

## **Arbeit und Technik 5**

**Wolfgang Dzida**

**Udo Konradt (Hrsg.)**

# **Psychologie des Software-Entwurfs**



ARBEIT UND TECHNIK: PRAXISORIENTIERTE  
BEITRÄGE AUS PSYCHOLOGIE UND INFORMATIK

Herausgegeben von Prof. Dr. Michael Frese und Prof. Dr. Horst Oberquelle

---

BAND V

Psychologie des  
Software-Entwurfs

herausgegeben von  
Dr. Wolfgang Dzida  
und  
Dr. Udo Konradt



Verlag für Angewandte Psychologie

Psychologie des  
Software-Entwurfs

herausgegeben von  
Dr. Wolfgang Dzida  
und  
Dr. Udo Konradt



Verlag für Angewandte Psychologie

## Weitere Veröffentlichungen aus der Reihe Arbeit und Technik

Band 1  
Kooperative Arbeit  
und Computerunterstützung  
herausgegeben von  
Prof. Dr. Horst Oberquelle

Band 3  
Anpaßbare Informationssysteme  
von Dr. Detlef Haaks

Band 2  
Expertenkönnen  
Erkennen und Vermitteln  
von Prof. Dr. Winfried Hacker

Band 4  
Mentale Modelle:  
Konstrukte des Wissens und Verstehens  
herausgegeben von Dr. Stephan Dutke

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

**Psychologie des Software-Entwurfs** / hrsg. von  
Wolfgang Dzida und Udo Konradt. - Göttingen ;  
Stuttgart : Verl. für Angewandte Psychologie, 1995  
(Arbeit und Technik ; Bd. 5)  
ISBN 3-8017-0889-6  
NE: Dzida, Wolfgang [Hrsg.]; GT

© Hogrefe-Verlag, Göttingen • Bern • Toronto • Seattle 1995  
Rohnsweg 25, D-37085 Göttingen



Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Gesamtherstellung: Dieterichsche Universitätsbuchdruckerei  
W. Fr. Kaestner GmbH & Co. KG, D-37124 Göttingen-Rosdorf  
Printed in Germany  
Auf säurefreiem Papier gedruckt

ISBN 3-8017-0889-6

## Inhalt

### Vorwort der Herausgeber

### Die Bedeutung des Entwurfs für die Software-Entwicklung

*Udo Konradt, Wolfgang Dzida*

### I. Paradigmen des Software-Entwurfs

1. Datenorientierte Modellierung und Entwicklung betrieblicher Anwendungssysteme unter Beteiligung der Benutzer  
*Stefan Albers, Hans Joachim Petzold*
2. Objektorientierte Systementwicklung  
*Ulrich Piepenburg, Heinz Züllighoven*

### II. Zur Validierung des Software-Entwurfs

3. Struktur begrifflichen Wissens im menschlichen Gedächtnis - Anregungen für den Software-Entwurf?  
*Hartmut Wandtke*
4. Überlegungen zur semantischen Validierung der objektorientierten Analyse  
*Constanze Clauser*
5. Konzeptuelle Validierung des Entwurfs objektorientierter Benutzungsschnittstellen  
*Geerd Philipsen*

### III. Analyse und Repräsentation von Arbeitsaufgaben

6. Erhebung von Aufgabenmodellen als Beitrag zur Softwaregestaltung  
*Roland Mangold-Allwinn, Conny Antoni und Ulrich Eisenecker*
7. Die aufgabenspezifische Gestaltung von Benutzungsoberflächen aus der Perspektive der Anwender im betrieblichen Umfeld  
*Ulrich Eisenecker, Wolfgang Köpf*
8. Externalisierung von Aufgabenstrukturen und ihre Berücksichtigung im Software-Entwurf  
*Wolfgang Dzida*

#### IV. Analyse des Entwurfsprozesses

- |                                                                                                             |     |
|-------------------------------------------------------------------------------------------------------------|-----|
| 9. Software-Entwicklung als Prozeß -<br>Kognitive, motivationale und soziale Faktoren<br><i>Udo Konradt</i> | 183 |
| 10. Methoden zur Untersuchung von Designprozessen<br><i>Sabine Sonnentag</i>                                | 203 |

#### V. Wissensakquisition für den Software-Entwurf

- |                                                                                                                                                        |     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 11. Entwicklung von Expertensystemen: Prototypen,<br>Tiefenmodellierung und kooperative Wissensrevolution<br><i>Franz Schmalhofer, Ludger van Elst</i> | 223 |
| 12. Handlungsorientierte Wissensaufbereitung zur Ermittlung<br>des Unterstützungsbedarfs<br><i>Nina Großmann</i>                                       | 245 |
| 13. Tätigkeitsregulierende Leistungsvoraussetzungen:<br>Ein methodischer Beitrag zum Software-Entwurf<br><i>Simone Teske-El Kodwa</i>                  | 261 |

|                     |     |
|---------------------|-----|
| <b>Sachregister</b> | 277 |
|---------------------|-----|

## Vorwort

Software zu entwerfen ist wie die Programmierung keine rein Angelegenheit. Im Software-Entwurf wird auch über die Gestal Arbeitsaufgaben der künftigen Benutzer entschieden, etwa darüber, inw semantischen Konzepte, die Benutzer von ihren Aufgaben, Arbeitsgegenst Arbeitsverfahren haben, durch Anwendungsprogramme widergespiegelt w Lösung dieser Gestaltungsaufgaben kann die Psychologie mit Kenntr Methoden beitragen. Das hier vorliegende Buch baut eine Brücke Psychologie und Informatik. Da die meisten Beiträge auf psychologische Sa des Software-Entwurfs konzentriert sind, werden einerseits Konzepte und der Psychologie in die Angewandte Informatik getragen, andererseits Probleme der Software-Designer zum Gegenstand psychologischer gemacht. Der Austausch zwischen den Disziplinen Psychologie, Infor Wirtschaftsinformatik wurde anlässlich einer Arbeitstagung in Bad Neuena 1993, begonnen und hat zu einem Ergebnis geführt, das in Form gründlich ter Vorträge hiermit vorgelegt werden kann.

Software und Psychologie stehen seit Ben Shneidermans Buch "Psychology" (1980) in einem multidisziplinären Verhältnis zueinander. A dieses Pionierwerkes gab es bereits einen kurzen Abschnitt über den Er Software zu lesen; psychologische Grundlagen wurden jedoch kaum er Vordergrund standen vielmehr die psychologischen Probleme des Progra etwa die Tätigkeit des Programmiers, sein Stil zu programmieren und di der Qualität seiner Arbeitsergebnisse. Mit der Entwicklung von Entw Konstruktionstechniken der Systemanalyse hat sich inzwischen der Softwa gegenüber der Programmierung als eine selbständige Tätigkeit herausgebild hinaus hat sich innerhalb des Software-Engineerings der Aktivitätsschwe Systemanalyse verschoben.

Bisher schien es unvermeidbar zu sein, beim Übergang vom En Programm die Konzepte eines Anwendungsgebietes durch technische K verzerren, die sich aus den Sachzwängen der Implementierung erga Abbildung des Anwendungsgebietes, z.B. dem Verständnis des Anwende chende Aufgaben und Organisationsformen, brächte jedoch für die Ben Rechners große ergonomische Vorteile. Die Software-Ergonomen fordern Eingaben des Benutzers und Ausgaben des Systems den Anforderungen gere sollen, die sich aus den Aufgabenerfordernissen ergeben. Diese Forderung nicht erst bei der Gestaltung der Benutzungsschnittstelle eingelöst, vie

#### IV. Analyse des Entwurfsprozesses

9. Software-Entwicklung als Prozeß -  
Kognitive, motivationale und soziale Faktoren 183  
*Udo Konradt*

10. Methoden zur Untersuchung von Designprozessen 203  
*Sabine Sonnentag*

#### V. Wissensakquisition für den Software-Entwurf

11. Entwicklung von Expertensystemen: Prototypen,  
Tiefenmodellierung und kooperative Wissensrevolution 223  
*Franz Schmalhofer, Ludger van Elst*

12. Handlungsorientierte Wissensaufbereitung zur Ermittlung  
des Unterstützungsbedarfs 245  
*Nina Großmann*

13. Tätigkeitsregulierende Leistungsvoraussetzungen:  
Ein methodischer Beitrag zum Software-Entwurf 261  
*Simone Teske-El Kodwa*

#### Sachregister 277

### Vorwort

Software zu entwerfen ist wie die Programmierung keine rein technische Angelegenheit. Im Software-Entwurf wird auch über die Gestaltung von Arbeitsaufgaben der künftigen Benutzer entschieden, etwa darüber, inwieweit die semantischen Konzepte, die Benutzer von ihren Aufgaben, Arbeitsgegenständen oder Arbeitsverfahren haben, durch Anwendungsprogramme widergespiegelt werden. Zur Lösung dieser Gestaltungsaufgaben kann die Psychologie mit Kenntnissen und Methoden beitragen. Das hier vorliegende Buch baut eine Brücke zwischen Psychologie und Informatik. Da die meisten Beiträge auf psychologische Sachverhalte des Software-Entwurfs konzentriert sind, werden einerseits Konzepte und Methoden der Psychologie in die Angewandte Informatik getragen, andererseits praktische Probleme der Software-Designer zum Gegenstand psychologischer Forschung gemacht. Der Austausch zwischen den Disziplinen Psychologie, Informatik und Wirtschaftsinformatik wurde anlässlich einer Arbeitstagung in Bad Neuenahr, im Mai 1993, begonnen und hat zu einem Ergebnis geführt, das in Form gründlich überarbeiteter Vorträge hiermit vorgelegt werden kann.

Software und Psychologie stehen seit Ben Shneidermans Buch "Software Psychology" (1980) in einem multidisziplinären Verhältnis zueinander. Am Schluß dieses Pionierwerkes gab es bereits einen kurzen Abschnitt über den Entwurf von Software zu lesen; psychologische Grundlagen wurden jedoch kaum erwähnt. Im Vordergrund standen vielmehr die psychologischen Probleme des Programmierens, etwa die Tätigkeit des Programmierers, sein Stil zu programmieren und die Messung der Qualität seiner Arbeitsergebnisse. Mit der Entwicklung von Entwurfs- und Konstruktionstechniken der Systemanalyse hat sich inzwischen der Software-Entwurf gegenüber der Programmierung als eine selbständige Tätigkeit herausgebildet; darüber hinaus hat sich innerhalb des Software-Engineerings der Aktivitätsschwerpunkt zur Systemanalyse verschoben.

Bisher schien es unvermeidbar zu sein, beim Übergang vom Entwurf zum Programm die Konzepte eines Anwendungsgebietes durch technische Konzepte zu verzerren, die sich aus den Sachzwängen der Implementierung ergaben. Eine Abbildung des Anwendungsgebietes, z.B. dem Verständnis des Anwenders entsprechende Aufgaben und Organisationsformen, brächte jedoch für die Benutzung des Rechners große ergonomische Vorteile. Die Software-Ergonomen fordern z.B., daß Eingaben des Benutzers und Ausgaben des Systems den Anforderungen gerecht werden sollen, die sich aus den Aufgabenerfordernissen ergeben. Diese Forderung wird aber nicht erst bei der Gestaltung der Benutzungsschnittstelle eingelöst, vielmehr sind

Voraussetzungen dafür schon beim Software-Entwurf zu schaffen. Die Psychologie des Software-Entwurfs bietet somit wichtige Grundlagen für eine ergonomische Gestaltung. Während sich die Software-Ergonomie bisher fast ausschließlich mit der *Benutzung* von Rechnern befaßt hat, besinnt sie sich nunmehr auf die psychologischen und technisch-konstruktiven Voraussetzungen ergonomischen Gestaltens.

Technische und konstruktive Voraussetzungen sind durch die gegenwärtig bevorzugten Paradigmen des Software-Entwurfs gegeben, d.h. Datenmodellierung, funktionale Dekomposition und objekt-orientierte Konstruktion. Im Unterschied zu den erstgenannten Paradigmen scheint die objekt-orientierte Konstruktion einen bruchlosen Übergang von der Analyse des Anwendungsbereichs, über den Entwurf, bis hin zur Programmierung zu bieten, so daß technische Begriffe erst spät im Entwicklungsprozeß relevant werden und das Anwendungsmodell nicht verzerren. Letztendlich beruht auch der objekt-orientierte Entwurf nicht auf dem Verständnis des Anwenders, sondern auf Kenntnissen, die der Designer von Aufgaben und Problemen eines Anwendungsgebietes erworben hat.

Sollen die Vorteile der Entwurfs-Paradigmen wirklich zuverlässig genutzt werden, so ist zu überlegen, wie man im Entwurfs- und Konstruktionsprozeß der Sichtweise des Anwenders möglichst nahe bleiben kann, und es ist zu prüfen, wo sich das "intuitive Verständnis" des Designers auf den Software-Entwurf in unkontrollierter Weise auswirkt. Die Psychologie kann dazu beitragen,

- daß unsicheres Wissen über das Anwendungsgebiet präzisiert wird,
- daß die Konzepte der Benutzer und Anwender die Grundlage des "semantischen Modells" werden,
- daß der Designer sein meist heuristisches Vorgehen besser begründen kann,
- daß Konzepte aus dem Anwendungsbereich nicht von Implementationsentscheidungen verzerrt werden und
- daß schließlich ein semantisch valider Entwurf entsteht.

Auf diese Weise könnte eine ergonomische Qualitätssicherung von Software bereits beim Entwurf beginnen, also vor der Programmierung und dem Test ansetzen. Eine psychologische Unterstützung des Entwurfsprozesses kann jedoch nicht nur prospektiv zur Qualitätsverbesserung des Entwurfs, sondern auch zur Steigerung der Effizienz der Designertätigkeit beitragen.

Das vorliegende Buch steht in der multidisziplinären Tradition der Software-Psychologie. Um den Einstieg in die Thematik des Software-Entwurfs zu erleichtern, haben die Herausgeber dieses Buches ein einleitendes Kapitel vorangestellt. Die Bedeutung, die heute der Software-Entwurf in der industriellen Software-Entwicklung bekommen hat, wird anschaulich beschrieben; hierbei werden psychologische Forschungsthemen angeschnitten, die teilweise in den nachfolgenden Aufsätzen vertieft werden. Weitere einführende Aufsätze liefern Stefan Albers und Hans Joachim Petzold

sowie Ulrich Piepenburg und Heinz Züllighoven. Sie wenden sich an Psychologen, die konkurrierende Paradigmen des Software-Entwurfs kennenlernen wollen, ohne mit komplizierten informatischen oder technischen Details überfordert zu werden. Dabei schält sich die "semantische Modellierung" als das psychologisch zu bearbeitende Kernproblem heraus. Hartmut Wandke referiert konzeptionelle und empirische Ergebnisse der psychologischen Grundlagenforschung zur semantischen Repräsentation im menschlichen Gedächtnis - ein Aufsatz, der sich an die Informatiker wendet, die sich mit psychologischen Grundlagen ihrer Entwurfstätigkeit auseinandersetzen wollen. Der folgende Aufsatz von Constanze Clauser setzt diese Grundlagen voraus und zeigt einige psychologische Probleme der semantischen Validität auf, aus denen sich besonders für den objekt-orientierten Designer Anregungen für sein oft noch heuristisches Vorgehen ergeben. Ergänzend hierzu reflektiert Geerd Philipsen die theoretischen und methodischen Möglichkeiten einer semantischen Validierung, und zwar am Beispiel des objekt-orientierten Entwurfs, speziell des Entwurfs von Benutzungsschnittstellen. Er untersucht Gründe, ob und inwieweit "semantische Validität" empirisch überhaupt feststellbar ist. Grundlage seiner Analyse ist das Konzept der "ökologischen" Wahrnehmung.

Im Mittelpunkt der semantischen Modellierung stehen die Aufgaben- und Organisationsanalyse, klassische Themen der Systemanalyse, die jedoch bisher den Psychologen unzugänglich erschienen. Zwar haben Arbeitspsychologen viele taugliche Verfahren der Arbeits- und Tätigkeitsbewertung entwickelt, ihr Beitrag zur Systemmodellierung mittels einer Beschreibungsmethode für Aufgabenstrukturen steht bisher noch aus. Eine methodische Grundlage für die Erhebung eines Aufgabenmodells liefert der Beitrag von Roland Mangold-Allwinn und Daniel König. Sie stellen ein erprobtes Aufgabenanalyse-Instrument vor, das unter Beteiligung der Benutzer angewendet werden kann und eine semi-formale Modellierung erlaubt, die einerseits die Benutzer nicht abschreckt, aber andererseits formal genug ist, um beim Systementwurf ernst genommen zu werden (siehe das anschließend dargestellte Industrieprojekt). Aus den praktischen Erfordernissen eines Industrieprojekts setzen Ulrich Eisenecker und Wolfgang Köpf ein Verfahren zum Entwurf von Aufgabenmodellen um, das für komplexe, integrierte Anwendungssysteme relevant ist, in denen sich Benutzer besser zurechtfinden sollen. Die psychologische und formale Analyse von Aufgaben wird schließlich durch einen grundlagenorientierten Aufsatz von Wolfgang Dzida ergänzt, der psychologische Konzepte der Gedächtnisorganisation (siehe Aufsatz von Wandke) und der Handlungsregulationstheorie in der Form von Petri-Netzen zusammenführt. Damit zeigt er, daß Petri-Netze eine semantische Modellierung von Aufgabenstrukturen ermöglichen, deren Formalismus keineswegs willkürlich ist und deshalb den Anspruch an eine valide Repräsentation erfüllen hilft.

Um als Psychologe multidisziplinär arbeiten zu können, sollten nicht nur die wichtigsten Paradigmen des Software-Entwurfs bekannt sein, sondern auch die Probleme des Entwurfsprozesses. Udo Konradt zeigt drei wesentliche Problemfelder des Software-Entwicklungsprozesses auf und analysiert kognitive, motivationale und soziale Faktoren, die sie bedingen. Für den Leser mit informatischem Hintergrund ist interessant zu erfahren, wie diese Probleme psychologisch bewertet werden und welche Maßnahmen zur Verbesserung des Entwicklungsprozesses beitragen. Sabine Sonntag stellt einige für solche Maßnahmen oder Untersuchungen geeignete

Methoden vor, insbesondere wenn es Probleme von Entwurfsprozessen in realen Arbeitssituationen zu erfassen gilt. Sie zeigt die Stärken der einzelnen Methoden und macht Vorschläge zur Kompensation der Schwachpunkte. Betreibt man Qualitätssicherung von Anfang an, also schon im Entwurfsprozess, so können Methoden zur Untersuchung der Effizienz von Qualitätssicherung dazu beitragen, das Qualitätsmanagement eines Herstellers zu beeinflussen.

Die Akquisition von Wissen ist bisher bei der Entwicklung von Expertensystemen am besten untersucht und erprobt worden. Aufgabenanalyse und Wissensakquisition sind verwandt, so daß die Erfahrungen aus dem Entwurf wissensbasierter Systeme für die semantische Modellierung ausgewertet werden können. Franz Schmalhofer stellt Entwurfstechniken vor, die der Akquisition des Wissens für Expertensysteme dienlich sind. Auch Schmalhofer kommt zu dem Schluß, daß die Tauglichkeit des Wissens von der Kommunikation zwischen den am Entwurf Beteiligten abhängt, die einen gemeinsamen Wissenstand durch evolutionäres Vorgehen erarbeiten (kooperative Wissensrevolution). Schmalhofer zeigt Parallelen zwischen den Entwurfstechniken für Expertensysteme und denen für Anwendungsprogramme. Für beide Arten von Software ist der Trend erkennbar, das beim Entwurf entstehende semantische Modell durch inkrementelle Anpassung und durch Konsens zwischen Entwicklern und Anwendern zu validieren (konsensuale Semantik). Nina Großmann stellt ein in der arbeitspsychologischen Tradition entstandenes Prinzip vor: Entwicklung von Technik, um die menschlichen Stärken zu fördern, anstatt diese durch Maschinen ersetzen zu wollen. Ihr Beitrag konkretisiert die mit dieser These verbundene Methodik, die als ein psychologischer Beitrag zur Systemanalyse entwickelt wurde und bereits praktisch eingesetzt wird. Die Methodik leistet ihren Beitrag zum Systementwurf insoweit, als sie den Unterstützungsbedarf der Benutzer und ihrer Organisationseinheit ermitteln hilft. Die Ergebnisse gehen in die Anforderungsanalyse ein. Im Anschluß an die Ermittlung des Unterstützungsbedarfs ist eine Methodik der Wissensakquisition angezeigt, die von den tätigkeitsregulierenden Leistungsvoraussetzungen ausgeht. Simone Teske-El Kodwa stellt eine hierfür geeignete Methodik der Wissensakquisition vor. Sie empfiehlt den Einsatz kombinierter Techniken in einem iterativen Prozeß der Erschließung aufgabenrelevanter Wissensselemente, und zwar im realen Aufgabenkontext und in der Interaktion mehrerer Sachexperten.

Die Herausgeber danken allen, die geholfen haben, dieses Buch zu veröffentlichen, vor allem der Firma Daimler-Benz für eine Spende zu den Druckkosten und Frau Ilona Reinhardt für die sorgfältigen Formatierungsarbeiten. Besonderer Dank gilt Frau Dr. Clauser, die zusammen mit den Herausgebern an der inhaltlichen Integration der Beiträge gearbeitet hat. Allen Autoren sei gedankt für die Bereitschaft, Änderungsvorschläge der Herausgeber zu berücksichtigen, so daß ein Buch für Leser aus verschiedenen Disziplinen entstehen konnte.

*Wolfgang Dzida*

*Udo Konradt*

# **Entwicklung von Expertensystemen: Prototypen, Tiefenmodellierung und kooperative Wissensevolution**

Franz Schmalhofer und Ludger van Elst

## **1. Einleitung**

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 (Dzida, in diesem Band) 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 dessen besonders 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. Desweiteren stellen wir punktuell einen Vergleich mit Verfahren bei der Entwicklung von Anwendungssoftware an.

## 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. Dieses kann für Expertensysteme der ersten Generation als typisch angesehen werden. 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 Medikamente, die von MYCIN verordnet wurden, mit den Verordnungen von neun menschlichen Experten. Als Aufgabenstellung waren zehn schwierige Fälle von Meningitis ausgewählt worden. Zwei Beurteilungskriterien waren maßgeblich. Erstens wurde bewertet, ob das verschriebene Medikament bei der aktuell vorhandenen 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 regelbasierten Expertensysteme ergeben sich jedoch einige Probleme. Als beispielsweise NEOMYCIN erstellt wurde, war es sehr schwierig, die Regeln aus MYCIN in das neue System zu übertragen. So berichtet Clancey (1983): "Als wir die Regeln von MYCIN nach NEOMYCIN übertrugen, brauchten wir viele Beratungen mit den ursprünglichen Autoren, um die volle Bedeutung der Regeln wieder aufzudecken. (...) Der Mangel an festgelegten Prinzipien für die Benutzung des Repräsentationsformalismus macht die Interpretation des Zwecks von Klauseln und Regeln schwer. Die Strategie und die Struktur des Systems mußten mit Hilfe von Diagrammen rekonstruiert werden" (Übers. nach Karbach und Linster, 1990). Durch die geringe Wartbarkeit dieser Expertensysteme deutete sich bereits an, daß die Vorgehensweise von Experten und ihr Verständnis von Regeln in diesen Systemen nicht explizit repräsentiert waren.

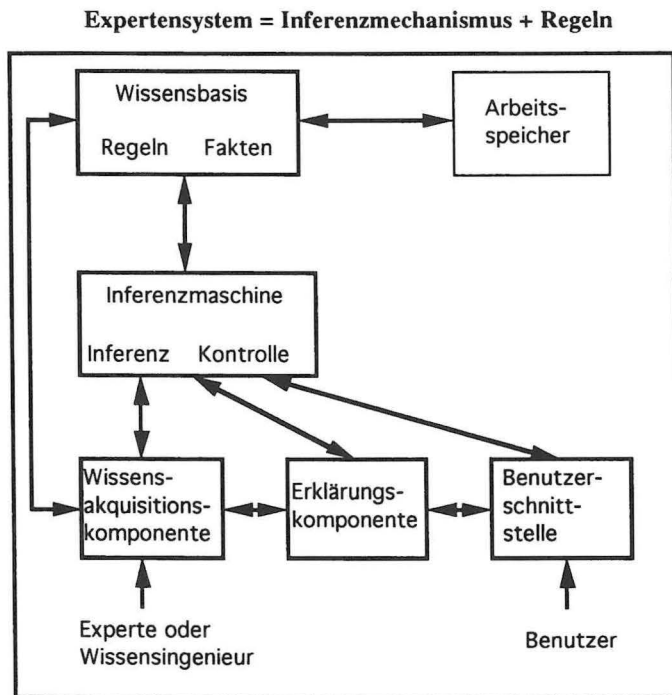
### 2.3 Die Systemarchitektur

Den Aufbau der Expertensysteme der ersten Generation zeigt Abbildung 1. Die Inferenzmaschine mit dem verwendeten Repräsentationsmechanismus und die Wissensbasis wurden dabei als die beiden zentralen Komponenten des Systems angesehen. Da diese Systeme im wesentlichen regelbasiert waren, ging man davon aus, daß ein Expertensystem aus einer Inferenzmaschine und einer Regelbasis besteht. Man betrachtete zunächst die Entwicklung der Inferenzmaschine mit dem Repräsentationsmechanismus als die Hauptarbeit der Expertensystementwicklung.

## 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 ähnlichen Resultaten.



**Abbildung 1:** Architektur der Expertensysteme der ersten Generation

### 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 Shell des Expertensystems 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 *Knowledge Engineering*. Im wesentlichen ergaben sich zwei unterschiedliche Auffassungen, warum Expertensysteme der ersten Generation nicht nach einem Standardverfahren entwickelt werden konnten:

- die Annahme vom verborgenen Wissen (*Mining View*)
- und den Ansatz der Tiefenmodellierung (*Model Building View*).

### 3.2 Verborgenes Wissen (*Mining View*)

Mit der Annahme vom verborgenen Wissen (*Mining View*) läßt sich die Vorstellung weiter aufrecht erhalten, daß ein Expertensystem aus einer von der Anwendungsdomäne unabhängigen Shell und einer bereichsspezifischen Wissensbasis besteht. Nach dieser Ansicht liegen die Schwierigkeiten beim Aufbau einer Regelbasis darin, daß Experten die Regeln ihres Verhaltens nicht unmittelbar mitteilen können. Es wird vermutet, daß das eigentliche Expertenwissen, welches das Problemlöseverhalten bestimmt, im Experten "tief verborgen" ist<sup>1</sup>.

<sup>1</sup> Zur 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

### 3.3 Tiefenmodellierung (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 werden<sup>2</sup>.

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 *definitiven Abstraktion* wird eine terminologische Umformulierung vorgenommen. In einem letzten Abstraktionsschritt wird dann *generalisiert*. Es schließt sich ein *heuristic 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.

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 dann das Designmodell (M2) erstellt, ein Modell des zu konstruierenden Experten-

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.

<sup>2</sup> Mangold-Allwinn, Antoni & Eisenecker (in diesem Band) 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.

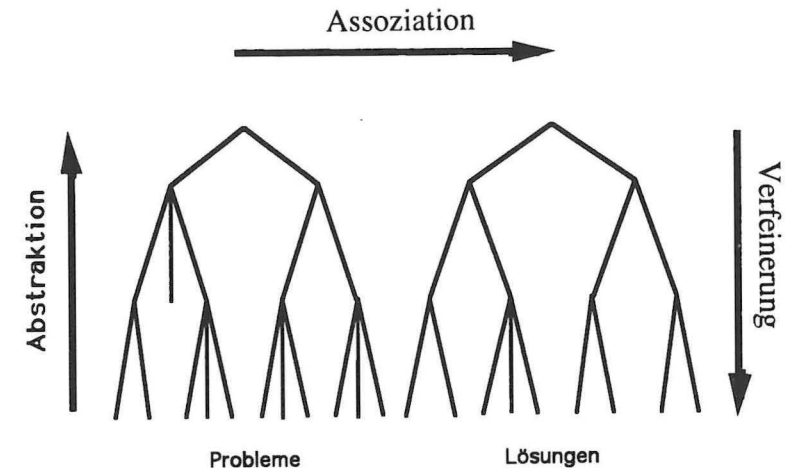


Abbildung 2: Schema der heuristischen Klassifikation (nach Karbach & Linster, 1990)

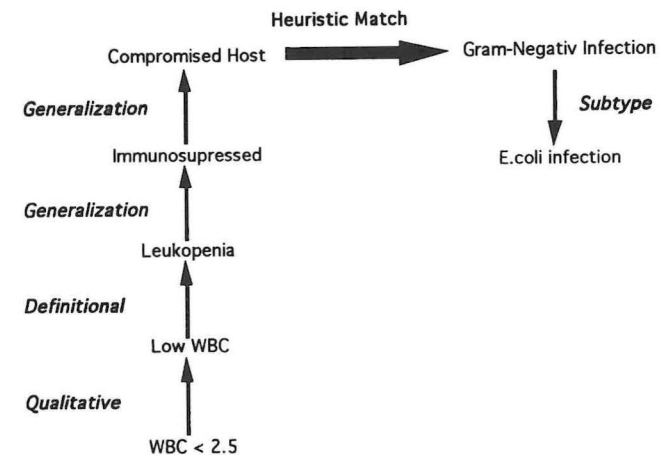


Abbildung 3: Die Inferenzstruktur von MYCIN als heuristische Klassifikation (nach Clancey, 1985)

systems. Solche Designmodelle können auch mit formalen Beschreibungssprachen dargestellt werden, so daß daraus ausführbare Spezifikationen des zu erstellenden Systems entstehen können (M3) (vgl. Fensel et al., 1993). Durch eine Top-Down-Implementierung kann aus dem so spezifizierten System dann ein voll funktionsfähiges Expertensystem entstehen. Im modellbasierten Vorgehen der Expertensystementwicklung zeigen sich also mehrere Gemeinsamkeiten zum traditionellen Software Engineering (vgl. auch die Artikel von Albers und Petzold sowie von Eisenecker und Köpf in diesem Band).

Models and the development space of expert system

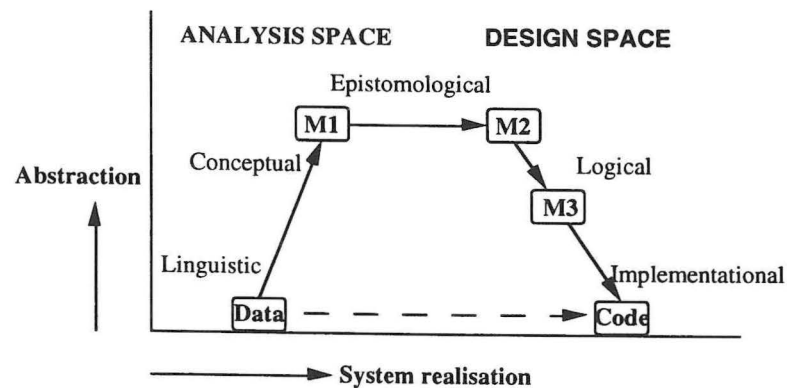


Abbildung 4: Entwicklungsraum für modellbasierte Expertensysteme (nach Breuker & Wielinga)

Nach der Erstellung eines angemessenen Modells wird zur Vervollständigung zusätzliches Detailwissen benötigt. Dieses muß erhoben und formalisiert werden, wobei die Verfahren der *Mining View* zur Anwendung kommen können. Es muß jedoch insbesondere darauf geachtet werden, daß Modell und erhobenes Detailwissen aufeinander abgestimmt sind.

Im Unterschied 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 sowie Selbsteinschätzungen über das repräsentierte Wissen bereitgestellt werden.

#### 4. Gemeinsame Wissensnutzung und Wiederverwendung

Sowohl der Prototypenansatz als auch das modellbasierte Verfahren haben das Ziel, Systeme zu erstellen, die selbstständig Analyse- oder Designaufgaben lösen können. Bei komplexen Domänen zeigen sich jedoch mehrere Probleme. Zum einen wünscht der Benutzer oftmals die Berücksichtigung von Faktoren, die während der Erstellung des Systems noch nicht bekannt waren. Zum anderen muß für jedes System das Domänenwissen komplett neu akquiriert werden. Die Folge davon ist, daß nur kleine oder mittelgroße Systeme entstehen, selbst wenn es viele Systeme innerhalb eines Domänenbereichs, z.B. der Medizin, gibt. Neches et al. (1991) sehen dafür im wesentlichen vier Gründe. So gibt es zum einen viele verschiedene Wissensrepräsentationssprachen, zum Teil aufgrund mangelnder Standardisierung, aber auch wegen unterschiedlicher Angemessenheit der einzelnen Repräsentationsformalismen für die jeweiligen Problemstellungen. Somit wird der Transfer des Wissens von einem System in ein anderes schwierig. Verstärkend kommt hier hinzu, daß es selbst innerhalb eines Repräsentationssystems verschiedene Dialekte gibt. Auch gibt es keine festgelegten Kommunikationsprotokolle, die den Aufbau von "Wissens-Servern" ermöglichen würden. Nicht zuletzt fehlt es auch an gemeinsamen Terminologien und Bezeichnungshierarchien, also Ontologien, die gemeinsame Wissensnutzung ermöglichen würden. In den Ansätzen der aktuellen Forschung und Entwicklung versucht man, solchen Problemen zu begegnen.

Im folgenden diskutieren wir den Einsatz von maschinellen Lernverfahren in bezug auf den *Knowledge Sharing and Reuse Effort*. Dabei interessiert uns insbesondere die Frage, ob ähnlich zur Entwicklung der menschlichen Expertise auch Expertensysteme mittels Lernen aus zunächst laienhaften Kenntnissen eines Wissensbasiskerns ein umfassenderes Verhaltenspotential aufbauen können.

Daher ist der Einsatz von sogenannten *kooperativen Lernverfahren* in komplexen Bereichen besonders erstrebenswert und bei einer Anpassung der einzelnen Techniken an die jeweilige Domäne sowie die Wünsche des Systembenutzers auch erfolgversprechend (Schank, 1991; Schmalhofer, Reinartz & Tschaischian, 1995). In den letzten Jahren wurden zwei grundsätzlich verschiedene Ansätze verfolgt, nämlich Lernen unter Verwendung von umfangreichem Alltagswissen (CYC-Ansatz) und gemeinsames Lernen eines Systems mit seinem Benutzer (z.B. kooperative Wissensrevolution). Die grundsätzlichen Unterschiede zwischen dem CYC-Ansatz und der kooperativen Wissensrevolution sind in Abbildung 5 durch eine Metapher veranschaulicht. 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 Wissensrevolution davon aus, daß maschinelles Lernen, ähnlich einem Fahrrad, immer auch der Mitwirkung eines menschlichen Benutzers bedarf.

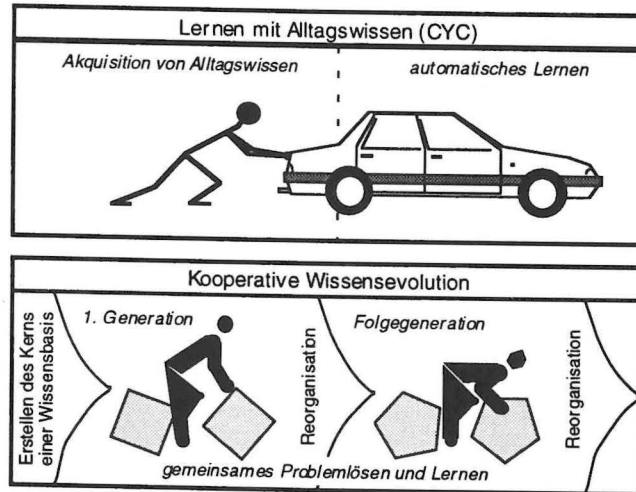


Abbildung 5: Vergleich von zwei verschiedenen Lernansätzen anhand einer Metapher

#### 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 Kreativitätsanlässe 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 *construction analyzer* analysiert und kritisiert den Inhalt

der aktuellen Problemlösung, indem er dem Benutzer Hinweise gibt sowie Einträge in die Argumentationskomponente macht. Ein *argumentation illustrator* hilft dem Benutzer, die Prinzipien im argumentativen Hypertext zu verstehen, indem Beispieldesigns oder -analysen dargestellt werden. Der *catalogue explorer* 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 Wissensrevolution 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 & Tschaitchian, 1992); formales und nicht-formales Wissen sind also nicht künstlich von einander 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 Wissensrevolution 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 Expliztheit bei der Manipulation von Wissen ermöglichen. Stefik 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." (Stefik, 1986, Seite 43)

### 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 möglich noch wünschenswert ist (vgl. Lesgold, 1993). Vielmehr können sich maschinelles und menschliches Lernen wechselseitig beeinflussen und ergänzen. Die Effektivität von maschinellen Lernverfahren könnte so auch durch die Meta-Kognition seiner Benutzer erhöht werden (vgl. Weinert, 1984). Dabei kommt es mit fortlaufender Zeit zu einer zunehmend besseren Anpassung der beiden Partner. Während Lenat vermutet, daß ähnlich einem Auto, das man über längere Zeit hin anschiebt, auch maschinelle Lernverfahren nach Zuführung von Alltagswissen schließlich "anspringen" werden (vgl. Abbildung 5), ist die kooperative Wissensrevolution mit einer Person vergleichbar, die sich zur Fortbewegung eines Fahrrades bedient, das nie von alleine fahren wird, aber von Anfang an ein nützliches Fortbewegungsmittel darstellt.

### 5. Kooperative Wissensrevolution in der Labormedizin

Der oben vorgestellte Ansatz der kooperativen Wissensrevolution ist grundlegend für das LAB-IDEAS Systems, das als intelligentes Dokumentations- und Kommunikationswerkzeug für medizinische Labors und behandelnde Ärzte konzipiert wurde. Es eignet sich also gut, um diesen Ansatz im Beispiel zu verdeutlichen.

Analysiert man das Verhältnis zwischen behandelnden Ärzten (Dienstleistungsnehmer) und medizinischen Labors (Dienstleistungsunternehmen), so stellt sich dieses als Partnerschaft zum Zweck einer möglichst guten Diagnosenstellung im Dienste des Patienten dar. Gleichzeitig ist dieses Verhältnis aber auch durch Informationsasymmetrien (gute Kenntnis des einzelnen Patienten vs. Spezialwissen im 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. Dadurch wird die Nutzung der vorhandenen Information im Hinblick auf die Ziele, Qualität und Effizienz der Diagnose, verbessert. 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 Patientenakte 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.

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

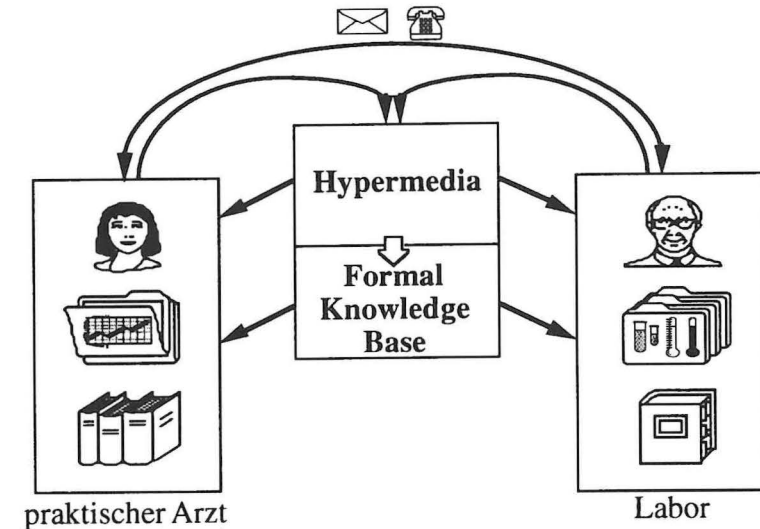


Abbildung 6: Kommunikationswege zwischen niedergelassenem Arzt und Labormediziner

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 kooperativen Wissensrevolution beruhenden Systems, bestehend aus Systemkonstruktion, Nutzungsphase mit Wachstum und schließlich Reorganisation, beschrieben. Nun konkretisieren wir diesen Ansatz für die hier behandelte Domäne, indem wir das allgemeine Kommunikationsmodell verfeinern.

## 5.1 Seeding

In einem ersten Schritt werden die Strukturen der Domäne und die Schritte der Problemlösung im Hinblick auf die Aufgabe der Diagnosenstellung spezifiziert. Es werden Ontologien für die Anwendung Labormedizin entwickelt, wobei darauf zu achten ist, daß aus praktischen Erwägungen heraus sowie aus Kostengründen möglichst schon vorhandene Begriffssysteme wiederbenutzt werden (Anforderung des *sharing* &

reuse). Der Gebrauch einheitlicher bzw. weit verbreiteter Wissensrepräsentations-sprachen unterstützt die praktischen Aspekte dieses Wiederverwendungsgedankens. Auf der Basis dieser Strukturen wird somit ein Modell der Aufgabe des zukünftigen Systems erstellt.

Anschließend wird aus bereits bekannten Fällen mittels der Beurteilung von Experten (hier: Labormediziner und niedergelassener Arzt) eine Fallhierarchie gebildet. Dazu stehen Werkzeuge wie das Wissensakquisitionswerkzeug CECoS (Case Experience Combination System) zur Verfügung, die aufgrund von Ähnlichkeitsurteilen eines Experten Problemklassenhierarchien konstruieren, die durch charakteristische Merkmale (Features) beschrieben werden. Neben der Gliederung der Problemklassen liefert CECoS auch die Abstraktionsregeln, nach denen aus der konkreten Fallbeschreibung die Merkmale der Problemklassen erkannt werden können (Reinartz & Schmalhofer, 1994).

Weiteres Wissen wird aus Fachbüchern und ähnlichen Quellen akquiriert. Dazu dienen Werkzeuge wie COKAM+ (Case Oriented Knowledge Acquisition Method from text; Schmidt, 1992), die durch die Verwendung von gelösten Fällen, also Diagnosen, die mit Hilfe der aus Fachtexten isolierten Wissensseinheiten erklärt werden, eine frühe Abschätzung der Relevanz und Vollständigkeit des erhobenen Wissens ermöglichen und die schrittweise Formalisierung der Wissensseinheiten unterstützen. Das Ergebnis ist eine strukturierte, hypertextartig vernetzte Wissensbasis bestehend aus natürlichsprachlichen und formalen Wissensseinheiten.

Aus dem anhand von Labor- und anderen patientenspezifischen Daten sowie Fachwissen erhobenen Domänenwissen über die Diagnose können nun unter Verwendung maschineller Lernverfahren aus konkreten Fällen allgemeine Prozedurschemata konstruiert werden, nach denen sich die in verschiedene Problemklassen fallenden Probleme lösen lassen. Dies geschieht mit Hilfe von Werkzeugen wie SP-GEN (Schmalhofer et al., 1991) oder PABS (Schmalhofer & Tschaitchian, 1993), die solche Prozedurschemata in Begriffswelten mit unterschiedlichem Detaillierungsgrad finden.

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.

## 5.2 Evolutionäres Wachstum

Das in der Konstruktionsphase erstellte initiale System kann nun als aktives Medium der Kommunikation zwischen Labormediziner und niedergelassenem Arzt bei der Interpretation der Laboranalysen genutzt werden.

Für den Laborarzt dient das System dabei hauptsächlich als Diskussionspartner bei der Entscheidungsfindung, weniger als automatisches Diagnosesystem. Relevante Daten über den einzelnen Patienten werden bereitgestellt, aber auch medizinisches Fachwissen, das bei der Beurteilung des aktuellen Falles hilfreich sein könnte. Darüber hinaus kann das System Vorschläge machen, die es aus dem Wissen über schon bear-

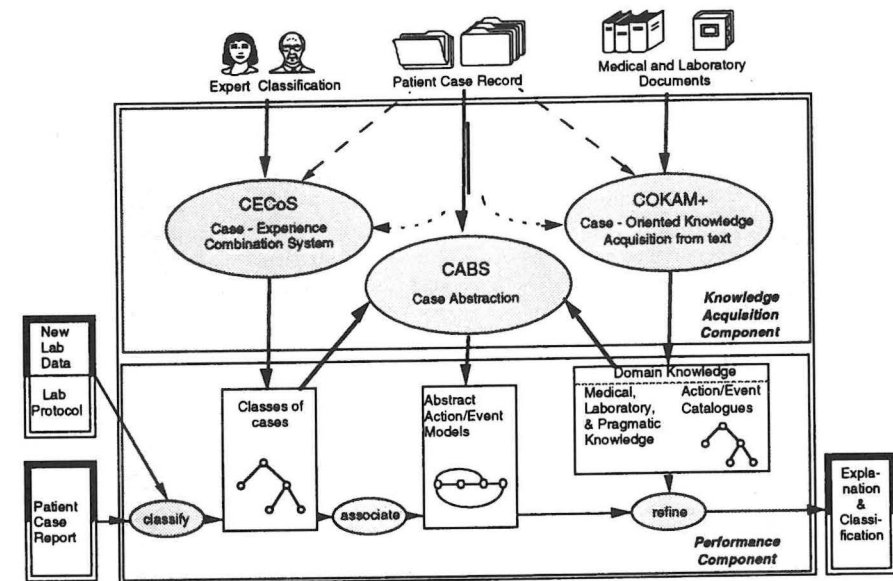


Abbildung 7: Integrierte Wissensakquisition (adaptiert nach Schmalhofer & Thoben, 1992)

beitete Fälle ableitet. Aus der Begründung, die das System für seine Vorschläge liefert, kann der Benutzer Ideen und Kreativitätsanstöße für den gerade zu bearbeitenden Fall ziehen. Indem der Labormediziner dann seine Entscheidung am 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 außerdem ständig neues Wissen zugeführt: Das System *wächst*. Dieses Wissen kann sowohl formaler als auch nicht-formaler Natur sein. Denkbar wäre hier neue Fachliteratur, die weitere Parameter bei der Diagnose bestimmter Krankheiten nennt,

neue Annahmen durch weiterentwickelte Analyseverfahren oder auch Wissen aus der mit der Zeit steigenden Erfahrung der Benutzer. Dieses Wissen wird dem System teils explizit während der Nutzung hinzugefügt, teils kann es aber auch automatisch vom System selbst gewonnen werden. Hier ist insbesondere der Einsatz maschineller Lernverfahren sinnvoll.

### 5.3 Reseeding

Das oben beschriebene Wachstum des Systems während der Nutzung erfordert in periodischen Abständen die Wartung und Reorganisation durch einen Wissensingenieur. Dabei kann zum Beispiel durch neuere Erkenntnisse überflüssig gewordenes Wissen aus dem System entfernt werden (aus Effizienzgründen, aber auch zur Erhaltung der Konsistenz) und vom Benutzer hinzugefügtes nicht-formales Wissen in die formalen Strukturen des Systems übersetzt werden. Möglicherweise können in dieser Phase auch Änderungen am Bearbeitungsmodell vorgenommen werden, die sich durch die Arbeit mit dem System oder durch externe Faktoren als sinnvoll erwiesen haben.

Insgesamt erhält man durch den Schritt des *reseeding* ein System der zweiten Generation, dessen Konstruktion weitestgehend den Belangen der Wiederverwendungsforderung Rechnung getragen hat. Mit dem neuen, verbesserten System kann sich nun eine weitere Nutzungsphase mit evolutionärem Wachstum anschließen. Die Felge des Fahrrades (Abbildung 5) ist wieder etwas runder geworden, die Fortbewegung, in unserem Fall die Diagnose durch Kommunikation zwischen Labor und niedergelassenem Arzt, kann leichter von staten gehen als zuvor.

## 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 relativ 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 mathematisiert sind. Ein Beispiel für ein solches System ist XCON/R1, das als Eingabe eine Liste von Rechnerkomponenten und eine gewünschte Rechnerfunktionalität bekommt, diese auf Vollständigkeit und Konsistenz prüft und dann die Komponenten in einem Gehäuse räumlich anordnet.

Wissen ist bei diesen Prototypen in Form von Regeln gespeichert. Problemlöseverhalten ergibt sich nur aus dem Zusammenspiel von Regelbasis und Inferenzmaschine, wird jedoch nicht explizit repräsentiert. Dies macht eine Anpassung an neue Gegebenheiten äußerst schwierig. Desweiteren stellen solche Systeme mit ihrem relativ festen Ein-/ Ausgabeschema nur ein unzureichendes Modell von menschlichen Experten dar. Gerade in sehr komplexen Domänen wie z.B. der Medizin ist Problemlösung durch Experten ein stark dialogorientierter Vorgang, bei dem oft bereits die exakte Problemformulierung zu Beginn kaum möglich ist. Vielmehr stellt es sich erst im Laufe der Arbeit heraus, wo das Problem genau liegt, welche Wissensquellen benötigt werden und welche Verfahren zur Lösung eingesetzt werden könnten. Bei der kooperativen Wissensrevolution wird dazu der oben genannte Dialogaspekt stärker betont. Auch die Problematik der Wissenserhebung, eine der Hauptschwierigkeiten der rein modellbasierten Systeme, wird hier schon im Ansatz behandelt. Da der Experte gleichzeitig auch Nutzer ist, gibt es keine Phase, nach der die Wissenserhebung vollständig abgeschlossen sein muß, sie kann vielmehr dauernd während der Arbeit mit dem System stattfinden. Gleichwohl ist ein solches System nicht geeignet, an Stelle eines Experten zu arbeiten (was in manchen Domänen durchaus erstrebenswert sein könnte), sondern diesen zu unterstützen. Dies ist insbesondere 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 Wissensniveaus, sollte auch noch weitere Forschungsarbeit geleistet werden.

Abschließend wollen wir zumindest punktuell die vorgestellten Verfahren mit Methoden bei der Entwicklung von Anwendungsprogrammen vergleichen<sup>3</sup>. Die modellbasierte Entwicklung von Expertensystemen und die dabei angestrebte Tiefenmodellierung besitzt viele Ähnlichkeiten mit der strukturierten Program-

<sup>3</sup> Die hier angestellten Überlegungen beruhen auf Anregungen von Wolfgang Dzida, wofür wir uns hiermit bedanken.

mierung beim Entwurf konventioneller Software (vgl. z. B. Jordan und Urban, 1978). Bei der Expertensystementwicklung wird aus einem konzeptuellen Modell einer Wissensdomäne das Design-Modell für das wissensbasierte System (vgl. Abb. 4) erstellt. Analog dazu beruht die Entwicklung von Anwendungsprogrammen auf der Erstellung von Datenmodellen (vgl. Artikel von Albers und Petzold, in diesem Buch). Dabei wird darauf Wert gelegt, daß sich das Datenmodell als zeitlich stabil erweist, eine Annahme, die auch den konzeptuellen Modellen von wissensbasierten Systemen zugrunde liegt.

Der Ansatz der kooperativen Wissensrevolution weist Ähnlichkeiten mit der semantischen Modellierung bei der objektorientierten Analyse auf. Sowohl die Wissensbasis als auch das semantische Modell (Klassenbibliothek) wird als Ergebnis eines inkrementellen Anpassungsprozesses gesehen, in dem die Nutzungserfahrung zur Anpassung der Modelle beitragen. Für objekt-orientierte Systeme wird behauptet, daß erst durch dieses software-technische Konstruktionsparadigma die evolutionäre Entwicklung von Anwendungsprogrammen lohnend wird, da der dynamische Ablauf des Programms nicht prinzipiell festgelegt werden braucht. Vielmehr ergibt er sich erst aus der vom Benutzer ausgeübten "external control", die bestimmt, welche Objekte des Systems miteinander "kommunizieren". Insofern können Nutzungserfahrungen der Benutzer Anlässe zur Erweiterung, Verfeinerung und Veränderung des Systems liefern.

Diese Überlegungen lassen es durchaus lohnend erscheinen, einen detaillierten Vergleich zwischen so unterschiedlicher Software wie Expertensystemen und Anwendungsprogrammen anzustellen.

## 7. Literatur

- Aschenbrenner, K.M., Gertzen, H., Kühn, O., Schmalhofer, F. und 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.
- Birk, A., Tschaitchian, B., Schmalhofer, F., Thüning, M., and Gertzen, H. (1994). Architecture and User-Interface of the IDEAS Intelligent Documentation System. In: W. Schuler, J. Hannemann, (Eds.): *Methodological Issues on the Design of Hypertext-based User Interfaces* (im Druck).
- Breuker, J.A., Wielinga, W.J. (1989). Model driven knowledge acquisition. In: P. Guida, G. Tasso, G. (Eds.): *Topics in the design of expert systems* (pp. 265-296). Amsterdam: North-Holland/Elsevier.
- Boose, J. H., 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, 215-251.
- Clancey, W.J. (1985). Heuristic Classification. *Artificial Intelligence* 27, 289-350.
- Clancey, W.J. (1991). Change and confluence in cognitive science. In: K. VanLehn (Ed.): *Architectures for intelligence. The Twenty-second Carnegie Mellon Symposium on Cognition* (pp. 357-423). Hillsdale, NJ: Lawrence Erlbaum.
- Fensel, D., Angele, J., Landes, D., and Studer, R. (1993). Giving structured analysis techniques a formal and operational semantics with KARL. In: H. Züllighoven (Ed.): *Proc. of Requirements Engineering RE '93 - Prototyping* (pp. 267-285). Stuttgart: Teubner.
- Fischer, G., McCall, R., Ostwald, J., Reeves, B., and Shipman, F. (1994). Seeding, evolutionary growth and reseeding: Supporting the incremental development of design environments. *CHI '94 Conference Proceedings, April, 24-28, Boston, MA* (pp. 292-298). 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, 311-332.
- Jordan, W., Urban, H. (1978). *Strukturierte Programmierung*. Berlin: Springer.
- Karbach, W., Linster, M. (1990). *Wissensakquisition für Expertensysteme: Techniken, Modelle und Softwarewerkzeuge*. München: 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, 7-28.
- Lenat, D.B. (1989). When Will Machines Learn?. *Machine Learning*, 4(3/4), 255-258.
- Lenat, D., Prakash, M., Shepherd, M. (1986). CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI Magazine*, 65-85.
- Lesgold, A. M. (1993). Beyond a commodity view of knowledge in instruction. In: G. Strube, K.F. Wender, (Eds.): *The cognitive psychology of knowledge* (pp. 425-433). Amsterdam: North-Holland/Elsevier.
- Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., & Swartout, W.R. (1991). Enabling Technology for Knowledge sharing. *AI Magazine*, 36-55.
- Reinartz, T., Schmalhofer, F. (1994). An integration of knowledge acquisition techniques and EBL for real-world production planning; erscheint in *Knowledge Acquisition*.
- Schank, R.C. (1991). Where's the AI? *AI Magazine*, 38-49.
- Schmalhofer, F., Bergmann, R., Kühn, O., and Schmidt, G. (1991). Using an integrated knowledge acquisition method to prepare sophisticated expert plans for the reuse in novel situations. *Proceedings of the 15th German Workshop on Artificial Intelligence*, 62-71.
- Schmalhofer, F., Reinartz, T., and Tschaitchian, B. (1992). Intelligent documentation as a catalyst for developing cooperative knowledge-based systems. In: T. Wetter, K.-D. Althoff, J. Boose, B.R. Gaines, M. Linster, and F. Schmalhofer (Eds.): *Current developments in knowledge acquisition - EKAW '92* (pp. 406-424). Berlin: Springer.
- Schmalhofer, F., Reinartz, T., and Tschaitchian, B. (1995). A unified approach to learning for complex real world domains. *Applied Artificial Intelligence, Vol.9*. (im Druck).

- Schmalhofer, F., Tschaitchian, 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: T. Christaller, H.-W. Hein und M.M. Richter (Hrsg.): *Künstliche Intelligenz: Theoretische Grundlagen und Anwendungsfelder* (pp. 245-291). Berlin: Springer.
- Schmidt, G. (1992). Knowledge acquisition from text in a complex domain. *Proceedings of the Fifth International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems*, 529-538.
- Shortliffe, E.H. (1976). *Computer-Based Medical Consultations: MYCIN*. Amsterdam: North Holland/Elsevier.
- Steels, L. (1990). Components of Expertise. *AI Magazine* 11(2), 28-49.
- Stefik, M. (1986). The Next Knowledge Medium. *AI Magazine* 7(1), 34-46.
- Weinert, F.E. (1984). Metakognition und Motivation als Determinanten der Lerneffektivität: Einführung und Überblick. In: F.E. Weinert and R.H. Kluwe, (Hrsg.): *Metakognition, Motivation und Lernen* (pp. 9-21). Stuttgart: Kohlhammer.
- Wilkins, D.C. (1991). A framework for knowledge acquisition and machine learning. *Proceedings of the 6th Banff Knowledge Acquisition for Knowledge-based Systems Workshop*, 37.1.-37.14.

## Handlungsorientierte Wissensaufbereitung zur Ermittlung des Unterstützungsbedarfs

Nina Großmann

### 1. Einleitung

Schwerpunkt dieses Beitrages ist der aufgabenbezogene Gestaltungsprozeß. Es wird ein Weg aufgezeigt wie der Unterstützungsbedarf ermittelt werden kann und welcher Art zu konzipierende Unterstützungsinstrumente sein können.

Es wird dargestellt wie eine Projektgruppe zusammengesetzt werden soll und welche Aufgaben der Projektgruppe im Gestaltungsprozeß zugeordnet werden müssen.

Ein Modellvergleich zwischen verschiedenen Konzepten des Handlungswissen wurde hinsichtlich ihres Nutzens bei der Erhebung und Darstellung von Wissen in drei Praxisbereichen durchgeführt. Anhand einer Untersuchung im chemischen Labor werden die Gestaltung der Arbeitsaufgabe beschrieben und die Ergebnisse zum Modellvergleich erläutert.

### 2. Unterstützungsbedarf im Arbeitsprozeß

#### 2.1 Der Mensch im Mittelpunkt des Gestaltungsprozesses

Die Unterstützung der Arbeitstätigkeit durch Werkzeuge ist ein wesentliches Merkmal der gesellschaftlich-geschichtlichen Entwicklung. Historisch betrachtet vollzog sich die Entwicklung der Arbeitstätigkeit auch dadurch, daß menschliche Arbeit durch immer kompliziertere Werkzeuge unterstützt und somit auch verändert wurde. Dabei gilt es folgende Fragen zu klären:

- Mit welchem Ziel wird Arbeit unterstützt?
- Welche Formen und Prinzipien sind für die Veränderung der menschlichen Arbeit akzeptabel bzw. human?

Zur Beantwortung dieser Fragen sei hier das Prinzip vorangestellt: "Die menschlichen Stärken schützen und fördern!" (Luczak et al., 1989). Dieses Prinzip fordert einerseits, menschliche Stärken zu erkennen und andererseits verlangt es, diese in den Mittelpunkt des Entwicklungsgeschehens zu stellen. Damit werden die Besonderheiten des Menschen gegenüber anderen (z.B. wirtschaftlichen, technischen) Systemen hervorgehoben (Luczak et al., 1989).