Fallbasiertes Problemlösen in Expertensystemen - begriffliche und inhaltliche Betrachtungen*

Klaus-Dieter Althoff & Stefan Weß†
Universität Kaiserslautern – FB Informatik
Postfach 3049 – W-6750 Kaiserslautern
e-mail: althoff *or* wess@informatik.uni-kl.de

Zusammenfassung

Im Bereich der Expertensysteme ist das Problemlösen auf der Basis von Fallbeispielen ein derzeit sehr aktuelles Thema. Da sich sehr unterschiedliche Fachgebiete und Disziplinen hiermit auseinandersetzen, existiert allerdings eine entsprechende Vielfalt an Begriffen und Sichten auf fallbasiertes Problemlösen. In diesem Beitrag werden wir einige für das fallbasierte Problemlösen wichtige Begriffe präzisieren bzw. begriffliche Zusammenhänge aufdecken. Die dabei verfolgte Leitlinie ist weniger die, ein vollständiges Begriffsgebäude zu entwickeln, sondern einen ersten Schritt in Richtung eines einfachen Beschreibungsrahmens zu gehen, um damit den Vergleich verschiedener Ansätze und Systeme zu ermöglichen. Auf dieser Basis wird dann der derzeitige Stand der Forschung am Beispiel konkreter Systeme zur fallbasierten Diagnose dargelegt. Den Abschluß bildet eine Darstellung bislang offener Fragen und interessanter Forschungsziele.

Abstract
For the field of expert systems, case-based problem solving is a hot research topic. Since it is within the focus of different fields of research there are many different views on this approach. In this paper, we want to go a first step towards the development of a simple framework which enables the comparison of different systems and approaches. We give an overview of the current state of the art based on an exemplary selection of approaches to case-based diagnosis. Finally, we point out some open questions and interesting research goals.

1 Einführung

Gegenstand dieses Beitrages ist das sehr aktuelle Thema des fallbasierten Problemlösens *(fallbasiertes Schließen, fallbasiertes Lernen; Case-Based Reasoning)*. In "erster Näherung" kann hierunter das Lösen von Problemen anhand von bereits bekannten Beispielen verstanden werden. Dazu werden Erfahrungen in Form von Fallbeispielen gesammelt und in das bereits vorhandene Erfahrungswissen eingeordnet. Aktuelle Probleme werden dann direkt unter Verwendung von ähnlichen Fallbeispielen gelöst.

Zur Präzisierung der zugrunde liegenden Begriffe bzw. zur Konkretisierung des skizzierten Verfahrens des fallbasierten Problemlösens muß berücksichtigt werden, daß sich unterschiedliche Fachgebiete und Disziplinen mit diesem Thema auseinandersetzen und (somit) eine entsprechende Vielfalt von Sichten und Begriffen existiert.

Während aus kognitionspsychologischer Sicht darunter ein Modell für menschliches Problemlösen zu verstehen ist, steht fallbasiertes Lernen aus Sicht des Maschinellen Lernens für Lernverfahren, die direkt auf die ihnen zugrunde liegenden Beispiele referenzieren. Für den Bereich des Knowledge Engineering impliziert fallbasiertes Problemlösen eine dynamische Sicht der Wissensmodellierung, die die scharfe Trennung zwischen Wissensakquisition und -applikation, wie sie z.B. derzeit in der modellbasierten Wissensakquisition existiert, aufhebt.

Für die Entwicklung von Expertensystemen bietet fallbasiertes Schließen darüber hinaus eine Reihe von Ansätzen zur Bewältigung realer Aufgabenstellungen, wie z.B. der Behandlung inhomogener Lösungsräume sowie der Repräsentation und Verarbeitung von Erfahrungswissen (Lern- und Anpassungsfähigkeit des Systems).

Wichtige Begriffe, die mit fallbasiertem Problemlösen verbunden werden sollten, sind somit Lernfähigkeit, Anpassung sowie Integration von Wissensakquisiton und -applikation. Die Automatisierung der Wissensakquisition und -anpassung stellt dabei den Übergang zum Lernen dar. Letzteres wurde bereits von Morik (1987) (im Sinne automatischer Wissensmodellierung) vorgeschlagen.

Im nachfolgenden Abschnitt wollen wir einige für das fallbasierte Problemlösen wichtige Begriffe präzisieren bzw. begriffliche Zusammenhänge aufdecken. Die dabei verfolgte Leitlinie ist weniger die, ein vollständiges Begriffsgebäude zu entwickeln, sondern einen ersten Schritt in Richtung eines einfachen Be-

*Positionspapier für den GWAI-91-Workshop: *Welche Rolle spielen Fälle für wissensbasierte Systeme?*

†Die hier vorgestellte Arbeit wurde zum Teil gefördert durch die Deutsche Forschungsgemeinschaft, SFB 314: "Künstliche Intelligenz - Wissensbasierte Systeme", Projekte X6 und X9.

schreibungsrahmens zu gehen, um damit den Vergleich verschiedener Ansätze und Systeme zu ermöglichen. Obwohl die verwendeten Beispiele und Motivationen häufig diagnostischen Problemstellungen entlehnt sind, sind die meisten Überlegungen doch von grundsätzlicher Art und können entweder direkt oder durch leichte Modifikation auch auf synthetische Fragestellungen angewendet werden.

Im dritten Abschnitt wird dann der derzeitige Stand der Forschung am Beispiel konkreter Systeme zur fallbasierten Diagnose dargelegt. Die Beschreibung und Bewertung exemplarischer Systeme orientiert sich dabei an dem in Abschnitt 2 entwickelten Beschreibungsrahmen. Den Abschluß bildet eine Darstellung bislang offener Fragen und interessanter Forschungsziele.

2 Terminologie

Grundlegend für fallbasiertes Problemlösen ist die Frage, was überhaupt unter einem Fall (Fallbeispiel) zu verstehen ist. Wir wollen als Fall ein in der Realität vorgekommenes Beispiel bezeichnen, d.h. Fälle sind Abstraktionen von zeitlich und räumlich eingrenzbaren Ereignissen bzw. Vorgängen. Derartiges Wissen wird auch als episodisches Wissen bezeichnet [vgl. Strube (1989)]. Sobald Klarheit über die zugehörige Abstraktionsabbildung besteht, findet häufig eine Identifizierung des Falles mit dem zugrunde liegenden Ereignis bzw. Vorgang statt.

Es existieren unterschiedliche Auffassungen über die formale Beschreibung von Fällen. Wir wollen hier von der folgenden sehr allgemeinen Definition ausgehen, die für unsere Zwecke ausreichend ist. Formal wollen wir unter einem Fall ein Tripel aus einer Problembeschreibung, der zugehörigen Lösung und einer Rechtfertigung der Lösung verstehen [vgl. hierzu z.B. Veloso & Carbonell (1989)]:

Definition: Fall
Ein Fall F ist gegeben als ein Tripel (P, R, L) mit einer Problembeschreibung P, einer Rechtfertigung R und einer Lösung L.

Häufig wird dabei der Lösungsweg als Rechtfertigung angegeben. Für die Diagnose kann ein Fall dann z.B. das folgende Aussehen haben: die gegebene Symptomatik beschreibt das Problem, die gestellte Diagnose ist die Lösung und die Folge der Symptombelegungen, die zum Stellen der Diagnose führten, liefert die Rechtfertigung.

Vom zentralen Begriff des Falls leiten sich nun einige für das fallbasierte Problemlösen wichtige Fragestellungen ab. Zum ersten sollte z.B. klar sein, auf welcher Abstraktionsebene Fälle Gegenstand der Betrachtungen sind. Von verschiedenen Autoren wurde hier eine Unterscheidung von zumindest drei Detaillierungsstufen vorgeschlagen [vgl. z.B. Anderson (1989) bzw. Richter (1989)]:

- **Kognitive oder Wissensebene** (Knowledge Level)
- **Repräsentationsebene** (Algorithmic Level)
- **Implementationsebene** (Implementation Level)

Es ist interessant zu wissen, was für Fälle in einem fallbasierten Problemlöser vorkommen und welchen Informationsquellen sie entstammen bzw. in welchem Kontext ein fallbasierter Problemlöser eingesetzt wird (analytische oder synthetische Problematik). Fälle können für vielfältige Aufgaben verwendet werden, z.B. zum Aufbau, zur Verfeinerung oder zur Überprüfung einer Wissenbasis bzw. zur Erklärung bestimmter Inferenzen [Bareiss (1989), Sharma & Sleeman (1988), Althoff (1991), Schank (1989)]. Es ist somit von Interesse, welche Schlüsse aus Fällen gezogen werden sollen, was aus ihnen gelernt bzw. inwieweit Hintergrundwissen zu ihrer Verarbeitung einbezogen werden kann.

Neben der Modellierung eines einzelnen Falles sind für fallbasiertes Problemlösen grundlegend die Repräsentation der Fallgesamtheit in einem Fallgedächtnis (case memory), die Bereitstellung (retrieval) von Fällen aus dem Gedächtnis, die zu einem gegebenen Problem ähnlich sind, sowie die Lösung dieses Problems auf der Basis von Analogien zu den bereitgestellten Fällen (Lösungstransfer).

In diesem Sinne ist fallbasiertes Problemlösen eine Spezialisierung analogiebasierten Problemlösens. Allgemein wird unter Analogie oder analogem Schließen die Transformation und Extension existierenden Wissens aus einer Domäne verstanden, um eine ähnliche Aufgabe in einer anderen Domäne mit ähnlichen Eigenschaften durchzuführen und z.B. ein ähnliches Problem zu lösen. Das Wissen über die bekannte Domäne wird als *Basis*, das in der anderen Domäne zu inferierende Wissen bzw. die zu erzielende Problemlösung wird als *Ziel* des Analogieprozesses bezeichnet. Wichtige Fragen sind nun, welche Bedeutung einander ähnliche Fälle haben, wie sie als solche erkannt und auf gegebene Probleme angewendet werden können. Das erste uns bekannte Modell für Analogie, das Forschungsarbeiten aus dem Bereich fallbasierten Problemlösens explizit berücksichtigte, ist das von Wolstencroft (1989).

Auf die Problematik, daß fallbasiertes Schließen bzw. Analogie bislang hauptsächlich von unterschiedlichen Forschergruppen bearbeitet wurden, verwies z.B. Kolodner (1989). Dabei lag der Forschungsschwerpunkt beim fallbasierten Schließen auf der Fallrepräsentation und -bereitstellung, bei der Analogieforschung dagegen auf dem Lösungstransfer. Dies ist u.a. in unterschiedlichen Grundannahmen begründet, nämlich ob beim Lösungstransfer ein Domänenwechsel stattfindet (Analogie) oder nicht (fallbasiertes Schließen). Eine Ausnahme hinsichtlich der getrennten Bearbeitung dieser Forschungsschwerpunkte bilden z.B. die Arbeiten von Carbonell (1983+1986), Veloso & Carbonell (1989+1991), Aamodt (1990) bzw. Althoff, Maurer & Weß (1990).

Die Art und Weise, wie Wissen über einen bekannten Fall für ein gegebenes Problem nutzbar gemacht wird, ist ein Klassifikationskriterium für fallbasierte Problemlöser. Wir wollen fallbasierte Problemlöser als *auswählend, klassifizierend* oder *fallvergleichend* (classification/interpretive/precedent-based case-based reasoning; case-matching System) bezeichnen, falls sie lediglich Lösungen akzeptieren oder ablehnen, die durch die bereits bekannten Fälle vorgegeben sind. Entsprechend heißt ein fallbasierter Problemlöser *transformierend, modifizierend* oder *fallanpassend* (problem solving case-based reasoning; case-adaptation System), wenn er Lösungen finden kann, die über die Menge der durch die bekannten Fälle vorgegebenen Lösungen hinausgeht [vgl. Hammond (1989), Rissland (1989), Puppe (1990)].

Da eine ganze Reihe von Verfahren existieren, die Fälle verarbeiten, wollen wir nun einige von ihnen auf einfache Weise von einander abgrenzen. Datenbankverfahren, fallbasiertes Problemlösen und Induktion haben z.B gemeinsam, daß sie für die Verarbeitung großer Beispielmengen gedacht sind. Sie unterscheiden sich aber in der Art ihrer Verarbeitung:

- Datenbankverfahren *greifen* lediglich auf Fälle *zu*.
- Fallbasiertes Schließen *interpretiert* die gegebenen Fälle, die im Gedächtnis abgelegt sind.
- Induktion *compiliert* die gegebenen Fälle z.B. in Regeln, Entscheidungsbäume etc.

Fallbasiertes Schließen versucht dabei "bottom-up" eine Theorie der jeweiligen Domäne aufzubauen, Induktion hingegen "top-down". Als Beispiel aus der Realität für die Unterscheidung dieser beiden Vorgehensweisen mögen hier die angelsächsische und die deutsche Rechtsprechung dienen. Während erstere sich in starkem Maße an Präzedenzfällen orientiert, ist in Deutschland die Regelung auf der Basis von Gesetzen wesentlich ausgeprägter.

Induktion, Analogie und Erklärungsbasiertes Lernen haben gemeinsam, daß sie aus gegebenen Fällen lernen. Induktion lernt dabei im allgemeinen aus vielen Fällen, Analogie dagegen aus einem bekannten Fall für einen zweiten, der lediglich unvollständig gegebenen ist. Erklärungsbasiertes Lernen (in seiner Reinform) lernt anhand eines Beispiels seinen deduktiven Prozeß zu fokussieren. Induktive Lernverfahren (similarity-based learning) haben mit analogiebasierten gemeinsam, daß sie Fälle auf der Basis von Ähnlichkeiten verarbeiten. Während bei der Induktion der Zweck die Generalisierung der Fälle ist, ist dies beim Lernen durch Analogie a priori nicht festgelegt. In diesem Sinne ist Analogie allgemeiner als Induktion. Darüber hinaus ist zu beachten, daß analoge, und damit fallbasierte Schlüsse, genau wie induktive Schlüsse im allgemeinen unsicher sind.

In der Statistik und der Mustererkennung werden ebenfalls Fälle aufgrund Ähnlichkeiten verarbeitet. Hierzu werden entweder Ähnlichkeits- oder Distanzmaße verwendet. Beide Darstellungsweisen sind allerdings auf einfache Wiese ineinander überführbar [vgl. Richter & Weß (1991)].

Statt dieser kompakten Beschreibung mit Hilfe einer mathematischen Funktion benutzen viele fallbasierte Problemlöser eine Graphstruktur zur expliziten Repräsentation der Ähnlichkeit zwischen Fällen (Fallgedächtnis). Der Zugriff auf die Fälle wird dabei durch massives Indizieren unterstützt. Eine derartige Struktur soll dabei, zumindest in einer einfachen Form, das menschliche Gedächtnis modellieren [vgl. Schank (1982), Kolodner (1983a+b)]. Diese Ansätze sind bislang von verschiedenen Seiten kritisiert worden [z.B. Thargard & Holyoak (1989), Waltz (1989)]. Eine wichtige Aufgabe eines Fallgedächtnisses ist z.B. die Fähigkeit zur automatischen Reorganisation zur Integration neuer Fälle. Das Fallgedächtnis paßt sich somit ständig an den jeweils bekannten Erfahrungsstand an.

Die Motivation für diese beiden grundsätzlichen Vorgehensweisen (Ähnlichkeitsmaß versus Fallgedächtnis) ist im ersten Fall mathematischer, im zweiten dagegen kognitionspsychologischer Natur. Dabei ist beiden Ansätzen gemeinsam, daß sie, wie auch immer, ähnliche Fälle bereitstellen. Da wir dies für einen zentralen Punkt beim fallbasierten Problemlösen halten, wollen wir diese Problematik mit Hilfe der oben eingeführten drei Abstraktionsebenen ein wenig genauer betrachten.

Die sich auf der Wissensebene stellende Aufgabe ist die *Bewertung der Ähnlichkeit* zwischen Fällen. Das zugrunde liegende Ziel ist dabei die Optimierung des Kosten-Nutzen-Verhältnisses für den betroffenen Problemlöser. Grundsätzlich kann nun entweder die *Beschreibung des Bewertungsvorganges* oder die *Beschreibung der Ähnlichkeit* in den Vordergrund gestellt, d.h. explizit gemacht werden.

Im Falle des Ähnlichkeitsmaßes wird dabei der *Bewertungsvorgang* beispielsweise durch eine mathematische Funktion bzw. durch eine Regel- oder Constraintmenge explizit repräsentiert und z.B. durch eine Funktion der jeweiligen Implementierungssprache bzw. durch eine entsprechende Inferenzmaschine operationalisiert. Beim Fallgedächtnis wird dagegen die Ähnlichkeit explizit durch Listen, Tabellen oder Fallstrukturen (Graphen) dargestellt.

Die Konsequenz daraus ist nun, daß mit Hilfe des Ähnlichkeitsmaßes die *Ähnlichkeit* zwischen Fällen lediglich zur Laufzeit berechnet werden kann. Entsprechendes gilt für den *Bewertungsvorgang* der Ähnlichkeit im Fallgedächtnis. Er wird hier lediglich implizit realisiert über die Bereitstellungs- und Einfügealgorithmen der jeweiligen Implementierungssprache.

Die Quintessenz ist also, daß es für beide Ansätze zutrifft, daß zum einen wichtige Information explizit dargestellt, zum anderen aber auch lediglich implizit kodiert wird. Hinsichtlich der Verständlichkeit und Transparenz derartiger Verfahren läßt sich somit unabhängig von einer Anwendungssituation keine Aussage machen. So können auf einem Fallgedächtnis basierende Verfahren Vorteile für Problemklassen mit eingeschränkter Komplexität bieten, in denen sich die Ähnlichkeitsbewertung auf einfache Weise aus der repräsentierten Ähnlichkeit ableiten läßt. Ähnlichkeitsmaße können dagegen aufgrund ihres hohen Abstraktionsgrades auch für komplexe Anwendungen eingesetzt werden. Die Problematik des Findens eines guten Ähnlichkeitsmaßes kann dabei durch adaptive Lernverfahren erheblich vereinfacht werden [vgl. Stanfill & Waltz (1986), Althoff & Weß (1991)].

Entsprechend der Gestaltung der Fallbereitstellung ergeben sich unterschiedliche Möglichkeiten zur Beurteilung der Anwendbarkeit eines Falles bzw. zur Übertragung des in ihm enthaltenen Wissens auf das gegebene aktuelle Problem. Realisiert werden kann dies durch die Partitionierung des Wertebereiches eines Ähnlichkeitsmaßes mit Hilfe von Schwellwerten, durch die explizite Repräsentation zusätzlichen Hintergrundwissens (z.B. Constraints) oder eine Kombination von beidem.

Die Möglichkeiten zur Beurteilung der Anwendbarkeit von Fällen bzw. der Übertragung des in ihnen enthaltenen Wissens hängt natürlich direkt von der "Reichhaltigkeit"der gewählten Fallrepräsentation ab. Darüber hinaus ist insbesondere für Plausibilitätsüberprüfungen die Tatsache von Interesse, ob der fallbasierte Problemlöser mit anderen Inferenzmechanismen integriert ist.

Fallbasiertes Problemlösen interpretiert also im Fallgedächtnis abgelegte Fälle direkt, indem mit Hilfe von Analogie Lösungen bekannter Fälle auf neue Probleme angewendet werden. Im folgenden wollen wir den für Analogie und fallbasiertes Problemlösen zentralen Begriff der Ähnlichkeit formal fassen. Wir lehnen uns dabei an Richter & Weß (1991) an.

3 Ähnlichkeit und Analogie

Der Ähnlichkeitsbegriff ist von zentraler Bedeutung für die Analogieschlüsse. Wie bei allen Begriffen hat er die beiden Aspekte

- inhaltliche Bedeutung;
- formale Axiomatisierung.

Zur inhaltlichen Bedeutung ist zu sagen, daß es keinen Sinn macht, absolut von *der Ähnlichkeit* zweier Objekte zu sprechen. Ist ein formaler Rahmen festgelegt, so ist auch die Gleichheit von Objekten determiniert. Hingegen läßt auch ein formaler Rahmen die Möglichkeit vieler Ähnlichkeitsmaße offen. Ähnlichkeit besteht also stets nur bezüglich bestimmter Aspekte von Objekten, und diese sind ganz wesentlich von ihrem Verwendungszweck, d.h. von der Pragmatik bestimmt. Betrachten wir ein Beispiel:

- *Möchte man mit einem Auto von Kaiserslautern nach Bonn fahren, so mag man dafür in Frage kommende Autos* ähnlich *nennen, wenn sie den gleichen Komfort, die gleiche Höchstgeschwindigkeit und eine vergleichbare Straßenlage haben.*
- *Sucht man allerdings in der Bonner Innenstadt einen Parkplatz, so wird man wohl Autos* ähnlich *nennen, die in ihren äußeren Maßen nicht stark voneinander abweichen.*

Die Einschränkung auf Aspekte wird durch eine Abstraktionsabbildung [vgl. Richter (1989)] geleistet. Typisch dafür ist die Auswahl gewisser Parameter, die meist zudem noch qualitativ betrachtet werden (d.h. man abstrahiert auch noch von den exakten numerischen Werten).

Die Pragmatik bringt einen weiteren Aspekt in die Diskussion. Wird die Ähnlichkeit für Analogieschlüsse gebraucht, so hat man meist eine gute Vorstellung davon, ob die erhaltene Lösung brauchbar ist. Das läßt einen a posteriori Rückschluß auf die Güte der verwandten Ähnlichkeit zu. Diese a posteriori Information ist aber a priori nicht direkt erhältlich.

Möchte man also zwei Autos als ähnlich bezeichnen, wenn sie eine vergleichbare Straßenlage haben, so läßt sich dies leicht durch Testfahren feststellen. Dem Konstrukteur, der ausgehend von einem bereits fertiggestellten Auto ein neues mit ähnlichen Eigenschaften konstruieren möchte, ist diese Information jedoch nicht zugänglich; ihm ist das zu entwerfende Auto lediglich durch technische und numerische Informationen gegeben und nur danach kann er eine Ähnlichkeitsdefinition erbringen. Das Problem des Konstrukteurs ist, solche Aspekte zu verwenden, die zu einer "guten" Ähnlichkeitsdefinition führen. Dies geschieht meist nicht auf einen Schlag, sondern iterativ;

man kann die Suche nach einem guten Ähnlichkeitsmaß selbst wieder als einen Lernprozeß auffassen.

Bei der Formalisierung des Ähnlichkeitsbegriffes kann man verschiedene Ausgangspositionen einnehmen. Wir wollen hier zwei verschiedene Ansätze verfolgen. Beim ersten Ansatz haben wir es mit einer Relation zu tun. Zunächst benötigen wir eine vierstellige Relation $R(x, y, u, v)$ gelesen als: "*x ist ähnlicher zu y als u zu v*", die den Bedingungen:

$$R(x, x, u, v)$$
$$R(x, y, u, v) \Leftrightarrow R(y, x, u, v) \Leftrightarrow R(x, y, v, u)$$

genügt. Die gewünschte Relation $S(x, y, z)$ ist dann durch $S(x, y, z) :\Leftrightarrow R(x, y, x, z)$ definiert.

Für das analoge Schließen benötigt man letzlich immer die Relation S. Sie wird verwendet, um zu einem x "*ein ähnlichstes y*" aus einer Menge von Fallbeispielen zu finden. Dies genügt dann der Bedingung $\forall z\, S(x, y, z)$. Dazu ist es prinzipiell nicht nötig, irgend etwas über numerische Werte von Ähnlichkeitsbeziehungen zu wissen. Dies ist ein Weg, der mit Hilfe einer reichhaltigen Indizierung häufig in den bereits erwähnten Fallgedächtnissen beschritten wird. In der Praxis ist es jedoch meist angeraten, den Umweg über die numerische Bewertung von Ähnlichkeitsbeziehungen zu gehen.

Damit sind wir bei unserem zweiten Ansatz. Hier wird nicht eine Relation wie z.B. "*x und y sind ähnlich*" axiomatisiert, sondern eine Funktion, die besagt, mit welchem Grade x und y ähnlich sind. Das trägt der Tatsache Rechnung, daß Ähnlichkeit sinnvoller Weise keine Relation ist, die zwischen Objekten entweder besteht oder nicht besteht, sondern besser mit den Begriffen "*mehr oder weniger*" erfaßt werden kann. Zu diesem Zwecke werden Ähnlichkeitsmaße eingeführt:

Definition: Ähnlichkeitsmaß
Ein Ähnlichkeitsmaß auf einer Menge $\mathcal{M}$ ist eine reellwertige Funktion $sim : \mathcal{M}^2 \to [0, 1]$ mit

$$\begin{aligned} sim(x, x) &= 1 \\ sim(x, y) &= sim(y, x) \end{aligned}$$

Die beiden Bedingungen entsprechen der Reflexivität und Symmetrie einer Ähnlichkeit; wichtig ist aber, daß der Grad der Ähnlichkeit über längere Ketten hin abnehmen kann; so ist auch etwa die Relation "*sehr ähnlich*" zu sein i.a. nicht transitiv. Die oben eingeführte "*ähnlicher*"- Relation R kann nun auf einfache Weise bestimmt werden:

$$R_{sim}(x, y, u, v) :\Leftrightarrow sim(x, y) \geq sim(u, v) \qquad (1)$$

Für einen weiteren Zugang zur Formalisierung des Ähnlichkeitsbegriffes über Distanzmaße, ihre Beziehung zu Ähnlichkeitsmaßen bzw. zur Darlegung von Zusammenhängen zwischen Ähnlichkeit und Unsicherheit vgl. Richter & Weß (1991). Einen Überblick über in der Literatur verwendete Ähnlichkeits- und Distanzmaße gibt z.B. Weß (1991).

4 Stand der Forschung

In diesem Abschnitt wollen wir einen Überblick über den Stand der Forschung am Beispiel fallbasierter Diagnosesysteme geben. Wir haben dabei eine Auswahl an Systemen getroffen, die uns in diesem Zusammenhang als sinnvoll erscheint und die wichtigsten Aspekte diagnostischer Aufgabenstellungen berücksichtigt. Bekannte Systeme wie z.B. CHEF, PRODIGY, MEDIATOR, JULIA oder SIZZLE werden hier also nicht berücksichtigt, da ihr Schwerpunkt eher die Bearbeitung synthetischer Fragestellungen ist. Ebenfalls nicht berücksichtigt wurden Systeme zum fallbasierten Lernen von Diagnosestrategien bzw. zur fallbasierten Verfeinerung von Wissensbasen wie z.B. BOLERO [Lopez & Plaza (1991)], REFINER [Sharma & Sleeman (1988)] bzw. GENRULE [Althoff et al.(1991)].

Grundsätzlich muß hier darauf verwiesen werden, daß die aus der Literatur bekannten Systeme nicht gleichermaßen ausführlich beschrieben sind. Das führt dazu, daß häufig Unklarheit über die Realisierung bestimmter Systemkomponenten besteht. Insbesondere ist hiervon die Beschreibung der formalen Bedeutung des Ähnlichkeitsbegriffs in den entsprechenden Systemen betroffen. Als prominentes Beispiel sei hier das CHEF-System von K. Hammond erwähnt.

4.1 PROTOS

PROTOS [Bareiss (1989)] ist ein System zur Problemlösung und Begriffsbildung in analytischen Aufgabenstellungen. Bareiss selbst bezeichnet seinen Ansatz als *Exemplar-Based Knowledge Acquisition.* Ziel von PROTOS ist die Zuordnung eines vom Benutzer präsentierten Falles zu einem dem System bereits bekannten Begriff. Ein Begriff wird dabei extensional durch die Menge der zu diesem Begriff gehörenden Fälle definiert.

Repräsentation:
Ein Fall im Sinne von PROTOS wird als ein Exemplar des Begriffs verstanden, das durch eine Menge von Attribut-Wert Paaren repräsentiert wird. Die dem System präsentierten Fälle werden von PROTOS dabei unverändert, d.h. insbesondere nicht generalisiert, gespeichert. Der Zugriff zu gespeicherten Fällen erfolgt durch eine Indizierung über die Attribute des Falles, die *remindings* genannt werden. Die Fallrepräsentation wird erweitert durch Relationen wie z.B. *part-of* oder *causes*, die zwischen Attributen und Begriffen bestehen können. Die von PROTOS vorgegebenen Relationen können dabei vom Benutzer nicht erweitert

werden. Relationen und Attribute werden in PROTOS durch numerische Werte gewichtet. Die Gewichtung repräsentiert dabei die Relevanz eines Merkmals für die Zugehörigkeit des Falles zum jeweiligen Begriff bzw. die *Erklärungsstärke (explanatory strength)* einer bestimmten Relation.

Verarbeitung:
Die Lösung eines aktuellen Problems (repräsentiert durch eine Menge von präsentierten Attributwerten) ist der Begriff, zu dem der ähnlichste Fall gehört, der PROTOS bekannt ist. Zum Retrieval von Fällen werden dabei Ketten von Ableitungen über den definierten Relationen, *explanations* genannt, verwendet. Ein Fall wird von PROTOS als *ähnlich* akzeptiert, wenn das kumulierte Gewicht der gefundenen Erklärung einen definierten Schwellwert überschreitet.

Lernphase:
Die Lernphase von PROTOS besteht darin, weitere Relationen einzufügen und die Gewichtung der *remindings* zu verändern. Wird die gefundene Klassifikation vom Benutzer abgelehnt, so wird der Benutzer aufgefordert, den vorliegenden Fall durch die Eingabe von weiteren Beziehungen zu Begriffen und Fällen genauer zu spezifizieren. Die Wissensbasis von PROTOS wird so schrittweise durch den Benutzer verfeinert.

Bewertung:
PROTOS repräsentiert Wissen in Form eines semantischen Netzwerks aus Begriffen, Attributen und Relationen, es kann dieses Wissen jedoch nur zur Klassifikation eines Falles, nicht aber zur Adaption einer gefundenen Lösung einsetzen. Die von PROTOS verwendete Repräsentation der Fälle ist flach. Obwohl mit PROTOS bereits diagnostische Aufgabenstellungen prototypisch gelöst wurden[1], kann die vom System verfolgte Diagnosestrategie nicht beeinflußt werden, d.h. PROTOS behandelt nicht die für reale Diagnoseaufgaben relevante Problematik der Auswahl des nächsten Tests.

4.2 Memory-Based Reasoning

Der *Memory-Based Reasoning* (MBR) Ansatz wurde 1986 von Craig Stanfill und David Waltz vorgestellt. Der Ansatz kann als paralleler, rein syntaktischer, fallbasierter Ansatz eingeordnet werden und unterscheidet sich damit wesentlich von anderen bekannten Ansätzen.

Repräsentation:
Fälle werden wie in PROTOS als Exemplare aufgefaßt, die flach und unverändert gespeichert werden.

Verarbeitung:
Der von Memory-Based Reasoning Systemen verfolgte Ansatz ist relativ einfach. Für alle dem System bekannten Fälle wird parallel eine vorgegebene Bewertungsfunktion berechnet. Das aktuelle Problem wird gelöst, indem die Lösung des ähnlichsten Falles übernommen wird. Das von MBR-Systemen verwendete Ähnlichkeitsmaß ist dabei jedoch nicht statisch, sondern kann in Abhängigkeit von den in der Fallbasis konkret enthaltenen Fälle dynamisch definiert werden[2].

Lernphase:
Gelöste Fälle werden in die Fallbasis des Systems aufgenommen und bei der nächsten Berechnung des dynamisch definierten Ähnlichkeitsmaßes mit berücksichtigt.

Bewertung:
Memory-Based Reasoning Ansätze sind bedingt durch die massiv parallele dynamische Bewertung der bekannten Fälle sehr schnell. MBR kann dabei auch als rein statistisch motivierter Klassifikationsansatz (ohne Hintergrundwissen) verstanden werden. Da bekannte Fälle nur interpretiert und nicht adaptiert werden können, gehören Memory-Based Reasoning Systeme zur Klasse der fallvergleichenden Systeme.

4.3 CASEY

CASEY [Koton (1988)] ist ein fallbasiertes Diagnosesystem für Herzerkrankungen, welches auf dem von Kolodner (1983a+b) publizierten Fallgedächnis aufsetzt. CASEY kombiniert dabei einen ähnlichkeitsbasierten Ansatz mit der Verwendung von kausalem Hintergrundwissen.

Repräsentation:
Ein Fall besteht in CASEY aus einer Menge von Attribut-Wert Paaren, mit denen eine kausale Erklärung und die entsprechende Diagnose assoziiert ist. Fälle werden von CASEY in einem hierarchisch organisierten Fallgedächnis gespeichert. Dabei wird innerhalb der Hierarchie top-down zwischen gemeinsamen und mehr diskriminierenden Attributen unterschieden.

Verarbeitung:
CASEY versucht zunächst unter Verwendung des Fallgedächnisses einen ähnlichen Fall zu finden. Ist die Diagnose des Falles für das vorliegende Problem noch nicht geeignet, so versucht CASEY die gefundene Diagnose durch Anwendung der Hintergrundtheorie an die aktuelle Situation anzupassen. Kann kein ähnlicher Fall in der Fallbasis gefunden werden, so versucht CASEY auf der Basis des vorhandenen Hintergrundwissens eine geeignete Diagnose zu bestimmen.

Lernphase:
Wird ein aktuelles Problem durch die Anwendung eines bereits bekannten Fall gelöst, so wird die Gewichtung der zur Lösung verwendeten Merkmale erhöht.

[1] Mit PROTOS wurde ein System zur Diagnose von Hörstörungen entwickelt.

[2] Ein sehr einfaches Beispiel für eine solche dynamische Definition ist die Approximation der Relevanz eines Merkmals durch die relative Häufigkeit mit der es in bestimmten Situationen aufgetreten ist.

Wird ein neues Problem durch Adaption einer bekannten Lösung bzw. durch die Anwendung des kausalen Hintergrundwissens erfolgreich gelöst, so sortiert CASEY den neuen Fall in das Fallgedächnis ein.

Bewertung:
Bei dem CASEY zugrunde liegenden Modell handelt es sich nicht um ein tiefes funktionales Modell im Sinne des *Qualitative Reasoning*, sondern "lediglich" um kausales Hintergrundwissen. Der fallbasierte Ansatz von CASEY ist vergleichbar mit dem erklärungsbasieren Lernen (EBL). CASEY setzt dabei "vollständiges" Hintergrundwissen voraus. Schlußfolgerungen sind dann nur im Rahmen dieser statischen Theorie möglich. In diesem Sinne operationalisieren die Fallbeispiele in CASEY die aus der Theorie ableitbaren Schlüsse. Verglichen mit PROTOS wird in CASEY die Beschreibungssprache auf die Relation *causes* eingeschränkt.

4.4 CREEK

CREEK [Aamodt (1990)] ist eine Weiterentwicklung des von PROTOS verfolgten fallbasierten Ansatzes. Ziel von CREEK ist die Einbettung des fallbasierten Problemlöseprozesses in eine homogene Gesamtarchitektur, die es ermöglicht, alle Wissensquellen für die fallbasierte Problemlösung zur Verfügung zu stellen. CREEK verwendet daher neben fallbasiertem auch modellbasiertes und regelbasiertes Wissen zur Kontrolle und Fokussierung des Problemlöseprozesses.

Repräsentation:
Fälle werden in CREEK als Sammlung von Begriffen repräsentiert. Die CREEK Architektur basiert dabei auf unabhängigen Moduln, die aber eine gemeinsame Basis, *conceptual knowledge fundament* genannt, besitzen. Aufgabenspezifische Begriffe wie *Diagnose*, *Symptom* bzw. *Index* werden in der Repräsentation nicht von domänenspezifischen Begriffen wie *Auto* und *Batterie* unterschieden. Die Basis der Repräsentationsprache von CREEK sind Frames.

Verarbeitung:
Der Problemlöseprozeß von CREEK kann durch die Phasen *activate, explain* und *focus* beschrieben werden. Zunächst werden von CREEK die Begriffe aktiviert, welche Teile der eingegebenen Problembeschreibung bzw. das vorgegebene Goal *matchen*. Im Anschluß daran werden durch einen Prozeß, *goal-focused spreading activation* genannt, weitere Begriffe aktiviert. CREEK versucht dann eine Menge von Fällen zu bestimmen und differierende Merkmale zu erklären. Ist ein ähnlichster Fall bestimmt, so wird die Lösung übernommen bzw. entsprechend angepaßt.

Falls zum präsentierten Problem ein ähnlicher Fall (Bewertung ist größer als vordefinierter Schwellwert) existiert, verfolgt CREEK eine fallbasierte Strategie, ansonsten wird eine regelbasierte Strategie angewendet. Scheitert auch diese, so wird versucht durch die Interpretation des repräsentierten Maschinenmodells eine Diagnose zu finden. Je nach Erfolg der angewendeten Methode wird der entsprechende Schwellwert dann angepaßt. Modellbasiertes Wissen wird in CREEK auch zur Kontrolle der ausgeführten Generalisierungen eingesetzt. Zur Repräsentation von heuristischem und strategischem Wissen können Regeln verwendet werden.

Lernphase:
Konnte die Lösung des ähnlichsten Falles für den aktuellen Fall übernommen werden, so werden die zum Retrieval verwendeten Indizes verstärkt und es wird versucht, die beiden Fälle durch die Generalisierung von Merkmalen zu einem neuen Fall zu vereinigen. Wurde das aktuelle Problem durch die Modifizierung der bekannten Lösung gelöst, so wird der neue Fall in die Fallbasis eingetragen und die Unterschiede explizit vermerkt.

Bewertung:
Die Kombination eines fallbasierten Ansatzes mit anderen Diagnoseansätzen ist sinnvoll. Die Entscheidung, ob ein bestimmter Problemlösungsmechanismus für ein gegebenes Problem anwendbar ist, wird in CREEK auf den einfachen Vergleich mit einem Schwellwert reduziert. Hier ist eine differenziertere Bewertung und eine Einschätzung "der eigenen Kompetenz" des Problemlösers nötig. Die CREEK Architektur ist eher als Konzept, denn als Prototyp zu verstehen. Einige Teilkomponenten wurden bereits realisiert. Die Realisierung weiterer Teilkomponenten (z.B. Lösungsadaption) bleibt allerdings unklar.

4.5 PATDEX/2

PATDEX/2 [Weß (1991), Althoff & Weß (1991)] wurde im Rahmen des MOLTKE Projektes [vgl. Richter (1991)] an der Universität Kaiserslautern entwickelt und stellt den fallbasierten Problemlöser der MOLTKE Werkbank zur Diagnose technischer Systeme dar. Ein Prototyp PATDEX/1 wurde bereits 1988 realisiert [vgl. Althoff et al. (1989)].

Repräsentation:
Fallbeispiele werden in PATDEX/2 als Protokolle des diagnostischen Problemlöseverhaltens von Serviceexperten aufgefaßt. Sie werden dabei als Liste von Symptom-Wert Paaren repräsentiert. In PATDEX/2 erfolgt, im Gegensatz zu den bisher vorgestellten Ansätzen, sowohl die Klassifikation als auch die Testauswahl fallbasiert.

Wir müssen also zwischen Diagnose- und Strategiefällen unterscheiden. Diagnosefälle modellieren dabei das (Fehler-) Klassifikationswissen und Strategiefälle das strategische Wissen eines Serviceexperten. Grundlage für beide Fallarten sind dabei die durch die Beobachtung eines Experten gewonnenen Fallbeispiele. Eine Trennung in unterschiedliche Fallarten ist sinnvoll, da so in PATDEX/2 das Klassifikationsverhalten und die vom System verfolgte Diagnosestrategie unabhängig voneinander verbessert werden können.

Die in PATDEX/2 verwendete Fallstruktur orientiert sich an dem durch OPS5 bekannten Rete-Netzwerk. Zu weiteren Steigerung der Effizienz werden darüber hinaus die Fälle dynamisch in verschiedene "Aktivierungsebenen" eingeordnet. Dadurch kann PATDEX/2 sich bei der Berechnung der Ähnlichkeit auf die Fälle der jeweils höchsten Aktivierungsebene beschränken.

Verarbeitung:
Die Bestimmung des ähnlichsten (Diagnose-) Falles erfolgt in PATDEX/2 nach Maßgabe eines dynamischen Ähnlichkeitsmaßes. Überschreitet die Bewertung eines Falles einen lokal definierten Schwellwert, so feuert die entsprechende Diagnose. Ansonsten wird versucht, die Informationssituation durch die Erhebung eines weiteren Symptoms so zu verbessern, daß anschließend eine Diagnose möglich wird. Ein Fall wird in PATDEX/2 umso eher als eine mögliche Lösung des aktuellen Problems akzeptiert, je niedriger die bei einer Fehldiagnose entstehenden "Kosten" sind.

Zur Berechnung der Ähnlichkeit zwischen Fall und vorliegender Situation wird in PATDEX/2 ein mit dem *Contrast Model* [vgl. Tversky (1977)] verwandtes Ähnlichkeitsmaß eingesetzt, welches auf dem Vergleich von übereinstimmenden, sich widersprechenden und noch unbekannten Symptomen aufbaut. Einzelne Symptome werden dabei gemäß ihrer Relevanz für eine bestimmte Diagnose gewichtet. Zur (initialen) Bestimmung der Relevanzen verwendet PATDEX/2 einen dem *competitive learning* [vgl. Rumelhart & Zipser (1985)] verwandten konnektionistischen Lernansatz. Für die Testauswahl erfolgt eine Anpassung aufgrund einer Schätzung der mittleren Diagnosekosten mit Hifle eines A^*-ähnlichen Verfahrens.

Für die Relevanz eines Symptoms ist es in PATDEX/2 auch von Bedeutung, ob ein Symptom unter Umständen Ausdruck eines normalen Systemverhaltens sein kann oder ob es einen ausgesprochen pathologischen Charakter besitzt. Dieses Wissen ist Teil der maschinenspezifischen Hintergrundtheorie im MOLTKE System und kann in PATDEX/2 angewendet werden.

Lernphase:
Wird eine Diagnose von PATDEX/2 gestellt, so wird der Benutzer gebeten, diese zu verifizieren. Ist die Diagnose korrekt, so wird die Relevanz der übereinstimmenden Symptome erhöht bzw. die Relevanz der sich widersprechenden Symptome verringert. Die von PATDEX/2 aktuell verfolgte Diagnosestrategie wird in Form von Strategiefällen in die Strategiefallbasis aufgenommen. Dabei werden unter Umständen andere Strategiefälle verdrängt. Die Diagnosekosten werden aktualisiert und der Fall als erfolgreich gelöst vermerkt. Unterscheidet sich die aktuelle Situation wesentlich von dem in der Fallbasis enthaltenen Fallbeispiel, so wird ein neuer Diagnosefall generiert und in die Fallbasis aufgenommen.

Wird die Diagnose vom Benutzer abgelehnt, so werden die Relevanzen der sich widersprechenden und die Relevanzen der noch unbekannten Symptome erhöht bzw. die Relevanz der übereinstimmenden Symptome verringert. PATDEX/2 löscht (temporär) alle Fälle mit der widerlegten Diagnose aus der Fallbasis, markiert den widerlegten Fall und versucht eine bessere Diagnose zu finden.

Bewertung:
PATDEX/2 ist stabil hinsichtlich der Verarbeitung vager Fälle und berücksichtigt unsicheres Wissen in Form von Defaultwerten für Symptome. Das PATDEX/2 System basiert auf zweijährigen Erfahrungen mit dem Prototypsystem PATDEX/1 und ist vollständig in die MOLTKE Werkbank integriert und dort für die Bearbeitung von Ausnahmen bzw. Sonderfällen zuständig. Entsprechend kann PATDEX/2 das in der Werkbank repräsentierte Wissen als Hintergrundwissen verwenden (z.B. zum Identifizieren abnormer Symptomwerte). Darüber hinaus ist PATDEX/2 aber auch als stand-alone System einsetzbar. PATDEX/2 besitzt somit fest definierte Rollen. In beiden Fällen wird von PATDEX/2 keine Lösungsadaption vorgenommen. Im stand-alone-Betrieb ist dies aus Effizienzgründen nicht sinnvoll, im Falle der Integration in die Werkbank wird diese Aufgabe durch die Diagnoseshell übernommen. Im Gegensatz zu vielen anderen Systemen wird in PATDEX/2 die Testauswahl explizit berücksichtigt.

5 Ausblick

Fallbasiertes Problemlösen erscheint uns aus den verschiedensten Gründen ein sehr vielversprechendes Forschungsthema zu sein. So sind z.B. Rechtsprechung, Medizin oder Wirtschaftswissenschaften prominente Bereiche, in denen fallbasiertes Problemlösen zum Alltag gehört. Aber selbst einer so formalen Disziplin wie der Mathematik ist, beispielsweise bei der Beweisfindung, eine Orientierung an bekannten Fallbeispielen nicht fremd [vgl. Kerber (1989)].

Es wurde bereits in der Einführung dargelegt, daß sich eine ganze Reihe unterschiedlicher Disziplinen aus einer spezifischen Sichtweise mit dem Thema des fallbasiertem Problemlösens auseinandersetzen. Dies läßt eine Vielzahl interessanter Ergebnisse erhoffen. Zum anderen sehen wir hierin die Möglichkeit zur Integration von Ansätzen aus Forschungsgebieten wie Knowledge Engineering/Wissensakquisition, Kognition, Maschinelles Lernen, Wissensrepräsentation, Nichtmonotones Schließen, Statistik und Mustererkennung. Auf den Nutzen, der von der Integration von Ansätzen dieser Gebiete zu erwarten ist, verwiesen u.a. Morik (1990), Janetzko & Strube (1991), Aamodt

(1990), Wrobel (1991) bzw. Althoff & Weß (1991).

Ein wichtiges Forschungsziel im Bereich des fallbasierten Problemlösens ist die Definition von "Rollen", die ein fallbasierter Problemlöser im Rahmen einer Gesamtarchitektur [z.B. einer Expertensystemwerkbank] "spielen" kann. Interessante Fragestellungen sind hier, welche Mechanismen innerhalb einer Architektur sinnvollerweise integriert werden sollten, wie fallbasiertes Schließen mit anderen Schlußfolgerungsmechanismen kombiniert bzw. wie eine realistische Kosten/Nutzen-Abschätzung hierfür gefunden werden kann. Eine Übersicht über die Integration von fall- und modellbasiertem bzw. compiliertem Wissen geben van Someren, Zheng & Post (1990).

Erste Vorschläge sind die Systeme PRODIGY [z.B. Veloso & Carbonell (1989+1991)], CREEK [Aamodt (1990)], CcC [Puppe (1990)] (Erweiterung von MED2/FAKTA) und MOLTKE [Althoff (1991)].

Die Tragfähigkeit fallbasierter Problemlösemechanismen wird sich daran messen lassen müssen, inwieweit es gelingt, Hintergrundwissen in die einzelnen Teilverarbeitungsschritte einzubringen, das einer "vernünftigen" Kosten-Nutzen-Analyse standhält. Erste Ansätze für den Bereich der Diagnose sind die Einbeziehung taxonomischen bzw. kausalen Hintergrundwissens [vgl. z.B. Koton (1988), Aamodt (1990), Weß (1991)]. Durch derartiges Hintergrundwissen werden einfache Anpassungen der Lösungen bzw. Lösungswege bereits bekannter Fälle an neue Gegebenheiten durchführbar. Leicht anpaßbare Fälle werden auch als *Softcases*, *Routinefälle* oder *Fälle mit kleiner Variation* bezeichnet. Dagegen sind *Fälle mit großer Variation*, sogenannte *Hardcases*, unter Kosten-Nutzen-Aspekten kaum behandelbar, da zu ihrer Verarbeitung entweder sehr detailliertes Hintergrundwissen oder in größerem Maße Common-Sense-Wissen erforderlich ist.

Eine weitere wichtige Forschungsaufgabe ist die Untersuchung von Gemeinsamkeiten und Unterschieden von fallbasierter Diagnose bzw. fallbasierter Planung, insbesondere für bestimmte (noch zu definierende) Problemklassen. Querbezüge sind z.B. gegeben durch die Strategieaufgabe innerhalb des Diagnoseprozesses, durch Klassifikationsteilaufgaben innerhalb des Planungsprozesses sowie durch die Einbeziehung von Modellen des qualitativen Schließens als Hintergrundwissen. Fallbasierte Mechanismen für synthetische Aufgabenstellungen sind z.B. beschrieben in Hammond (1989) [CHEF], Veloso & Carbonell (1989+1991) [PROGIGY], Cunis & Günter (1991) [PLAKON] bzw. Humm, Schulz, Radtke & Warnecke (1991) [MOLTKE-P].

"Last, but not least" ist mit Spannung abzuwarten, inwieweit Ergebnisse aus dem Bereich der Kognitionsforschung weitere bzw. detailliertere Modellvorschläge für fallbasiertes Problemlösen liefern. Aus psychologischer Sicht interessante Forschungsziele wurden z.B. in Strube & Janetzko (1990) dargelegt. Grundsätzlich sind wir der Meinung, daß die weitgreifendsten Fortschritte für fallbasiertes Problemlösen im Rahmen einer interdisziplinären Zusammenarbeit zu erwarten sind, wo sich die Wissenschaftler der beteiligten Gebiete weniger an einem "Input-Output"-Denken orientieren, sondern an interaktiver Zusammenarbeit [vgl. hierzu auch Schmalhofer (1991)].

6 Danksagung

Wir möchten uns hier bei Herrn Prof. Dr. Michael M. Richter bedanken, ganz besonders für die Unterstützung bei der Formalisierung bestimmter Fragestellungen sowie das Aufzeigen von Querbezügen zu anderen Fach- bzw. KI-Disziplinen. Eine große Hilfe war uns auch Frank Maurer durch seine vielen kritischen Anmerkungen und seine weiterführenden Ideen. Mike Stadler und Ralph Traphöner waren in großem Maße am ursprünglichen Entwurf von PATDEX bzw. GENRULE beteiligt.

7 Literatur

Aamodt, A. (1990). A Computational Model of Knowledge-Intensive Learning and Problem Solving. Proc. EKAW-90, 1-20

Althoff, K. -D. (1991). Eine fallbasierte Lernkomponente als integrierter Bestandteil der MOLTKE -Werkbank zur Diagnose technischer Systeme. Dissertation, Universität Kaiserslautern (in Vorbereitung)

Althoff, K. -D. , Kockskämper, S. , Maurer, F. , Stadler, M. & Weß, S. (1989). Ein System zur fallbasierten Wissensverarbeitung in technischen Diagnosesituationen. In: Retti, J. & Leidlmeier, K. (eds.) (1989), 5. Österreichische Artificial-Intelligence- Tagung, 65-70, Berlin, Heidelberg: Springer Verlag

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. (1990). Die Lernkomponente der MOLTKE Werkbank zur Diagnose technischer Systeme. In: Morik (1991), 58-64 **Althoff, K. -D. , Maurer, F. & Weß, S. (1990).** Case-Based Reasoning and Adaptive Learning in the MOLTKE Workbench. Eingereicht.

Althoff, K. -D. & Weß, S. (1991). Case-Based Knowledge Acquisition, Learning and Problem Solving in Diagnostic Real World Tasks. Eingereicht.

Anderson, J. R. (1989). A Theory of the Origins of Human Knowledge. Artificial Intelligence - Special Volume on Machine Learning - , 40, 313-352

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

Carbonell, J. G. (1983). Learning by analogy: formulating and generalizing plans from experience. In: Michalski, R. S. Carbonell, J. G. & Mitchell, T. M. (eds.), Machine Learning. Palo Alto: Tiogo Publishing Co.

Carbonell, J. G. (1986). Derivational Analogy in Problem Solving and Knowledge Acquisition. In: Michalski, R. S. Carbonell, J. G. & Mitchell, T. M. (eds.), Machine Learning, Vol II. Los Altos: Morgan Kaufmann

Cunis, R. & Günter, A. (eds.)(1991). Das PLAKON Buch. Springer Verlag

Hammond, K. (ed.)(1989). Proc. of the 2nd DARPA Workshop on Case-Based Reasoning. Holliday Inn, Pensacola Beach: Morgan Kaufmann

Humm, B. , Schulz, C. , Radtke, M. & Warnecke, G. (1991). A System for Case-Based Process Planning. Proc. 1st CIRP Workshop on Learning in Intelligent Manufacturing Systems (IMS), Budapest

Janetzko, D. & Strube, G. (1991). Fallbasiertes Schließen und modellbasierte Wissensakquisition. Schriftliche Ausarbeitung zum gemeinsamen Workshop der GI-Fachgruppen Knowledge Engineering und Kognition, Kaiserslautern

Kerber, M. (1989). Some Aspects of Analogy in Mathematical Reasoning, SEKI-Report SR-89-12, Universität Kaiserslautern

Kolodner, J. L. (1983a). Maintaining Organization in a Dynamic Long-Term Memory. Cognitive Science, 7, 243-280

Kolodner, J. L. (1983b). Reconstructive Memory: A Computer Model. Cognitive Science, 7, 281-328

Kolodner, J. L. (ed.)(1988). Proc. of a DARPA Workshop on Case-Based Reasoning. Holliday Inn, Clearwater Beach: Morgan Kaufmann

Kolodner, J. L. (1989). The MEDIATOR: Analysis of an Early Case-Based Problem Solver. Cognitive Science, 13, 507-549

Koton, P. (1988). Reasoning about evidence in causal explanations. Proc. AAAI-88, 256-261

Lopez, B. & Plaza, E. (1991). BOLERO: Case-based Learning of Strategic Knowledge. Proc. EWSL-91, 398-411

Morik, K. (1987). Sloppy Modeling. In: Morik (1989a), 107-134

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

Morik, K. (1990). Integrating Manual and Automatic Knowledge Acquisition - BLIP. In: McGraw & Westphal (eds.). Readings in Knowledge Acquisition - Current Practices and Trends, 213-232, Ellis Horwood

Morik, K. (ed.) (1991). Sonderheft Maschinelles Lernen. KI, Nr. 1, FBO-Verlag, Baden-Baden

Puppe, F. (1990). Problemlösungsmethoden in Expertensystemen. Springer Verlag

Richter, M. M. (1989). Prinzipien der Künstlichen Intelligenz. Teubner Verlag

Richter, M. M. (ed.) (1991). Das MOLTKE Buch. Springer Verlag (in Vorbereitung)

Richter, M. M. & Weß, S. (1991). Similarity, Uncertainty and Case-Based Reasoning in PATDEX. Erscheint in: Festschrift for Woody Bledsoe, Kluwer Academic Publishers

Rumelhart, D. E. & Zipser, D. (1985). Feature discovery by competitive learning. Cognitive Science, 9, 75-112

Schank, R. C. (1982). Dynamic Memory: A Theory Of Learning in Computers and People. Cambridge University Press

Schank, R. C. (1989). Case-Based Explanation. Artificial Intelligence, 40, Special Volume on Machine Learning, 353-385

Schmalhofer, F. (1991). Export/Import im Vergleich zu interdisziplinärer Zusammenarbeit. Proc. des ersten gemeinsamen Workshops der GI-Fachgruppen Knowledge Engineering und Kognition, Kaiserslautern, Springer Verlag

Sharma, S. & Sleeman, D. (1988). REFINER: A Case-Based Differential Diagnosis Aide for Knowledge Acquisition and Knowledge Refinement. Proc. EWSL-88, 201-210

Stanfill, C. & Waltz, D. L. (1986). Towards memory based reasoning. Communications of the ACM, 29, 1213-1229

Strube, G. (1989). Episodisches Wissen. Arbeitspapiere der GMD, 385, 10-26

Strube, G. & Janetzko, D. (1990). Episodisches Wissen und fallbasiertes Schließen: Aufgaben für die Wissensdiagnostik und die Wissenspsychologie. Schweizerische Zeitschrift für Psychologie, 49 (4), 211-221

Thagard, P. & Holyoak, K. J. (1989). Why Indexing is the Wrong Way to Think about Analog Retrieval. In: Hammond (1989), 36-40

Tversky, A. (1977) Features of Similarity. Psychological Review, 84, 327-362

Veloso, M. M. & Carbonell, J. G. (1989). Learning Analogies by Analogy - The Closed Loop of Memory Organization and Problem Solving. In: Hammond (1989), 153-158

Veloso, M. M. & Carbonell, J. G. (1991). Learning by Analogical Replay in PRODIGY: First Results. Proc. EWSL-91, 375-390

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

Waltz, D. L. (1989). Is Indexing Used for Retrieval ? In: Hammond (1989), 41-44

Weß, S. (1991). PATDEX/2: ein System zum adaptiven, fallfokussierenden Lernen in technischen Diagnosesituationen. SEKI Working Paper SWP-91-01, Universität Kaiserslautern

Wolstencroft, J. (1989). Restructuring, Reminding, Repair: What's missing from Models of Analogy. AI Communications, 2, 58-71

Wrobel, S. (1991). Beitrag zur Podiumsdiskussion über "Evaluating and Changing Representation in Concept Acquisition", EWSL-91, Porto