



Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Document
D-94-03

**Maschinelles Lernen:
Eine kognitionswissenschaftliche
Betrachtung**

Franz Schmalhofer

März 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

**Maschinelles Lernen:
Eine kognitionswissenschaftliche Betrachtung**

Franz Schmalhofer

DFKI-D-94-03

**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 a licence with payment of fee to Deutsches Forschungszentrum für Künstliche Intelligenz.

ISSN 0946-0098

MASCHINELLES LERNEN:
EINE KOGNITIONSWISSENSCHAFTLICHE BETRACHTUNG

Franz Schmalhofer
Deutsches Forschungszentrum
für Künstliche Intelligenz (DFKI)
Postfach 2080
D-67608 Kaiserslautern
Germany

Inhaltsverzeichnis

1. Einleitung	1
2. Exemplarische Darstellung von Lernverfahren.....	2
2.1 Lernen aus Fällen	2
2.1.1 Fallbasiertes Schließen in der Rechtsprechung	3
2.1.2 Das JUDGE-System	3
2.2 Lernen durch Induktion	6
2.2.1 Induktives Schließen.....	7
2.2.2 Das INDUCE-System	8
2.2.3 Lernsysteme als Grundlage psychologischer Modelle.....	10
2.3 Lernen aus Texten	11
2.3.1 Das GENESIS-System	12
2.3.2 Das C-I-Modell	16
3. Eine vereinheitlichende Betrachtung von Lernsystemen	19
3.1 Betrachtung auf der Wissensebene	21
3.1.1 Parameterausprägungen	25
Woraus wird was gelernt? - Zweck.....	26
Womit wird gelernt? - Wissen.....	28
Wie wird gelernt? - Fertigkeiten.....	28
Wo wird gelernt? - Einsatzgebiet.....	29
3.1.2 Verhaltensbeschreibung	30
3.2 Fall-, such- und verstehensorientierte Lernsysteme	31
3.2.1 Fallorientierte Systeme.....	32
CHEF	32
PRODIGY	32
3.2.2 Suchorientierte Systeme	34
SOAR	35
Die Versionenraummethode (VR).....	35
ACT-R: Wissenskompilation und Analogien.....	36

3.2.3 Verstehensorientierte Systeme	36
CASCADE.....	37
KIWi / PABS.....	38
Theorie-Revision (TR).....	39
4. Wissensakquisition.....	39
4.1 Lernen mit Alltagswissen: Der CYC-Ansatz.....	40
4.2 Gemeinsames Lernen.....	41
4.2.1 Lehrlingslernen	41
4.2.2 Kooperative Wissensrevolution	41
4.3 Vergleichende Betrachtung	42
5. Diskussion.....	44
6. Literatur	45

1. EINLEITUNG

Auf dem Gebiet des maschinellen Lernens wurden vor allem in den letzten zehn Jahren entscheidende Fortschritte erzielt, die sowohl für das Gebiet der kognitiven Psychologie als auch bei der Erstellung von Expertensystemen erhebliche Auswirkungen zeigen. Die Idee des maschinellen Lernens selbst ist jedoch bereits in der Anfangsphase der Forschungen zur Künstlichen Intelligenz vorhanden. So hat Alan Turing schon in den vierziger Jahren postuliert, daß Computersysteme, die als digitale informationsverarbeitende Maschinen anzusehen sind, lernfähig sein müßten, um intelligentes Verhalten zeigen zu können. Lange vor dem praktischen Einsatz von Computersystemen hat er dazu theoretische Überlegungen angestellt (Turing, 1947). In den fünfziger Jahren entstanden dann Computerprogramme, die in klar definierten Mikrowelten, (z.B. in einfachen Spielen wie Checkers) ihr Verhalten durch Lernen verbessern konnten (Samuel, 1959).

Nachdem es in den nächsten Jahrzehnten wegen der zu hoch gesteckten Erwartungen und den darauffolgenden Enttäuschungen auf dem Gebiet des maschinellen Lernens etwas ruhiger geworden war, werden seit Anfang der achtziger Jahre sowohl in theoretischer als auch in praktischer Hinsicht wieder vielfältige Fortschritte erzielt. So können nun auch Lernprogramme erstellt werden, die den Einsatz in sehr komplexen Situationen des realen Lebens ermöglichen. Dabei zeigt sich allerdings auch, daß unser Verständnis von maschinellem Lernen dem Wandel unterliegt.

Bei der Entwicklung von Lernprogrammen wurden vielfach Erkenntnisse der Psychologie genutzt. Es ist daher vielleicht nicht besonders überraschend, daß diese Programme ihrerseits nun auch wiederum als Modelle des menschlichen Lernens betrachtet werden können. In manchen Fällen wurden maschinelle Lernverfahren bereits mit Verhaltensdaten verglichen. Dadurch kann eine Prüfung der psychologischen Validität dieser Computermodelle erfolgen. Solche Computermodelle werden dann als kognitive Modellierungen des menschlichen Lernens bezeichnet (Opwis, 1992; Schmalhofer & Wetter, 1988; Strube et al., 1993).

Mit diesem Beitrag soll nun ein erster Einblick in das sehr aktive Forschungsgebiet des maschinellen Lernens gegeben werden. Dabei werden zunächst verschiedene Lernverfahren exemplarisch dargestellt. Am Beispiel des JUDGE-Systems (Bain, 1986) wird veranschaulicht, wie Erfahrungen über komplexe Zusammenhänge der Rechtsprechung durch fallbasiertes Schließen aufbereitet und dann wiederverwendet werden können. Die Funktionsweise von induktiven Lernverfahren wird anhand des INDUCE-Systems (Michalski, 1983) erläutert, das bereits als psychologisches Lernmodell evaluiert wurde. Zum Lernen aus Texten werden zwei Systeme vorgestellt, nämlich das GENESIS-System (Mooney & DeJong, 1985) und das C-I-Modell (Kintsch, 1988; Mross & Roberts, 1992).

Im dritten Abschnitt wird eine Klassifikation von Lernsystemen auf der Wissensebene (vgl. Newell, 1982) erarbeitet, und es werden daraufhin weitere Lernsysteme vorgestellt, die für die einzelnen Klassen von Lernsystemen (mehr oder weniger) prototypisch und/oder für die Modellierung menschlicher Lernprozesse bedeutsam sind. In dem darauffolgenden Abschnitt wollen wir dann maschinelles Lernen in komplexen Bereichen der realen Welt diskutieren. Im Gegensatz zu der Modellierung von Lernprozessen in gut verstandenen und klar beschriebenen Laborsituationen (Mikrowelten) wird hier nun der praktische (und auch kommerzielle) Nutzen von Lernmodellen thematisiert. In der abschließenden Diskussion wird die Beziehung von maschinellem Lernen und Künstlicher Intelligenz zur psychologischen Lernforschung und der Kognitiven Psychologie erläutert.

2. EXEMPLARISCHE DARSTELLUNG VON LERNVERFAHREN

In diesem Abschnitt werden anhand von ausgewählten Systemen drei Lernverfahren vorgestellt, die unterschiedliche Leistungen erbringen: Mit dem JUDGE-System werden Erfahrungen in einer objektiven Weise als Fälle auf Rechnern abgelegt, um sie später wiederverwenden zu können. Mit INDUCE werden aus einer Menge von klassifizierten Beispielen (z. B. positive und negative Beispiele) allgemeingültige Regeln gefunden, mit Hilfe derer alle Beispiele einer Klasse von den Beispielen der anderen Klassen unterschieden werden können. Die Systeme GENESIS und C-I beschreiben schließlich, wie aus Texten ein allgemeingültiges Schema und Sachkenntnisse erworben werden können.

2.1 Lernen aus Fällen

Unter dem Thema Lernen aus Fällen und fallbasiertes Schließen werden Arbeiten zusammengefaßt, die Anfang der 80er Jahre in der Arbeitsgruppe von Roger Schank begonnen wurden und in den letzten Jahren zunehmend auch in Europa und Deutschland an Bedeutung gewonnen haben (Aamodt, 1991; Althoff et al., 1992; Hesse, 1991; Janetzko & Strube, 1992; Kolodner, 1993; Schmalhofer & Thoben, 1992; Weber et al., 1988; Weber, 1993). Diese Lernverfahren sollen nun am Beispiel der Rechtsprechung mit Hilfe des JUDGE-Systems vorgestellt werden. JUDGE (Bain, 1986; Riesbeck & Schank, 1989) ist nämlich ein typisches fallbasiertes System, das aufgrund einer Modellierung der Informationsverarbeitungsprozesse bei der amerikanischen Rechtsprechung entwickelt wurde.

2.1.1 Fallbasiertes Schließen in der Rechtsprechung

Die Entwicklung von JUDGE begann mit einer umfangreichen Wissensakquisition bei Rechtsexperten. In dieser Phase wurden mehreren amerikanischen Richtern sexuelle Mißhandlungen von Kindern geschildert. Jeder Fall wurde nur sehr dürftig beschrieben, so daß die Richter durch Nachfragen die für die Rechtsprechung relevanten Informationen erst erkunden mußten. Dabei wurde festgestellt, daß amerikanische Richter vergleichsweise wenig Paragraphen der Strafgesetzbücher verwenden, sondern vielmehr aufgrund ihres (reichhaltigen) Erfahrungsschatzes urteilen. Sie entscheiden dabei oft aufgrund besonders typischer Merkmale. Von dem dargestellten Tatbestand werden so vor allem die Vorstrafen des Täters, mögliche Erklärungen für sein kriminelles Verhalten, und die Größe des Schadens beim Opfer berücksichtigt. So wird beispielsweise eine sexuelle Mißhandlung für schwerwiegender erachtet als eine einfache Körperverletzung. Ferner wird ein Urteil nicht unmittelbar gefällt, sondern es werden eine Reihe möglicher Urteile aufgezählt und die an sie geknüpften Bedingungen genannt. Dann wird geprüft, welche dieser Bedingungen durch weitere Zeugenaussagen gestützt wird, und anschließend ein endgültiges Urteil gefällt. Das Verhalten eines amerikanischen Richters kann so als fallbasiertes Schließen modelliert werden.

Fallbasiertes Schließen (engl.: Case-Based Reasoning) beinhaltet im wesentlichen die folgenden Schritte: erst sucht der Urteilssuchende (oder allgemeiner, der Problemlöser) in seinem Gedächtnis nach einem ähnlichen Fall, den er in der Vergangenheit bearbeitet hat. Seine Erinnerung ist dabei nicht auf den Fall alleine beschränkt, sondern er kann sich auch an die

früher verwendeten Strategien erinnern. Dann paßt er seine Strategie an die Anforderungen des neuen Problems an und merkt sich das so entstandene Paar, das aus der Problemstellung und der Problemlösung besteht (u.U. mit den aufgetretenen Unterschieden zu dem früheren Fall).

2.1.2 Das JUDGE-System

Das JUDGE-System ist in der Lage, zu einem vorgelegten Tatbestand einer kriminellen Handlung aufgrund der Erfahrungen der bisherigen Rechtsprechung ein Urteil über Schuld und Unschuld des Täters, den es zunächst vom Opfer zu unterscheiden gilt, zu fällen, sowie ein angemessenes Strafmaß zu bestimmen.

Ein typischer Fall, der JUDGE vorgegeben wird, ist beispielsweise FALL1:

"First, Randy struck Chuck with his fists several times. Chuck was only slightly hurt. Next, Chuck struck Randy with his fists several times. Randy was only slightly hurt. Then, Randy slashed at Chuck with a knife one time. Chuck's skin was cut. Next, Chuck slashed at Randy with a knife one time. Randy's skin was cut. Finally, Randy stabbed Chuck with a knife several times. Chuck died."

JUDGE¹ versucht nun zunächst, die Motivationen der Täter für die einzelnen Handlungen zu erklären, versucht daraufhin, sich an einen ähnlichen Fall zu erinnern, und paßt schließlich das erinnerte Strafmaß an den vorgegebenen aktuellen Fall an.

Beschreibung der Verarbeitungsphasen: Das System JUDGE realisiert ein fallbasiertes Modell der Rechtsprechung. Es beginnt mit einer Menge einfacher Strategien, um Urteile zu fällen. Aber bereits nach wenigen Fällen erinnert sich JUDGE an die bereits gelösten Fälle und versucht diese wiederzuverwenden. Das JUDGE System hat fünf Phasen: 1. die Interpretationsphase, 2. die Erinnerungsphase, 3. die Vergleichsphase, 4. die Anwendungs- und Modifikationsphase und schließlich 5. die Generalisierungsphase.

In der *Interpretationsphase* inferiert JUDGE die Motivationen der handelnden Personen, die in der Fallbeschreibung vorkommen. Die Eingabe in das Programm besteht nur aus den Aktionen, wie z.B. eine Person schlägt eine andere, und den daraus entstandenen direkten und indirekten Folgen, z.B. die andere Person wird niedergestreckt oder verliert ihr Bewußtsein.

JUDGE kennt drei Arten von Interpretationen: *initiale Interpretationen* motivieren den Beginn eines Kampfes, *intentionale Interpretationen* beurteilen die darauf folgenden Aktionen (die relative Kraft der Handlung als Antwort auf die Aktion des anderen (stärker, ähnlich stark, oder weniger stark) und den eigentlichen Grund für die Aktion (Selbstverteidigung oder Rache)), und schließlich die *Ergebnis-Interpretation*, die das Resultat einer Aktion relativ zu der beabsichtigten Konsequenz bewertet (stärker als beabsichtigt, wie beabsichtigt oder schwächer als beabsichtigt). Dabei bleibt JUDGE möglichst unvoreingenommen und objektiv.

Bei der initialen Interpretation des oben beschriebenen Falles (FALL1) wird zunächst ein einzelnes Ereignis (bestehend aus einer Aktion und deren Konsequenz) betrachtet. "First, Randy struck Chuck with his fists several times. Chuck was only slightly hurt." wird so als erstes Einzeleignis (engl. EVENT) oder Episode bezeichnet. JUDGE sucht und findet dann

eine EPISODE in seinem Gedächtnis, die mit der dargestellten Beschreibung weitgehend übereinstimmt. In dem speziellen Beispiel wird der Beginn des Kampfes dadurch also als unprovokierte Verletzung des Gegenübers eingestuft.

In der folgenden *Erinnerungsphase* sucht JUDGE nun mit Hilfe der bereits gefundenen Episode nach einem ähnlichen Fall in seinem Gedächtnis und findet dabei den folgenden als FALL0 bezeichneten Eintrag:

"First, Ted slashed at Al with a knife one time. Al's skin was slightly cut. Next, Al slashed at Ted with a knife one time. Ted's skin was slightly cut. Finally, Ted stabbed Al with a knife several times. Al died."

Zum Auffinden dieses Falles mußte zunächst entschieden werden, welches die wichtigsten Merkmale der Situationsbeschreibung sind, die bei einer Bewertung der Ähnlichkeit die vorherrschende Rolle einnehmen. Zu diesem Zweck ist das System mit einer Reihe von Kriterien ausgestattet. Das wichtigste Merkmal ist etwa, ob der Täter als Erwachsener oder als Jugendlicher eingestuft wird, denn demzufolge ist entweder das normale Strafgesetz oder das Jugendstrafgesetz maßgeblich. Darüber hinaus ist entscheidend, wer den Kampf begonnen hat (der Täter oder das Opfer).

Der im Gedächtnis abgelegte Fall kann dem neu zu beurteilenden Fall mehr oder weniger ähnlich sein. Es muß daher ein Mindestmaß an Ähnlichkeit gewährleistet werden, bevor ein bereits abgelegter Fall zur Beurteilung des neuen Falles herangezogen wird. Findet JUDGE keinen hinreichend ähnlichen Fall, so muß das System versuchen, ein Urteil aufgrund einer der vorhandenen Strategien zu fällen, die auch bei einer leeren Fallbasis angewendet werden. Stehen mehrere geeignete Fälle zur Verfügung, so muß das System aufgrund eines Präferenzkriteriums einen auswählen. Ein solches Kriterium ist z.B. die Aktualität eines Falles oder eine besonders hohe Übereinstimmung in den entscheidenden Merkmalen.

In der *Vergleichsphase* ermittelt JUDGE nun die Unterschiede zwischen dem aktuellen Fall und dem Fall aus der Fallbasis. Wenn sich keinerlei Unterschiede fänden, würde das Urteil aus dem alten Fall unverändert übernommen. Im allgemeinen sind aber Unterschiede gegeben, und es werden Modifikationsstrategien eingesetzt, um das Urteil des alten Falles angemessen an den neuen Fall anzupassen.

Bei dem Vergleich von FALL1 mit FALL0 stellt sich bezüglich des Schadens am Opfer kein Unterschied heraus:

IN BOTH CRIMES, THE VICTIM WAS KILLED. NOT ONLY WERE BOTH OF THESE OUTCOMES THE DIRECT INTENTIONS, BUT THE ACTORS INTENDED AND CAUSED THE SAME AMOUNT OF HARM.

Grundsätzlich vergleicht JUDGE zwei Ereignisse solange, bis merkliche Unterschiede gefunden sind, oder eines der beiden (oder beide) Ereignisse vollständig geprüft ist.²

In der *Anwendungs- und Modifikationsphase* wendet JUDGE mehrere Modifikationsregeln auf das Ergebnis der Vergleichsphase an. Das alte Urteil wird dadurch so verändert, daß daraus das

² Es handelt sich hierbei im Prinzip um einen kriteriumsabhängigen Vergleichsprozess, der auch bei vielen psychologischen Prozeßmodellen eine wichtige Rolle spielt (z.B. Schmalhofer, Albert, Aschenbrenner, Gertzen, 1986).

Urteil für den neuen Fall entsteht. Wenn die Fälle beispielsweise nicht identisch, aber doch sehr ähnlich sind, so wird das ausgesprochene Urteil in Bezug auf das Mindest- und Höchstmaß der Strafdauer proportional skaliert. Sonst wird je nach Grad der Ähnlichkeit beispielsweise eine Regel der Art "IF comparison = AGREED or SIMILAR-RESULT and the crimes are not immediately different THEN increase or decrease the old sentence by 25% as appropriate." angewendet.

In der *Generalisierungsphase* schließlich formt JUDGE aus mehreren ähnlichen Fällen allgemeine Regeln für die Rechtsprechung. Bei einer ersten Generalisierung, die auf zwei Fällen beruht, wird das Urteil einfach als Konklusion der Regel übernommen, und die gemeinsamen Merkmale der beiden Fälle dienen als Prämissen. Eine solche Generalisierung von Einzelbeobachtungen zu einer allgemeinen Regel wird als induktiver Schritt bezeichnet. Induktive Lernprozesse werden im folgenden Abschnitt anhand des INDUCE Systems noch ausführlich dargestellt.

Bei der Erstimplementierung von JUDGE konnte verständlicherweise nicht sämtliches Wissen, das einerseits zum bereichsunabhängigen Schließen und andererseits zur Rechtsprechung selbst benötigt wird, vollständig repräsentiert werden. JUDGE wurde so für Delikte wie Mord und Totschlag implementiert und mit 55 Fällen von Kampfeshandlungen geprüft. Dabei berücksichtigte JUDGE aber nur die relative Abscheulichkeit der Tat, die z.B. am persönlichen Schaden und an der mangelnden moralischen Handlungsrechtfertigung gemessen wird.

2.2 Lernen durch Induktion

Ein wichtiger Lernvorgang besteht darin, aus einer begrenzten Anzahl von speziellen Vorgaben das allgemein Gültige zu erkennen. Wenn bei mehreren vorgegebenen Beispielen die unwesentlichen Merkmale außer acht gelassen, und die allgemeingültigen Regeln erkannt werden, so wird dadurch ganz sicherlich neues Wissen erworben und nicht nur eine andersartige aber doch gleichwertige Darstellung von bereits vorhandenem Wissen erzeugt. Im Gegensatz zu Lernvorgängen die nur die Effizienz der Wissensnutzung steigern (Etzioni, 1992), findet beim induktiven Lernen ein Wissenserwerb im engeren Sinne des Wortes statt (Dietterich, 1986). Es gibt eine Vielzahl von maschinellen Lernverfahren, bei denen induktiv gelernt wird. Hier sollen nun zuerst einige Probleme des induktiven Lernens dargestellt werden

um dann anschließend das System INDUCE (Michalski, 1983) vorzustellen. Dieses System ist auch deshalb besonders interessant, weil es von Medin, Wattenmaker & Michalski (1987) als psychologisches Prozeßmodell aufgefaßt und durch psychologische Experimente evaluiert wurde.

2.2.1 Induktives Schließen

Die Fertigkeit des induktiven Schließens wird beispielsweise in Intelligenztests geprüft. Wenn bei der Vervollständigung von Zahlenreihen ein Proband das nächste Element der Sequenz "1,2,4" bestimmen soll, so erfordert dies eine induktive Generalisierung (vgl. Induktionsproblem, Waldmann, 1994). Dabei könnte als nächstes Element beispielsweise "5" genannt werden, nämlich dann, wenn als allgemeine Regel "ansteigende Sequenz von ganzzahligen Zahlen, die nicht durch 3 teilbar sind", induziert wurde. Alternativ dazu könnte aber das nächste Element auch 8 lauten (allgemeine Regel: $a_{n+1} = 2 \cdot a_n$; $a_1 = 1$) oder 14 (allgemeine Regel: $a_{n+1} = a_n^2 - a_n + 2$, $a_1 = 1$). Sogar "A" könnte als nächstes Element in der Sequenz genannt werden, denn es wurde ja keineswegs gesagt, daß es sich um Zahlen aus dem dezimalen System handelt, und "1,2,4,A,B,D" ist ohne weitergehende Einschränkungen genauso eine zulässige Sequenz wie die zuvor genannten Sequenzen oder beliebig andere.

Zum induktiven Lernen werden daher zusätzliche Restriktionen oder Randbedingungen benötigt, so daß von den (unbegrenzt) vielen Generalisierungen nur eine einzige entsteht (oder wenigstens nur begrenzt viele). Man bezeichnet diese Randbedingungen als Lernbevorzugungen (learning bias). Diese Lernbevorzugungen werden bei induktiven Lernverfahren durch die zur Darstellung der Beispiele und Hypothesen gewählten Repräsentationsformen³ (representational bias oder restricted hypothesis space bias) und durch die Verfahren zur Erzeugung und Prüfung von Hypothesen (search bias oder preference bias) bestimmt (Rendell, 1986).

Nun erscheinen manche Lernbevorzugungen sicherlich natürlicher und zweckmäßiger als andere. Nach dem Prinzip von Occam (Einfachheitskriterium) kann man beispielsweise fordern, daß Lernbevorzugungen, die einfachere Regeln erzeugen, gegenüber denjenigen vorzuziehen sind, die komplexe Regeln erzeugen (vgl. Blumer et al., 1987). In ähnlicher Weise können Lernverfahren bevorzugt konjunktive Verknüpfungen erzeugen und disjunktive Verknüpfungen vermeiden.⁴

Generalisierungen können auch so gebildet werden, daß man aufgrund eines Merkmals möglichst gut die Klassenzugehörigkeit des Objekts (d.h. möglichst große cue validity) oder aufgrund der Klassenzugehörigkeit möglichst gut die Merkmale des Objekts (möglichst große category validity) bestimmen kann. Diese Kriterien spielen bei der Kategorienbildung eine wichtige Rolle und werden von Anderson (1990) und Eckes (dieser Band) weitergehend erläutert. Des weiteren kann die Verwendung von Negationen erlaubt oder vermieden werden

³ Einen Überblick über verschiedene Wissensrepräsentationsformen und die damit verbundenen Methoden und Ziele geben Opwis & Lüer (1994) sowie Tack (1987).

⁴ Beispielsweise würden deshalb die Veröffentlichungen eines Autors bevorzugt beschrieben als "Objekte, die den Namen des Autors tragen und aus Text bestehen und allgemein zugänglich sind", und nicht als "Zeitschriftenartikel oder Beiträge zu herausgegebenen Büchern oder selbstverfaßte Bücher".

(z. B. der Mensch wird definiert als *nicht* gefiederter Zweibeiner versus der Mensch als aufrecht gehender Säuger). Schließlich lassen sich solche und andere Lernbevorzugungen durch das angestrebte und oft domänenspezifisch beschriebene Lernergebnis (Keil, 1981) oder durch vermutlich allgemeingültige und domänenunabhängige Eigenschaften des Lernprozesses (Mitchell, 1982) charakterisieren.

2.2.2 Das INDUCE-System

Das INDUCE System kann aus gegebenen (positiven und negativen) Beispielen⁵ allgemeine Beschreibungen eines Zielkonzepts erzeugen, das durch die Klasse der positiven Beispiele bestimmt ist. Die erzeugten Beschreibungen können dann zur Klassifikation weiterer Beispiele verwendet werden (vgl. Bruner, Goodnow & Austin, 1956; Haygood & Bourne, 1965). Als Eingabedaten erhält das INDUCE System Beschreibungen von Beispielen, die a priori ein oder mehreren Zielkonzepten zugeordnet sind (d.h. klassifizierte Beispiele). So können etwa zwei Gruppen von Zügen, die den Zielkonzepten *Ost-Züge* und *West-Züge* zugeordnet sind, als solche Beispiele fungieren (siehe Abbildung 1). Jedes Beispiel (d.h. ein Zug) wird dabei durch mehrere Eigenschaften beschrieben. So werden die Wagen der Züge als *kurz* oder *lang*, durch ihre Form, und/oder durch die *Anzahl der Räder* charakterisiert. Dabei können sowohl nominale (z. B. links, rechts), ordinale (z. B. kurz, mittel, lang), metrische (z. B. Längenangaben in cm) als auch hierarchisch strukturierte Attribute verwendet werden (z. B. ein Zug besteht aus einer Lokomotive und mehreren Waggons; jeder Waggon wiederum besteht aus Rädern, dem Fahrgestell, Aufsatz und Dach; Dächer haben die Ausprägung Farbe, etc.).

Allgemeine Beschreibung INDUCE wendet nun verschiedene Regeln an, um dadurch allgemeinere Beschreibungen für die jeweiligen Zielkonzepte zu erzeugen. Solche Generalisierungen können auf unterschiedliche Weise erfolgen: 1) Ersetzen von Konstanten durch Variablen (z. B. "rot" wird ersetzt durch "beliebige Farbe"); 2) Beschreibungsmerkmale werden ersatzlos gestrichen (d.h. es wird davon ausgegangen, daß dieses Merkmal irrelevant ist); 3) Bei metrischen Bezeichnungen werden aus den vorliegenden Werten ganze Intervalle erzeugt. Die Beschreibung, daß ein West-Zug 2 *oder* 4 Waggons hat, wird so generalisiert zu "zwischen 2 und 4 Waggons". 4) Bei hierarchisch strukturierten Beschreibungen erfolgt eine Generalisierung oft dadurch, daß man in dem Generalisierungsbaum eine Stufe höher steigt. Aus VW *oder* OPEL *oder* BMW kann so die Bezeichnung "deutsches Auto" entstehen. 5) Um kürzere Beschreibungen zu erhalten kann in besonderen Fällen die Negation eingesetzt werden.

⁵ Der Unterschied zwischen einem Beispiel und einem Fall besteht eigentlich darin, daß ein Beispiel immer auch ein Beispiel für etwas ist, (wie etwa für eine allgemeine Regel). Ein Fall kann dagegen für sich alleine stehen. Wenn wir von einem Beispiel für Erpressung sprechen, unterstellen wir, daß es allgemeine Regelmäßigkeiten gäbe für die vorliegende Erpressung ein Beispiel ist, für einen Fall von Erpressung muß dies dagegen nicht gelten. In der englischsprachigen Literatur werden die Bezeichnungen Beispiel und Fall oft aber auch anders gebraucht (vgl. Reimann & Schult, 1992).

Aus der Beschreibung "Dreieck *oder* Fünfeck *oder* Elypse *oder* Kreis" ergibt sich so beispielsweise die Bezeichnung "*kein* Quadrat". 6) Wenn ein Beispiel des Zielkonzepts (z. B. Ost-Zug) und ein Gegenbeispiel (z. B. West-Zug) gegeben sind, kann durch eine Generalisierungsregel die Negation aller Eigenschaften der Gegenbeispiele, die bei dem Beispiel selbst nicht vorliegen, erzeugt werden. In dieser Situation erhält man die allgemeinste Beschreibung des Zielkonzepts, welche die Gegenbeispiele ausschließt (Michalski, 1983).

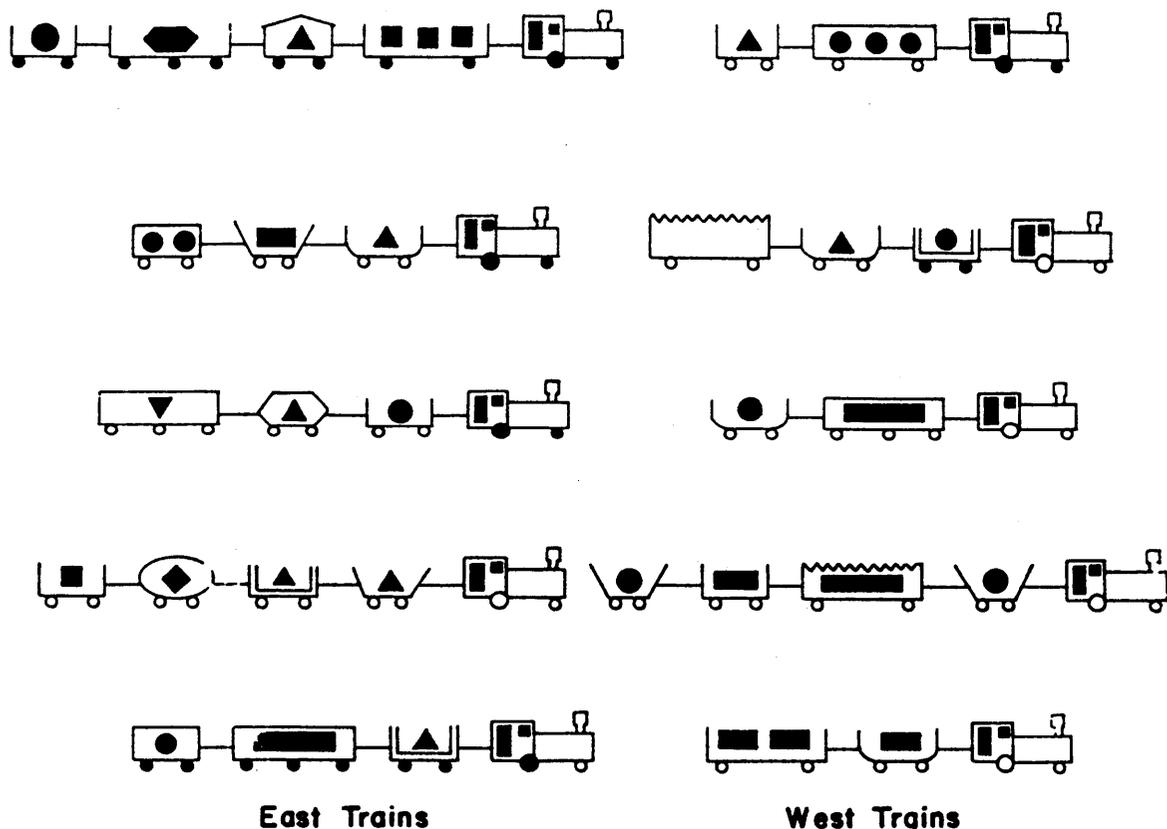


Abbildung 1: Zwei Gruppen von Zügen (Ost- und West-Züge), die als positive und negative Beispiele von dem INDUCE-Verfahren bearbeitet werden. In insgesamt 4 Experimenten wurden diese Züge von Versuchspersonen als Lernmaterial verwendet (nach Medin et al., 1987).

Darüberhinaus gibt es die *konstruktive Generalisierung*. Ein Beispiel dafür ist die sogenannte Zählregel, die eine Auszählung der Merkmale durchführt und bei mehrmaligem Vorliegen eines Merkmals eine Häufigkeitsangabe erzeugt. Aus "Lokomotive *und* Waggon *und* Waggon *und* Waggon *und* Waggon" wird so "Lokomotive *mit* (d.h. *und*) 4 Waggons". In ähnlicher Weise werden mit einer anderen konstruktiven Generalisierungsregel aus einer Sequenz von Entitäten

Bezeichnungen wie das erste Element, das mittlere Element oder das letzte Element erzeugt.

Induktionsalgorithmus: Als Induktionsverfahren wird von INDUCE die sogenannte "Stern-Methode" (englisch: star algorithm) (Michalski, 1983; Morik, 1993, S. 266f) verwendet. Dabei wird zuerst zufällig ein positives Beispiel aus der Menge der Beispiele für das Zielkonzept ausgewählt (z.B. der erste Ost-Zug aus Abbildung 1) und als Kern (englisch: seed) bezeichnet. Dieses Beispiel wird dann mit Hilfe von Generalisierungsprozessen auf unterschiedliche Weisen so beschrieben, daß die erzeugten Beschreibungen dabei aber nicht auf die Gegenbeispiele (d.h. hier die West-Züge) zutreffen (z. B. langer Zug mit Sechseck-Ladung). Die Gesamtheit der so erzeugten Beschreibungen wird als "Stern" bezeichnet. In einem dritten Schritt werden diese Beschreibungen nun hinsichtlich verschiedener Kriterien (z. B. Konjunktionen werden bevorzugt) beurteilt (vgl. hierzu die eingangs besprochene Lernbevorzugung) und es wird dann mit der bestbewerteten Beschreibung weitergearbeitet.

Falls die so erzeugte Beschreibung bereits alle (positiven) Beispiele umfaßt, ist der Lernvorgang nun mit der konjunktiven Verknüpfung von Beschreibungstermen (z. B. langer Zug mit Sechseck-Ladung) schon beendet. Ansonsten wird von den positiven Beispielen, die nicht unter die bisherige Konzeptbeschreibung fallen, ein Beispiel als weiterer Kern hinzugenommen (z. B. der zweite Ost-Zug; er hat keine Sechseck-Ladung). Das Verfahren wird so mit diesem Kern wiederholt und die aus beiden Kernen entstandenen Konzeptbeschreibungen werden dann disjunktiv verknüpft. Als Lernergebnis erzeugt INDUCE dadurch eine Disjunktion, deren Terme selbst wiederum aus Konjunktionen bestehen (z. B. West-Züge haben 2 Waggons *mit* schwarzen Ladungen *oder* ein gezacktes Dach).

Aus der Beschreibung des Algorithmus wird ersichtlich, wie das Lernergebnis von INDUCE von den speziellen Lernbevorzugungen abhängt: Der induktive Schluß hängt also sowohl von

Personen in der speziellen Lernsituation benutzte Repräsentationsbevorzugung erfaßt werden. Darüberhinaus mußten die Versuchspersonen die Beispiele in Klassen aufteilen und Regeln für die Klassenzugehörigkeit aufstellen. Es zeigte sich, daß es in verschiedener Hinsicht Übereinstimmungen zwischen INDUCE und der menschlichen Vorgehensweise gibt. So wurden auch von den Versuchspersonen einfache Merkmale und konjunktive Verknüpfungen von Merkmalen bevorzugt verwendet. Ebenso waren die von den Versuchspersonen erstellten Beschreibungen der Beispiele mit den von INDUCE verwendeten Beschreibungen durchaus vergleichbar.

Zwischen INDUCE und dem Verhalten der Versuchspersonen ergab sich jedoch auch eine wesentliche Abweichung: Versuchspersonen benutzten oft vollständige Konzeptbeschreibungen, die aber insgesamt widersprüchlich waren. INDUCE erzeugt dagegen zuerst Beschreibungen, die zwar unvollständig dafür aber immer konsistent sind. Die Tatsache, daß Menschen sehr viel besser mit Widersprüchen leben und arbeiten können, ist auch aus anderen Untersuchungen bekannt und ist ein Anreiz für neuere Entwicklungen auf dem Gebiet des maschinellen Lernens.

Des weiteren zeigte sich, daß die Versuchspersonen spezifischere Konzeptbeschreibungen erzeugten (z. B. zwei Räder *und* Dreiecksladung *und* langer Zug) als aufgrund des Einfachheitskriteriums erwartet würde (z. B. langer Zug). Die durchgeführten Experimente ergaben auch, daß für die Versuchspersonen "category validity" ein wichtigeres Kriterium war als "cue-validity" (vgl. Eckes, dieser Band). Insgesamt konnten Medin et al. aber zeigen, wie maschinelle Lernverfahren als Grundlage zur Modellierung psychologischer Lernprozesse eingesetzt werden können. Dafür waren mehrere psychologische Experimente erforderlich. Mit Hilfe der experimentellen Ergebnisse konnte dann eine neue Version von INDUCE erzeugt werden, die nun eine höhere psychologische Validität aufweist.

2.3 Lernen aus Texten

In dem vorausgegangenen Abschnitt wurde bereits gezeigt, wie das Lernresultat ganz entscheidend von den gewählten Wissensrepräsentationen beeinflußt wird. Beim Lernen aus Texten wird nun als wichtigster Teil des Lernprozesses die Erstellung von geeigneten Repräsentationen unmittelbar thematisiert. Es sollen hier zwei Lernverfahren vorgestellt werden, die sich in vielerlei Hinsicht ähneln, in einigen wichtigen Annahmen aber ganz grundsätzlich unterscheiden. Das GENESIS-System lernt aus Texten, indem es durch die Erwartung einer wohldefinierten Kette von Verknüpfungen (z. B. Kausalbeziehungen) geleitet wird, wodurch eine systematisch aufgebaute Wissensstruktur erzeugt wird. Das C-I Modell dagegen geht davon aus, daß in einer sog. *Konstruktionsphase* zunächst völlig planlos verschiedene Repräsentationseinheiten konstruiert und mit dem Lernmaterial sowie untereinander verknüpft werden. Das so konstruierte Wissen ist oft redundant und kann auch in

sich widersprüchlich sein. Erst in einer darauffolgenden *Integrationsphase* wird dann aufgrund der vorliegenden Verknüpfungen und Beziehungen eine für den speziellen Kontext angemessene Repräsentation bestimmt. Die beiden Systeme sollen im folgenden einander gegenüber gestellt werden.

2.3.1 Das GENESIS-System

GENESIS steht für "*GENERALizing Explanations of Stories Into Schemata*" (Mooney & DeJong, 1985) und ist ein System, das englischsprachige Geschichten analysiert und das zugrundeliegende Handlungsschema erzeugt. Der dabei verwendete Verallgemeinerungsprozess ist eine wissensbasierte Analyse der kausalen Zusammenhänge der einzelnen Ereignisse innerhalb der Geschichte. Die folgende Geschichte, die GENESIS als Eingabe dient, handelt von einer Entführung mit Lösegelderpressung. Nach der Verarbeitung der Geschichte kann GENESIS auch Fragen, z.B. nach dem Grund für die Gefangennahme Marys und dem dabei verfolgten Ziel, beantworten.

"Fred is the father of Mary and is a millionaire. John approached Mary. She was wearing blue jeans. John pointed a gun at her and told her he wanted her to get into his car. He drove her to his hotel and locked her in his room. John called Fred and told him John was holding Mary captive. John told Fred if Fred gave him \$ 250000 at Trenos then John would release Mary. Fred gave him the money and John released Mary."

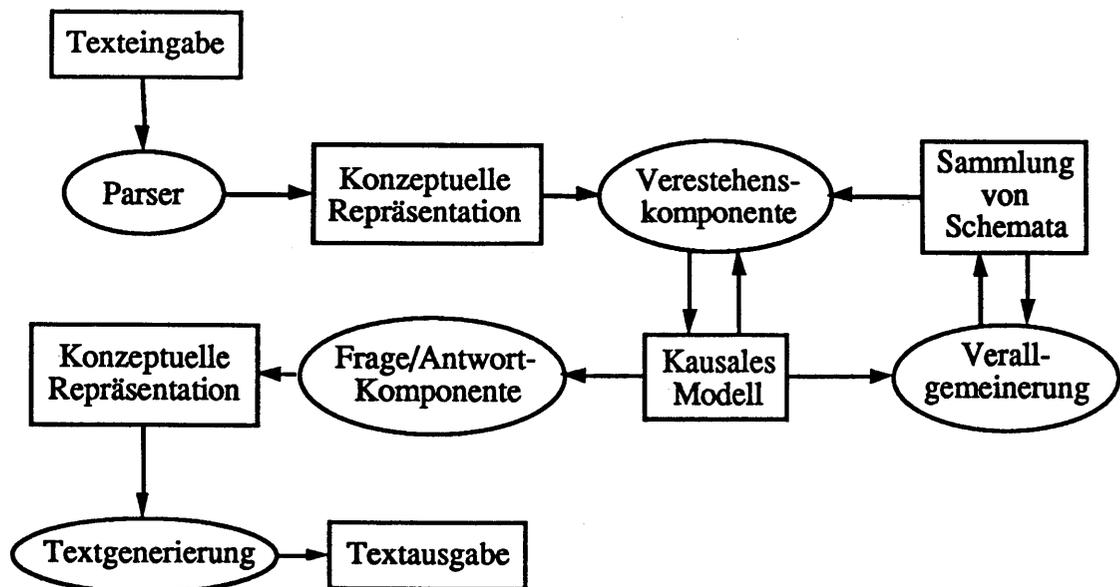


Abbildung 2: Architektur des GENESIS-Systems (nach Mooney & DeJong, 1985).

Die Architektur des GENESIS-Systems ist in Abbildung 2 dargestellt. Die englische Texteingabe wird zunächst von einem Parser (Dyer, 1983) in eine konzeptuelle Representation mit semantischen Primitiven übersetzt (vgl. Schank, 1982; Schank & Abelson 1977; Kapitel 1). Die Hauptaufgabe der Verstehenskomponente besteht darin, eine vollständige Repräsentation der kausalen Zusammenhänge innerhalb der Geschichte zu erzeugen. Um die Aussagen des Textes in einen ursächlichen Zusammenhang zu bringen, sind verschiedene Inferenzschritte erforderlich (vgl. van den Broek, 1990; Singer et al., 1992). Dafür ist Hintergrundwissen

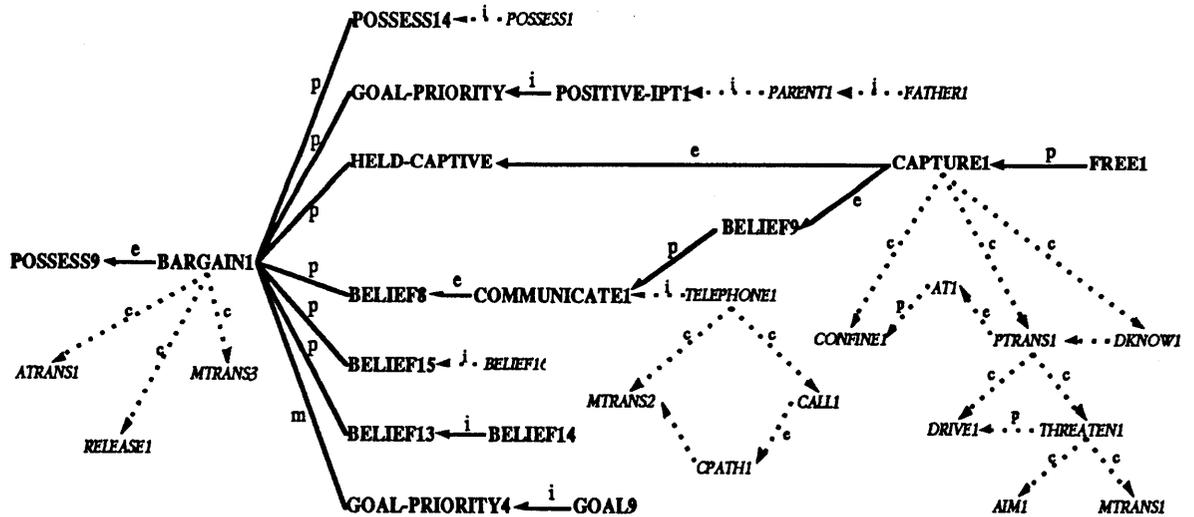
Komponente dient dazu, unterschiedliche Fragen zu beantworten.

Die von GENESIS verwendete Methode des Erklärens und Verallgemeinerns ist eine Form des 'Erklärungs-Basierten Lernens' (EBL; DeJong & Mooney, 1986), die auch im EBG-System (Mitchell et al., 1986) verwendet wird. Bei EBL und EBG wird die Zugehörigkeit eines vorgelegten Beispiels (hier im Text beschrieben) zu einem ebenfalls vorgegebenen Zielkonzept mit Hilfe einer sogenannten Domänentheorie erklärt (*Erklärungsphase*). Anschließend wird mittels der so erzeugten Erklärung eine operationale Beschreibung erzeugt. Es sind dies Beschreibungen, die einerseits allgemeiner als das Beispiel sind, andererseits aber viel spezieller als das vorgegebene Zielkonzept (*Verallgemeinerung des Beispiels*).

Die Domänentheorie, die der Verstehenskomponente von GENESIS zur Verfügung steht, enthält eine Spezifikation der Objekte und ihrer Eigenschaften, die in den Geschichten auftreten können, allgemeine Inferenzregeln zum Ableiten weiterer Eigenschaften und Relationen, sowie eine Sammlung von Problemlösungsoperatoren und bereits bekannten Schemata.

Zur Bearbeitung der vorliegenden Entführungsgeschichte muß die Domänentheorie

ergeben, während *GENERAL* die Substitutionen so allgemein wie möglich hält, ohne wichtige Bedingungen für die Anwendbarkeit der Operatoren zu verletzen. Tabelle 1 und Abbildung 3 zeigen auszugsweise die vollständige Erklärungsstruktur (alle Knoten und Kanten) der Entführungsgeschichte in Form eines Baumes.



i: inference; p: precondition; m: motivation; e: effect; c: composed of

Abbildung 3: Gezeigt ist die vollständige Erklärungsstruktur der Entführungsgeschichte (alle Kanten und Knoten), als auch die generalisierte Erklärungsstruktur (nur fettgedruckte Kanten und Knoten). Die Beschreibungen der einzelnen Knoten sind in Tabelle 1 angegeben (nach DeJong & Mooney, 1986).

Im zweiten Schritt der Erklärungsphase werden Operatoren entfernt, die für die Erreichung des eigentlichen verfolgten Zieles irrelevant sind. In der Entführungsgeschichte dient alles dem Ziel des Lösegeldgewinns mit Ausnahme der Tatsache, daß Mary Bluejeans trägt. Dieser Knoten wird demzufolge aus der Erklärungsstruktur entfernt.

Verallgemeinerungsphase: Instantiierungen bereits bekannter allgemeiner Schemata werden entfernt. Anschließend werden Aktionen und Zustände entfernt, die nur zur Herleitung abstrakter Aktionen und Zustände dienen. Dadurch werden z. B. die Knoten FATHER1 und PARENT1 entfernt. Das bis dahin generierte Schema wird dann noch optimiert. Zuletzt wird nun die allgemeinste Substitution *GENERAL*, die während der Erklärung erzeugt wurde, angewendet. So ergibt sich das Handlungsschema, das in Tabelle 1 und Abbildung 3 durch die fettgedruckten Eintragungen gekennzeichnet ist.

POSSESS9	John hat 250000 \$. (Person1 hat Geld1.)
BARGAIN1	John macht einen Handel mit Fred, bei dem John Mary freiläßt und Fred 250000 \$ an John gibt. (Person1 macht einen Handel mit Person2, bei dem Person1 Person3 freiläßt und Person2 Geld1 an Person1 gibt.)
MTRANS3	John sagt Fred, daß er Mary freilassen wird, wenn er ihm 250000 \$ gibt.
RELEASE1	Fred läßt Mary frei.
ATRANS1	Fred gibt John 250000 \$.
POSSESS14	Fred hat 250000 \$. (Person2 hat Geld1.)
POSSESS1	Fred hat mehrere Millionen Dollar.
GOAL-PRIORITY5	Fred bevorzugt, daß Mary frei ist, als 250000 \$ zu haben. (Person2 bevorzugt daß Person3 frei ist, als Geld1 zu haben.)
POSITIVE-IPT1	Fred hat eine positive zwischenmenschliche Beziehung zu Mary. (Person2 hat eine positive zwischenmenschliche Beziehung zu Person3.)
PARENT1	Fred ist ein Elternteil von Mary.
FATHER1	Fred ist Marys Vater.
HELD-CAPTIVE1	John hält Mary gefangen. (Person1 hält Person3 gefangen.)
CAPTURE1	John fängt Mary. (Person1 fängt Person3.)
D-KNOW1	John findet heraus, wo Mary ist.
PTRANS1	John bringt Mary zu seinem Hotelzimmer.
DRIVE1	John fährt Mary zu seinem Hotelzimmer.
THREATEN1	John droht Mary zu erschiessen, wenn sie nicht in sein Auto steigt.
AIM1	John zielt mit einer Waffe auf Mary.
MTRANS1	John sagt Mary, er möchte, daß sie in sein Auto steigt.
AT1	Mary befindet sich in Johns Hotelzimmer.
CONFINE1	John schließt Mary in seinem Hotelzimmer ein.
FREE1	Mary ist frei. (Person3 ist frei.)
•	•
•	•
•	•
BELIEF14	John glaubt, daß Fred Marys Vater ist. (Person1 glaubt, daß Person2 der Vater von Person3 ist.)
GOAL-PRIORITY4	John möchte lieber 250000 \$ haben als Mary gefangenhalten. (Person1 möchte lieber Geld1 haben als Person3 gefangenhalten.)
GOAL9	John möchte 250000 \$ haben. (Person1 möchte Geld1 haben.)
ATTIRE1	Mary trägt Blue Jeans.

Tabelle 1: Beschreibung der Knoten der in Abbildung 3 gezeigten Erklärungsstruktur des GENESIS-Systems. Die fettgedruckten Eintragungen zeigen in Klammern die Knotenbeschreibungen die durch die Substitution GENERAL entstehen.

2.3.2 Das C-I-Modell

Das C-I-Modell wurde nicht so sehr zum Zweck der Untersuchung von Lernkompetenzen, sondern vielmehr als psychologisches Prozeßmodell entwickelt. Deshalb wurden bei seiner Implementierung auch die wichtigsten empirischen und theoretischen Erkenntnisse aus der Gedächtnispsychologie und der Textforschung berücksichtigt (vgl. van Dijk & Kintsch, 1983). Um einen besseren Vergleich mit dem GENESIS-System zu ermöglichen, wollen wir das C-I-Modell auf eine ähnliche Weise beschreiben. C-I steht für Construction-Integration. Das C-I-Modell kann auf verschiedene Arten von Texten (Kintsch, 1988, Mannes & Kintsch, 1991) und andere Lernmaterialien (Mannes & Doane, 1994; Schmalhofer, 1994) angewendet werden.

Beispielsweise beschreiben Kintsch et al. (1990), wie Geschichten, denen ein Handlungsschema zugrundeliegt, mit dem C-I Modell bearbeitet werden. Der Text, der mit dem C-I-Modell bearbeitet wurde, war ca. 200 Worte lang und folgte einem Handlungsschema, das hier als Skriptnorm bezeichnet wird (z. B. dem Skript "Going to Movies", Galambos, 1983). Die beiden ersten Sätze der Geschichte lauteten: "Nick decided to go to the movies. He looked at the newspaper to see what was playing." Mit dem C-I-Modell kann auch die Beantwortung von Gedächtnis und Sachfragen (z. B. "Did Nick pay for his movie ticket?") simuliert werden.

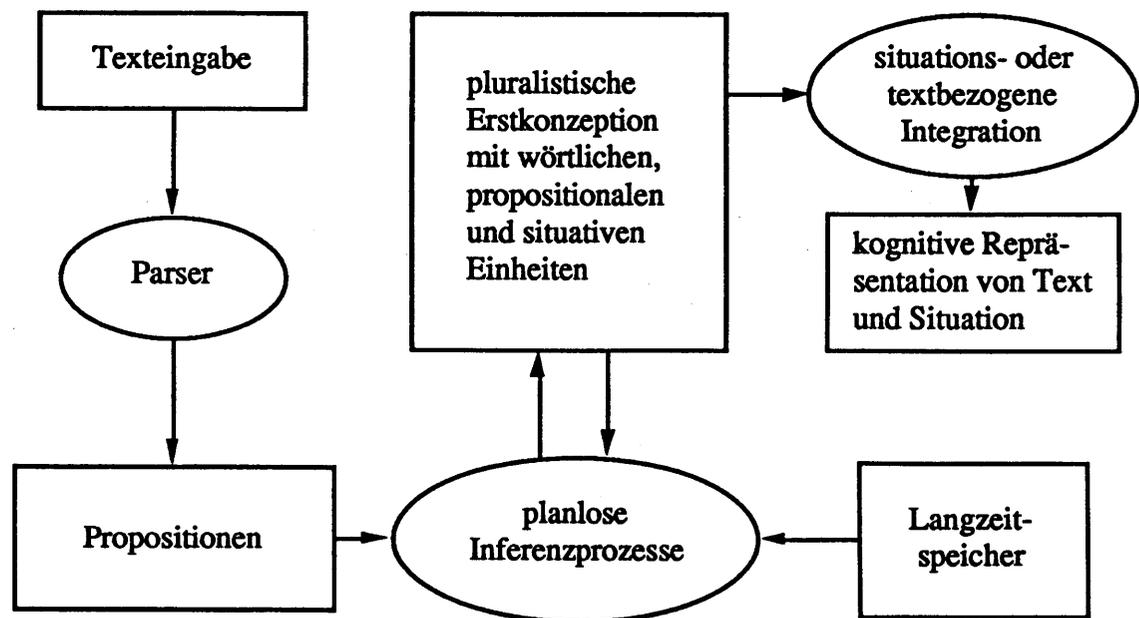


Abbildung 4: Allgemeine Inferenzstruktur des C-I Modells.

Die Architektur des C-I-Systems ist in Abbildung 4 dargestellt. Beim C-I-Modell besteht das

bezeichnet, durch die die Bedeutung von Texten dargestellt wird (vgl. McKoon & Ratcliff, 1980). Durch weitgehend unkontrollierte Inferenzprozesse wird dann mit dem im Langzeitspeicher abgelegten Sprach- und Sachwissen, eine vernetzte Struktur von wörtlichen, propositionalen und situativen Einheiten konstruiert, die man als eine pluralistische Erstkonzeption bezeichnen könnte. Dabei werden insbesondere drei Repräsentationen unterschieden, nämlich wörtliche, propositionale und situative (Fletcher, 1994).

Durch einen Integrationsprozeß, der mittels eines *spreading activation process* simuliert wird, entsteht dann eine im Sinne des Konnektionsmus verteilte Repräsentation von Text und Situation (siehe Mangold, dieser Band). Die Beantwortung von Testsätzen erfolgt im Prinzip auf die gleiche Weise wie das Verstehen eines Satzes im Text, so daß im Gegensatz zu GENESIS hier keine separate Antwortkomponente erforderlich wird.

Nick	decided	to go to	the movies.
<i>L1</i>	<i>L2</i>	<i>L3</i>	<i>L4</i>
(NICK)	(DECIDE, P1, P3)	(GO, P1, P4)	(MOVIES)
<i>P1</i>	<i>P2</i>	<i>P3</i>	<i>P4</i>
<i>M1</i>	<i>M2</i>	<i>M3</i>	<i>M4</i>

He	looked at	the newspaper		to see	what	was playing.
<i>L10</i>	<i>L5</i>	<i>L6</i>		<i>L8</i>	<i>L11</i>	<i>L9</i>
	(LOOK, P1, P6)	(NEWSPAPER)	(CAUSE, P5, P8)	(SEE, P1, P9)		(PLAY, IN MOVIES)
	<i>P5</i>	<i>P6</i>	<i>P7</i>	<i>P8</i>		<i>P9</i>
	<i>M5</i>	<i>M6</i>	<i>M7</i>	<i>M8</i>		<i>M9</i>

Tabelle 2: Beschreibung der einzelnen Knoten der in Abbildung 6 dargestellten Repräsentation von Text und Situation im C-I Modell.

Beim Bearbeiten der Geschichte über Nick werden in der Konstruktionsphase für die ersten beiden Sätze die in Tabelle 2 aufgeführten Oberflächensegmente des Textes (L1 bis L11), Propositionen (P1 bis P9), und Situationseinheiten (M1 bis M10) konstruiert und durch Kanten miteinander verknüpft. Die Verknüpfung der Oberflächensegmente erfolgt gemäß der syntaktischen Satzstruktur und ergibt das in Abbildung 5 (unten) dargestellte Resultat. Ähnlich

werden die Propositionen aufgrund ihrer referentiellen Beziehungen miteinander verknüpft (Kintsch, 1974). Daraus ergibt sich der in Abbildung 5 (Mitte) dargestellte Textbasisausschnitt. Auch die Situationseinheiten werden verknüpft. Diese Verknüpfungen richten sich nach den sachlichen Abhängigkeiten zwischen den einzelnen Einheiten (ähnlich wie in GENESIS; z. B. kausale, zeitliche, motivationale Zusammenhänge etc.), die gemäß der betreffenden Skriptnorm gelten. So erhält man den in Abbildung 5 (oben) dargestellten Ausschnitt eines Situationsmodells.

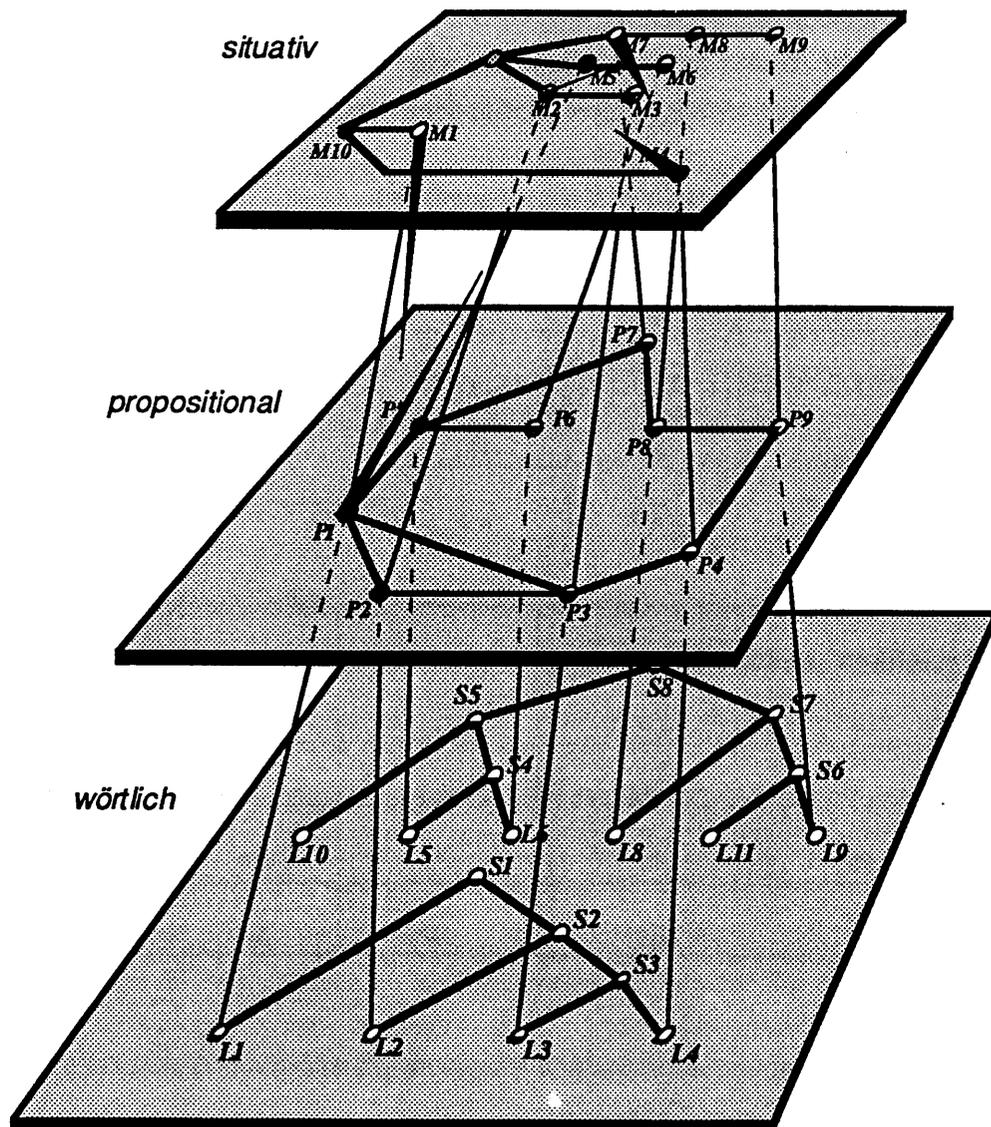


Abbildung 5: Drei-Ebenen Darstellung der vom C-I Modell erzeugten Repräsentation von Text und Situation. Die unterschiedlichen Schwärzungen der einzelnen Knoten zeigen die jeweiligen Aktivierungsgrade der Knoten nach der Integrationsphase. Die Beschreibungen der einzelnen Knoten sind in Tabelle 2 angegeben (nach Kintsch et al. (1990) Abb. 3, 4, und 5).

In dem letzten Schritt der Konstruktionsphase werden korrespondierende Knoten der drei Repräsentationsebenen miteinander verbunden (z. B. die linguistische Einheit L1 mit der Proposition D1 und die Proposition D1 mit der Situation S1). D1 ist in L1

erscheinenden Fachzeitschrift 'Machine Learning' im Jahre 1986 waren dafür entscheidende Meilensteine. Inzwischen gibt es neben verschiedenen Jahrestagungen, bei denen maschinelle Lernverfahren u.a. vorgestellt werden (e.g. Wetter et al., 1992; Musen & Gaines, 1994), eine einschlägige europäische Jahrestagung mit ca. 50 Beiträgen (e.g. Kodratoff, 1991), eine jährlich stattfindende internationale Konferenz zum maschinellen Lernen mit ca. 120 Beiträgen (e.g. Birnbaum & Collins, 1991) und die internationale KI-Tagung (s.u.), bei denen jährlich neue Lernverfahren und Lernsysteme vorgestellt, vergleichend evaluiert und Fragen der prinzipiellen Erlernbarkeit diskutiert werden. Mit 36 Publikationen war "Maschinelles Lernen" 1993 auch auf der internationalen Konferenz für Künstliche Intelligenz (Bajcsy, 1993) wiederum das am stärksten vertretene Teilgebiet.

Um sich überhaupt einen aktuellen Überblick über das Gebiet des maschinellen Lernens verschaffen zu können, ist es daher erforderlich, Ordnungsgesichtspunkte und Klassifikationen einzuführen. So haben beispielsweise Habel und Rollinger (1984) *Lernen als Erinnern, Lernen aus Instruktionen, Lernen durch Analogien, Lernen aus Beispielen, und Lernen durch Beobachtung* differenziert.

Carbonell (1989) unterscheidet ebenfalls zwischen fünf verschiedenen Lernparadigmen: Beim *induktiven Lernparadigma* wird die Induktion einer allgemeinen Begriffsbeschreibung aus positiven und negativen Beispielen beschrieben (vgl. INDUCE). Das *analytische Paradigma* umfasst Lernen aus wenigen (oft nur einem einzigen) Beispielen und einer umfangreichen Hintergrundtheorie (vgl. GENESIS). Das *genetische Paradigma* entstand in Anlehnung an die Beschreibungen der Mutation in biologischen Systemen und in Anlehnung an die Darwinschen Auswahlstrategien (z.B. 'survival of the fittest'). Das *konnektionistische Paradigma* (auch: *Neuronale Netze* oder *parallel verteilte Systeme*) beschreibt Verbindungsstrukturen vieler kleiner autonomer Einheiten, die jede für sich eine Teilaufgabe übernehmen, und insgesamt zum Beispiel bestimmte Muster darstellen können (vgl. C-I-Modell; Mangold, dieser Band). Die Idee konnektionistischer Ansätze entstammt dem Versuch, Computermodelle in Anlehnung an die neuronale Informationsverarbeitung zu entwickeln (vgl. McClelland & Rumelhart, 1986; Strube, 1990). Die genetischen und konnektionistischen Techniken werden auch als *subsymbolische* Methoden zusammengefaßt und gegen die anderen *symbolischen* Verfahren abgegrenzt.

Eine schon nahezu klassisch zu bezeichnende Einteilung in drei Gruppen nehmen Michalski und Kodratoff (1990) vor. Sie unterscheiden *auf Analogie beruhende* Techniken (vgl. JUDGE) von *synthetischen* (vgl. INDUCE) und *analytischen Techniken* (vgl. GENESIS). Diese Unterscheidung, in der die bei dem Lernverfahren zugrunde liegende Inferenzmethode im Vordergrund steht, wurde auch von Morik (1993) und Richter und Wendel (1993) aufgegriffen. Dabei differenziert Morik etwas feiner zwischen *Lernen als Suche* (z. B. Mitchell, 1982), *induktivem Lernen* (z. B. INDUCE), *deduktivem Lernen* (z. B. GENESIS), *Logik-orientiertem induktivem Lernen* (Muggleton, 1992), *Lernen als nicht-monotonem Schluß*

während Richter und Wendel lediglich die auf Analogie beruhenden Verfahren abweichend als *fallbasiertes Lernen* bezeichnen.

Unter dem Aspekt des praktischen Einsatzes von Lernsystemen (z. B. zur Entwicklung und Wartung von Expertensystemen; vgl. Buchanan, 1989) wird oft die Art des Aufgabenwissens (vgl. Breuker & Wielinga, 1989), das erlernt oder erweitert werden soll als Ordnungskriterium herangezogen. So können Lernsysteme zur qualitativen Verbesserung der *Planung*, der *Diagnose*, der *Klassifikation*, der *Erklärung* oder zur reinen *Effizienzsteigerung* eingesetzt werden. Zusätzlich werden bei der Diskussion des praktischen Einsatzes von Lernsystemen meist vollständig automatisch ablaufende Lernverfahren von den sog. Lehrlingslernsystemen (Tecuci & Kodratoff, 1990) unterschieden, bei denen System und Benutzer gemeinsam ein Lernresultat erarbeiten. Anhand der aufgeführten Kategorisierungen kann man leicht erkennen, daß eine Systematisierung entweder mehr vom grundlagenwissenschaftlichen Interesse der Erforschung von Lernvorgängen oder aber von praktischen Zielsetzungen geleitet sein kann.

3.1 Betrachtung auf der Wissensebene

Lernsysteme sind in der Regel sehr komplexe Computerprogramme, die in einer spezifischen Sprache (z. B. LISP, PROLOG oder C) implementiert sind und nur in einer bestimmten Hard- und Softwareumgebung ausgeführt werden können. Ähnlich wie bei anderen KI-Programmen und kognitiven Modellierungen sind viele technische Einzelheiten für die Lauffähigkeit eines Systems erforderlich. Für die eigentlichen Fragen des maschinellen Lernens sind diese Implementierungsdetails aber meist von geringem Interesse. Auch eine Beschreibung der verschiedenen Algorithmen auf der Symbolebene (Newell, 1980) kann manchmal wegen der damit verbundenen großen Komplexität die wesentlichsten Unterschiede zwischen

Nach Newell werden im einzelnen folgende Zuschreibungen gemacht: 1) Ein (Wissens-)System ist ein Akteur, der die Fähigkeit hat, eine Menge von Aktionen in einer bestimmten Umgebung auszuführen. 2) Das (Wissens-)System verfolgt Ziele, die sich auf eine gewünschte Beschaffenheit seiner Umgebung beziehen. 3) Das System hat einen Wissenskörper (englisch: "a body of knowledge"). Dieses Wissen bezieht sich auf die Umgebung, seine Ziele, seine Aktionen und deren Beziehungen zueinander. Darüber hinaus kennt das System auch alles, was aus seinem Wissen abgeleitet werden kann. (D.h. die inferentielle Hülle seines explizit angegebenen Wissens, also alle logisch möglichen Deduktionen, zählen ebenfalls zum Wissenskörper.⁹ 4) Das Rationalitätsprinzip ist das einzige Gesetz seines Verhaltens. Es besagt: *"If an agent has knowledge that one of its actions will lead to one of its goals, then the agent will select that action."* (Newell, 1982, S. 102). Im Prinzip handelt es sich bei Beschreibungen auf der Wissensebene um Beschreibungen von Computersystemen (oder der menschlichen Informationsverarbeitung), die von einem sogenannten "theoretician observer" (Clancey, 1993) vorgenommen werden, der das System als einen "idealized rational agent" betrachtet. Ein System *"is viewed as having a body of knowledge and a set of goals, so that it takes actions in the environment that its knowledge indicates will attain its goals"* (Newell, 1992, S. 426).

Dietterich (1986) hat nun versucht, Lernsysteme auf der Wissensebene zu charakterisieren. Zunächst könnte man vermuten, daß sich hierbei Lernen als ein Wissenszuwachs darstellen würde. Die Beschreibung eines Systems vor einem Lernvorgang würde sich von der Beschreibung des Systems nach dem Lernvorgang dadurch unterscheiden, daß dem System nach dem Lernvorgang mehr Wissen zugeschrieben wird (vgl. Kaelbling, 1987). Doch Dietterichs Analysen ergaben ein anderes Resultat.

Einer großen Anzahl von Lernsystemen wie z. B. LEX (Mitchell, Utgoff & Banjerji, 1983) oder EBG (Mitchell et al., 1986) wurde nach dem Lernvorgang das gleiche Wissen zugeschrieben wie vor dem Lernvorgang.¹⁰ Eine so wichtige Veränderung wie der Übergang

⁹ Wie später noch ausgeführt werden wird, bedeutet dies, daß auf der Wissensebene die Kenntnis der Spielregeln als Beschreibung des Wissens von Schachexperten schon hinreichend ist und auf der Wissensebene kein Wissensunterschied zwischen Schachexperten und Schachnovizen besteht. Dieses Manko nimmt Newell (1982, S. 105) mit dem folgenden Argument in Kauf: "Sometimes behavior can be predicted by the knowledge level description; often it cannot. The incompleteness is not just a failure in certain special situations or in some small departures. The term radical is used to indicate that entire ranges of behavior may not be describable at the knowledge level, but only in terms of systems at a lower level (namely, the symbolic level). However, the necessity of accepting this incompleteness is an essential aspect of this level."

¹⁰ Warum dies so ist, läßt sich leicht an dem folgenden Beispiel verdeutlichen: Vergleichen wird beispielsweise einen Schachcomputer1, der nur die Spielregeln beherrscht und nur durch Suche einen Zug auswählt mit einem Schachcomputer2, der sich ähnlich wie ein Schachmeister verhält und extrem viele Brettkonstellationen und Zugfolgen gespeichert hat. Auch wenn der Schachcomputer2 in der Realität unvergleichbar besser spielt, so unterscheidet er sich wegen der dabei vorgenommenen Idealisierung in seiner Beschreibung auf der Wissensebene in keinsten Weise von dem Schachcomputer1. Da angenommen wird, daß der Schachcomputer1 alles weiß, was aus den Spielregeln deduziert werden kann (s.o. Abschnitt 3), weiß er auch alles was man über das Spiel überhaupt wissen kann. Nach dem Rationalitätsprinzip wird er daher das perfekte Schachspiel spielen und sich nicht von dem Schachcomputer2 unterscheiden. Die Unterschiede zwischen Schachcomputer1 und Schachcomputer2 treten also erst bei der Beschreibung auf der Symbolebene zutage.

von einem Schachnovizen zu einem Schachmeister stellt nach Dietterichs Analysen keinerlei Wissenszuwachs dar.

Bei einer zweiten Gruppe von Lernverfahren konnte wegen der spezifischen Abhängigkeiten von der darunterliegenden Symbolebene (vgl. *representational bias* und *search bias*) die beim Lernen auftretenden Veränderungen nicht auf der Wissensebene beschrieben werden: Da die Wissensveränderungen hier von der Symbolebene abhängt, die Symbolebene aber nicht Bestandteil der zulässigen Beschreibungsmittel ist, können diese Veränderungen auf der Wissensebene nicht dargestellt werden.¹¹ In diese Gruppe fielen die induktiven Lernverfahren wie z. B. ID3 (Quinlan, 1986) oder INDUCE. Diese Ergebnisse sind durchaus unerwünscht und scheinen auf eine Fehlkonzeption der Wissensebene hinzuweisen (vgl. Bourne, 1969; Bourne, Ekstrand & Dominowski, 1971; Schmalhofer, Aitken & Bourne, 1994).

Nur bei der dritten und letzten Gruppe konnte die Veränderung durch den Lernvorgang als Wissenszuwachs beschrieben werden. In diese Gruppe fallen Systeme, die völlig neue Informationen aufnehmen.¹² Zu dieser dritten Gruppe zählen deduktive Datenbanksysteme, wie z. B. MRS (Russell, 1985). Auch fallbasierte Systeme (vgl. JUDGE), die durch die Speicherung eines neuen Falles neues Wissen aufnehmen, könnte man dieser Gruppe zuordnen.

Insgesamt zeigten Dietterichs Analysen, daß bei der Verwendung von Newell's Wissensebene Lernen nur in sehr wenigen Fällen durch den Zuwachs von Wissen charakterisiert werden kann. Dies liegt daran, daß Newell's Wissensebene auf einer unrealistischen Idealisierung beruht. Deshalb sind selbst essentielle Wissensunterschiede (wie z. B. zwischen dem Wissen von Schachnovizen und Schachexperten, vgl. Chase & Simon, 1973) auf einer so konzipierten Wissensebene nicht darstellbar.

Zur besseren Charakterisierung von intelligenten Systemen wurden daher von Schmalhofer, Aitken & Bourne (1994) eine Erweiterung der Wissensebene vorgeschlagen, die als Verhaltensbeschreibung (engl.: behavior descriptions) bezeichnet wird. Wie Newell gehen auch diese Autoren davon aus, daß einem intelligenten System Wissen und Ziele zugeschrieben werden sollten. Darüber hinaus sind aber mindestens zwei weitere Parameter erforderlich, um intelligentes Verhalten angemessen zu charakterisieren, nämlich entsprechende Fertigkeiten

¹¹ Dabei wird von Dietterich definiert: "A system is said to be deductively describable at the knowledge level if its behavior can be captured (and predicted) as deductive inference over a set of sentences that do not refer to symbol level entities."

¹² Wenn ein System beispielsweise bisher weiß "Alle Menschen sind sterblich", und nun in einem Lernvorgang "Sokrates ist ein Mensch" aufnimmt, so stellt dies einen Zuwachs an Wissen dar. Das darauffolgende Lernen von "Sokrates ist sterblich" würde dagegen keine Veränderung auf der Wissensebene bewirken, weil sich dieser Satz bereits durch Deduktion ergibt. (Da die inferenzielle Hülle bereits zum Wissen gehört, kann diese nicht mehr neu gelernt werden).

(engl.: skills) und eine tatsächliche Aktionsausführung in einem bestimmten Kontext oder Einsatzgebiet (englisch: performance).

Das *Wissen* ist dabei die vollständige Menge von Unterscheidungen, Fakten und verfügbaren Begriffen, die ein Wissenssystem aus früheren Erfahrungen erworben hat. Doch Wissen existiert nicht unabhängig von der Art und Weise, wie es eingesetzt wird (vgl. Bourne, Ekstrand & Dominowski, 1971, Kapitel 1). Deshalb wird für jede Wissensseinheit eine korrespondierende *Fertigkeit* postuliert, die seine Verwendung beschreibt. In ähnlicher Weise sind auch Ziele nicht unabhängig von Wissen und den Fertigkeiten. Die drei Parameter Wissen

Verhaltensbeschreibung zu erhalten, muß deshalb noch eine tatsächliche Aktionsausführung vorliegen. Außerdem müssen die Verhaltensparameter über eine oder mehrere Relationen zueinander so in Beziehung stehen, daß sich eine kohärente und konsistente Verhaltensbeschreibung ergibt.

Die Verhaltensbeschreibung *B* eines Lernsystems kann somit notiert werden als: $B =$

3.1.1 Parameterausprägungen

Die Ausprägungen der Parameter von Verhaltensbeschreibungen können selbst wieder komplexe Symbolstrukturen sein (vgl. Schmalhofer et al., 1994). Da wir hier aber nur an recht globalen Verhaltensbeschreibungen interessiert sind, sollen diese Symbolstrukturen nicht dargestellt, sondern nur benannt werden; auf diese Weise ergeben sich als Parameterausprägungen verschiedene Bezeichnungen. Diese Ausprägungen lassen sich durch die Beantwortung von parameterspezifischen Fragen bestimmen: 1. Durch die Beantwortung der Frage 'Woraus wird was gelernt?' wird der Zweck eines Lernsystems und damit der G-Parameter bestimmt. 2. Das Wissens eines Systems (d.h. der K-Parameter) wird durch die Frage 'Womit wird gelernt?' ermittelt. 3. Die Frage 'Wie wird gelernt?' bestimmt die Fertigkeiten und damit den S-Parameter. Nachdem durch diese Parameter das Verhaltenspotential eingegrenzt ist, wird mit der Frage 'Wo wird gelernt?' die tatsächliche Aktionsausführung und damit das Einsatzgebiet (P-Parameter) bestimmt.

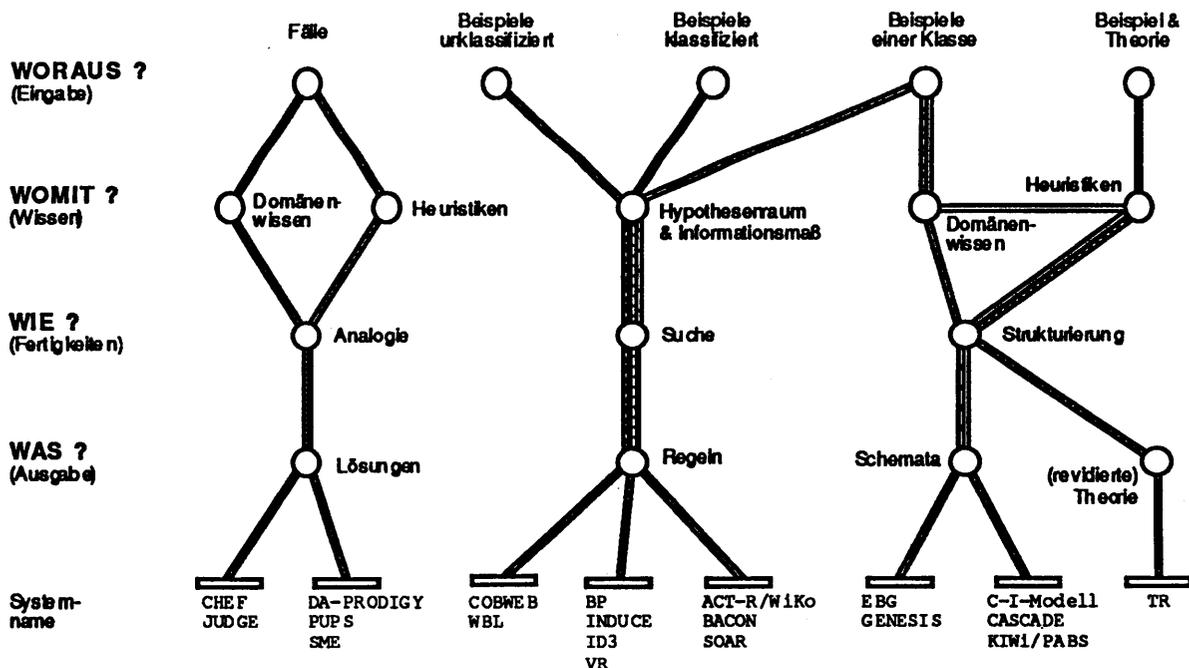


Abbildung 6: Beschreibungen von verschiedenen Lernsystemen auf der erweiterten Wissensebene (vgl. Fußnote 14).

Die Verhaltensbeschreibungen und die Relation R, die sich aufgrund unserer Analysen der ausgewählten Lernsysteme ergaben, sind in Abbildung 6 dargestellt. Zur übersichtlicheren Darstellung wurden dabei die Ausprägungen des P-Parameters nicht aufgenommen und der G-

Parameter wurde in seine Komponenten Eingabe und Ausgabe untergliedert. Abbildung 6 kann im Prinzip wie das Streckennetz von IC-Zügen gelesen werden.¹⁴

Woraus wird was gelernt? - Zweck

Hinsichtlich des Zwecks von Lernsystemen ergaben sich sechs verschiedene Ausprägungen. Systeme wie JUDGE oder DA-PRODIGY lernen aus Fällen komplette Lösungen. So lernt JUDGE in dem Bereich der Rechtsprechung aus alten, bekannten und bereits beurteilten Fällen, die als eine Menge von Präzedenzfällen eingesetzt werden. Mittels dieser bekannten Erfahrungen wird ein Urteil für einen neu vorgelegten Tatbestand gefällt. Die so entstandenen neuen Fälle können wiederum gemerkt werden, so daß sich das Wissen des Systems sukzessive vergrößert (*closed-loop-learning*, vgl. Wrobel, 1989). In ähnlicher Weise, allerdings mehr domänenunabhängig, verwendet DA-PRODIGY konkrete Erfahrungen aus selbst erzeugten Fällen, um Lösungen für neue Probleme zu erstellen. Die SME lernt aus dem Aufbau des Sonnensystems wie Atome aufgebaut sein könnten und welche strukturellen Beziehungen zwischen den einzelnen Elektronenschalen und dem Atomkern bestehen.

Eine zweite Gruppe von Systemen wie z. B. COBWEB oder WBL lernen aus Beispielen ein oder mehrere Regeln, die zur Klassifikation von Objekten eingesetzt werden können. Die Beispielmenge für COBWEB ist eine unstrukturierte Menge von Objekten, die in Form von Merkmalsbeschreibungen repräsentiert sind. Die Ausgabe des COBWEB Systems ist eine mit Regeln beschreibbare Begriffshierarchie, in der die einzelnen Gruppierungen in Unter-Obermengen-Beziehungen zueinander stehen. Beim WBL werden bestimmte Eingabemuster (= Aktivitätszustände der Eingabeneuronen) vorgegeben. Anschließend werden die Gewichte der Verbindungen einzelner Neuronen so eingestellt, daß bei Vorlage eines Eingabe-Musters die relevanten Merkmale (als Menge von Aktivitäten in der Ausgabeschicht) herausgefiltert werden können, was sich wiederum mit Regeln beschreiben läßt.

Eine dritte Gruppe erzeugt Regeln aus Beispielen, die von Anfang an bereits klassifiziert vorliegen. So erhält INDUCE eine Menge von Objektbeschreibungen in Form von Attribut-Wert-Paaren, die jeweils einer bestimmten Klasse (z.B. der Klasse der OST- oder WEST-ZÜGE) zugeordnet sind. Ähnlich verhält es sich bei den Eingaben von ID3 und VR. Der Unterschied zwischen INDUCE, ID3 und VR liegt einerseits darin, daß bei den ersten beiden Systemen die Beispiele nicht-inkrementell und bei VR inkrementell vorgegeben werden, und andererseits darin, daß INDUCE und ID3 mehrere Klassen gleichzeitig betrachten können,

¹⁴ Für ein bestimmtes System (wie z. B. INDUCE) erhält man seine Verhaltensparameter, indem man den entsprechend markierten Pfad verfolgt und dabei die Ausprägungen (also, Regeln, Suche, Hypothesen und Informationsmaß klassifizierte Beispiele) abliest. Umgekehrt kann man einen beliebigen Pfad verfolgen um dann zu erfahren, auf welche Systeme die entsprechenden Ausprägungen zutreffen (z.B. {(Fälle), (Problemlösung)} {Heuristiken} {Analogie}...) trifft auf SME, DA-PRODIGY, und PUPS zu.

während VR immer nur einen Begriff erkennt.¹⁵ INDUCE erzeugt ähnlich wie VR mit Regeln eine Konzeptbeschreibung aufgrund der entdeckten relevanten Attributausprägungen. ID3 erzeugt ebenfalls Konzeptbeschreibungen, allerdings in Form eines sogenannten Entscheidungsbaumes, dessen Knoten Attribute und dessen Äste Verzweigungen gemäß der verschiedenen Attributausprägungen sind. Solche Entscheidungsbäume sind im Prinzip eine Sammlung von Regeln (vgl. Quinlan, 1987). An den Blättern steht der Name einer Klasse, so daß sich aus jedem Pfad eine Konzeptbeschreibung ergibt. RP lernt aus einer vorgegebenen

Menge von Ein-/Ausgabe-Paaren für gewünschte Aktivierungszustände der entsprechenden Neuronen. Die dargebotenen Beispiele können auch zu einem Begriff gehören, ohne daß dies explizit angegeben ist. BP ist nach dem Lernprozess (ähnlich wie das WBL durch Einstellen der Gewichte) dazu in der Lage, zu einem Eingabemuster das gewünschte Ausgabemuster zu erzeugen. Auch dies läßt sich bei einer Betrachtung auf der Wissensebene durch Regeln charakterisieren (vgl. McMillan & Smolenski, 1988).

Systeme wie BACON oder SOAR erhalten nur einheitliche Beispiele (d.h. Beispiele einer einzigen Klasse) als Eingabe, aus denen sie dann Regeln erzeugen. So erhält BACON als Eingabe die Paare einer Wertetabelle aus einem Physik-Experiment. BACON erkennt dann funktionale Abhängigkeiten und erzeugt so eine Funktionsvorschrift zur Berechnung weiterer Werte. SOAR lernt neben der Lösung des gegebenen Problems, die sich als Pfad im Zustandsraum ergibt, neue Operatoren hinzu, die mehrere Einzelschritte einer Lösung

zusammenfassen.

Eine fünfte Gruppe von Lernsystemen erzeugen aus Beispielen Schemata. Zu diesen Systemen gehören GENESIS, das aus einer Beispielerzählung und einfachen Regelzusammenhängen lernt, EBG, das aus einem Beispiel eines Konzeptes und einer Hintergrundtheorie über den betrachteten Problembereich neues Wissen erwirbt, und schließlich das C-I-Model, das aus einem Text eine Textrepräsentation und ein Situationsmodell erstellt. GENESIS erzeugt ein allgemeines Handlungsschema zu dem in der Beispielerzählung enthaltenen Geschehen. In ähnlicher Weise generiert EBG eine allgemeine Konzeptbeschreibung, die effizienter eingesetzt werden kann.¹⁶

Womit wird gelernt? - Wissen

Dies ist die Frage welches Wissen beim Lernen eingesetzt wird, also welche Ausprägungen der K-Parameter annimmt. Dabei ergaben sich vier verschiedene Ausprägungen. 1) Systeme wie JUDGE verwenden neben den bereits bekannten Fällen allgemeine Grundsätze und Verordnungen der Gesetzgebung, um gegebenenfalls ohne Vorlage eines Präzedenzfalles ein Urteil fällen zu können. Sie besitzen also *Domänenwissen*.

2) Systeme wie SME und PRODIGY lernen mit Hilfe von *Heuristiken*, die ggf. bereichsspezifisch erweitert werden können und die Erinnerung an eine bekannte Struktur oder ein bereits gelöstes Problem leiten.

3) Eine weitere Gruppe von Lernsystemen, zu der SOAR, COBWEB, WBL, INDUCE und ID3 zählen, setzen unterschiedliche Hypothesenräume ein, in denen eine Art von Informationsmaß definiert ist (z.B. Kennwerte für die Parsimonie einer Hypothese, oder das Informationsmaß von ID3). SOAR kennt einen Raum von Zuständen, die durch Operatoren verbunden sind. Der Hypothesenraum von COBWEB ist durch alle möglichen Einteilungen der vorgelegten Objekte in verschiedene Begriffe gegeben. WBL verwendet eine Menge von Verbindungsstärken zwischen einzelnen Neuronen, um geeignete Gewichte für die einzelnen Merkmale der vorgelegten Muster zu finden. INDUCE, ID3 und VR steht ein Raum möglicher Konzeptbeschreibungen in Form von konjunktiv verknüpften Attribut-Wert-Paaren zur Verfügung. BP lernt (ähnlich wie WBL) geeignete Verbindungsstärken durch eine Minimierung einer Fehlerfunktion.

4) Lernsystemen wie z. B. GENESIS, EBG, C-I-Modell, CASCADE, KIWI/PABS und TR stehen teils Domänenwissen und teils Heuristiken zur Verfügung. GENESIS und EBG verfügen über eine Domänentheorie, die aus Fakten und Regeln des behandelten Problembereiches besteht. Auch hier lassen sich die einzelnen Ausprägungen partiell ordnen. So können wir beispielsweise feststellen, daß ein System, dem nur Heuristiken zur Verfügung stehen, weniger Wissen hat als ein System, dem bereichsspezifisches Domänenwissen und Heuristiken zur Verfügung stehen.

Wie wird gelernt? - Fertigkeiten

Die Beantwortung dieser Frage ergab insgesamt drei Ausprägungen für die Fertigkeiten von maschinellen Lernverfahren (S-Parameter): 1. Eine erste Gruppe erkennt Analogien zwischen bekannten und neuen Aufgaben. So lernen JUDGE, SME und PRODIGY durch das Auffinden einer vergleichbaren bereits bekannten Situation und einer Modifikation dieser alten Lösung (falls erforderlich).

2. Eine andere Gruppe (z. B. COBWEB, WBL) sucht in dem gegebenen Hypothesenraum nach einer hinsichtlich des Informationsmaßes optimalen Lösung. So versuchen COBWEB und

WBL Gemeinsamkeiten der verschiedenen dargebotenen Beispiele zu finden, und die Objekte werden dann aufgrund dieser entdeckten Merkmale gruppiert. Bei COBWEB werden diese Gruppen so gebildet, daß sich zwischen den Objekten einer Gruppe eine möglichst große Ähnlichkeit und zwischen Objekten verschiedener Gruppen eine möglichst geringe Ähnlichkeit ergibt. Dagegen werden die Gruppen beim WBL nicht explizit erstellt, sondern das neuronale Netz stellt nach dem Lernvorgang nur einen Merkmalsdetektor dar. INDUCE sucht gemeinsame Generalisierungen der Beispiele, so daß sich eine allgemeine Charakterisierung der einzelnen Beispiele ergibt. ID3 bewertet den Informationsgehalt eines jeden Attributes, und legt eine Reihenfolge fest, in der die Attribute eines neuen Objektes getestet werden können, indem es den noch zu erwartenden Informationsgewinn nach Testen des Attributes minimiert. Die VR durchläuft den Raum möglicher Konzeptbeschreibungen gleichzeitig aus der Richtung des allgemeinsten und des speziellerten Begriffes kommend. Nach Vorlage eines neuen Beispiels wird die Menge der noch mit den bisherigen Beispielen verträglichen Begriffe sukzessive eingeschränkt, bis schließlich nur noch ein Begriff übrig bleibt. BP verwendet ebenso wie ID3 eine Minimumsuche zum Lernen. Im Gegensatz zu ID3 wird bei BP allerdings die Fehlerabweichung zwischen tatsächlicher aktueller Ausgabe und der vorgegebenen gewünschten Ausgabe minimiert. Diese allgemein auch als 'Hill-Climbing' bezeichnete Technik der Suche nach einem Optimum stellt auch eine der (verschiedenen) Heuristiken dar, die BACON verwendet.

3. Eine dritte Fertigkeit ist die Erstellung von Erklärungen unter Verwendung des vorhandenen Wissens. So begründet GENESIS mit Hilfe des Hintergrundwissens die verschiedenen Zusammenhänge einer Erzählung, während EBG die Zugehörigkeit des dargebotenen Beispiels zu dem Begriff mit Hilfe der Hintergrundtheorie erklärt (beispielsweise wird erklärt, daß ein vorgegebenes Objekt eine Tasse ist). Und schließlich dient das Verständnis der dargebotenen Beispiele aufgrund des vorhandenen Wissens dem C-I-Modell dazu, eine kohärente Struktur in der Form einer Textbasis und eines Situationsmodells aufzubauen.

Selbstverständlich kann ein System auch mehrere dieser Fertigkeiten haben. Wir würden dann sagen, daß ein System, welches alle Lernfertigkeiten eines anderen Systems umfaßt, mehr Lerndispositionen besitzt. Ein "multistrategy learning system" (Michalski, 1993), das sowohl Analogien finden als auch einen Hypothesenraum absuchen kann, hätte somit umfassendere Lerndispositionen als ein System, das nur Analogien finden kann.

Wo wird gelernt? - Einsatzgebiet

Bei der Beantwortung dieser Frage ergaben sich viele unterschiedliche Anwendungen von Lernsystemen. Daher soll hier keine vollständige Aufzählung der Parameterausprägungen gegeben werden, sondern nur einige genannt werden. JUDGE lernt auf dem Gebiet der Rechtsprechung. Die SME erkennt physikalische Zusammenhänge zwischen Elektronen und dem Atomkern. DA-PRODIGY lernt Kontrollregeln für die Fertigungsplanung im

Maschinenbau. SOAR wurde hauptsächlich in Mikrowelten wie z. B. dem 8-Puzzle oder dem Tower-of-Hanoi-Problem eingesetzt. BACON entdeckt aus Experimentaldaten klassische Gesetze, wie z.B. das dritte Keplersche Gesetz und die (chemischen Gesetze von Dalton und Gay-Lussac).

3.1.2 Verhaltensbeschreibung

In dem vorausgegangenen Abschnitt wurden die verschiedenen Ausprägungen der Parameter G, K, S und P aufgezählt, die sich bei einer globalen Beschreibung von ausgewählten Lernsystemen ergaben. Die Parameterausprägungen lassen sich wie folgt notieren:

$$G = \{(\{Fälle\}, \{Problemlösung\}), (\{Beispiele\}, \{Regeln\}), \\ (\{Beispiele\} \text{ Klassifikation} \} \{Regeln\}), (\{Beispiele\} \text{ Selektion} \} \{Regeln\}), (\{Beispiele\} \\ \text{Selektion} \} \text{Schema}), (\{Beispiele\} \text{ Theorie} \} \text{Theorie})\}$$

$$K = \{(\{Domänenwissen\}, \{Heuristiken\}), (\{Hypothesenraum\}, \{Informationsmaß\}), \\ (\{Domänenwissen\} \text{ Heuristiken} \} \}$$

$$S = \{(\{Analogie\}), (\{Suche\}), (\{Erklärung\})\}$$

$$P = \{(\{Rechtsprechung\}), (\{Atommodell\}), (\{Turm_von_Hanoi\}), \\ (\{Fertigungsplanung\} \text{ im Maschinenbau}), (\{physikalische\} \text{ Gesetze}) \dots\}$$

Das kartesische Produkt der Parameter (d.h. $G \times K \times S \times P$) und die für jeden Parameter möglichen Ausprägungen definieren nun einen Raum für mögliche Verhaltensbeschreibungen. Dabei ist allerdings nicht jede beliebige Kombination von Parameterausprägungen sinnvoll. Eine Kombination wird erst dann als sinnvoll erachtet, wenn bereits ein implementiertes System vorliegt, auf das die entsprechenden Ausprägungen zutreffen. Um die sinnvollen Ausprägungskombinationen von den bedeutungslosen zu unterscheiden wird die Relation R benötigt (die als Teilmenge von $G \times K \times S \times P$ definiert ist). Wir sagen, daß die Relation R zwischen den Ausprägungen g_i , k_i , s_i und p_i einer Verhaltensbeschreibung B_i erfüllt ist, falls ein System i implementiert vorliegt, dem diese Ausprägungen zugeschrieben werden, so daß sich daraus eine kohärente und vollständige Systembeschreibung auf einer (mehr oder weniger) abstrakten Ebene (d.h. der erweiterten Wissensebene) ergibt. Darüber hinaus sollten zwischen der Symbolebene des Systems und der Beschreibung auf der Wissensebene Zustands- und Sequenzabstraktionsabbildungen definierbar sein (vgl. Schmalhofer et al., 1993, S. 213-215).

Durch die vorliegenden Beschreibungen auf der erweiterten Wissensebene werden Äquivalenzklassen von Lernsystemen bestimmt. Die einzelnen Systeme einer Äquivalenzklasse mögen sich zwar aufgrund einer Beschreibung auf der Symbolebene unterscheiden, hinsichtlich

der Beschreibung auf der Wissensebene sind sie aber identisch. Durch die weiter oben angegebenen Ordnungsbeziehungen auf den verschiedenen Parameterausprägungen lassen sich nach den Verfahren der neueren Wissensdiagnostik (vgl. Lukas & Albert, 1993; Irtel & Schmalhofer, 1981/82) die einzelnen Lernsysteme anhand ihrer Verhaltensbeschreibungen auch partiell ordnen.¹⁷

Auf der erweiterten Wissensebene konnten so Beschreibungen für alle Lernsysteme erstellt werden. Bei diesen Verhaltensbeschreibungen zählt die inferentielle Hülle des Wissens (s.o.) nur insoweit zum Wissen, als die Inferenzen von den angegebenen Fertigkeiten (S-Parameter) erzeugt werden können. Das Wissen eines Schachexperten unterscheidet sich daher so sehr wohl von dem Wissen eines Schachnovizen. Auch induktive Lernverfahren wie ID3 konnten so auf der Wissensebene beschrieben werden. Selbst bei Lernverfahren wie EBG und CASCADE erkennt man, daß hier Lernen durch einen Zuwachs an Wissen darstellt und somit Lernen auf der Wissensebene stattfindet.

3.2 Fall-, such- und verstehensorientierte Lernsysteme

Die in Abbildung 6 dargestellten Beschreibungen der verschiedenen Lernsysteme zeigten, daß sich auch auf der Wissensebene ohne Beachtung des P-Parameters bereits sieben verschiedene Lernsysteme ergaben. (Jeder Pfad charakterisiert einen Typ von Lernsystem). Die sieben Lernsysteme lassen sich aber leicht in drei Gruppen einteilen, nämlich 1) Systeme, die aus Fällen durch analoges Schließen Lösungen erstellen, 2) Systeme die aus verschiedenen gearteten Mengen von Beispielen durch Suche geeignete allgemeine Regeln finden und 3) Systeme die durch die Erzeugung kohärenter Erklärungen Regeln, Schemata oder revidierte Theorien konstruieren. Diese drei Gruppen bezeichnen wir im folgenden als fallorientierte, suchorientierte bzw. verstehensorientierte Lernsysteme. Um die Funktionsweise solcher Systeme zu veranschaulichen, werden nun für jede dieser drei Gruppen einige typische Lernsysteme vorgestellt: CHEF und DA-PRODIGY als fallorientierte Systeme, SOAR und VR als suchorientierte Systeme und CASCADE, KIWi/PABS und TR als verstehensorientierte Systeme. Darüber hinaus haben wir noch entsprechende Komponenten des ACT-R-Systems mit aufgenommen. Da ACT-R als integrative kognitive Architektur mehrere Lernverfahren umfaßt, läßt sich dieses System, ähnlich wie das PRODIGY-System, nicht eindeutig zuordnen, sondern verwendet selbst wieder mehrere unterschiedliche Lernverfahren.

¹⁷ Die umweltbezogene Signifikanz eines Lernsystems könnte beispielsweise folgendermaßen definiert werden (vgl. Wrobel, 1991). Ein System S_i hat eine gleichgroße oder größere umweltbezogene Signifikanz (also $U(S_i) \geq U(S_j)$), falls die folgende lexikographische Ordnungsrelation erfüllt ist: $(p_i, g_i, k_i, s_i) \geq (p_j, g_j, k_j, s_j)$. Die einzelnen Parameterausprägungen (z.B. p_i und p_j) sind dabei Elemente einer Potenzmenge, so daß ihre Ordnung auf einfache Weise durch die Mengeneinklusion definiert ist (vgl. Lukas & Albert, 1993, S. 146).

3.2.1 Fallorientierte Systeme

Die Grundidee aller fallorientierten Lernverfahren besteht im wesentlichen darin, bei der Lösung neuer Probleme alte Erfahrungen einzusetzen, um eine vollständige neue Erarbeitung einer Lösung (z.B. mit Hilfe allgemeiner Regeln) zu vermeiden (vgl. auch JUDGE). Dazu ist es notwendig, sich bereits gelöste Probleme geeignet zu merken, und für die Lösung eines neuen Problems angemessen anzupassen. Existierende fallorientierte Systeme unterscheiden sich in diesen grundlegenden Komponenten, die auch als 'Indizieren', 'Erinnern' (Retrieval) und 'Modifikation' bezeichnet werden. Diese Vorgehensweise entspricht in gewisser Hinsicht dem menschlichen Problemlöseverhalten, denn auch der Mensch versucht, alte Erfahrungen in sich neu ergebenden Situationen einzusetzen.

CHEF

CHEF (Hammond, 1986; Hammond, 1989) ist ein fallorientiertes Planungssystem, das nach gegebenen Wünschen (etwa Geschmack, bestimmte Zutaten o.ä.) Kochrezepte erzeugt, indem es alte Rezepte aufbewahrt und diese versucht an die neuen Ansprüche anzupassen. CHEF arbeitet in sechs Phasen: 1. Problemvorhersage, 2. Planerinnerung, 3. Planmodifikation, 4. Planreparatur, 5. Fehleranalyse und 6. Planspeicherung. Bei der *Problemvorhersage* versucht CHEF, aufgrund der gestellten Ansprüche und alter Erfahrungen zu erwartende Probleme aufzudecken, und fügt die Vermeidung dieser als zusätzliche Ziele hinzu. Die *Planerinnerung* sucht daraufhin in der Fallbasis alter Erfahrungen nach einem geeigneten Rezept, das in möglichst vielen Anforderungen mit dem neuen gesuchten Rezept übereinstimmt. Dabei werden sowohl die Ziele als auch die zu erwartenden Fehler zur Indizierung der Rezepte genutzt. Während der *Planmodifikation* wird dann das geeignetste alte Rezept so verändert, daß auch die noch nicht erreichten Ziele erfüllt werden. Die Ausführung dieses neuen Rezeptes wird nun mittels allgemeiner Regeln aus dem Bereich des Kochens simuliert. Treten dabei keine Schwierigkeiten auf, so kann das neue Rezept abgespeichert werden. Ergeben sich jedoch fehlerhafte Teile des Rezeptes, so sucht CHEF bei der *Planreparatur* und der *Fehleranalyse* nach den Ursachen für die aufgetretenen Probleme, und behebt diese mit Hilfe allgemeiner Reparaturregeln. Das funktionsfähige Rezept wird anschließend abgespeichert.

PRODIGY

PRODIGY (Carbonell, 1986; Minton 1986; Minton et al. 1989) ist ein domänenunabhängiger Problemlöser, der neues Wissen durch Analyse seiner Erfahrungen und Interaktion mit einem Benutzer erwirbt. Während sich CHEF lediglich ganze Problemlösungen merkt, versucht

einer heuristischen Suche im Problemraum, die durch eine Reihe von Lernmethoden unterstützt wird. Abbildung 7 zeigt die Gesamtarchitektur des PRODIGY Systems.

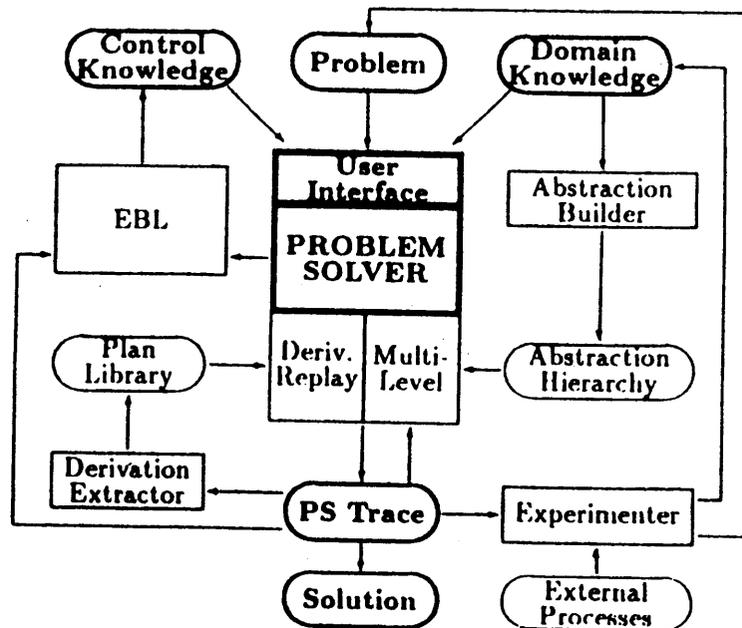


Abbildung 7: Die Gesamtarchitektur des PRODIGY-Systems (nach Minton et al. 1989).

Der Problemlöser baut sukzessive einen Suchraum auf, dessen Wurzel einen Initialzustand darstellt, und dessen Kanten durch Operatoren beschriftet sind, die einen Zustand in einen anderen (Folge)Zustand überführen. Das Ziel ist es, zu einem Blatt des Suchraumes zu gelangen, das dem vorgegebenen Zielzustand entspricht. Eine *Benutzerschnittstelle* ermöglicht die Interaktion mit einem Experten, der Hilfestellungen anbieten soll, wenn PRODIGY alleine nicht weiterkommt. Eine *Erklärungsbasierte Lernkomponente* erzeugt aus einer Problemlösung allgemeine (Kontroll)Regeln, die im folgenden (wieder)verwendet werden können, um die Suche nach einem geeigneten Pfad im Problemraum zu beschleunigen. Die ('derivational') *Analogie-Komponente* (Carbonell, 1986) unterstützt den Problemlöser, indem alte (Teil)Lösungen vollständig genutzt werden können. Die Möglichkeit der (Multi-Level) *Abstraktion* erweitert diese Wiederverwendung, so daß nicht nur konkrete, sondern auch

allgemeinere (Teil)Lösungen benutzt werden können. Durch *Experimentieren* kann PRODIGY schließlich Fehler in der bisherigen (Teil)Lösung ermitteln (und gegebenenfalls korrigieren).

3.2.2 Suchorientierte Systeme

Newell (1990, S. 111) zählt neben *Repräsentation*, *Wissen* und *Symbolen* auch *Suchen* zu den Grundlagen intelligenten Verhaltens. Um eine bestimmte Aufgabe zu bewältigen, oder allgemeiner ein gegebenes Problem zu lösen, muß insbesondere auch ein System einen Weg suchen, um ein gewünschtes Ziel zu erreichen. Diese Suche vollzieht sich in einem *Problemraum*, der mögliche Situationen enthält, und Verbindungen zwischen verschiedenen Situationen, wenn diese auf die eine oder andere Weise auseinander hervorgehen können. Bestimmte Situationen in diesem Raum entsprechen dann einem gestellten Problem, und andere Situationen korrespondieren zu Situationen, in denen das Problem gelöst ist. Die Suche besteht nun im wesentlichen darin, einen Weg zwischen diesen Situationen zu finden (vgl. Dörner, 1979).

Das eine Extrem dieser Suche ist die *blinde Suche*. Ohne jegliches Wissen einzusetzen oder eine Strategie zu verwenden, bewegt das System sich in dem Problemraum, bis es (eher zufällig) die Lösung gefunden hat. *Tiefen-* und *Breitensuche* sind typische Beispiele solch *uninformierter Suche*. Das andere Extrem ist im Grunde keine Suche mehr. Das System hat so viel Wissen, daß es direkt den richtigen Weg im Problemraum findet.

Zwischen diesen beiden Extremen bewegen sich sogenannte *informierte Suchverfahren*. Die Information, die ein System bei der Suche durch den Problemraum verwendet, stellt (Hintergrund)Wissen dar, und so ergibt sich ein zweiter Suchprozeß, der mit der Suche nach einem Lösungsweg unmittelbar verbunden ist. Es ist die *Suche nach dem in der aktuellen Situation sinnvoll einzusetzenden Wissen*. Die Qualität des eingesetzten (oder überhaupt verfügbaren) Wissens, durch das die Problemlösungssuche gesteuert wird, bestimmt maßgeblich die Geschwindigkeit oder Effizienz der Suche. So unterscheiden sich existierende Systeme der Künstlichen Intelligenz erstens nach dem zu durchsuchenden Problemraum, zweitens nach der verwendeten Suche und schließlich drittens nach dem dabei eingesetzten Wissen.

Suchorientierte Lernverfahren gehen nun davon aus, daß (sämtliches) Lernen durch das Auffinden einer Lösung in einem Problemraum und eine anschließende Zusammenfassung der gefundenen Lösung erfolgt. Auf diese Weise bildet SOAR chunks, und chunking wird von Newell als globales Lernverfahren betrachtet. Bei Mitchells VR-Methode findet die Suche in einem Versionenraum statt, der als Problemraum fungiert. Als Suchergebnis erhält man dann die gewünschte Konzeptbeschreibung. Während die Erzeugung von Produktionsregeln durch Wissenskompilation in ACT-R mit der Erzeugung von chunks in SOAR durchaus vergleichbar ist, ist das PUPS-Verfahren von ACT-R eher den fallorientierten Lernverfahren zuzurechnen.

Als integrierte kognitive Architektur ist ACT-R somit ein multi-strategy-Lernsystem, das über Lernverfahren aus verschiedenen Klassen verfügt.

SOAR

SOAR (Laird et al., 1984; Rosenbloom et al., 1986) ist eigentlich ein Problemlösesystem, das auf heuristischer Suche in Problemräumen basiert, in dem bei jeder Problemlösung aber auch Lernen stattfindet. Der zugrunde liegende Problemraum besteht aus Zuständen und Operatoren, die einen Zustand in einen anderen überführen. Ausgehend von einem Initialzustand sucht SOAR nach einem bestimmten Zielzustand. Im Idealfall kann dabei immer unmittelbar eindeutig ein bestimmter Operator angewendet werden. In Situationen, in denen eine eindeutige Wahl eines Operators nicht möglich ist, wird ein Teilziel erzeugt, um den richtigen Operator zu finden. Die (generalisierte) Lösung dieses Teilziels merkt sich SOAR in Form eines sogenannten 'chunks'. 'Chunks' sind im Prinzip Produktionsregeln, d.h. Regeln der Form 'WENN Bedingung₁ & ... & Bedingung_n, DANN Aktion_{xy}'. Die Erzeugung der 'chunks' ist also letztlich nichts anderes als das Speichern einer zielorientierten sequentiellen

Strukturbildung (vgl. Hoffmann, 1994). Diese 'chunks' können dann in Zukunft bei der Lösung desselben oder ähnlicher Probleme eingesetzt werden, und erhöhen somit die Performanz des Problemlösers. Die Menge der gelernten 'Chunks' stellt eine Art Langzeitgedächtnis dar und bildet das bei der Suche vorwiegend eingesetzte Wissen.

Die Versionenraummethode (VR)

Bei der Versionenraummethode (Mitchell, 1982; Morik, 1993, S. 256-262) ist das einsetzbare Wissen in Form von positiven und negativen Beispielen eines bestimmten Konzeptes gegeben. Die Aufgabe besteht darin, eine konsistente (d.h. kein negatives Beispiel umfassende) und vollständige (d.h. jedes positive Beispiel umfassende) Konzeptbeschreibung zu finden. Der Problemraum ist durch die Menge der möglichen Konzeptbeschreibungen (z.B. in Form von konjunktiv verknüpften Attribut-Wert-Paaren) gegeben. Bei der Suche wird der Problemraum auf den Versionenraum eingeschränkt, der lediglich die Menge konsistenter und vollständiger

ACT-R: Wissenskompilation und Analogien

Anderson (1993) hat vor kurzem das System ACT-R vorgestellt, das einerseits auf dem Vorgänger ACT* (Anderson, 1983) aufbaut und andererseits zur Erstellung von Analogien um das PUPS Verfahren (Anderson & Thompson, 1989) erweitert wurde. ACT-R und ACT* sind umfassende psychologische Theorien über die Entstehung menschlichen Wissens (Anderson,

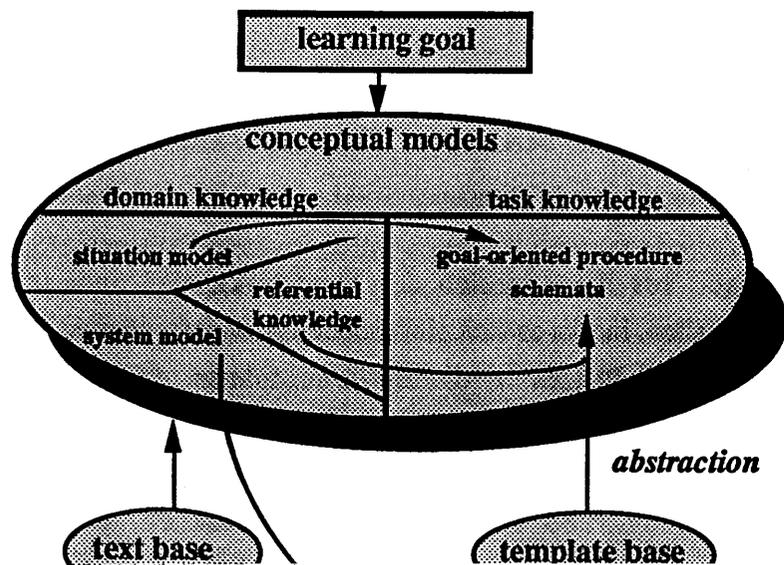
/Obermengen-Relationen statt), während bei einer Abstraktion eine Übersetzung von einer konkreten in eine abstrakte Begriffswelt stattfindet. (Michalski & Kodratoff, 1990). Bei den verstehensorientierten Verfahren wird dabei die abstrakte Begriffswelt aus den unterschiedlichsten Vorkenntnissen erst strukturiert. So wird beispielsweise beim C-I-Modell aus dem im Langzeitspeicher abgelegten Hintergrundwissen durch Konstruktions- und Interpretationsprozesse ein allgemeines Situationsmodell erstellt. Auf eine vergleichbare Weise erstellt CASCADE eine kohärente Erklärung von Beispielen. Anders formuliert erstellen oder verändern verstehensbasierte Lernverfahren eine (abstrakte) Theorie aus Hintergrundwissen und den gegebenen (konkreten) Erfahrungen (vgl. Schmalhofer & Tschaitchian, 1993).

CASCADE

Verschiedene empirische Untersuchungen zeigen, daß Studenten besser lernen, wenn sie beim Studium von Beispielen Erklärungen erzeugen (Chi, Bassok, Lewis, Reimann & Glaser, 1989). Die besseren Studenten versuchen also, Problemlösungen zu *verstehen*, wenn sie ein Problem betrachten. CASCADE (VanLehn et al., 1991; VanLehn et al., 1992) simuliert dieses als *Selbsterklärungseffekt* bezeichnete Phänomen. Dieses Lernverfahren basiert auf der Annahme, daß bei dem Selbsterklärungseffekt Lernen auf der Wissensebene stattfindet. Dabei erwirbt CASCADE zwei Arten von neuem Wissen: einerseits neue domänenspezifische Regeln (in diesem Falle aus dem Bereich der Mechanik), und andererseits aber auch neues Problemlöseverhalten, indem es sich eingesetzte Lösungswege merkt. Im wesentlichen besteht CASCADE aus zwei Phasen: 1. die Erklärungsphase, und 2. die Problemlösungsphase. CASCADE verwendet einen regelbasierten, rückwärts verketteten Theorembeweiser (ähnlich zu PROLOG). Als Eingabe dient ein Beispiel aus der Mechanik mit den zugehörigen Fakten der Problembeschreibung, und eine Reihe von Aussagen, die das zugrunde liegende Kräftediagramm und die Problemlösung repräsentieren. In der *Erklärungsphase* beweist CASCADE nun jede dieser Aussagen mittels der gegebenen Fakten und der bis dahin bewiesenen Aussagen. In der *Problemlösungsphase* wird CASCADE eine Reihe von Fakten und eine Aussage vorgelegt, und die Erklärung wird benutzt, um einen Beweis der Aussage herzuleiten. Für den Fall, daß während des Beweises mehrere Regeln anwendbar sind oder gar keine Regel angemessen ist, verwendet CASCADE Analogie-basierte Methoden, um trotzdem eine nützliche Regel zu finden (Carbonell, 1986). Obgleich diese Vorgehensweise für viele komplexere Anwendungsgebiete zu einfach ist, reicht sie für den betrachteten Bereich der Mechanik und ähnlich einfache Domänen mit monotonen Schlußweisen¹⁸ aus. Die in CASCADE eingesetzte Methode wird auch mit *Erklärungsbasiertem Lernen von Korrektheit* bezeichnet (VanLehn, Ball & Kowalski, 1990).

¹⁸ Eine Schlußweise heißt *monoton*, wenn durch neues Wissen früher bewiesene Aussagen gültig bleiben.

Im KIWi-Modell (*Konstruktion und Integration von Wissen*) (Bergmann et al., 1992; Schmalhofer et al., 1990) wird die Unterscheidung zwischen konkreter und abstrakter Repräsentationsebene explizit gemacht. So erzeugen Versuchspersonen beispielsweise aus Assembler-Code (auf der Ebene der Register und elementarer Registeroperationen) zunächst eine Beschreibung der dadurch realisierten Funktion (auf der Ebene einer höheren Programmiersprache) (etwa Addition oder Mittelwertbildung). Hier wird der Unterschied zwischen Generalisierung und Abstraktion besonders deutlich: Eine höhere Programmiersprache enthält neue (Sprach)Konstrukte, die auf der Maschinenebene nicht existieren (Schmalhofer et al. 1993). Abbildung 8 zeigt das KIWi-Modell zur Akquisition von Domänen- und Aufgabenwissen.



konzeptuelle Modell von den ursprünglichen Materialien unabhängig. Die Bildung von Domänen- und Aufgabenwissen erfolgt wie beim C-I-Modell in zwei Schritten: 1. einem Konstruktionsprozess und 2. einem Integrationsprozess. In dem *Konstruktionsprozess* wird zunächst neues, unter Umständen aber noch inkonsistentes Wissen erzeugt, das in dem *Integrationsprozess* zu kohärentem und konsistentem Wissen geformt wird (konzeptuelles Modell). Das Ziel des gesamten Lernprozesses ist also die Erzeugung neuen, und leicht auf zukünftige Probleme anwendbaren Wissens in allgemeinerer und von den konkreten Texten und/oder Beispielen abstrahierender Form. Diese resultierende Form des Wissens wird auch als *Prozedurschema* bezeichnet (vgl. Vorberg & Goebel, 1991). Eine formale Charakterisierung des Konstruktionsprozesses als Erzeugung (konkreter und abstrakter) Zustände, sowie (konkreter und abstrakter) Operatoren, die diese Zustände ineinander überführen, und des Integrationsprozesses als Abhängigkeitsanalyse zwischen einzelnen Zuständen und Operatoren, sowie einer Generalisierung der abstrakten Operatoren zu einem Prozedurschema führte schließlich zu einer prototypischen Implementierung des PABS (*Plan oder Program ABStraction*) Systems. PABS realisiert eine *Erklärungsbasierte Abstraktion* (vgl. Schmalhofer et al., 1995).

Theorie-Revision (TR)

Die sogenannte *Theorie-Revision* (z.B. Richards & Mooney, 1991) ist ein weiteres Beispiel für verstehensorientiertes Lernen. Ausgehend von (einer unter Umständen leeren) Theorie (im wesentlichen einer Menge von Formeln) der Domäne und konkreten Beispielen wird durch Revision der alten Theorie eine neue Theorie erzeugt, die für die gegebenen Beispiele vollständig und konsistent ist, d.h. die neue Theorie stellt ein globales und abstraktes Verständnis der Domäne dar, das ausreicht, um die konkreten Beispiele zu erklären.

4. WISSENSAKQUISITION

In der Künstlichen Intelligenz versteht man unter Wissensakquisition den Prozeß der Konstruktion eines Expertensystems (vgl. Clancey, 1989). Zur Entwicklung von Expertensystemen wurden bisher drei Methodiken erarbeitet: 1) In den 70er Jahren wurden Expertensysteme zunächst prototypisch implementiert (Shortliffe & Buchanan, 1975; Buchanan & Shortliffe, 1984). 2) In den 80er Jahren wurden dann allgemeine Inferenzmechanismen als Expertensystemschalen (Barstow et al., 1983) und in der Form von generischen Aufgabenlösern (Breuker & Wielinga, 1989; Chandrasekaran, 1986) bereitgestellt. Bei diesem Ansatz wird die explizite Wissensmodellierung betont. 3) In den 90er Jahren (Neches, Fikes, Finin, Gruber, Patil, Senator, & Swartout, 1991) wird nun die gemeinschaftliche Nutzung und die Wiederverwendung der Wissensbasen von Expertensystemen (englisch: "knowledge

sharing and reuse effort") untersucht. Bei dem Einsatz von Expertensystemen soll so eine größere Kompatibilität und Interoperabilität erreicht werden.

Der Einsatz von den in Forschungslaboren entwickelten maschinellen Lernverfahren (s.o.) soll nun hauptsächlich in Bezug auf den "Knowledge Sharing and Reuse Effort" diskutiert werden. Insbesondere interessiert uns dabei die Frage, ob ähnlich zu der Entwicklung der menschlichen Expertise (vgl. Mandl & Gruber, 1994; Ericsson, Krampe & Tesch-Römer, 1993) auch Expertensysteme mittels Lernen aus den zunächst laienhaften Kenntnissen eines Wissensbasiskerns ein Verhaltenspotential aufbauen können, das dem eines menschlichen Experten entspricht.

Durch einen solchen Einsatz von maschinellen Lernverfahren wird darüber hinaus geprüft, ob die bisher nur in Mikrowelten geprüften Verfahren auch für reale komplexe Domänen brauchbar sind. Von wenigen Ausnahmen abgesehen (z.B. ID3), benutzen kommerzielle Expertensysteme derzeit keine maschinellen Lernverfahren (Buchanan, 1989). Es stellt sich also die Frage, ob maschinelle Lernverfahren in komplexeren Bereichen der realen Welt generell versagen müssen. Da KI-Verfahren sehr gut in Mikrowelten funktionieren, bei einem praktischen Einsatz in realen komplexen Domänen aber oft Schwierigkeiten auftreten, ist diese Prüfung in der KI auch als das Skalierungsproblem (englisch: scaling-up problem) bekannt geworden (Schank, 1991). Die komplexen Anwendungsdomänen von Expertensystemen bieten daher eine wichtige Testumgebung für die tatsächliche Leistungsfähigkeit und den praktischen Nutzen von maschinellen Lernverfahren.

Trotz der vorerst negativen Bilanz ist der Einsatz von maschinellen Lernverfahren in komplexeren Bereichen erstrebenswert und bei einer Anpassung der einzelnen Verfahren an die jeweilige Domäne sowie die Wünsche des Systembenutzers auch erfolgversprechend (Schank, 1991; Schmalhofer et al., 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 Wissensevolution).

-4.1 Lernen mit Alltagswissen: Der CYC-Ansatz

Dieser hauptsächlich in den USA verfolgte Ansatz postuliert, daß automatisches Lernen in komplexen Bereichen ohne ein sehr umfangreiches Alltagswissen nicht möglich ist. Deshalb wird zunächst von Hand (d.h. mit den gewöhnlichen Methoden des Knowledge Engineering) eine Wissensbasis mit Alltagswissen erstellt. Danach soll sich die so aufgebaute Wissensbasis durch maschinelles Lernen dann automatisch erweitern.

Dieser Weg wird in dem auf zwanzig Jahre angelegten CYC-Projekt besprochen (Lenat, Prakash & Shepard, 1986?). Von 1984 bis 1994 wird dabei eine Wissensbasis aufgebaut, die

neben Fakten, Algorithmen, Heuristiken, Geschichten, Repräsentationen u.s.w. auch die Methoden zum effizienten Ableiten neuer Erkenntnisse umfaßt. 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 weiterlernen. Folgerichtig lautet Lenats präzise Antwort auf die Frage, wann Maschinen endlich lernen werden: "*... since CYC started in September 1984, and the crossover to automated KA was and still is scheduled to occur ten years later than that, a tongue in cheek answer is... September, 1, 1994.*" (Lenat, 1989; S.257).

4.2 Gemeinsames Lernen

Der Ansatz des gemeinsamen Lernens bezieht dagegen den menschlichen Benutzer in den Lernprozeß mit ein (Lajoie & Lesgold, 1989; Lesgold, Lajoie, Bunzo & Eggen, 1992; Wilkins, 1991). Beim Lehrlingslernen kann ein Lernsystem beispielsweise die Rolle eines Lehrlings und der menschliche Experte die Rolle des Lehrmeisters übernehmen. Die kooperative Wissensrevolution ist eine Weiterentwicklung des Lehrlingslernens, bei dem der Experte und das Lernsystem ein Tandem bilden und gemeinsam voneinander lernen.

4.2.1 Lehrlingslernen

Lehrlingslernen wird beim Lösen von konkreten Aufgabenstellungen eingesetzt. Unterschieden

Einsatz (Reinartz & Schmalhofer, 1994). Der Benutzer denkt und lernt meist auf der informellen, abstrakten Ebene, während das System eher mit formalisiertem Wissen arbeitet. Ein zentrales Anliegen dieses Ansatzes ist die Verbesserung der Koordination zwischen beiden Partnern. So soll dem Problem Rechnung getragen werden, daß es für den menschlichen Experten zumeist schwierig und unpraktisch ist, sich auf einer stark formalen Ebene mit der Maschine zu verständigen.

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 (englisch: editorial assistant), der das Wissen in die richtige Form bringt und auf bestimmte Konventionen beim Ablegen des Wissens achtet.

Nach diesem Ansatz werden drei Phasen unterschieden (vgl. Fischer, McCall, Oswald, Reeves & Shipman, 1994), die auch in Abbildung 9 unten, dargestellt sind. Zunächst wird 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 voneinander getrennt, sondern ergänzen einander. In der anschließenden Wissensnutzungs- und Wachstumsphase wird das System in der ersten Generation praktisch eingesetzt. Dabei wird die Wissensbasis teils automatisch und teils vom Anwender um fehlendes Wissen erweitert. Hier kommen u.a. auch Verfahren des Lehrlingslernens zum Tragen. Im Laufe der weiteren Nutzung muß die Wissensbasis in periodischen Abständen auch vom Wissensingenieur "gewartet" werden, wobei eine Reorganisation und Erneuerung der Wissensbasis stattfindet.

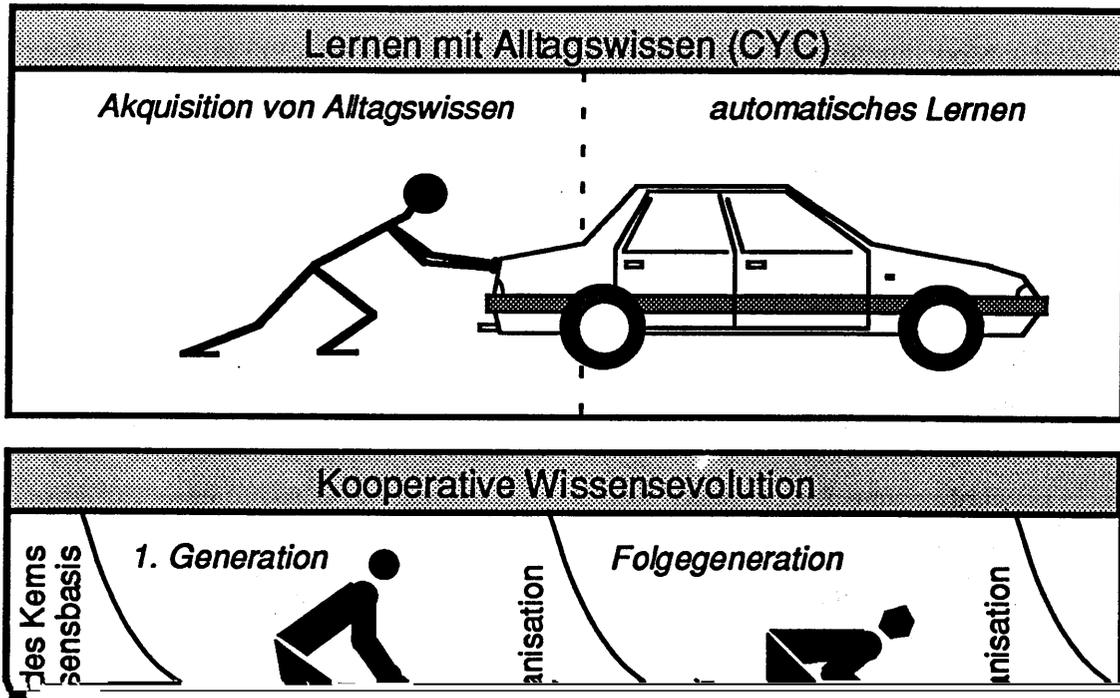
Da die darauf folgende Wissensnutzung auf den vorausgegangenen Erfahrungen aufbaut, sind Benutzer und System in den Folgegenerationen nun besser aufeinander abgestimmt. Dadurch wird das Anwender-System-Tandem zunehmend effizienter. An diesem Wechselspiel von Wissensnutzung, -wachstum und Reorganisation wird ein wesentlicher Vorteil der kooperativen Wissensrevolution deutlich: die Möglichkeit, sich an veränderte Umweltbedingungen anzupassen (vgl. Clancey, 1991; Fischer et al., 1994). Dieser Ansatz wird bei der Entwicklung des IDEAS-Systems verfolgt (Birk et al., 1994).

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 dann auch in komplexen Domänen selbsttätig lernen.



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 9), ist die kooperative Wissens evolution 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. DISKUSSION

Die Modellierung von menschlichem Lernen (und Verhalten) mit Computersimulationen hat im letzten Jahrzehnt auch in Deutschland stark an Bedeutung gewonnen (vgl. Mangold, dieser Band; Strube & Wender, 1993). So werden in der Kognitionswissenschaft (Collins & Smith, 1988; Habel, Kanngießer & Strube, 1990) Computermodelle entwickelt und in psychologischen Experimenten geprüft (z. B. Möbus, 1988; Tack, Wallach, Unz, Henss & Egler, 1993; Schmalhofer, Kühn & Boschert, 1994; Schmalhofer & Polson, 1986).

In diesem Beitrag wurden nun mehrere computerimplementierte Lernsysteme vorgestellt, die sich für verschiedene Lernaufgaben und mehr oder weniger gut auch als Grundlage für psychologische Lernmodelle eignen. Wie von Herrmann vor kurzem dargelegt wurde, spielen psychologische Experimente mit einer systematischen Bedingungsvariation auch für die empirische Prüfung solcher Prozeßmodelle eine entscheidende Rolle (Herrmann, 1990; Kluwe, 1993). Dabei kann die Bedingungsvariation ganz gezielt auf die theoretischen Konzepte der Prozeßmodellierungen abgestimmt werden. So haben z.B. Schmalhofer & Glavanov (1986) die Drei-Ebenen-Repräsentationsannahme des C-I-Modells (vgl. Abb. 5) herangezogen, um Bedingungsmanipulationen in ihren Experimenten festzulegen. Mit Hilfe von Lesezeitmessungen (Experiment 1) und Gedächtnismaßen (Experimente 2 und 3) konnten die Repräsentationsannahmen des C-I-Modells dadurch dann bestätigt werden. Beim Einsatz von Computermodellen wird sich so die Experimentiermethodik eher weiterentwickeln, als daß sie an Bedeutung verlöre.

Einige der dargestellten Lernsysteme sind von einer detaillierten empirischen Prüfung aber noch weit entfernt. Manche Systeme sollte man zunächst auch eher als rein theoretische Ausarbeitungen betrachten (z.B. Mitchells VR-Methode oder Richards & Mooneys TR-System). An dieser Stelle kann aber vermutlich jeder Leser schon selbst entscheiden, wo Computersysteme zu psychologischen Prozeßmodellen ausgearbeitet und erweitert werden können und wo die eingesetzten Techniken einer Prüfung in psychologischen Experimenten kaum standhalten würden.

Jedenfalls steht das Gebiet des maschinellen Lernens zu den Lern- und Gedächtnisforschungen der allgemeinen experimentellen Psychologie in einer sehr viel engeren und harmonischeren

Beziehung, als dies von den Vertretern der Künstlichen Intelligenz und der experimentellen Psychologie zur Zeit vermutet wird. Es handelt sich dabei nämlich um die zwei Seiten einer Medaille, die man als Kognitionswissenschaft bezeichnet. Da die ersten KI-Systeme dadurch entstanden, daß man versuchte, menschliche Verhaltensweisen nachzubilden, und andererseits die so entstandenen Techniken und Werkzeuge wieder herangezogen werden, um menschliches Verhalten zu beschreiben (Gigerenzer, 1988; 1991), ist diese enge Beziehung vielleicht gar nicht einmal so überraschend.

6. LITERATUR

Aamodt, A. (1991). A knowledge-intensive, integrated approach to problem solving and sustained learning. Doctoral Dissertation, University of Trondheim.

Althoff, K.D., Weiß, S., Bartsch-Spörl, B., Maurer, F., Janetzko, D., & Voß, A. (1992). Fallbasiertes Schließen in Expertensystemen. KI, 6(4), pp. 14-21.

Anderson, J.R. (1983). The architecture of cognition. Cambridge, MA: Harvard University Press.

Anderson, J.R. (1989). A theory of the origins of human learning. Artificial Intelligence, 40, pp. 313-351.

Anderson, J.R. (1990). The adaptive character of thought. Hillsdale, NJ: Lawrence Erlbaum.

Anderson, J.R. (1993). Rules of the mind. Hillsdale, NJ: Lawrence Erlbaum.

Anderson, J.R., & Thompson, R. (1989). Use of analogy in a production system architecture. in: Vosniadou, S., & Ortony, A. (Eds.) Similarity and analogical reasoning. Cambridge, England: Cambridge University Press, pp.267-297.

Barstow, D.R., et al. (1983). Languages and Tools for Knowledge Engineering. in: Hayes-Roth, F., Waterman, D.A., & Lenat, D.B. (eds.) Building Expert Systems, Vol.1. Reading, MA: Addison-Wesley.

Bergmann, R., Boschert, S. & Schmalhofer, F. Das Erlernen einer Programmiersprache: Wissenserwerb aus Texten, Beispielen und komplexen Programmen. In K. Reiss, M. Reiss und H. Spandl, Hrsg, Maschinelles Lernen - Modellierung von Lernen mit Maschinen. Springer-Verlag, Berlin, 1992, S. 204-224.

Bain, W.M. (1986). Case-Based Reasoning: A Computer Model of Subjective Assessment. PhD thesis, Yale University.

Bajcsy, R. (Ed.) (1993). Proceeding of the 13th International Conference on Artificial Intelligence. August, 28 - September, 3. Chamberry, France. San Mateo, CA: Morgan Kaufmann.

Birk, A., Tschaitshian, B., Schmalhofer, F., Thüring, 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.

Birnbaum, L.A., & Collins, G.C. (1991). Machine Learning, Proceedings of the Eighth International Workshop (ML91). San Mateo, CA: Morgan Kaufmann.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M.K. (1987). Occam's razor. Information Processing Letters, 24, pp. 377-380. ebenso veröffentlicht in: Shavlik, J.W., & Dietterich, T.G. (Eds.) (1990). Readings in machine learning, pp. 201-204. San Mateo, CA: Morgan Kaufmann.

Bourne, L.E., Jr. (1969). Concept learning and thought: behavior not process. in: Voss, J. (ed.) Approaches to thought, pp. 167-195. Columbus, OH: Merrill.

Bourne, L.E., Jr., Ekstrand, B.R., & Dominowski, R.L. (1971). The psychology of thinking. Englewood Cliffs, NJ: Prentice-Hall.

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.

Bruner, J.S., Goodnow, J.J., & Austin, G.A. (1956). A study of thinking. New York: Wiley.

Buchanan, B.G. (1989). Can Machine Learning Offer Anything to Expert Systems?. Machine Learning, 4(3/4), pp. 251-254.

Buchanan, B.G., & Shortliffe, E.H. (1984). Rule-based expert systems: The MYCIN experiments of the Stanford heuristic programming project. Reading, MA: Addison-Wesley.

Carbonell, J.G. (1986). Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. in: Michalski, R.S., Carbonell, J.G., & Mitchell, T.M. (Eds.) Machine Learning, pp. 171-216. Reading, MA: Addison-Wesley.

Dörner, D. (1979). Problemlösen als Informationsverarbeitung. Stuttgart: Kohlhammer.

Dyer, M. (1983). In Depth Understanding. Cambridge, MA: MIT Press.

Eckes, T. (1994). Die Bildung von Kategorien. dieser Band.

Etzioni, O., (1992). An asymptotic analysis of speedup learning. in Sleeman, D. & Edwards, P. (eds) Machine Learning: Proceedings of the Ninth International Workshop pp. 120-126. San

Fischer, G., McCall, R., Ostwald, J., Reeves, B., & Shipman, F. (1994). Seeding, evolutionary growth and reseeding: Supporting the incremental development of design environments. CHI'94 Conference Proceedings, April, 24-28, Boston, MA. Reading, MA: Addison-Wesley.

Ericsson, K.A., Krampe, R. Th., & Tesch-Römer, C. (1993). The role of deliberate practice in the acquisition of expert performance. Psychological Review, Vol. 100, No. 3, pp. 363-406.

Fletcher, Ch., R. (1994) Levels of representation in memory for discourse. in: Gernsbacher, M. A. (Ed) Handbook of Psycholinguistics, New York: Academic Press.

Galambos, J.A. (1983). Normative studies of six characteristics of our knowledge of common activities. Behaviour Research Methods & Instrumentation, 15(3), pp. 327-340.

Gigerenzer, G. (1988). Woher kommen Theorien über kognitive Prozesse? Psychologische Rundschau, 39, pp. 91-100.

Gigerenzer, G. (1991). From tools to theories: A heuristic of discovery in cognitive psychology. Psychological Review, 98, pp. 254-267.

Habel, C., Kanngießer, S., & Strube, G. (1990). Editorial. Kognitionswissenschaft, 1, pp. 1-3.

Habel, C., & Rollinger, C.-R. (1984). Lernen und Wissensakquisition. In: Habel, C. (Hrsg.) Künstliche Intelligenz, Repräsentation von Wissen und natürlichsprachliche Systeme. Frühjahrsschule, Dassel (Solling), März 1984, pp. 249-320. Berlin: Springer-Verlag.

Hammond, K.J. (1986). CHEF: A model of case-based planning. Proceedings of the American Association for Artificial Intelligence (AAAI '86), pp. 261-271.

Hammond, K.J. (1989). Case-based planning: Viewing planning as a memory task. Perspectives in artificial intelligence. Boston, MA: Academic Press.

- Hoffmann, A. (1991). Die Theorie des Lernbaren - ein Überblick. KI, 1, pp. 7-11.
- Hoffmann, J. (1994). Sequentielle Strukturbildung. dieser Band.
- Irtel, H., & Schmalhofer, F.. (1981/82). Psychodiagnostik auf Ordinalskalenniveau: Meßtheoretische Grundlagen, Modelltest und Parameterschätzung. Arch. Psychol., 134, pp. 197-218.
- Janetzko, D., & Strube, G. (1992). Case-based reasoning and model-based knowledge acquisition. in: Schmalhofer, F., Strube, G., & Wetter, T. (Hrsg.) Contemporary knowledge engineering and cognition, Heidelberg: Springer-Verlag, pp. 99-114.
- Kaelbling, L.P. (1987) Learning as an Increase of Knowledge. Manuscript, Artificial Intelligence Center, SRI International.
- Keil, F.C. (1981). Constraints on knowledge and cognitive development. Psychological Review, 88, pp. 197-227.
- Kintsch, W. (1974). The representation of meaning in memory. Hillsdale, NJ: Lawrence Erlbaum.
- Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction-integration model. Psychological Review, Vol. 95, No. 2, pp. 163-182.
- Kintsch, W., Welsch, D., Schmalhofer, F., & Zimny, S. (1990). Sentence memory: A theoretical analysis. Journal of Memory and Language, 29, pp. 133-159.
- Kluwe, R.H., (1993). Knowledge and performance in complex problem solving. in: Strube, G., & Wender, K.F. (eds.) The Cognitive Psychology of Knowledge, pp. 401-423. Amsterdam, North-Holland: Elsevier.
- Kodratoff, Y. (1991). Machine Learning - EWSL-91. European Working Session on Learning. Porto, Portugal, March 1991, Proceedings. Berlin: Springer-Verlag.
- Kolodner, J.L. (1993). Case-Based Reasoning. San Mateo, CA: Morgan Kaufmann.
- 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.
- Laird, J.E., Newell, A., & Rosenbloom, P.S. (1984). Towards chunking as a general learning mechanism. Proceedings of the American Association of Artificial Intelligence (AAAI '84). Austin, Texas: Morgan Kaufmann.
- 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., Lajoie, S.P., Bunzo, M., & Egan, G. (1992). SHERLOCK: A coached practice environment for an electronics troubleshooting job. in: Larkin, J., & Chabay, R. (Eds.) Computer assisted instruction and intelligent tutoring system: Shared issues and complementary approaches. Hillsdale, NJ: Lawrence Erlbaum.
- 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.
- Lewis, C.H. (1988). Why and how to learn why: Analysis-based generalization of procedures. Cognitive Science, 12, pp. 211-256.

Lukas, J. & Albert, D. (1993). Knowledge assessment based on skill assignment and psychological task analysis. in: Strube, G., & Wender, K.F. (Eds.) The cognitive psychology of knowledge, pp. 139-159. Amsterdam, North-Holland: Elsevier.

Mandl, H., & Gruber, H. (1994). Das Entstehen von Expertise. dieser Band.

Mangold, R. (1994). Lernen in Netzwerken. dieser Band.

Mannes, S., & Doane, S. (1994). Beyond discourse: Applications of the construction-integration model. to appear in: Weaver, C., Fletcher, C., & Mannes, S. (Eds.) Discourse comprehension: Strategies and processing revisited. Hillsdale, NJ: Lawrence Erlbaum.

Mannes, S., & Kintsch, W. (1991). Routine computing tasks: Planning as understanding. Cognitive Science, 15(3), pp. 305-342.

Marcus, S. (1989). Introduction: A Sampler in Knowledge Acquisition for the Machine Learning Community. Machine Learning, 4(3/4), pp. 247-250.

McClelland, J.E., & Rumelhart, D.E. (1986). Parallel distributed processing. Cambridge, MA: MIT Press.

McKoon, G., & Ratcliff, R. (1980). Priming item recognition: The organization of propositions in memory for text. Journal of Verbal Learning and Verbal Behavior, 19, pp. 369-386.

McMillan, C., & Smolenski, P. (1988). Analyzing a connectionist model as a system of soft rules. The Tenth Annual Conference of the Cognitive Science Society, 17-19 August 1988, Montreal, Quebec, Canada, pp. 62-68. Hillsdale, NJ: Lawrence Erlbaum.

Medin, D.L., Wattenmaker, W.D., & Michalski, R.S. (1987). Constraints and Preferences in

- Mitchell, T.M. (1982). Generalization as search. Artificial Intelligence, 18, pp. 203-226.
- Mitchell, T.M., Keller, R., & Kedar-Cabelli, S. (1986). Explanation-based generalization: A unifying view. Machine Learning, 1, pp. 47-80.
- Mitchell, T.M., Utgoff, P.E., & Banjerji, R. (1983). Learning by experimentation: Acquiring and refining problem-solving heuristics. in: Michalski, R.S., Carbonell, J.G., & Mitchell, T.M. (Eds.). Machine learning. Los Altos, CA: Morgan Kaufmann.
- Möbus, C. (1988). Zur Modellierung kognitiver Prozesse mit daten- bzw. zielorientierten Regelsystemen. In: Mandl, H., & Spada, H. (Hrsg.) Wissenspsychologie, pp. 423-465. München: Psychologie Verlags Union.
- Mooney, R., & DeJong, G. (1985). Learning schemata for natural language processing. Proceedings of the Ninth International Joint Conference on Artificial Intelligence, pp. 681-687. Los Angeles, CA: Morgan Kaufmann.
- Morik, K. (1993). Maschinelles Lernen. In: Görz, G. (Hrsg.) Einführung in die künstliche Intelligenz, pp. 247-301. Bonn: Addison-Wesley.
- Mross, E.F., & Roberts, J.O. (1992). The construction-integration model: A program and manual. Technical Report, Institute of Cognitive Science, University of Colorado.
- Muggleton, S. H. (1992). Editor. Inductive Logic Programming. Academic Press, New York. NY.
- Musen, M., & Gaines, B. (1994). Proceedings of the 8th Banff Knowledge Acquisition for Knowledge-based Systems Workshop
- Nathan, M., Kintsch, W., & Young, E. (1992). A theory of algebra-word-problem comprehension and its implications for the design of learning environments. Cognition and Instruction, 9(4), pp. 329-389.
- Neches, R., Fikes, R., Finin, T., Gruber, T. Patil, R., Senator, T., & Swartout, W.R. (1991). Enabling technology for knowledge sharing. AI Magazine, Fall.
- Newell, A. (1980). Physical symbol systems. Cognitive Science, 4, pp. 135-183.
- Newell, A. (1982). The knowledge level. Artificial Intelligence, 18, pp. 87-127.
- Newell, A. (1990). Unified theories of cognition. Cambridge, MA: Harvard University Press.
- Newell, A. (1992). Precis of unified theories of cognition. Behavioral and Brain Sciences, 15, pp. 425-437.
- Opwis, K. (1992). Kognitive Modellierung - Zur Verwendung wissensbasierter Systeme in der psychologischen Theoriebildung. Bern: Huber.
- Opwis, K., & Lüer, G. (in press). Modelle der Repräsentation analogen Wissens im Gedächtnis. in: Albert, D., & Stapf, K.H. (Eds.). Enzyklopädie der Psychologie. Gedächtnispsychologie: Erwerb, Nutzung und Speicherung von Information. Göttingen: Hogrefe.
- Perfetti, (1994). Where do propositions come from? to appear in: Weaver, C., Fletcher, C., & Mannes, S. (Eds.) Discourse comprehension: Strategies and processing revisited. NJ: Hillsdale.
- Quinlan, J.R. (1986). Induction of decision trees. Machine Learning, 1, pp. 81-106.

Quinlan, J.R. (1987). Generating Production Rules from Decision Trees. Proceedings of the Tenth International Joint Conference on Artificial Intelligence, pp. 304-307. Milano, Ital.: Morgan Kaufmann.

Reimann, P. & Schult, T.J. (1992) Transforming examples into cases. in Schmalhofer, F., Strube, G. & Wetter, Th. (eds) Contemporary Knowledge Engineering and Cognition. Berlin: Springer-Verlag. pp. 139-145.

Reinartz, T., & Schmalhofer, F. (1994). An integration of knowledge acquisition techniques and EBL for real-world production planning. erscheint in Knowledge Acquisition.

Rendell, L. (1986). A general framework for induction and a study of selective induction. Machine Learning, 1, pp. 177-226.

Richards, B.L., & Mooney, R.J. (1991). First-order theory revision. Proceedings of the Eighth International Workshop on Machine Learning, pp. 447-451. Evanston, IL, June 1991.

Richter, M.M., & Wendel, O. (1993). Lernende Systeme. Informatik-Vorlesungsskript, Universität Kaiserslautern.

Riesbeck, C.K., & Schank, R.C. (1989). Inside Case-Based Reasoning. Hillsdale, NJ: Lawrence Erlbaum.

Rodenhausen, H. (1992). Mathematical aspects of Kintsch's model of discourse comprehension. Psychological Review, Vol. 33. No. 3, pp. 547-549.

Rosenbloom, P.S., Laird, J.E., Newell, A., Golding, A., & Unruh, A. (1986). Current research on learning in SOAR. in: Mitchell, T.M., Carbonell, J.G., & Michalski, R.S. (Eds.) Machine Learning: A guide to current research, pp. 281-290. Boston: Kluwer Academic Publishers.

Russell, S.J. (1985). The Compleat Guide to MRS. Technical Report No. STAN-CS-85-1080, Department of Computer Science, Stanford University.

Samuel, A.L. (1959). Some studies in machine learning using the game of checkers. IBM Journal Research and Development, Vol.3, pp. 210-229. reprinted in: Feigenbaum & Feldman (eds.) Computers and Thought. New York: McGraw-Hill.

Schank, R. (1982). Dynamic memory: A theory of learning and computers and people. New York: Cambridge University Press.

Schank, R.C. (1991). Where's the AI? AI Magazine, Winter, pp. 38-49.

Schank, R., & Abelson, R. (1977). Scripts, plans, goals and understanding. Northvale, NJ: Erlbaum.

Schiefele, U. (1991) Interesse und Textrepräsentation - Zur Auswirkung des thematischen Interesses auf unterschiedliche Komponenten der Textrepräsentation unter Berücksichtigung kognitiver und motivationaler Kontrollvariablen. Zeitschrift für Pädagogische Psychologie, 5, 1991, Heft 4, pp. 245-259.

Schmalhofer, F. (1986). Verlaufscharakteristiken des Informationsabrufs beim Wiedererkennen

Schmalhofer, F., Aitken, S., & Bourne, L.E. (1994). Beyond the knowledge level: Descriptions of rational behavior for sharing and reuse. Research Report RR 94-05. German Research Center for Artificial Intelligence, Germany.

Schmalhofer, F., Albert, D., Aschenbrenner, K.M., & Gertzen, H. (1986). Process traces of binary choices: Evidence for selective and adaptive decision heuristics. The Quarterly Journal of Experimental Psychology, 38A, pp. 59-76.

Schmalhofer, F., Bergmann, R., Boschert, S., & Thoben, J. (1993). Learning program abstractions: Model and empirical validation. in: Strube, G., & Wender, K.F. (Eds.) The cognitive psychology of knowledge, pp. 203-231. Amsterdam, North-Holland: Elsevier.

Schmalhofer, F., Boschert, S., & Kühn, O. (1990). Der Aufbau allgemeinen Situationswissens aus Text und Beispielen. Zeitschrift für pädagogische Psychologie, 4, pp. 177-186.

Schmalhofer, F., & Glavanov, D. (1986). Three components of understanding a programmer's manual: Verbatim, propositional, and situational representations. Journal of Memory and Language, 25, pp. 279-294.

Schmalhofer, F., Kühn, O., & Boschert, St. (1994). The Acquisition and Utilization of Knowledge in Beginners and Advanced Learners. in: Wender, K.F., Schmalhofer, F., & Böcker, H.D. (eds.) Cognition and computer programming. Norwood, NJ: Ablex.

Schmalhofer, F., & Polson, P.G. (1986). A production system model for human problem solving. Psychological Review, 48, pp. 113-122.

Schmalhofer, F., Reinartz, T., & Tschaitschian, B. (1992). Intelligent documentation as a catalyst for developing cooperative knowledge-based systems. in: Wetter, T., Althoff, K.-D.

- Strube, G., & Wender, K.F. (Eds.) (1993). The Cognitive Psychology of Knowledge. Amsterdam, North-Holland: Elsevier.
- Tack, W.H. (1987). Ziele und Methoden der Wissensrepräsentation. Sprache und Kognition, 6, pp. 150-163.
- Tack, W.H., Wallach, D., Unz, D., Henss, R., & Egler, R. (1993). Design and implementation of a representation system for human knowledge. in: Strube, G., & Wender, K.F. (eds.) The Cognitive Psychology of Knowledge, pp. 351-366. Amsterdam, North-Holland: Elsevier.
- Tecuci, G., (1992). Cooperation in knowledge base refinement. in Sleeman, D. & Edwards, P. (eds) Machine Learning. Proceedings of the Ninth International Workshop. pp. 445-450. San Mateo, CA: Morgan Kaufmann.
- Tecuci, G., & Kodratoff, Y. (1990). Apprenticeship learning in imperfect domain theories. in: Kodratoff, Y. and Michalski, R.S. (eds.). Machine Learning: An artificial intelligence approach. Vol 3., pp. 514-551. San Mateo, CA: Morgan Kaufmann.
- Turing, A.M.. (1947). The automatic computing engine.
- Turner, A.A. (1987). The Propositional Analysis System Version 1.0. Technical Report No. 87-2, Institute of Cognitive Science, University of Colorado.
- van den Broek, P. (1990). Causal Inferences and the Comprehension of Narrative Texts. in: Graesser, A.C., & Bower, G.H. (eds.) Inferences and Text Comprehension, pp. 175-196. San Diego: Academic Press.
- van Dijk, T.A., & Kintsch, W. (1983). Strategies of discourse comprehension. New York: Academic Press.
- VanLehn, K., Ball, W., & Kowalski, B. (1990). Explanation-based learning of correctness. in: Minton, S., & Langley, P. (Eds.) Planning, scheduling and learning. Los Altos, CA: Morgan Kaufmann.
- VanLehn, K., Jones, R.M., & Chi, M.T.H. (1991). Modelling the self-explanation effect with Cascade 3. Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society. Hillsdale, NJ: Erlbaum.
- VanLehn, K., Jones, R.M., & Chi, M.T.H. (1992). A model of the self-explanation effect. Journal of the learning sciences, 2, pp. 1-59.
- Vorberg, D., & Goebel, R. (1991). Das Lösen rekursiver Programmierprobleme: Rekursionsschemata. Kognitionswissenschaft, 1, pp. 83-95.
- Waldmann, M. (1994) Wissensgeleitetes Lernen. dieser Band.
- Weber, G. (1993) Fallbasiertes Lernen und Analogien. Habilitationsschrift. Universität Trier.
- Weber, G., Waloszek, G., & Wender, K.F. (1988). The role of episodic memory in an intelligent tutoring system. in: Self, J. (Ed.) Artificial intelligence and human learning: Intelligent computer-aided instruction, London: Chapman & Hall, pp. 141-155.
- Weinert, F.E. (1984). Metakognition und Motivation als Determinanten der Lerneffektivität: Einführung und Überblick. In: Weinert, F.E., & Kluwe, R.H. (Hrsg.) Metakognition, Motivation und Lernen, pp. 9-21. Stuttgart: Kohlhammer.
- Wetter, T., Althoff, K.D., Boose, J., Gaines, B.R., Linster, M., & Schmalhofer, F. (Eds.) (1992). Current developments in knowledge acquisition - EKAW '92. Berlin: Springer-Verlag.

Wilkins, D.C. (1991). A framework for knowledge acquisition and machine learning. Proceedings of the 6th Banff Knowledge Acquisition for Knowledge-based Systems Workshop, pp. 37.1.-37.14

Wrobel, St. (1989). Demand-driven concept formation. in Morik, K., (ed) Knowledge Representation and Organization in Machine Learning. Berlin: Springer-Verlag. pp. 289-319.

Wrobel, St. (1991). Die Umweltverankerung von Begriffsbildungsprozessen. KI, 1, pp. 22-26.

Zwaan, R.A. Effect of genre Expectations on text comprehension. Journal of Experimental Psychology: Learning, Memory, and Cognition.



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.

Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

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.

The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-93-09

Philipp Hanschke, Jörg Würtz:
Satisfiability of the Smallest Binary Program
8 pages

RR-93-10

Martin Buchheit, Francesco M. Donini, Andrea Schaerf: Decidable Reasoning in Terminological Knowledge Representation Systems
35 pages

RR-93-11

Bernhard Nebel, Hans-Jürgen Bürckert:
Reasoning about Temporal Relations:
A Maximal Tractable Subclass of Allen's Interval Algebra
28 pages

RR-93-12

Pierre Sablayrolles: A Two-Level Semantics for French Expressions of Motion
51 pages

RR-93-13

Franz Baader, Karl Schlechta:
A Semantics for Open Normal Defaults via a Modified Preferential Approach
25 pages

RR-93-14

Joachim Niehren, Andreas Podelski, Ralf Treinen:
Equational and Membership Constraints for Infinite Trees
33 pages

RR-93-15

Frank Berger, Thomas Fehrle, Kristof Klöckner, Volker Schölles, Markus A. Thies, Wolfgang Wahlster: PLUS - Plan-based User Support Final Project Report
33 pages

RR-93-16

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

RR-93-17

Rolf Backofen:
Regular Path Expressions in Feature Logic
37 pages

RR-93-18

Klaus Schild: Terminological Cycles and the Propositional μ -Calculus
32 pages

RR-93-20

Franz Baader, Bernhard Hollunder:
Embedding Defaults into Terminological Knowledge Representation Formalisms
34 pages

RR-93-22

Manfred Meyer, Jörg Müller:
Weak Looking-Ahead and its Application in Computer-Aided Process Planning
17 pages

RR-93-23

Andreas Dengel, Ottmar Lutz:
Comparative Study of Connectionist Simulators
20 pages

RR-93-24

Rainer Hoch, Andreas Dengel:
Document Highlighting —
Message Classification in Printed Business Letters
17 pages

RR-93-25

Klaus Fischer, Norbert Kuhn: A DAI Approach to Modeling the Transportation Domain
93 pages

RR-93-26

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

RR-93-27

Hans-Ulrich Krieger:
Derivation Without Lexical Rules
33 pages

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

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

*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-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 Benutzungsoberflächen 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 Wissensrevolution

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

Maschinelles Lernen: Eine kognitionswissenschaftliche Betrachtung
Franz Schmalhofer

D-94-03
Document