# Bestimmung der Ähnlichkeit in der fallbasierten Diagnose mit simulationsfähigen Maschinenmodellen

Gerd Pews, Frank Weiler & Stefan Weß
Fachbereich Informatik
Universität Kaiserslautern

## 1 Einleitung

Fallbasierte Ansätze spielen bei der Entwicklung von Diagnoseexpertensystemen eine immer größere Rolle. Mit fallbasierten Systemen wie z.B. PATDEX [AW91] oder CcC+ [PG91] wurden zwar erste Erfolge erzielt, man stößt allerdings schnell auf einige grundsätzliche Probleme:

- Eine echte Lösungstransformation ist nicht möglich, d.h. obwohl ein bekannter Fehler in der Baugruppe A gefunden wird, ist es nicht möglich, den gleichen Fehler in einer völlig identischen zweiten Baugruppe A' zu finden.
- Es ist schwer möglich, die Relevanz eines Symptoms für ein bestimmtes Fehlverhalten einzuschätzen. Die Bewertung erfolgt daher oft auf statistischer Basis (vgl. PATDEX). Dabei wird implizit dem Benutzer die Vorauswahl der für eine Fehlersituation relevanten Symptome übertragen. Unter Umständen wird dann einem Symptom, das genau auf den Fehler hindeutet, die gleiche Bedeutung zugemessen wie einem irrelevanten Symptom, das gar nichts mit dem eigentlichen Fehler zu tun hat.

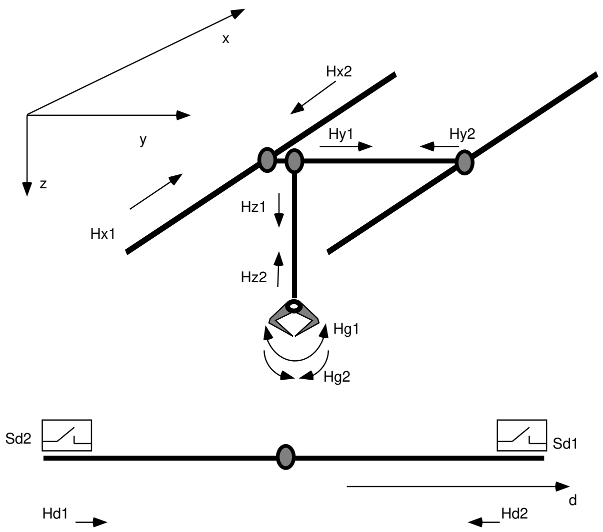


Abbildung 1: Der Werkzeuggreifer

Daher liegt es nahe, fallbasierte Diagnosesysteme durch eine weitere Komponente zu ergänzen, die - allerdings mit höherem Aufwand - diese Probleme umgeht und daher dann eingesetzt werden kann, wenn durch Case-Matching kein zufriedenstellendes Ergebnis erzielt wird. Ein Grund für die oben genannten Probleme ist das Nichtvorhandensein von Wissen über die Maschine. Die Idee für die in dieser Arbeit vorgestellte Komponente liegt in einer Verschmelzung von fallbasierter und modellbasierter Diagnose (vgl. [Kot88, Aam91]). Die Fallbasis ermöglicht eine schnelle

Fokussierung auf die Fehlerursachen und liefert Erfahrungswerte, mit denen fehlende Informationen über das Fehlverhalten ausgeglichen werden kann. Das Maschinenmodell ermöglicht die Lösungstransformation auf Baugruppen mit gleicher oder ähnlicher Funktion. Zusätzlich können die Informationen über das Fehlverhalten ihrer Relevanz entsprechend bewertet und so besonders beachtet bzw. ignoriert werden. Um Lösungstransformationen durchfüren zu können, sind folgende Informationen notwendig:

- Eine Fallbasis mit bereits gelösten Diagnoseproblemen
- Wissen über die Struktur der Maschine
- Wissen über die Funktion der einzelnen Bauteile (konkret und abstrakt)

Die hier vorgestellte Komponente setzt dabei auf die im Rahmen des Moltke-Projektes entwickelten Systeme Patdex[Wes91] (fallbasierte Diagnose) und iMake [Sch92] bzw. Make [Reh91] (modellbasierte Generierung von Moltke-Wissensbasen) auf.

# 2 Funktionsweise

Die Maschine, die wir betrachten wollen, ist ein Werkzeuggreifer, der sich in x-, y- und z-Richtung bewegen läßt (vgl. Abb. 1). Obwohl er recht einfach anmutet, ist sein innerer Aufbau komplex genug, um das System realistisch zu testen. Der Aufbau für die Bewegung in einer Richtung ist in Abb. 2 dargestellt. Um diesen Werkzeuggreifer mit Patdex diagnostizieren zu können, müssten Fallbeispiele für jede mögliche defekte Komponente in jeder der drei Bewegungsachsen eingegeben werden. Durch die Transformation von Lösungen kann man sich nun auf Fallbeispiele einer einzigen Richtung beschränken. Die Fehlverhalten in den anderen Bewegungsrichtungen können dann aus den gegebenen Fallbeispielen und dem Maschinenmodell abgeleitet werden.

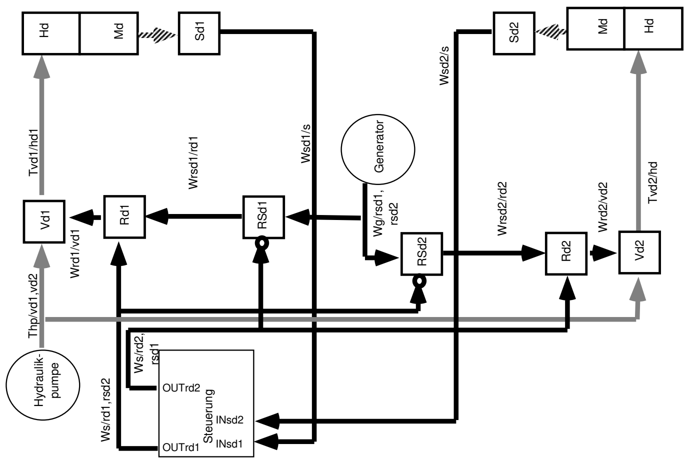


Abbildung 2: Innerer Aufbau einer Schiene

Ein eingegebener Fall wird zunächst abstrahiert, indem man die Symptome auf die abstrakten Symptome "erfüllt Funktion“ bzw. "Fehlfunktion“ abbildet. (Diese Symptome bedeuten nicht, daß das Bauteil in Ordnung bzw. defekt ist, sie sagen nur etwas darüber aus, ob das gewünschte Verhalten erzielt wird) Die Abbildung erfolgt durch eine Verhaltenssimulation mit Hilfe von iMake. Die gewünschten Zustände der Bauteile lassen sich so berechnen und mit den angegebenen Symptomen vergleichen. Weitere Stufen der Abstraktion sind das Abstrahieren der einzelnen Bauteile und das Zusammenfassen von mehreren Bauteilen aufgrund ihrer bekannten Baugruppenbeziehung. Wird nun eine Anfrage an das System gestellt, die nicht mit einem Fall der Fallbasis identisch ist, so werden die Anfrage und der Fall soweit abstrahiert, bis sie identisch sind. Das bedeutet auch, daß man mit Hilfe des Modells fehlende Symptome, die sich aus den bekannten Symptomen ergeben, herleitet und so den Fall und die Anfrage zur Deckung bringt. Um einen

Fall auf eine ähnliche Situation zu übertragen, werden dann die Bauteile, die im Fall vorkommen, abstrahiert. Der Gedanke, der hinter dieser Operation steht, ist der, daß *Ähnlichkeit nichts anderes ist als Gleichheit auf einer höheren Abstraktionsstufe.* Bei der Suche nach der passenden Abstraktion helfen:

- Symptome, die in keiner Beziehung zur Diagnose stehen (also unwichtig sind) können aussortiert werden.
- Das Wissen über Baugruppen bzw. Unterbauteile. Bei gleichen Modulen in einer Maschine sind zwar die Namen der Bauteile verschieden, aber schon bei einer Abstraktion erster Stufe (z. B. Abbildung auf "öffnendes Relais", "Sicherheitsventil") kann Gleichheit erzielt werden.

# 3 Repräsentation des Wissens

Eine Datenstruktur, die unseren Anforderungen genügt, ist die im folgenden dargestellte Netzwerkstruktur. Das Netz besteht aus Objekten, die ein reales oder ein abstraktes Ding darstellen, und typisierten Verbindungsobjekten, die je zwei Netzobjekte verbinden und dadurch eine Funktion oder Beziehung zwischen diesen Objekten repräsentieren. Diese Verbindungsobjekte können auch wieder miteinander verbunden werden. Ausgangspunkt der Repräsentation sind die real existierenden Bauteile und Baugruppen der Maschine, wie etwa ein Relais mit dem Namen R1 oder ein Ventil mit dem Namen V2, die durch entsprechende Netzobjekte repräsentiert werden.

**Wissen über die Verbindungen der Bauteile** Falls in der Maschine ein Bauteil ein anderes Bauteil beeinflußt, werden diese Bauteile durch ein Verbindungsobjekt vom Typ "physikalische Wirkung" verbunden. (Beispiel: Falls R1 das Ventil V1 ansteuert, werden die beiden Objekte durch ein entsprechendes Verbindungsobjekt verknüpft).

**Erklärungen zu den Bauteilen** Jedes real existierende Objekt ist durch ein Verbindungsobjekt mit mindestens einem Netzobjekt verbunden, das die abstrakte Vorstellung des Objekts repräsentiert. Diese Abstraktion kann ihrerseits wieder abstrahiert werden. Dabei ist anzumerken, daß diese Abstraktionen keine Hierarchie bilden müssen. (Beispiel: das Relais R1 ist mit seiner Abstraktion "schließendes Relais" verbunden, diese wiederum mit ihrer Abstraktion "Relais", diese wieder mit "Schalter", etc.)

**Wissen über die räumliche Gruppierung der Bauteile** Zu jedem Objekt kann angegeben werden, zu welcher Baugruppe es gehört, bzw. unter welchem Aspekt man es in der Maschine eingruppieren kann. (Die Bauteile R1 und V1 wären beispielsweise Bauteile einer Schubeinheit für die x-Richtung Sx, sie sind aber auch Bauteile einer Ansteuerung für eine Hydraulik)

**Die Fallbasis** Die Fallbasis besteht aus einer Menge von Netzobjekten "Fall", die mit Verbindungsobjekten auf die Netzobjekte weisen, die die im Fall vorkommenden Bauteile repräsentieren. Die Verbindungsobjekte sind vom Typ Symptom und tragen die Symptomausprägung (Beispiel: Fall1 ist verknüpft mit R1 durch das Verbindungsobjekt "Symptom: `Eingang p1 von R1 hat 3,5V`"). Mindestens ein Verbindungsobjekt ist vom Typ Diagnose und weist auf das in dem Fall defekte Bauteil. Weiter ist es sinnvoll, die Umstände, wie es zu dem Störfall gekommen ist, mit in den Fall aufzunehmen, d.h. die Steuerimpulse der Maschinensteuerung sollten bekannt sein. Diese sind auch in der Regel einfach zu erhalten oder können vom System selbst generiert werden, wenn es weiß, welche Aktion an der Maschine vorgenommen werden sollte (z.B. `Greifer auf Position x = xmax fahren`)

**Die Funktionsweise der Bauteile** Das System kann über ein spezielles Netzobjekt, das als Interface zu einem beliebigen Modell – in unserem Fall ist es das Maschinenmodell iMAKE – dient, dieses Modell benutzen und einfache Vorgänge in der Maschine simulieren. (Beispiel: `Wenn der Eingang p1 des Bauteils R1 Spannung erhält, wird der Ausgang p2 des verbundenen Bauteils V1 Druck führen`).

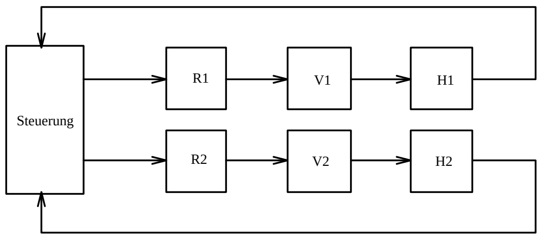


Abbildung 3: Die Beispielmaschine

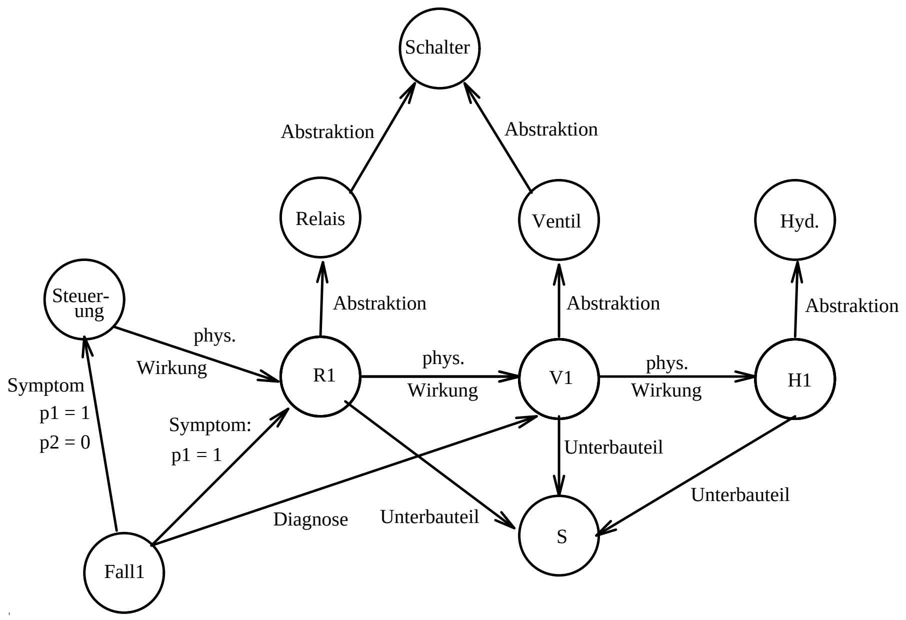


Abbildung 4: Das Netz für die Beispielmaschine

## 3.1 Beispiel

Wir werden als Beispiel im folgenden eine vereinfachte Version des Werkzeugreifers betrachten, die nur aus einer Steuerung und zwei gleichen Modulen besteht (Abb. 3). Diese Maschine wird durch ein Netzwerk repräsentiert, wie es ausschnittsweise in Abb. 4 dargestellt wird. Fall1 beschreibt ein Fehverhalten der Bauteilreihe Steuerung, R1 und V1. Dieser Fall ist in Abb. 5 oben links dargestellt. Daneben befindet sich der Netzausschitt für eine Systemanfrage, die ein Fehlverhalten der baugleichen Bauteilreihe Steuerung, R2 und H2 beschreibt. Zunächst erfolgt die Abbildung der Symptomwerte auf: "erfüllt Funktion" und "Fehlfunktion". Danach werden die Bauteile abstrahiert (R1 zu "Relais", R2 ebenfalls zu "Relais", etc. ). Nun ist ein Matching möglich, da die Fehlfunktion der Hydraulik aus der Fehlfunktion des Ventils hergeleitet werden kann.

# 4 Bewertung

Der Einsatz der vorgestellten Komponente kann durch die durchzuführende Verhaltenssimulation sehr zeitaufwendig werden. Auch ist apriori nicht bekannt auf welcher Abstraktionsebene zwei Fälle verglichen werden müssen, um Lösungen übertragen zu können. Außerdem wird zusätzliches Wissen (in Form eines konkreten Maschinenmodells) benötigt, das separat neben den Fallbeispielen erfasst werden muß. Die vorgestellte Komponente bietet sich daher an, wenn auf dem Wege des direkten Fallvergleichs durch Patdex keine Lösung gefunden werden kann.

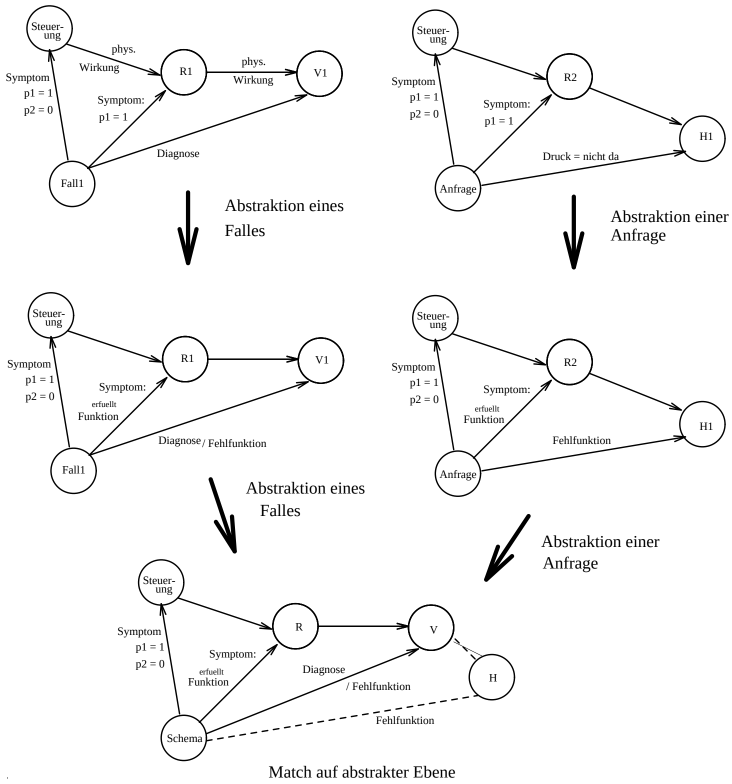


Abbildung 5: Die Abstraktionsschritte, um Fall und Anfrage zu matchen

# Literatur


[Aam91] Agnar Aamodt. *A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning*. Dissertation, University of Trondheim, 1991.

[AW91] Klaus-Dieter Althoff und Stefan Wess. Case-Based Knowledge Acquisition, Learning and Problem Solving for Diagnostic Real World Tasks. In Duncan Smeed, Hrsg, *Proceedings of the 5th European Knowledge Acquisition Workshop EKAW'91*. Springer Verlag, 1991. SEKI-Report SR-91-07 (SFB).

[Kot88] P. Koton. Reasoning about Evidence in Causal Explanation. In Janet L. Kolodner, Hrsg, *Proceedings Case-Based Reasoning Workshop*, Seiten 260–170, San Mateo, California, 1988. DARPA, Morgan Kaufmann Publishers. Clearwater Beach, Florida, USA, May 10–13, 1988.

[PG91] Frank Puppe und Klaus Goos. Improving Case-Based Classification with Expert Knowledge. In Thomas Christaller, Hrsg, *Proceedings of the German Workshop on Artificial Intelligence 1991*, Berlin, September 1991. Springer-Verlag.

[Reh91] Robert Rehbold. *Integration modellbasierten Wissens in technische Diagnostik-Expertensysteme*. Dissertation, Fachbereich Informatik, Universität Kaiserslautern, 1991.

[Sch92] Andreas Schuch. *iMake: Inkrementelle Modellierung und Simulation technischer Geräte zur Generierung einer Wissensbasis für* Moltke *3.0*. Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1992. Betreuer: A. de la Ossa.

[Wes91] Stefan Wess. Patdex/2: Ein System zum fallfokusierenden Lernen in technischen Diagnosesituationen. SEKI-Working Paper SWP91/01, Dept. of Computer Science, University of Kaiserslautern, Germany, Januar 1991.