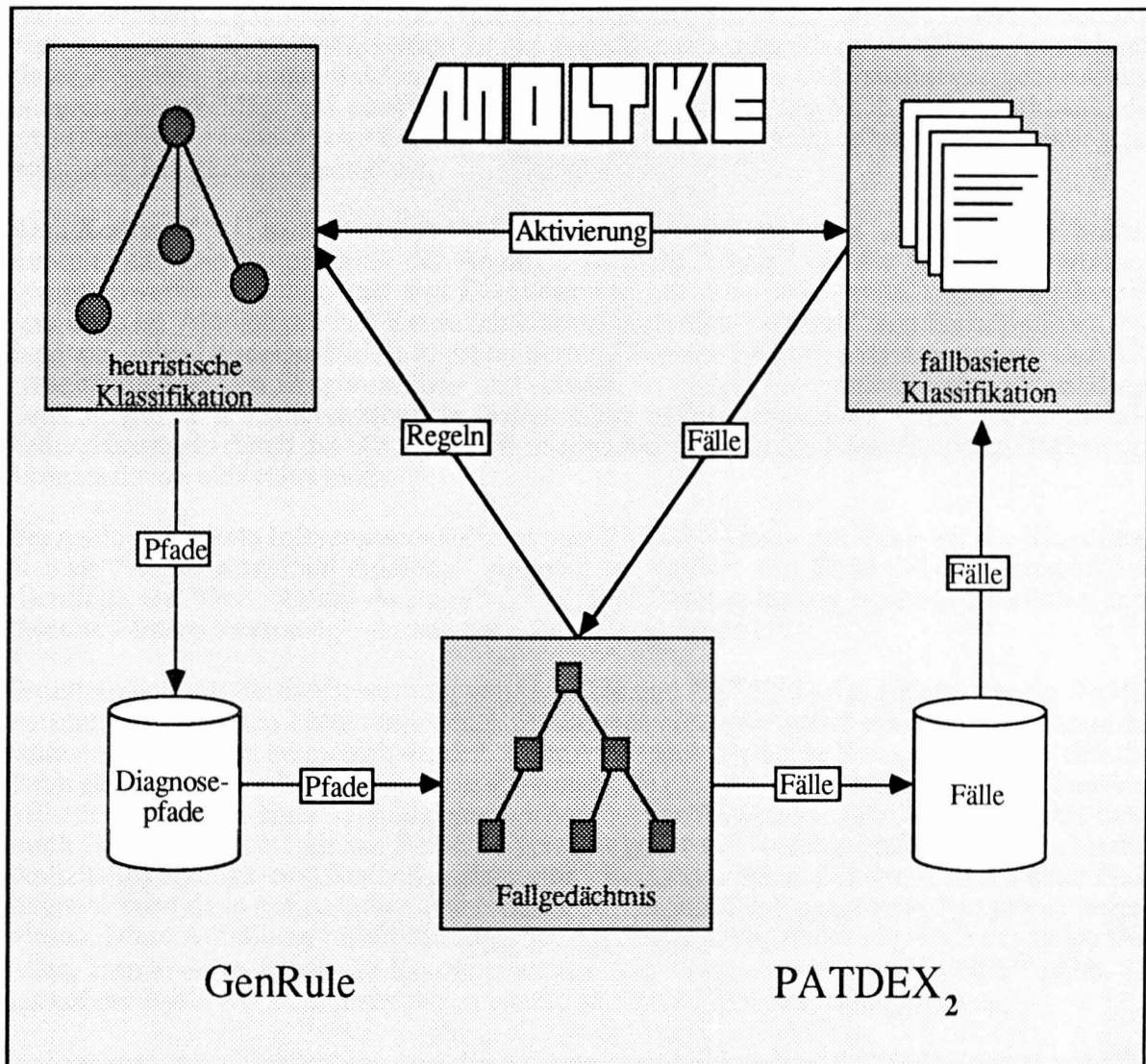


Abb. 3 – Die Lernkomponente innerhalb der MOLTKE₃-Werkbank

6. Bewertung und Diskussion

Die Diagnoseschale ist das Ergebnis einer mehrjährigen Kooperation mit einem anerkannten Ingenieurinstitut und erfüllt die Anforderungen, die seitens des Instituts gestellt wurden ([17], [15]). In diese Kooperation fließen auch theoretische Vorarbeiten im Bereich der Modellierung diagnostischen Problemlösens mit ein [22], die für den praktischen Einsatz umgesetzt wurden [13]. Aufbauend auf den Ergebnissen der Kooperation wurde die MOLTKE₃-Werkbank entwickelt [5].

Für die Lernkomponente diente die ursprüngliche Anwendung (Diagnose von CNC-Bearbeitungszentren: siehe [4]) als Medium zur Integration verschiedener Wissensakquisitions- und Lernverfahren. Der Aspekt der Zweckmäßigkeit der eingesetzten Lernverfahren war dabei von Anfang an wesentlich. Dies führte zu den spezialisierten Teilkomponenten GenRule und PATDEX₂ und deren vollständiger Eingliederung in die Werkbank. Insbesondere die Inte-

gration der Lernergebnisse von GenRule zusammen mit denen der fall- und modellbasierten¹⁸ bzw. manuellen Wissensakquisition ist ein ausschlaggebendes Merkmal für die Attraktivität dieses Ansatzes. So unterstreichen z.B. die in [20] vorgestellten Untersuchungsergebnisse und Integrationsvorschläge auf eindrucksvolle Weise die Qualität der MOLTKE₃-Werkbank, die somit den State-of-the-Art im Bereich Wissensintegration darstellt, mit der Einschränkung auf technische Diagnoseproblematiken.

Nach Kodratoff [11] handelt es sich bei GenRules Analogien um "gute" Analogien mit "voller" Komplexität. Die Komplexität der Analogie ist nicht "trivial", da die Kausalität zwischen Symptomwerten (Situationen) und Diagnose eng mit dem definierten Ähnlichkeitsbegriff verwoben ist. Ähnlichkeit und Kausalität können auch nicht "einfach" kombiniert werden und dann auf jedes beliebige Ziel, d.h. präsentierten Diagnosefall, angewendet werden. Der Allgemeinheitsgrad von Diagnosefällen und -pfaden ist gleich. Übergeneralisierungen sind mit GenRule nur eingeschränkt möglich. Insbesondere ist die Behandlung von Abkürzungs- und Redundanzregeln durch die Diagnoseschell relativ unkritisch, da die Klassifikationsfähigkeit des Systems davon unberührt bleibt.

Der analogiebasierte Inferenzmechanismus von PATDEX₂ kann mit Blick auf die Klassifikation als "transformational Analogy" eingeordnet werden, mit Sicht auf die Testselektion allerdings als "Derivational Analogy" ([9], [10]). Darüber hinaus bestehen Parallelen zum "Memory-Based Reasoning"-Ansatz nach Stanfill und Waltz [18].

Grundsätzlich gilt für die Bewertung von GenRule und PATDEX₂ das gleiche wie für die Bewertung der gesamten Lernkomponente. Sie können sinnvollerweise nur aus dem Gesamtzusammenhang heraus betrachtet werden. Charakteristisch für beide Komponenten ist, daß die durch sie eingesetzten Lernverfahren in starkem Maße die Testselektion beeinflussen. Die Klassifikationsfähigkeit einer WB dagegen wird entweder "erweitert" (ähnlichkeitsbasiert bzw. durch Generierung zusätzlicher Reihenfolgeregeln) oder nur "bedingt verkürzt", indem hierfür explizit Abkürzungs- und Redundanzregeln repräsentiert werden. Bei der Ausgabe einer Enddiagnose wird dann auf unsicher abgeleitete bzw. für überflüssig erachtete Symptome hingewiesen. Diese Aufteilung resultiert zum einen aus Anforderungen der zugrunde liegenden Domäne, zum anderen ist sie die Konsequenz aus dem Vorhandensein von MAKE¹⁹ (Abb. 4), das auf der Basis von Maschinenplänen bereits eine MOLTKE-WB erzeugen kann.

In ihrer konkreten Zielrichtung sind die Ansätze von GenRule und PATDEX₂ neu sowohl für das Verfeinern von WB's als auch für den Bereich des fallbasierten Schließens. Insbesondere konzentrieren sich existierende Systeme meist auf den Klassifikationsaspekt, während z.B. in PATDEX₂ sowohl die Klassifikation als auch die Testauswahl mit fallbasierten Mechanismen realisiert sind. Damit ist auch eine fallbasierte Alternative zu den bislang beschriebenen Testselektionsverfahren der Diagnoseschell möglich.

Hinsichtlich der exponentiellen Komplexität der eingesetzten Verfahren (Zeit- und Speicherkomplexität beim Einfügen ins Fallgedächtnis, Zeitkomplexität beim Generieren der Diagnosepfade durch den KnowledgeBaseExpander) nutzt die Lernkomponente in entscheidendem Maße die Dekomponierbarkeit der Domäne der CNC-Bearbeitungszentren aus. Dies stellt somit eine Einschränkung für die Verallgemeinerbarkeit der erzielten Resultate dar²⁰.

¹⁸ "Modellbasiert" im Sinne eines tiefen, simulationsfähigen Modells des zugrunde liegenden technischen Systems.

¹⁹ Model-Based Automated Knowledge Extractor: hierzu siehe [5] bzw. [16].

²⁰ Die von technischen Systemen allerdings meistens erfüllt wird.

Für eine genauere Bewertung und Diskussion der Lernkomponente siehe [1]. Alle hier beschriebenen Komponenten der MOLTKE₃-Werkbank sind implementiert²¹. Insbesondere ist PATDEX₂ auch als "stand alone"-System einsetzbar.

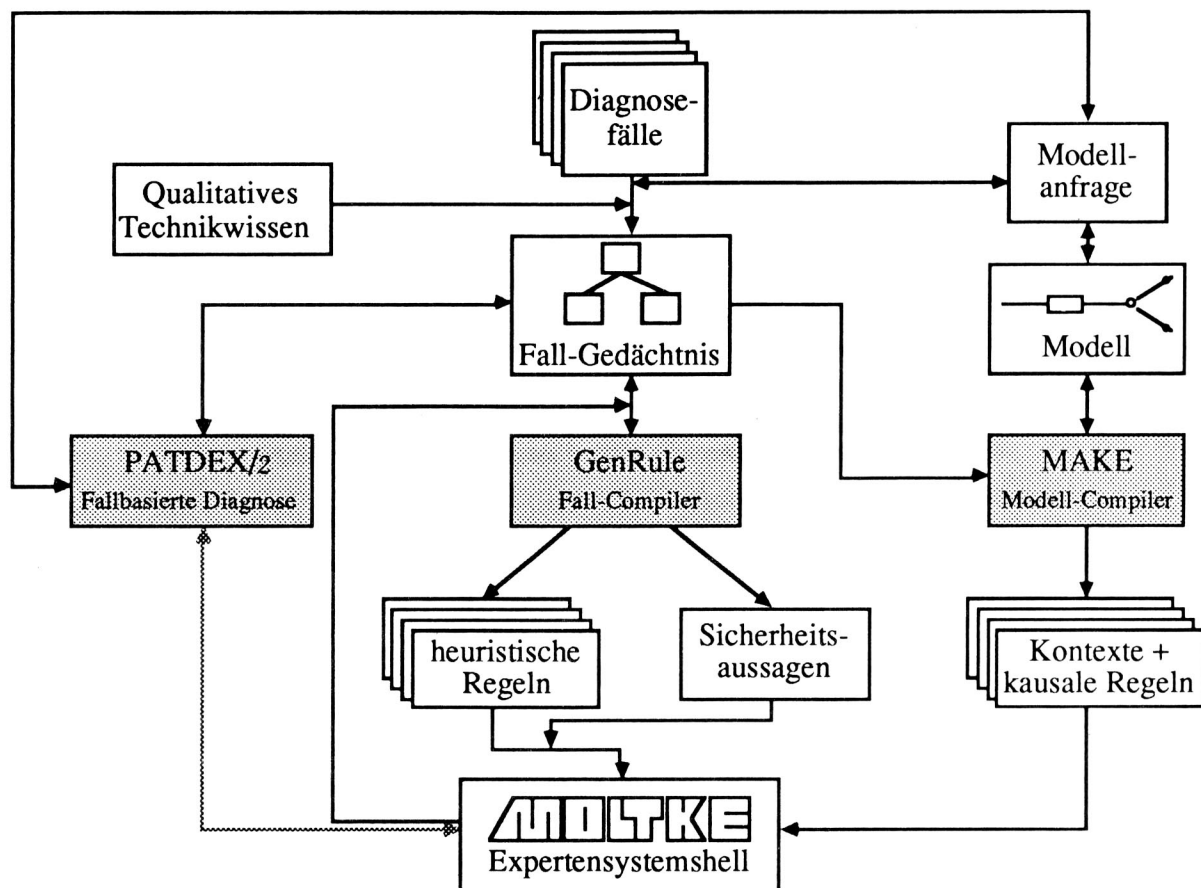


Abb. 4 – Der Aufbau der MOLTKE₃-Werkbank

7. Weitere Aktivitäten

Im Rahmen dieses Beitrages konnte nicht auf die Verwendung zusätzlichen Hintergrundwissens, z.B. allgemeinen qualitativen Technikwissens²² (Abb. 4), zur Verbesserung der Lernverfahren eingegangen werden. Dadurch kann z.B. die Rechtfertigung von GenRule generierter Regeln vereinfacht werden, da hierzu dann gegebenenfalls weniger Fälle erforderlich sind. Weitere wichtige Punkte sind die Erweiterung bzw. Abänderung des Kontextgraphen auf Basis des Fallgedächtnisses sowie die Integration nichtmonotoner Schlußfolgerungsverfahren.

²¹ Die Implementierungssprache ist Smalltalk-80. Entsprechend ist die Werkbank auf fast allen Workstations und größeren PC's verfügbar.

²² Darunter fallen z.B. Zusammenhänge wie: "Relais schalten Ventile, diese wiederum steuern die Hydraulik, etc."

8. Danksagung

Unser Dank gilt in besonderem Maße Herrn Prof. Dr. M. M. Richter, sowie der ML-Arbeitsgruppe in Kaiserslautern, für die unzähligen, engagierten Diskussionen, die die Lernkomponente in der hier beschriebenen Art und Weise erst möglich machten.

9. Literatur

- [1] Althoff, K.-D.: Eine fallbasierte Lernkomponente als integrierter Bestandteil der MOLTKE₃-Werkbank zur Diagnose technischer Systeme, 1991 (in Vorbereitung)
- [2] Althoff, K.-D., De la Ossa, A., Maurer, F., Stadler, M., Weiß, S.: Adaptive Learning in the Domain of Technical Diagnosis, in: Proc. Workshop on Adaptive Learning, FAW Ulm, 1989
- [3] Althoff, K.-D., De la Ossa, A., Maurer, F., Stadler, M., Weiß, S.: Case-Based Reasoning for Real World Applications, interner Bericht, Universität Kaiserslautern, 1990
- [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. EKAW 1989, S. 180-195
- [5] Althoff, K.-D., Maurer, F., Rehbold, R.: Multiple Knowledge Acquisition Strategies in MOLTKE, in: Proc. EKAW 1990, S. 21-40
- [6] Althoff, K.-D., Traphöner, R.: GenRule: Learning of Shortcut-Oriented Diagnostic Problem Solving in the MOLTKE₃-Workbench, interner Bericht, Universität Kaiserslautern, 1990
- [7] Bock, H.: Automatische Klassifikation, Göttingen, 1974
- [8] J.Breuker, B.Wielinga: Model-Driven Knowledge Acquisition: Interpretation Models, Memo 87, Deliverable task A1, Esprit Project 1098; 1987
- [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, Morgan Kaufmann, Los Altos, CA, 1986
- [11] Kodratoff, Y.: Combining Similarity and Causality in Creative Analogy, in: Proc. ECAI 1990, S. 398-403
- [12] Kolodner, J.L.: Maintaining Organization in a Dynamic Long-Term Memory, Cognitive Science (7), S. 243-280, 1983
- [13] Maurer, F.: FOMEX: ein fehlerorientiertes, modulares Expertensystem zur Diagnose von CNC-Bearbeitungszentren, Diplomarbeit, Universität Kaiserslautern, 1989
- [14] Morik, K.: Sloppy Modeling, in: [Morik, K. (ed.): Knowledge Representation and Organization in Machine Learning, Springer Verlag: Berlin, Heidelberg 1989], S. 107-134
- [15] Pfeifer, T. Held, H.-J., Faupel, B.: Aufbau einer Wissensbasis für Fehlerdiagnosesysteme von Bearbeitungszentren, VDI-Z VDI-Verlag 10, 1988
- [16] Rehbold, R.: Integration modellbasierten Wissens in technische Diagnostikexpertensysteme, Dissertation, Universität Kaiserslautern, 1991 (in Vorbereitung)
- [17] Richter, M. M., Pfeifer, T., Althoff, K.-D., Faupel, B., Nökel, K., Rehbold, R.: Abschlußbericht Teilprojekt X6, Sonderforschungsbereich "Künstliche Intelligenz - Wissensbasierte Systeme", Kaiserslautern: 1990

- [18] Stanfill, C., Waltz, D.: The memory based reasoning paradigm, Proc. DARPA Workshop on Case-Based Reasoning, Morgan Kaufmann, 1988
- [19] Traphöner, R.: Ein Konzept zur Verarbeitung von Erfahrungswissen in MOLTKE, Projektarbeit, Universität Kaiserslautern, 1990
- [20] van Someren, M. W., Zheng, L. L., Post, W.: Cases, Models or Compiled Knowledge; a Comparative Analysis and Proposed Integration, in: Proc. EKAW 1990, S. 339-355
- [21] Weiß, S.: PATDEX₂: ein System zum adaptiven, fallfokussierenden Lernen in technischen Diagnosesituationen, Diplomarbeit, Universität Kaiserslautern, 1990
- [22] Wetter, T.: Ein modallogisch beschriebenes Expertensystem, ausgeführt am Beispiel von Ohrenerkrankungen, Dissertation, RWTH Aachen, 1984

