



Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Document
D-94-04

**Entwicklung von Expertensystemen:
Prototypen, Tiefenmodellierung
und kooperative Wissensrevolution**

Franz Schmalhofer, Ludger van Elst

Juni 1994

**Deutsches Forschungszentrum für Künstliche Intelligenz
GmbH**

Postfach 20 80
67608 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
66123 Saarbrücken, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern and Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Atlas Elektronik, Daimler-Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Mannesmann-Kienzle, SEMA Group, and Siemens. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- Intelligent Engineering Systems
- Intelligent User Interfaces
- Computer Linguistics
- Programming Systems
- Deduction and Multiagent Systems
- Document Analysis and Office Automation.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Dr. Dr. D. Ruland
Director

Entwicklung von Expertensystemen: Prototypen, Tiefenmodellierung und kooperative Wissensevolution

Franz Schmalhofer, Ludger van Elst

DFKI-D-94-04

Diese Arbeit wurde finanziell unterstützt durch das Bundesministerium für
Forschung und Technologie (FKZ ITW-8902 C4 and 413-5839-ITW 9304/3).

© Deutsches Forschungszentrum für Künstliche Intelligenz 1994

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Deutsches Forschungszentrum für Künstliche Intelligenz, Kaiserslautern, Federal Republic of Germany; an acknowledgement of the authors and individual contributors to the work; all applicable portions of this copyright notice. Copying, reproducing, or republishing for any other purpose shall require

Inhaltsverzeichnis

1. Einleitung	1
2. Entwicklung von Prototypen.....	2
2.1 Ein typisches System.....	2
2.2Wartung	3
2.3 Die Systemarchitektur	3
3. Tiefenmodellierung durch modellbasiertes Vorgehen.....	4
3.1 Expertensystem-Shells.....	5
3.2 Mining View	5
3.3 Model Building View	6
4.Gemeinsame Wissensnutzung und Wiederverwendung	9
4.1 Lernen mit Alltagswissen: Der CYC-Ansatz	10
4.2 Kooperative Wissensevolution.....	11
4.3Vergleichende Betrachtung	13
5. Kooperative Wissensevolution in der Labormedizin.....	13
5.1 Seeding.....	16
5.2 Evolutionäres Wachstum	17
5.3 Reseeding.....	18
6. Diskussion.....	19
7. Literatur	20

Entwicklung von Expertensystemen: Prototypen, Tiefenmodellierung und kooperative Wissensevolution

Franz Schmalhofer und Ludger van Elst

1. Einleitung

Dieser Beitrag beschreibt drei wichtige Vorgehensweisen bei der Entwicklung von Expertensystemen. Es wird erläutert, wie sich nach dem experimentellen Prototyping der frühen Jahre modellbasierte Entwicklungsmethoden durchgesetzt haben. Als Wissensdarstellungen, die sich durch den menschlichen Betrachter sehr leicht erfassen lassen, spielen dabei Objektmodelle eine wichtige Rolle. Wegen häufig stattfindender Veränderungen in der Anwendungsdomäne müssen sich solche Systeme evolutionär anpassen können. Als dritter Ansatz werden deshalb die Verfahren betrachtet, die im Hinblick auf den sharing & reuse effort entwickelt werden. Diese ermöglichen ein evolutionäres Wachstum von modellbasierten Expertensystemen. Beispielfhaft verdeutlichen wir diesen Ansatz an einigen Konzepten des LAB-IDEAS Systems. In einer abschließenden Diskussion grenzen wir die einzelnen Entwicklungsverfahren hinsichtlich ihrer möglichen Einsatzgebiete, Operabilität und Wartbarkeit voneinander ab.

Nachdem die industrielle Revolution die körperliche Arbeit, wie es sie vorher gab, weitgehend abgeschafft hatte, kam in den 70er Jahren die Vision von der "Abschaffung des Denkens" durch die Künstliche Intelligenz auf. Man ging davon aus, daß die kognitiven Leistungen des Menschen berechenbar sind, und schloß mit der Church'schen These von der Äquivalenz der Berechnungsmodelle, daß sich solche Berechnungen insbesondere auch mit Computern durchführen lassen müßten. Als Teilgebiet der Künstlichen Intelligenz kristallisierte sich die Expertensystemforschung heraus, deren Ziel es war, Computerprogramme zu entwickeln, die das Wissen und die Verarbeitungsstrategien menschlicher Experten widerspiegeln (Schmalhofer & Wetter, 1988). Solche Systeme sollten durch die Externalisierung von Aufgabenstrukturen Probleme lösen können, die bis dahin nur von einem menschlichen Experten bearbeitet werden konnten (z.B. die Erstellung medizinischer Diagnosen). Schon 1977 begründeten Goldstein und Papert eine wissensorientierte Theorie der Intelligenz, indem sie die Frage nach der Repräsentation großer Mengen an Wissen als zentral einstufen und nicht so sehr die Forschung an einigen wenigen mächtigen "Techniken der Intelligenz" für wichtig erachteten. Teilweise wird heute sogar das visionäre Ziel "Entwicklung von intelligenten, autonomen Programmen" zugunsten der Entwicklung eines "Wissensmediums Expertensystem" (Stefik, 1986) modifiziert. Expertensysteme stellen somit weniger die designierten Nachfolger der "Revolution Dampfmaschine" als vielmehr des gedruckten Buches dar.

Im folgenden geben wir einen Überblick über verschiedene Methoden für die Entwicklung von Expertensystemen. Wir stellen zuerst die Architektur der Prototypen vor, die unter der Zielsetzung der 70er Jahre entwickelt wurden. Anschließend gehen wir auf den modellbasierten Ansatz ein, der die Mängel der Prototypen bei der Wartbarkeit des Wissens aufheben sollte, indem explizite Modellierungen des Problemlöseverhaltens und des Wissens die Basis des jeweiligen Systems bilden. Expertensysteme müssen aber oft auch an Innovationen angepaßt werden, die bei der Entwicklung des Systems nicht vorhergesehen werden konnten. Deshalb müssen während des Systemeinsatzes die aktuellen Problemvorstellungen des Benutzers und die ursprünglich erstellten Modellierungen erneut aufeinander abgestimmt werden. Dieser Ansatz der Veränderung und Anpassung eines Wissensmodells an neue Situationsfaktoren und die damit verbundene Wissenserweiterung wird unter dem Stichwort des *sharing & reuse effort* und der *kooperativen Wissensrevolution* dargestellt. Beispielhaft stellen wir die Konzeption des LAB-IDEAS Systems vor, anhand derer besonders auch die Funktion von Expertensystemen als Kommunikationsmedium verdeutlicht wird. In einem abschließenden Abschnitt grenzen wir die verschiedenen Entwicklungsmethoden hinsichtlich ihrer möglichen Einsatzgebiete, Operabilität sowie Wartbarkeit gegeneinander ab, so daß vorhandene Systeme leichter eingeordnet werden können und die Entscheidung für einen der Ansätze im Hinblick auf den praktischen Einsatz oder weitere Forschungsaktivitäten vereinfacht wird.

2. Entwicklung von Prototypen

2.1 Ein typisches System

Zu den bekanntesten experimentellen Prototypen zählen INTERNIST und MYCIN (Shortliffe, 1976). MYCIN wurde für die Diagnose und Behandlung von bakteriellen Infektionskrankheiten eingesetzt. Es handelt sich hierbei um ein regelbasiertes System, was für Expertensysteme der ersten Generation als typisch angesehen werden kann. Mit den vorliegenden Symptomen eines Patienten werden die wahrscheinlichen Diagnosen eingegrenzt. MYCIN stellt Fragen an den Benutzer, bestimmt den Organismus, der die Infektion verursacht haben kann, und wählt ein geeignetes Medikament zur Abwehr aus.

Neben der Tiefensuche mit Rückwärtsverkettung verwendet MYCIN sogenannte *Sicherheitsfaktoren*. Jeder Regel ist ein bestimmter Sicherheitsfaktor zugeordnet, der die Zuverlässigkeit der Regel kennzeichnet. MYCIN kann einem Benutzer die Ableitung einer Diagnose dadurch erklären, daß die einzelnen verwendeten Regelverkettungen und die dazugehörigen Sicherheitsfaktoren mitgeteilt werden.

Den Erfolg von MYCIN stellt am besten die Aussage eines Experten dar, der die Qualität von MYCIN auf folgende Weise beurteilte: "The system behaves as I do on my best days". Dieses Urteil wurde auch in einer kontrollierten Untersuchung bestätigt. In einem Doppelblindversuch verglichen acht unabhängige Experten die

Infektion effektiv sein würde, und zweitens wurde evaluiert, ob das verschriebene Medikament auch für andere Infektionen, die aufgrund der zuerst erhobenen Laborergebnisse möglicherweise hätten vorliegen können, wirksam wäre. Im Hinblick auf das erste Kriterium waren die Verordnungen von MYCIN und den neun Fachexperten gleich wirksam. Es zeigten sich weder Unterschiede zwischen den einzelnen Experten noch schnitten diese besser ab als MYCIN. Bei der zweiten Beurteilung ergab sich, daß MYCIN in 65% der Fälle ein erfolgreiches Medikament verschrieben hatte, während die menschlichen Experten nur in 42.5% bis 62.5% der Fälle effektive Medikamente verordneten.

Für diesen Erfolg von MYCIN lassen sich im wesentlichen drei Gründe angeben:

- Die Wissensbasis von MYCIN wurde von den besten Fachexperten erstellt und ist extrem detailliert. Speziell für Meningitis ist das in MYCIN enthaltene Wissen so umfassend wie bei keinem einzigen menschlichen Experten.
- MYCIN hat ein perfektes Gedächtnis, vergißt nichts und übersieht nichts. Diesbezüglich ist es dem menschlichen Experten überlegen.
- MYCIN trifft keine voreiligen Schlußfolgerungen. Selbst in Fällen, die den Experten offensichtlich und trivial erscheinen, geht es systematisch vor.

2.2 Wartung

Bei der Wartung solcher regelbasierter Expertensysteme...

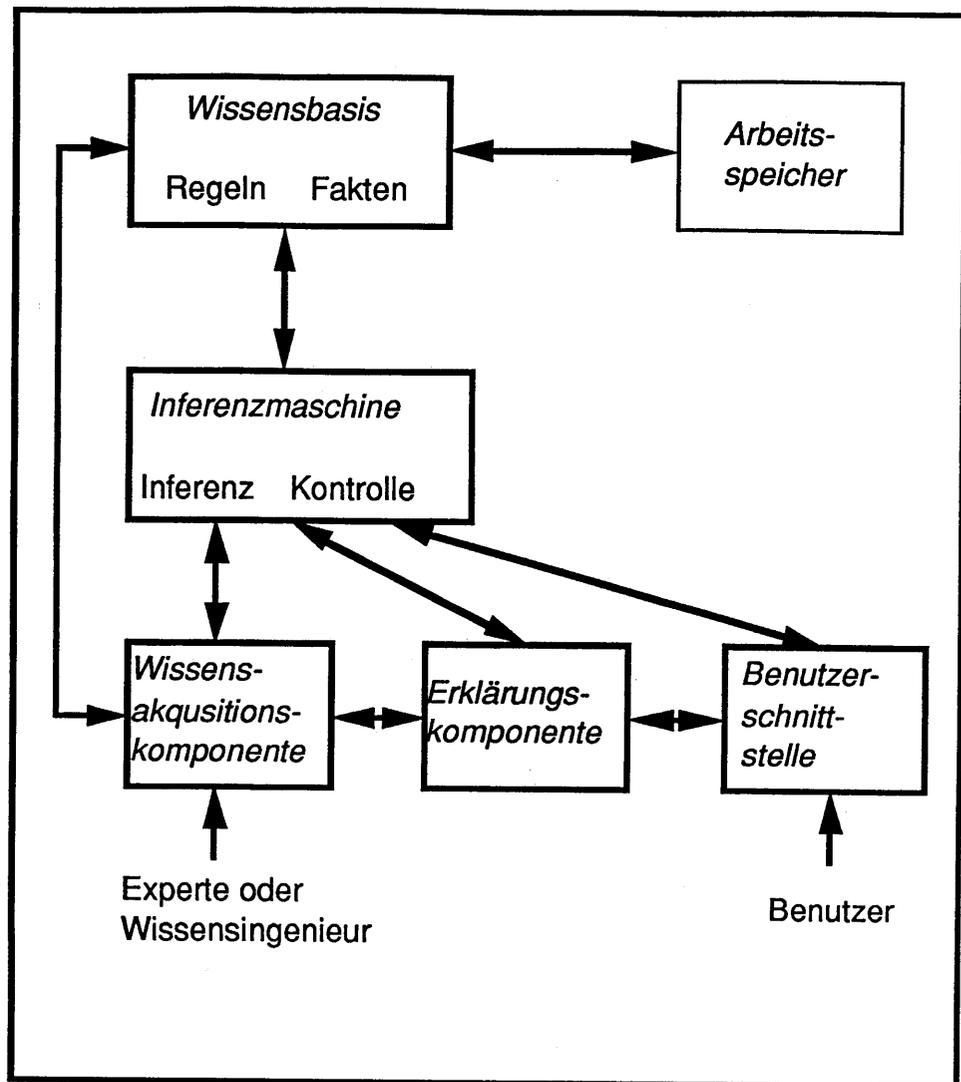


Abbildung 1: Architektur der Expertensysteme der ersten Generation

3. Tiefenmodellierung durch modellbasiertes Vorgehen

Mit diesen ersten funktionsfähigen Prototypen wurde somit gezeigt, daß es möglich ist, Systeme zu entwickeln, die menschliche Expertise in Teilen simulieren können. Die dabei benutzten Techniken versuchte man dann, auch anderen Domänen zugänglich zu machen, indem man die Inferenzmaschinen von den Regeln abkoppelte und in Expertensystem-Shells einbaute. Diesen Ansatz stellen wir in Abschnitt 3.1 vor. Desweiteren analysierte man die ersten prototypischen Expertensysteme und stellte fest, daß dort eigentlich implizit Modelle menschlichen Expertenverhaltens benutzt werden. In der folgenden Forschung wurde die Modellbildung stärker betont, indem man dazu überging, die Modelle explizit zu formulieren (vgl. Abschnitt 3.3). Clancey (1991) zeigte mit der Einführung des *model building operators* im wesentlichen die Äquivalenz zwischen Shells und modellbasierten Ansätzen. Man kommt also mit den beiden unterschiedlichen Betrachtungsweisen zu einem ähnlichen Resultat.

3.1 Expertensystem-Shells

Indem man die Regelbasis aus den prototypischen Systemen eliminierte, entstanden Expertensystem-Shells, von denen man vermutete, daß sie domänenunabhängig und dadurch allgemein einsetzbar sein würden. So wurde aus MYCIN die Expertensystem-Shell E-MYCIN, was Essential-MYCIN oder auch Empty-MYCIN bedeuten sollte. Aus der Bezeichnung *Essential-MYCIN* kann man noch erkennen, daß die Entwicklung des Inferenzmechanismus als die Hauptleistung beim Expertensystembau angesehen wurde. Es bestand damals die Vermutung, daß sich mit solchen Expertensystem-Shells sehr leicht vollständige Expertensysteme erstellen ließen, indem ein Domänenexperte sein Wissen in Regelform mitteilt. Das Erfassen der Expertenregeln bezeichnete man als *Wissensakquisition*. Die gewonnenen Regeln wurden dann beispielsweise über einen Editor (*Wissenseditor*) in das System eingegeben.

Für MYCIN wurde beispielsweise das Akquisitionssystem Teiresias entwickelt. Dieses geht davon aus, daß Wissen aus Regeln besteht, die im System dargestellt werden müssen. In Teiresias sind Regelmodelle und Vorschriften über die Struktur von Regeln repräsentiert. Dadurch kann es syntaktisch falsche Regeln entdecken und dem Benutzer mitteilen, wie eine Regel korrekt eingegeben werden sollte.

Die Wissenseingabe in Expertensystem-Shells erwies sich jedoch als sehr viel schwieriger als man ursprünglich dachte. So ist kein einziger Fall bekannt, in dem die Inferenzmaschinen aus Expertensystemen der ersten Generation erfolgreich auf eine andere komplexe Anwendungsdomäne übertragen wurden. Die Inferenzmaschine war somit nicht wiederverwendbar. Daraufhin wurde von Feigenbaum zusammenfassend festgestellt, daß die Wissensakquisition der "Flaschenhals der Expertensystementwicklung" sei. Diese Feststellung ist die Geburtsstunde der Wissensakquisition als zentraler Forschungs- und Entwicklungsbereich innerhalb des

Expertenwissen, welches das Problemlöseverhalten bestimmt, im Experten "tief verborgen" ist¹.

3.3 Model Building View

Eine andere Auffassung sieht die eigentliche Leistung bei der Entwicklung eines Expertensystems in der Erstellung eines adäquaten Modells des zu erstellenden Systems. Wissensakquisition ist demnach eine Modellierungstätigkeit und besteht nicht in erster Linie aus der Erhebung von detailliertem Wissen. Nach dieser Ansicht sollen Expertensysteme modellbasiert entwickelt werden².

Da die Expertensysteme der ersten Generation nicht modellbasiert entwickelt wurden, mußte man nun nachträglich analysieren, durch welche Modelle sich die schon vorhandenen Systeme beschreiben lassen. Bei der Analyse von MYCIN wurde das Modell der *heuristischen Klassifikation* entwickelt (Clancey, 1985), das in Abbildung 2 dargestellt ist.

Nach diesem Schema werden zuerst Abstraktionen durchgeführt. Dann erfolgt eine heuristische Assoziation auf eine Hierarchie abzählbarer Lösungen und eine Verfeinerung innerhalb dieser Hierarchie. Charakteristisch für die heuristische Klassifikation ist, daß die Lösungen aus einer abzählbaren Menge von bekannten Lösungsmöglichkeiten ausgewählt werden. Abbildung 3 zeigt, wie ein Ausschnitt von MYCIN in das Schema der heuristischen Klassifikation eingeordnet wurde.

Es gibt drei verschiedene Arten von Abstraktionen. Bei der *qualitativen Abstraktion* werden aus quantitativen Daten qualitative Aussagen generiert. Bei der *definitorisches Abstraktion* wird eine terminologische Umformulierung vorgenommen. In einem letzten Abstraktionsschritt wird dann *generalisiert*. Es schließt sich ein *heuristischer Match* an, bei dem aus den erzeugten Abstraktionen Verknüpfungen zu einer Menge von Lösungen oder Lösungsabstraktionen erzeugt werden. Bei MYCIN waren diese Assoziationen aus typischen Fällen abgeleitet.

¹Zur Unterstützung dieser Auffassung können einige psychologische Befunde angeführt werden. So haben Nisbett und Wilson gezeigt, daß nachträgliche Beschreibungen von Personen teilweise aus Rationalisierungen bestehen können, die mit dem eigentlichen Verhalten der Person nichts zu tun haben. Auch Ergebnisse aus Untersuchungen des impliziten Gedächtnisses des Menschen können zur Unterstützung dieser Ansicht genutzt werden. So zeigte sich, daß die Handlungen von Personen oft durch vorausgegangene Ereignisse beeinflusst werden, ohne daß sich die Person selbst explizit an diese Ereignisse erinnern kann. Es läßt sich also vermuten, daß manches Wissen im Menschen in kompilierter Form vorliegt. Er kann es daher nicht verbalisieren. Trotzdem ist dieses kompilierte Wissen verhaltenswirksam, indem es die Aktionen und Problemlösungen des Experten bestimmt. Man benötigt also besondere "Fördermechanismen", die das verborgene Wissen hervorholen, um es explizit in Expertensystemen repräsentieren und nutzen zu können. Eine solche Methode ist die des lauten Denkens, das *Repertory-Grid-Verfahren*, das sich insofern besonders gut eignet, da die Struktur des mit ihr erhobenen Wissens dem Repräsentationsmechanismus von Shells wie NEOMYCIN entspricht und direkt übernommen werden kann. Boose und Gaines (1989) berichten, daß sich mit diesem Verfahren für einfache Anwendungsbereiche ein Expertensystem in zwei Stunden entwickeln lasse.

²Mangold-Allwinn, Antoni, König & Eisenecker (1993) nutzen diese modellbasierte Sichtweise auch im Hinblick auf die Erstellung traditioneller Software. Die dort vorgestellten Verfahren zur Erhebung des Wissens können natürlich auch beim Entwurf von Expertensystemen genutzt werden.

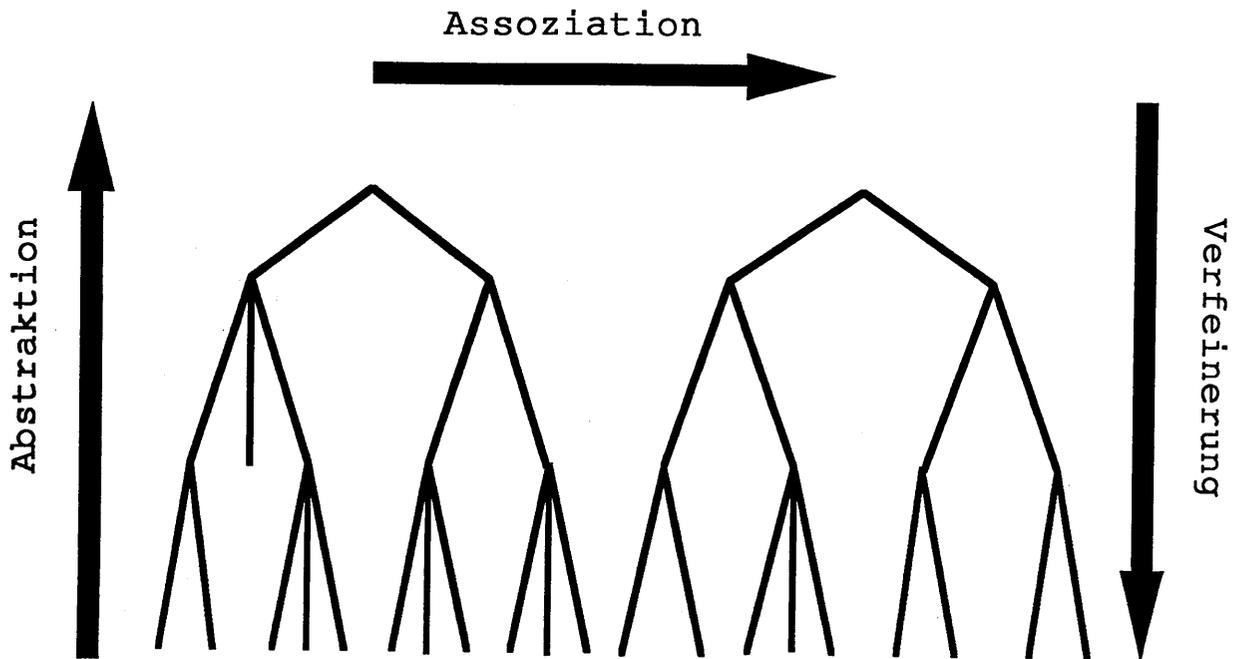


Abbildung 2: Schema der heuristischen Klassifikation (nach Karbach und Linster, 1990)

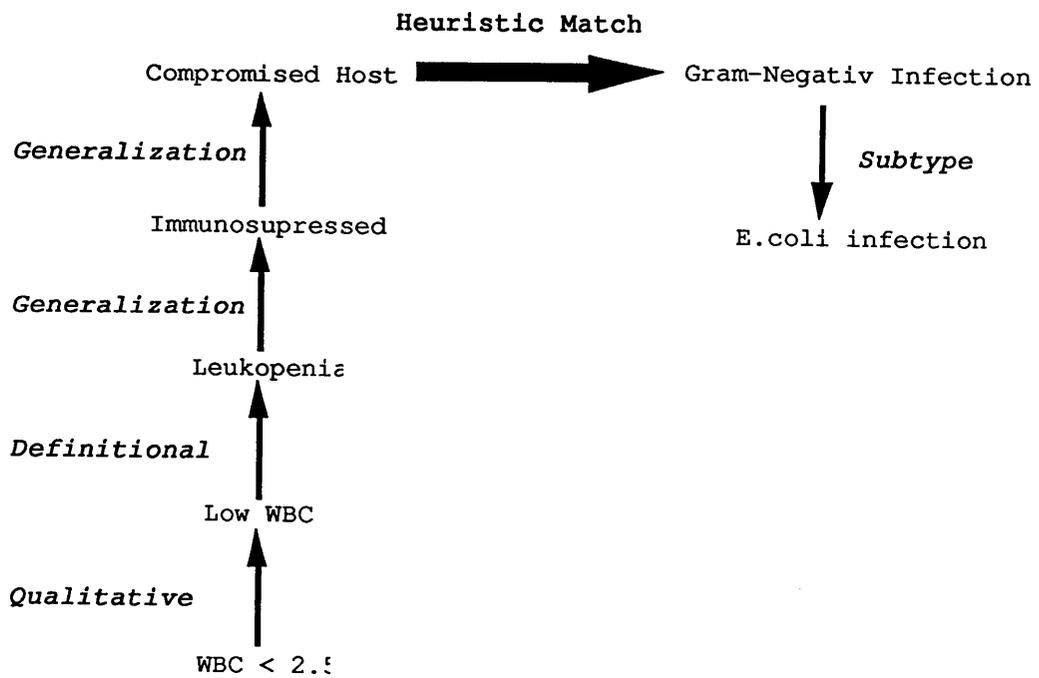


Abbildung 3: Die Inferenzstruktur von MYCIN als heuristische Klassifikation (nach Clancey, 1985)

Eines der wichtigsten modellbasierten Verfahren zur Entwicklung von Expertensystemen ist die KADS-Methodologie (Breuker & Wielinga, 1989). In dieser wird das Tiefenwissen in Form expliziter Domänen- und Objektmodelle und Inferenzstrukturen auf diesen Modellen (Steels, 1990) betont. Abbildung 4 gibt einen Überblick über dieses Verfahren. Nach KADS kann die Expertensystementwicklung in mehrere Phasen unterteilt werden. In der ersten Phase wird reales oder gewünschtes Expertenverhalten analysiert. Daraus wird ein konzeptuelles Modell (M1) der Expertise in der Anwendungsdomäne entwickelt. Aus dem konzeptuellen Modell wird

jedoch insbesondere darauf geachtet werden, daß Modell und erhobenes Detailwissen aufeinander abgestimmt sind.

Im Gegensatz zu den flachen Wissensbasen der Expertensysteme erster Generation enthalten die Systeme der zweiten Generation, die als *wissensbasiert* bezeichnet werden, unterschiedliche Wissensarten, die durch verschiedene Repräsentationsformalismen dargestellt und multiple Inferenztechniken bearbeitet werden. Es sollten außerdem Erklärungen auf verschiedenen Abstraktionsebenen

schaulicht. Nach dem CYC-Ansatz sollte ein System, ähnlich wie bei einem Auto, das zuerst angeschoben werden muß, nach einer umfangreichen Akquisition von Alltagswissen durch maschinelle Lernverfahren vollständig selbsttätig lernen können. Dem gegenüber geht die kooperative Wissensevolution davon aus, daß maschinelles Lernen, ähnlich einem Fahrrad, immer auch der Mitwirkung eines menschlichen Benutzers bedarf.

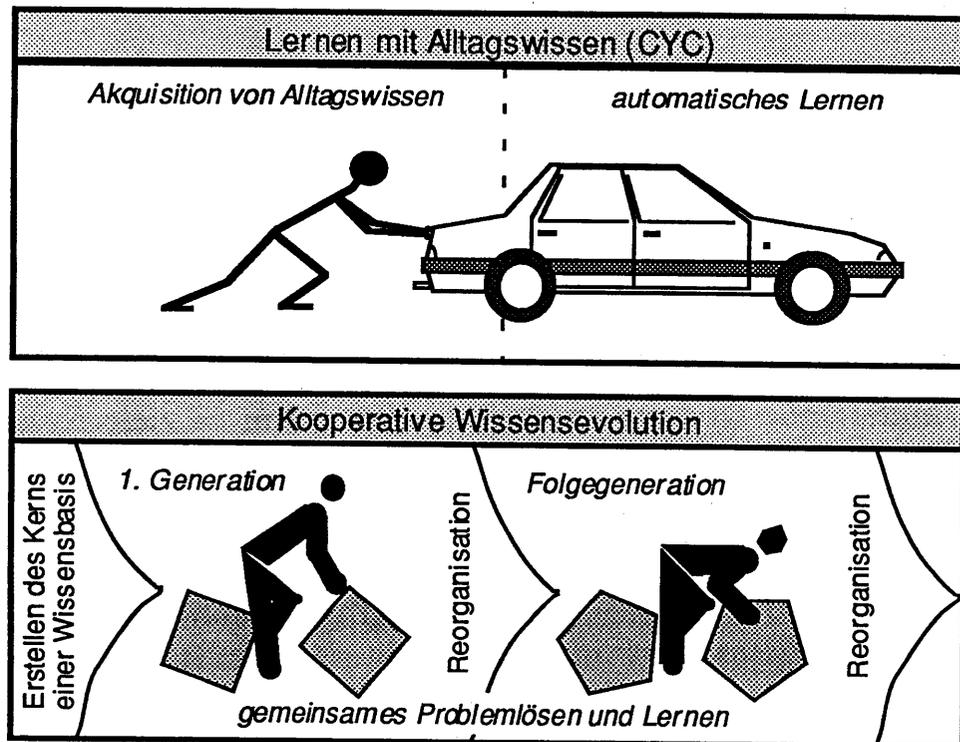


Abbildung 5: Vergleich von zwei verschiedenen Lernansätzen anhand einer Metapher

4.1 Lernen mit Alltagswissen: Der CYC-Ansatz

Dieser hauptsächlich in den USA verfolgte Ansatz fordert, daß zum automatischen Lernen in komplexen Bereichen umfangreiches Alltagswissen eingesetzt werden muß. Deshalb wird zuerst eine Wissensbasis mit Alltagswissen von Hand, d.h. mit den gewöhnlichen Methoden des Knowledge Engineering, erstellt. Danach soll sich die so aufgebaute Wissensbasis durch maschinelles Lernen automatisch erweitern.

Dieser Ansatz wird in dem auf zwanzig Jahre angelegten CYC-Projekt verfolgt (Lenat, Prakash & Shepard, 1986). Von 1984 bis 1994 wird dabei eine Wissensbasis aufgebaut, die neben Fakten, Algorithmen, Heuristiken, Geschichten, Repräsentationen usw. auch die Methoden zum effizienten Ableiten neuer Erkenntnisse enthält. Im Jahre 1989 enthielt die Wissensbasis bereits fünfhunderttausend Einträge; bis Mitte 1994

sollen es ca. 100 Millionen sein. Danach soll das CYC-System dann selbsttätig lernen (Lenat, 1989).

4.2 Kooperative Wissensrevolution

Der Ansatz der kooperativen Wissensrevolution bezieht dagegen den menschlichen Benutzer in den Lernprozeß mit ein (Lajoie & Lesgold, 1989; Wilkins, 1991).

Bei der *kooperativen Wissensrevolution* betrachtet man Computersystem und Benutzer als Partner bei der Problemlösung und beim Lernen (Fischer et al., 1994). Dieser Ansatz baut damit auf dem Lehrlingslernen auf, betont jedoch darüber hinaus, daß sowohl das System als auch der Benutzer in einer gemeinsamen Lernumgebung Probleme lösen und lernen. Dabei kommen vorzugsweise wissensintensive Lernverfahren zum Einsatz (Reinartz & Schmalhofer, 1994). Problemlösende Systeme werden also als Ansammlung von Akteuren gesehen, von denen einige Menschen sind, andere wiederum Software sein können. Es wird erkannt, daß Mensch und Programm komplementäre Stärken und Schwächen haben. Diese Asymmetrie versucht man auszunutzen. Der Benutzer bringt seinen "gesunden Menschenverstand" ein, seine Fähigkeiten zur Zieldefinition, Problemzerlegung und ähnliches. Der Computer stellt dagegen den "Externspeicher" für den Benutzer dar, gewährleistet die Konsistenz von Wissen und Lösungen, faßt Informationen zusammen, verbirgt irrelevante und visualisiert wichtige Information, etc. Man könnte solche kooperativen Problemlösungssysteme somit als "kognitive Verstärkung" des Menschen betrachten. Ein zentrales Anliegen dieses Ansatzes ist die Verbesserung der Koordination zwischen den beiden Partnern.

In der Wissensakquisitionsphase übernimmt ein Fachmann oder Experte der Anwendungsdomäne (z.B. ein Arzt für eine medizinische Wissensbasis) die Rolle eines Autors, der sein Wissen mit Hilfe des Expertensystems dokumentiert und verschiedene maschinelle Lernverfahren übernehmen die Rolle eines Lektors (engl.: editorial assistant), der das Wissen in die richtige Form bringt und auf bestimmte Konventionen beim Ablegen des Wissens achtet.

Fischer und Reeves (1992) schlagen eine Architektur zur Speicherung von Domänenwissen und Unterstützung von kooperativer Problemlösung vor, die aus fünf Komponenten besteht. Der zentrale Bestandteil ist ein *Konstruktions-* oder *Analysewerkzeug*, das die eigentliche Arbeitsfläche für die Tätigkeit des Benutzers darstellt. In einer *Spezifikationskomponente* werden die High-level-Anforderungen der Aufgabe und ihre Wichtigkeit repräsentiert. Ein argumentatives *Hypermedia-System* verwaltet das Wissen über die Domäne sowie Ergebnisse, Antworten und Argumente des Systems zusammen mit Beschreibungen, die es dem Benutzer ermöglichen, die Systemvorschläge zu verstehen. Die Verwaltung dieses Wissens geschieht in einem Hypertext, der die Definition von Begriffen, die Erzeugung von Kreuzverweisen sowie von Zeigern auf Information in anderen Systemkomponenten erlaubt. Der vierte Bestandteil dieser Architektur ist der *Katalog*. Hier werden die Objekte des Designs oder der Analyse in Form von Teil- oder Gesamtlösungen gesammelt. Sogenannte *constructive idioms* dienen dann als Startpunkte, Beispiele oder Kreativitätsanstöße für

die Problemlösung. Eine *Simulationskomponente* stellt das Komplement zur Argumentationskomponente dar. Mit ihrer Hilfe kann der Benutzer den bisherigen Stand der Problemlösung in verschiedenen Situationen ausprobieren und dadurch einschätzen, ob diese Lösung bereits angemessen ist bzw. wo sich Schwachpunkte zeigen.

Die Integration der genannten Komponenten geschieht in dieser Architektur durch drei Hilfsmittel. Der *Konstruktionsanalyser* analysiert und kritisiert den Inhalt der aktuellen Problemlösung, indem es dem Benutzer Hinweise gibt sowie Einträge in die Argumentationskomponente macht. Ein *Argumentationsillustrator* hilft dem Benutzer, die Prinzipien im argumentativen Hypertext zu verstehen, indem Beispieldesigns oder -analysen dargestellt werden. Der *Katalogexplorer* stellt einen Browser durch den Katalog dar. Er findet ähnliche Beispiele und ordnet diese im Hinblick auf die Angemessenheit für die aktuelle Situation.

Die Wartungsproblematik der Wissensbasis wird bei der kooperativen Wissens evolution durch ein life-cycle-Modell dargestellt, dem ein Drei-Phasen-Modell (vgl. Fischer et al. 1994) zugrunde liegt, das auch im unteren Teil von Abbildung 5 repräsentiert wird. In der *Seeding-Phase* wird zunächst der Wissensbasiskern eines Expertensystems von einem Experten und einem Wissensingenieur gemeinsam erstellt. Zur Vermeidung des Kommunikationsproblems werden neben dem formalen Wissen dabei auch natürlichsprachliche Informationen in einem mit dem formalen Wissen verknüpften Hypertextsystem abgelegt (Schmalhofer, Reinartz & Tschaitshian, 1992); formales und nicht-formales Wissen sind also nicht künstlich voneinander getrennt, sondern ergänzen sich gegenseitig. Die *Phase des evolutionären Wachstums* ist die der Anwendung durch den Benutzer, wo durch die Arbeit des Problemlösens neues Wissen im System angesammelt wird. Dieses Wissen kann unter anderem auch in nicht-formalisierter Form vorliegen. Durch Verfahren des Lehrlingslernens u.ä. findet aber auch eine automatische Erweiterung der Wissensbasis statt. Nach längerer Arbeit mit diesem System der ersten Generation ist die Repräsentation des Wissens oder auch das Problemlösemodell jedoch nicht mehr angemessen, und in einer Phase des *reseeding* wird mittels eines Knowledge Engineers das gewonnene Wissen formalisiert und das Modell gegebenenfalls angepaßt, um neue Anforderungen zu integrieren. Es wird also in periodischen Abständen eine Wartung durch einen Wissensingenieur durchgeführt, wobei eine Reorganisation und Erneuerung der Wissensbasis stattfindet. Da die darauf folgende Wissensnutzung im System zweiter Generation auf den vorausgegangenen Erfahrungen aufbaut, sind Benutzer und System in den Folgegenerationen nun besser aufeinander abgestimmt. Dadurch wird das Anwender-System-Tandem zunehmend effizienter. An diesem Wechselspiel von Wissensnutzung, -wachstum und Reorganisation wird ein wesentlicher Vorteil der kooperativen Wissens evolution deutlich, die Möglichkeit, sich an veränderte Umweltbedingungen anzupassen (vgl. Clancey, 1991; Fischer et al., 1994). Dieser Ansatz wurde auch bei der Entwicklung des IDEAS-Systems verfolgt (Aschenbrenner et al., 1993; Birk et al., 1994).

Der Computer stellt somit interaktive Werkzeuge zur Verfügung, die Speicher- und Retrieval- sowie Gruppenkommunikationsdienstleistungen erbringen. Weiterhin gibt es eine Sprache und Tools, die Präzision und Explizitheit bei der Manipulation von Wissen ermöglichen. Stefik (1986) schätzt die Möglichkeiten eines solchen Systems folgendermaßen ein: "A knowledge medium based on AI technology is part of

a continuum. Books and other passive media can simply store knowledge. At the other end of the spectrum are expert systems which can store and also apply knowledge. In between are a number of hybrid systems in which the knowledge processing is done mostly by people. There are many opportunities for establishing human-machine partnerships and for automating tasks incrementally."

4.3 Vergleichende Betrachtung

Mit dem CYC-Ansatz wie auch mit dem Ansatz des kooperativen Lernens wird das Ziel verfolgt, in der Praxis einsetzbare Expertensysteme für komplexe Anwendungen zu entwickeln und diese Entwicklung soweit wie möglich durch Verfahren des maschinellen Lernens zu unterstützen. Die beiden Ansätze unterscheiden sich hauptsächlich darin, wie die Arbeit zwischen den maschinellen Lerntechniken und dem Menschen aufgeteilt ist.

Der CYC-Ansatz geht von der Annahme aus, daß den bisher entwickelten maschinellen Lernverfahren für einen erfolgreichen Einsatz in komplexen Bereichen nur ein ausreichend großer Grundstock an Alltagswissen fehlt. Indem man zuerst umfangreiches Alltagswissen formalisiert, können die maschinellen Lerntechniken damit auch in komplexen Domänen selbsttätig lernen.

Der Ansatz der kooperativen Wissensrevolution beruht auf der Erfahrung, daß in komplexen Bereichen maschinelles Lernen ohne menschliche Unterstützung weder

Bereich der Labormedizin) gekennzeichnet, die einer optimalen Diagnosequalität entgegenstehen. Um diese Asymmetrien im Sinne einer umfassenden Wissensnutzung auszugleichen, ist es nötig, den Informationsfluß zwischen den beiden Partnern zu verbessern. Da passive Medien jedoch vielfältige Nachteile wie zum Beispiel Diskontinuität des Datenflusses (Datenbanksysteme), mangelnde Flexibilität (Brief) oder hoher personeller Aufwand (Telefon) haben, sind diese nur unzureichend geeignet. Stattdessen bieten sich wissensbasierte Systeme als Medium zur Erhöhung der Bandbreite der Kommunikation zwischen niedergelassenem Arzt und Labormediziner an. Diese können bei der Problemlösung die Eigenschaft von Akteuren annehmen und ergänzen somit das Kommunikationsnetzwerk um ein weiteres handelndes Element, um die Nutzung der vorhandenen Information im Hinblick auf die Ziele, Qualität und Effizienz der Diagnose, zu verbessern. Wissensbasierte Systeme können im Gegensatz zu passiven Medien eine logische Pufferung leisten, da Wissen nicht nur einfach gespeichert, sondern auch für weiteren Gebrauch vorbereitet und strukturiert wird.

In Abbildung 6 sind die Kommunikationswege zwischen niedergelassenem Arzt und Labormediziner dargestellt. Im Gegensatz zur traditionellen Struktur von Expertensystemen, die zwischen System, Experte und Benutzer unterscheidet, läßt sich diese Einteilung hier nicht mehr aufrecht erhalten. Vielmehr gibt es hier zwei Benutzer: den Labormediziner und den niedergelassenen Arzt. Beide sind gleichzeitig Experten und bringen spezielles, für die optimale Problemlösung notwendiges Wissen ein. Zwischen diesen beiden *Experten-Nutzern* befindet sich das wissensbasierte System als aktives Kommunikationsmedium.

Zum einen existieren in diesem Modell die herkömmlichen Kommunikationslinien zwischen Labormediziner und niedergelassenem Arzt auf dem Wege der üblichen passiven Medien wie Telefon und Brief (traditionell oder rechnergestützt via Email bzw. Mailbox). Weiterhin stehen beiden die Daten des konkreten Falles in Form von Laboranalysen auf der einen sowie der Patientenkartei auf der anderen Seite zur Verfügung. Die Überbrückung der räumlichen Distanz bei der Datenerhebung kann zum Beispiel durch Datenfernübertragung erfolgen. Außerdem können sowohl Labor- als auch niedergelassener Arzt Fachbücher, Zeitschriften u.ä. Informationsquellen nutzen. Unabhängig von der technischen Speicherform, gedrucktes Buch oder CD-ROM, Tagungsbericht oder Mailbox-Eintrag, handelt es sich auch hier um passive Medien. Dieser Teil der Kommunikation findet auf dem Hintergrund gemeinsamer Vorstellungen über das zu lösende Problem (Diagnose) sowie medizinischen Basiswissens statt und stellt bis hierher den herkömmlichen Teil der Kommunikation dar.

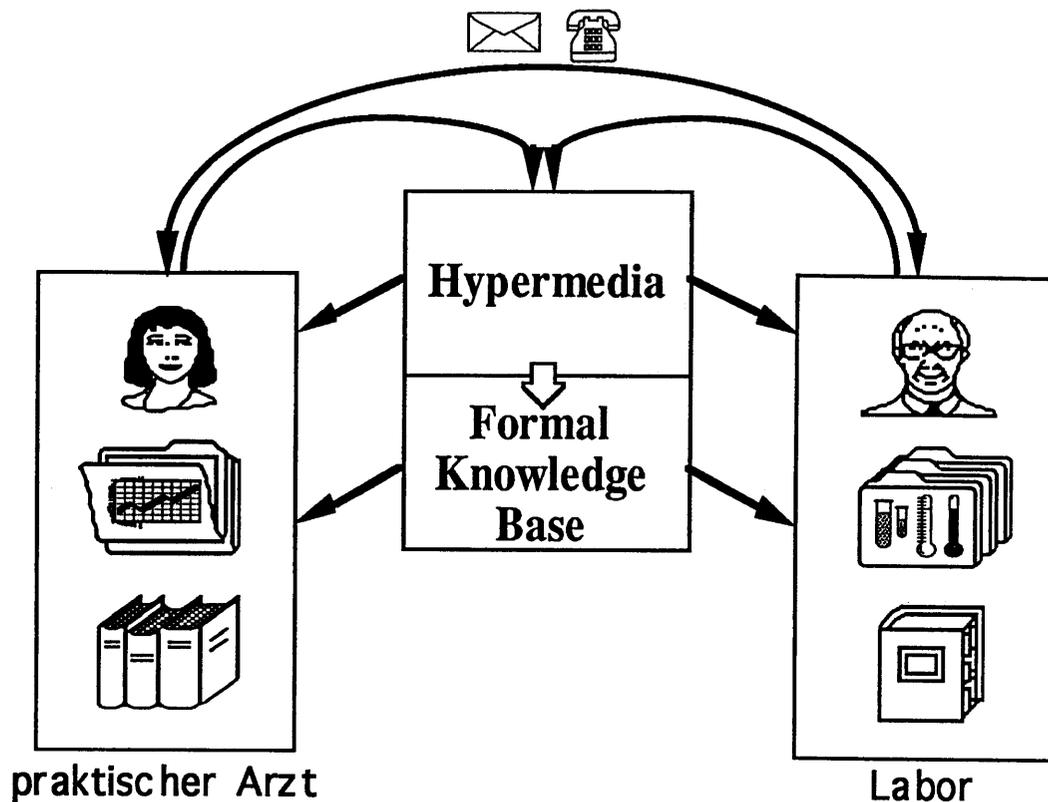


Abbildung 6: Kommunikationswege zwischen niedergelassenem Arzt und Labormediziner

Diese Struktur wird in unserem Modell um das aktive Medium *wissensbasiertes System* ergänzt. Auch das wissensbasierte System hat Zugriff auf die Falldaten und Fachinformationen aus Büchern u.ä.. Gleichzeitig arbeiten jedoch auch Labormediziner und niedergelassener Arzt bei der Lösung der Aufgabenstellung mit dem System. Dadurch fließen Informationen von beiden Benutzern in das System und umgekehrt auch vom System zu den Benutzern. Faßt man diese beiden Schritte zusammen, so ergibt sich ein Kommunikationsweg zwischen Labor und niedergelassenem Arzt, der jetzt allerdings nicht mehr direkt sondern über das wissensbasierte System verläuft. Die Kommunikation zwischen den beiden Benutzern ist also *entkoppelt*. Diese Entkopplung kann man nutzen, um das wissensbasierte System als Akteur, als handelnde Komponente, einzusetzen. Dabei findet die Entkopplung sowohl zeitlich als auch logisch statt. Zum einen wird also eine Pufferung in der eigentlichen Datenübermittlung erreicht, wie sie z.B. auch von Email, Anrufbeantworter oder Fax geleistet wird. Zum anderen jedoch gibt es eine inhaltliche Pufferung, da das System Antworten des Kommunikationspartners bereithält, auch wenn dieser zum aktuellen Zeitpunkt selber gerade nicht erreichbar ist und die Antworten nicht explizit bereitgestellt, sondern seine Arbeit lediglich mit dem System dokumentiert hat. Eine solche logische Entkopplung können die üblichen passiven Medien aufgrund ihrer mangelnden Flexibilität, insbesondere wegen ihres sequentiellen Charakters, nicht leisten.

Kooperative wissensbasierte Systeme können also Fragen von Experten beantworten, indem sie das von anderen Experten zur Verfügung gestellte Wissen anwenden.

In Abschnitt 4.2 haben wir den Lebenszyklus eines auf dem Prinzip der koope-

Nach dieser Phase der Wissensakquisition, die in Abbildung 7 noch einmal zusammenfassend illustriert ist, kann nun ein erstes einsatzfähiges wissensbasiertes System implementiert werden und die erste Nutzungsperiode schließt sich an.

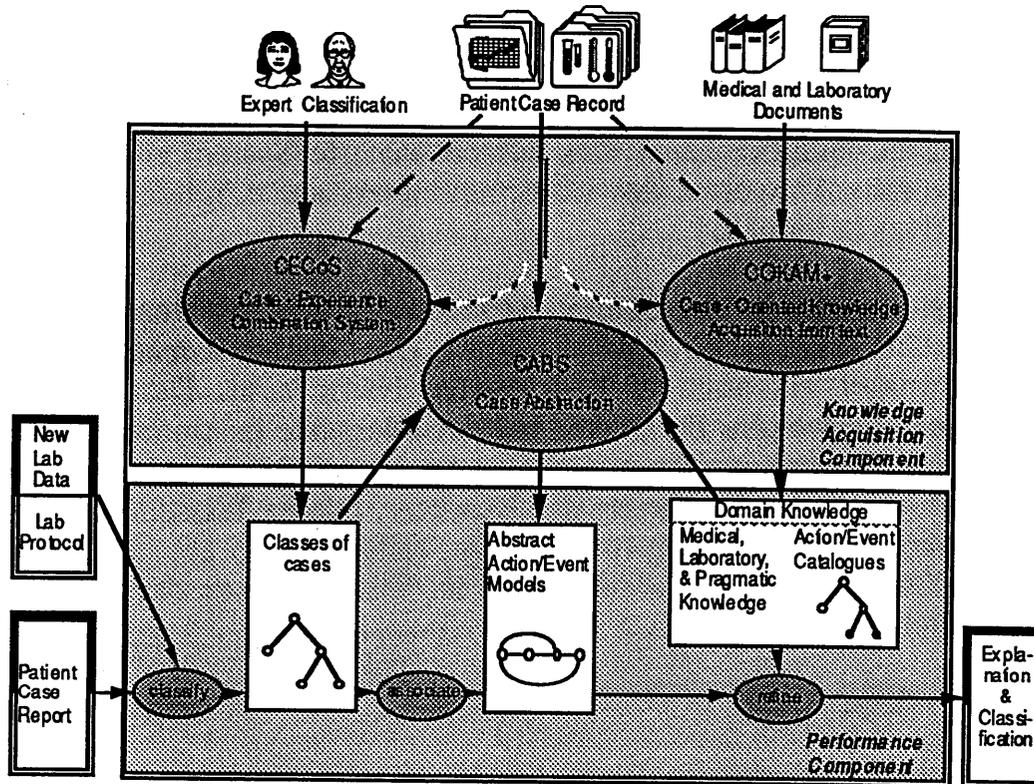


Abbildung 7: Integrierte Wissensakquisition (adaptiert nach Schmalhofer & Thoben, 1992)

5.2 Evolutionäres Wachstum

Das in der Konstruktionsphase erstellte initiale System kann...

System und mit dem System fällt, wird diese gleichzeitig dokumentiert und dadurch nachvollziehbar.

Diese Dokumentation der Entscheidung des Labors dient dann dem niedergelassenen Arzt als Basis für seine Arbeit mit dem System. Er kann die Argumentationsstruktur nachvollziehen und dann aus seiner Sicht überprüfen, ergänzen oder korrigieren, um anschließend zu seiner Diagnose zu kommen. Dem Arzt steht somit ein entscheidungsunterstützendes Beratungssystem zur Verfügung. Dabei findet eigentlich Kommunikation zwischen dem niedergelassenen Arzt und dem Labor statt, die lediglich durch das wissensbasierte System entkoppelt wurde, indem dieses im voraus Antworten des Labormediziners auf mögliche Fragen des niedergelassenen Arztes bereithält.

Durch die Arbeit der beiden Benutzergruppen mit dem System wird diesem

6. Diskussion

Auch wenn bei den oben dargestellten Entwurfstechniken für wissensbasierte Systeme (Prototypenansatz, modellbasierter Ansatz und kooperative Wissensrevolution) eine Entwicklung festzustellen ist, bei der jeder neue Ansatz versuchte, die Unzulänglichkeiten seines Vorgängers zu überwinden, so hat doch jedes Verfahren seine spezifischen Vorteile auch gegenüber neueren Ansätzen. Wenn über den Einsatz von Expertensystemen nachgedacht wird, lohnt es sich daher immer, alle bekannten Techniken in Betracht zu ziehen und die für den konkreten Verwendungsbereich optimale auszuwählen. Um dies zu erleichtern, stellen wir in diesem Abschnitt die Stärken und Schwächen der drei Entwurfsverfahren gegenüber. Dazu diskutieren wir sie insbesondere im Hinblick auf ihre Operabilität, Wartbarkeit sowie mögliche Einsatzgebiete.

Der Prototypenansatz mit seinem einfachen Aufbau aus Regeln und Inferenzmaschine bietet den Vorteil, daß ein solcher Prototyp, ist er einmal fertiggestellt, sehr operabel ist. Er hat ein relatives starres Ein-/ Ausgabeverhalten, was die Arbeit mit ihm leicht macht. Als Eingabe bekommt er Fakten, die die aktuelle Problemsituation beschreiben, und die Ausgabe des Systems soll dann die Lösung der entsprechenden Design- oder Analyseaufgabe sein. In Domänen, deren Wissen gut formalisierbar ist und bei denen es einige wenige klare Fragestellungen gibt, kann man mit diesem Ansatz durchaus Systeme erhalten, die sehr gute Ergebnisse liefern. Im wesentlichen sind dies technische Anwendungsgebiete, die ihrer Natur nach schon stark

sondere dann sinnvoll, wenn ein Teil des benötigten Expertenwissens nur schwer oder überhaupt nicht formalisierbar ist.

So ist letztlich das Hauptargument für den Ansatz der kooperativen Wissensrevolution ein pragmatisches: Man kann und sollte bei der Lösung komplexer Probleme ebenso wenig auf die starken (meta-)kognitiven Fähigkeiten von menschlichen Experten verzichten wie auf die Fähigkeiten von Rechnern zur Speicherung, Darstellung und Bearbeitung insbesondere formalisierten Wissens. Es wird jedoch auch deutlich, daß die Entwicklung der kooperativen Techniken noch nicht abgeschlossen ist. Sowohl bezüglich der Adaption der Methoden, die bei den Prototypen und modellbasierten Ansätzen entwickelt wurden, als auch hinsichtlich der

Darstellung und Bearbeitung von unterschiedlich stark formalisiertem Wissen, also verschiedener Wissens Ebenen, sollte auch noch weitere Forschungsarbeit geleistet werden.

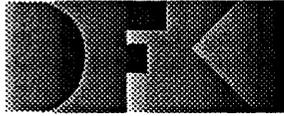
7. Literatur

- Albers, S., & Petzold, H. J. (1993). Modellbildung und Modellvalidierung im Rahmen eines partizipativen Entwicklungsansatzes betrieblicher Anwendungssysteme. Proceedings des Workshops Psychologie des Software-Entwurfs, 14.-15. Mai, Bad Neuenahr, Deutschland.
- Aschenbrenner, K.M., Gertzen, H., Kühn, O., Schmalhofer, F., & Schmidt, J. (1993). IDEAS: Unterstützung der Arzneimittelsicherheit durch eine umfassende Wissensnutzung mittels neuer Informationstechnologien. Projektbeschreibung. Hoechst AG, Frankfurt und Deutsches Forschungszentrum für Künstliche Intelligenz, Kaiserslautern, Deutschland.
- Birk, A., Tschaitchian, B., Schmalhofer, F., Thüning, M., & Gertzen, H. (1994). Architecture and User-Interface of the IDEAS Intelligent Documentation System. in: Schuler, W., & Hannemann, J. (Eds) Methodological Issues on the Design of Hypertext-based User Interfaces. Berlin, Springer-Verlag.
- Breuker, J.A., & Wielinga, W.J. (1989). Model driven knowledge acquisition. in: Guida, P., & Tasso, G. (Eds.) Topics in the design of expert systems, pp. 265-296. Amsterdam, North-Holland: Elsevier.
- Boose, J. H., and Gaines, B. R. (1989). Knowledge acquisition of knowledge-based systems: Notes on the state-of-the-art. Machine Learning, 4: 377-394.
- Clancey, W.J. (1983). The Epistemology of a Rule-Based Expert System - A Framework for Explanation, Artificial Intelligence 20, p. 215-251.
- Clancey, W.J. (1985). Heuristic Classification, Artificial Intelligence 27, p. 289-350.
- Clancey, W.J. (1991). Change and confluence in cognitive science. in: VanLehn, K. (ed.) Architectures for intelligence. The Twenty-second Carnegie Mellon Symposium on Cognition, pp. 357-423. Hillsdale, NJ: Lawrence Erlbaum.
- Eisenecker, U. W., & Köpf, W. (1993). Die aufgabenspezifische Gestaltung von Benutzungsoberflächen aus der Perspektive der Anwender im betrieblichen Umfeld.

- Engineering RE '93 - Prototyping, Bonn, Züllighoven, H. (Ed.), Stuttgart: Teubner-Verlag.
- Fischer, G., McCall, R., Ostwald, J., Reeves, B., & Shipman, F. (1994). Seeding, evolutionary growth and reseeded: Supporting the incremental development of design environments. CHI'94 Conference Proceedings, April, 24-28, Boston, MA. Reading, MA: Addison-Wesley.
- Fischer, G., & Reeves, B. (1992). Beyond intelligent interfaces: Exploring, analyzing, and creating success models of cooperative problem solving. J. Appl. Intel. 1, Kluwer Academic Publishers, pp. 311-332.
- Goldstein, I., & Papert, S. (1991). Artificial Intelligence, language, and the study of knowledge. Cognitive Science 1 (1): 84-123
- Karbach, W. & Linster, M. (1990). Wissensakquisition für Expertensysteme: Techniken, Modelle und Softwarewerkzeuge. München, Wien: Hanser.
- Lajoie, S., & Lesgold, A. (1989). Apprenticeship training in the workplace: Computer coached practice environment as a new form of apprenticeship. Machine-Mediated Learning, 3, pp. 7-28.
- Lenat, D.B. (1989). When Will Machines Learn?. Machine Learning, 4(3/4), pp. 255-258.
- Lenat, D., Prakash, M., & Shepherd, M. (1986). CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. AI Magazine, pp. 65-85.
- Lesgold, A. M. (1993) Beyond a commodity view of knowledge in instruction. in: Strube, G., & Wender, K.F. (Eds.) The cognitive psychology of knowledge, pp. 425-433. Amsterdam, North-Holland: Elsevier.
- Mangold-Allwinn, R., Antoni, C, König, D., & Eisenecker, U. (1993). DEI: Ein wissensdiagnostisches Verfahren zur Erhebung von Aufgabenmodellen. Proceedings des Workshops Psychologie des Software-Entwurfs, 14.-15. Mai, Bad Neuenahr, Deutschland.
- Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W.R. (1991). Enabling Technology for Knowledge Sharing. AI Magazine, 12(1), pp. 3-14.

Schmalhofer, F., & Tschaitshian, B. (1993). The acquisition of a procedure schema from text and experiences. Proceedings of the 15th Annual Conference of the Cognitive Science Society, pp. 883-888. Hillsdale, NJ: Lawrence Erlbaum.

Schmalhofer, F., & Wetter, T. (1988). Kognitive Modellierung: Menschliche Wissensrepräsentation und Verarbeitungsstrategien. In: Christaller, T., Hein, H.-W., & Richter, M.M. (Hrsg.) Künstliche Intelligenz: Theoretische Grundlagen und



**Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH**

DFKI
-Bibliothek-
PF 2080
67608 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse oder per anonymem ftp von ftp.dfki.uni-kl.de (131.246.241.100) unter pub/Publications bezogen werden.

DFKI Publications

The following DFKI publications or the list of all published papers so far are obtainable from the above address or via anonymous ftp from ftp.dfki.uni-kl.de (131.246.241.100) under pub/Publications.

zeichnet, kostenlos abgegeben.

DFKI Research Reports

RR-93-00

RR-93-16

Gert Smolka, Martin Henz, Jörg Würtz: Object-Oriented Concurrent Constraint Programming in Oz

RR-93-26

Jörg P. Müller, Markus Fischel: The Agent Architecture InteRRaP: Concept and Application
99 pages

RR-93-27

Hans-Ulrich Krieger:
Derivation Without Lexical Rules
33 pages

RR-93-28

Hans-Ulrich Krieger, John Nerbonne, Hannes Pirker: Feature-Based Allomorphy
8 pages

RR-93-29

Armin Laux: Representing Belief in Multi-Agent Worlds via Terminological Logics
35 pages

RR-93-30

Stephen P. Spackman, Elizabeth A. Hinkelman: Corporate Agents
14 pages

RR-93-31

Elizabeth A. Hinkelman, Stephen P. Spackman: Abductive Speech Act Recognition, Corporate Agents and the COSMA System
34 pages

RR-93-32

David R. Traum, Elizabeth A. Hinkelman: Conversation Acts in Task-Oriented Spoken Dialogue
28 pages

RR-93-33

Bernhard Nebel, Jana Koehler: Plan Reuse versus Plan Generation: A Theoretical and Empirical Analysis
33 pages

RR-93-34

Wolfgang Wahlster: Verbmobil Translation of Face-To-Face Dialogs
10 pages

RR-93-35

Harold Boley, François Bry, Ulrich Geske (Eds.): Neuere Entwicklungen der deklarativen KI-Programmierung — Proceedings
150 Seiten

Note: This document is available only for a nominal charge of 25 DM (or 15 US-\$).

RR-93-36

Michael M. Richter, Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner, Gabriele Schmidt: Von IDA bis IMCOD: Expertensysteme im CIM-Umfeld
13 Seiten

RR-93-38

Stephan Baumann: Document Recognition of Printed Scores and Transformation into MIDI
24 pages

RR-93-40

Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt, Andrea Schaerf: Queries, Rules and Definitions as Epistemic Statements in Concept Languages
23 pages

RR-93-41

Winfried H. Graf: LAYLAB: A Constraint-Based Layout Manager for Multimedia Presentations
9 pages

RR-93-42

Hubert Comon, Ralf Treinen: The First-Order Theory of Lexicographic Path Orderings is Undecidable
9 pages

RR-93-43

M. Bauer, G. Paul: Logic-based Plan Recognition for Intelligent Help Systems
15 pages

RR-93-44

Martin Buchheit, Manfred A. Jeusfeld, Werner Nutt, Martin Staudt: Subsumption between Queries to Object-Oriented Databases
36 pages

RR-93-45

Rainer Hoch: On Virtual Partitioning of Large Dictionaries for Contextual Post-Processing to Improve Character Recognition
21 pages

RR-93-46

Philipp Hanschke: A Declarative Integration of Terminological, Constraint-based, Data-driven, and Goal-directed Reasoning
81 pages

RR-93-48

Franz Baader, Martin Buchheit, Bernhard Hollunder: Cardinality Restrictions on Concepts
20 pages

RR-94-01

Elisabeth André, Thomas Rist: Multimedia Presentations: The Support of Passive and Active Viewing
15 pages

RR-94-02

Elisabeth André, Thomas Rist: Von Textgeneratoren zu Intellimedia-Präsentationssystemen
22 Seiten

RR-94-03

Gert Smolka:

A Calculus for Higher-Order Concurrent Constraint Programming with Deep Guards

34 pages

RR-94-05

Franz Schmalhofer,

J. Stuart Aitken, Lyle E. Bourne jr.:

Beyond the Knowledge Level: Descriptions of Rational Behavior for Sharing and Reuse

81 pages

RR-94-06

Dietmar Dengler:

An Adaptive Deductive Planning System

17 pages

RR-94-07

Harold Boley: Finite Domains and Exclusions as First-Class Citizens

25 pages

RR-94-08

Otto Kühn, Björn Höfling: Conserving Corporate Knowledge for Crankshaft Design

17 pages

RR-94-10

Knut Hinkelmann, Helge Hintze:

Computing Cost Estimates for Proof Strategies

22 pages

RR-94-11

Knut Hinkelmann: A Consequence Finding Approach for Feature Recognition in CAPP

18 pages

RR-94-12

Hubert Comon, Ralf Treinen:

Ordering Constraints on Trees

34 pages

RR-94-13

Jana Koehler: Planning from Second Principles — A Logic-based Approach

49 pages

RR-94-14

Harold Boley, Ulrich Buhrmann, Christof Kremer: Towards a Sharable Knowledge Base on Recyclable Plastics

14 pages

RR-94-15

Winfried H. Graf, Stefan Neurohr: Using Graphical Style and Visibility Constraints for a Meaningful Layout in Visual Programming Interfaces

20 pages

RR-94-16

Gert Smolka: A Foundation for Higher-order Concurrent Constraint Programming

26 pages

DFKI Technical Memos

TM-92-04

Jürgen Müller, Jörg Müller, Markus Pischel, Ralf Scheidhauer:

On the Representation of Temporal Knowledge

61 pages

TM-92-05

Franz Schmalhofer, Christoph Globig, Jörg Thoben:

The refitting of plans by a human expert

10 pages

TM-92-06

Otto Kühn, Franz Schmalhofer: Hierarchical skeletal plan refinement: Task- and inference structures

14 pages

TM-92-08

Anne Kilger: Realization of Tree Adjoining Grammars with Unification

27 pages

TM-93-01

Otto Kühn, Andreas Birk: Reconstructive Integrated Explanation of Lathe Production Plans

20 pages

TM-93-02

Pierre Sablayrolles, Achim Schupeta: Conflict Resolving Negotiation for COoperative Schedule Management

21 pages

TM-93-03

Harold Boley, Ulrich Buhrmann, Christof Kremer: Konzeption einer deklarativen Wissensbasis über recyclingrelevante Materialien

11 pages

TM-93-04

Hans-Günther Hein:

Propagation Techniques in WAM-based Architectures — The FIDO-III Approach

105 pages

TM-93-05

Michael Sintek: Indexing PROLOG Procedures into DAGs by Heuristic Classification

64 pages

TM-94-01

Rainer Bleisinger, Klaus-Peter Gores:

Text Skimming as a Part in Paper Document Understanding

14 pages

TM-94-02

Rainer Bleisinger, Berthold Kröll:

Representation of Non-Convex Time Intervals and Propagation of Non-Convex Relations

11 pages

DFKI Documents**D-93-08**

Thomas Kieninger, Rainer Hoch:
Ein Generator mit Anfragesystem für strukturierte
Wörterbücher zur Unterstützung von Texterkennung
und Textanalyse
125 Seiten

D-93-09

Hans-Ulrich Krieger, Ulrich Schäfer:
TDL ExtraLight User's Guide
35 pages

D-93-10

Elizabeth Hinkelman, Markus Vonerden, Christoph Jung: Natural Language Software Registry
(Second Edition)
174 pages

D-93-11

Knut Hinkelmann, Armin Laux (Eds.):
DFKI Workshop on Knowledge Representation
Techniques — Proceedings
88 pages

D-93-12

*Harold Boley, Klaus Elsbernd,
Michael Herfert, Michael Sintek, Werner Stein:*
RELFUN Guide: Programming with Relations and
Functions Made Easy
86 pages

D-93-14

Manfred Meyer (Ed.): Constraint Processing –
Proceedings of the International Workshop at
CSAM'93, July 20-21, 1993
264 pages

Note: This document is available only for a
nominal charge of 25 DM (or 15 US-\$).

D-93-15

Robert Laux:
Untersuchung maschineller Lernverfahren und
heuristischer Methoden im Hinblick auf deren
Kombination zur Unterstützung eines Chart-Parsers
86 Seiten

D-93-16

*Bernd Bachmann, Ansgar Bernardi, Christoph
Klauck, Gabriele Schmidt:* Design & KI
74 Seiten

D-93-20

Bernhard Herbig:
Eine homogene Implementierungsebene für einen
hybriden Wissensrepräsentationsformalismus
97 Seiten

D-93-21

Dennis Drollinger:
Intelligentes Backtracking in Inferenzsystemen am
Beispiel Terminologischer Logiken
53 Seiten

D-93-22

Andreas Abecker:
Implementierung graphischer Benutzungsober-
flächen mit Tcl/Tk und Common Lisp
44 Seiten

D-93-24

Brigitte Krenn, Martin Volk:
DiTo-Datenbank: Datendokumentation zu
Funktionsverbgefügen und Relativsätzen
66 Seiten

D-93-25

Hans-Jürgen Bürckert, Werner Nutt (Eds.):
Modeling Epistemic Propositions
118 pages
Note: This document is available only for a
nominal charge of 25 DM (or 15 US-\$).

D-93-26

Frank Peters: Unterstützung des Experten bei der
Formalisierung von Textwissen
INFOCOM:
Eine interaktive Formalisierungskomponente
58 Seiten

D-93-27

*Rolf Backofen, Hans-Ulrich Krieger,
Stephen P. Spackman, Hans Uszkoreit (Eds.):*
Report of theEAGLES Workshop on
Implemented Formalisms at DFKI, Saarbrücken
110 pages

D-94-01

Josua Boon (Ed.):
DFKI-Publications: The First Four Years
1990 - 1993
75 pages

D-94-02

Markus Steffens: Wissenserhebung und Analyse
zum Entwicklungsprozeß eines Druckbehälters aus
Faserverbundstoff
90 pages

D-94-03

Franz Schmalhofer: Maschinelles Lernen:
Eine kognitionswissenschaftliche Betrachtung
54 pages

D-94-04

Franz Schmalhofer, Ludger van Elst:
Entwicklung von Expertensystemen:
Prototypen, Tiefenmodellierung und kooperative
Wissensevolution
22 pages

D-94-06

Ulrich Buhrmann:
Erstellung einer deklarativen Wissensbasis über
recyclingrelevante Materialien
117 pages

D-94-08

Harald Feibel: IGLOO 1.0 - Eine grafikunterstützte
Beweisentwicklungsumgebung
58 Seiten

**Entwicklung von Expertensystemen:
Prototypen, Tiefenmodellierung und kooperative Wissensevolution**
Franz Schmalhofer, Ludger van Elst

D-94-04
Document