

## IX

# Systemkomponenten zum Wissenserwerb

Wolf-Fritz Riekert

Viele der heute mit Hilfe des Computers bearbeiteten Aufgaben erfordern die Lösung von schlecht strukturierbaren Problemen [Newell 69]. Daher ist die erste Realisierung eines Softwaresystems in aller Regel unzulänglich und entspricht nicht den tatsächlichen Bedürfnissen der Anwender. Zu allem Überfluß wandeln sich die Anforderungen an ein System mit dessen Benutzung. Die Folge ist eine nicht enden wollende Reihe von Änderungswünschen. Diese Änderungen kann der Benutzer meist nicht selbständig vornehmen; vielmehr ist er auf die Hilfe von Softwareexperten angewiesen.

Dies ist die Schwachstelle herkömmlicher Anwendersysteme: Die Anpassung an ursprünglich nicht erkannte Anforderungen wird durch das System selbst nicht unterstützt. Das gesamte Expertenwissen, das beim Design des Systems eingebracht wurde, liegt nur in der kodierten Form eines Computerprogramms vor und ist deshalb allein einem Eingeweihten verständlich. Dieser benötigt Programmiersprachen, Editoren und Compiler, um solches Wissen aufzubauen, zu erweitern und zu revidieren. Revisionen erfordern lange Entwicklungs- und Generierzeiten und sind schwierig, riskant und vor allen Dingen teuer. Meist unterbleiben deshalb die notwendigen Programmrevisionen, und der Benutzer muß sich notgedrungen selbst an ein unzulängliches System anpassen.

Ein wichtiges Kriterium für benutzergerechte Computersysteme ist der Grad ihrer *Konvivialität* [Fischer 81c; Illich 73]. Konviviale Systeme zwingen ihrem Benutzer kein fremdes Schema der Arbeitsorganisation auf, sondern lassen sich an dessen eigene Bedürfnisse anpassen. Oberstes Ziel muß sein, daß der Benutzer seine eigene Sichtweise des Problems und seine eigenen Vorstellungen von den Arbeitsabläufen in der rechnerunterstützten Problemlösung wiederfindet. Niemand kennt jedoch diese spezifischen Anforderungen besser als der Benutzer selbst. Daher ist es unabdinglich, daß der Benutzer Kontrolle über die Arbeitsweise des Systems besitzt und die dem System innewohnenden Schemata selbst umgestalten kann.

Einen Weg hin zu diesem Ziel eröffnen sogenannte wissensbasierte Systeme. Das Verhalten dieser Systeme ist im wesentlichen nicht durch Programme bestimmt, sondern vielmehr durch eine *Wissensbasis*, die das anwendungsspezifische Wissen enthält. In dieser liegt das Wissen sozusagen in objektivierter Form vor, es ist zugreifbar und veränderbar. Es ist daher möglich, für wissensbasierte Systeme Komponenten zu entwerfen, die dieses Wissen selbst zum Gegenstand haben. Solche Systemkomponenten werden als *Metasysteme* bezeichnet. Mit Hilfe geeigneter Mensch-Computer-Schnittstellen ermöglichen Metasysteme das Untersuchen, Einbringen, Aktualisieren und Überprüfen des Wissens, das einem Anwendungssystem zugrunde liegt.

Die Anpassung eines Systems an neue Anforderungen erhält so einen gänzlich neuen Charakter: Es ist nicht mehr erforderlich, Programme zu verstehen und zu modifizieren, um das Systemverhalten zu verändern. Die Umprogrammierung eines Systems mit Hilfe von externen Software-Entwicklungswerkzeugen wird abgelöst durch den Vorgang des *Wissenserwerbs* in einem zur Laufzeit umgestaltbaren System. Der Erwerb des Wissens wird dank einer integrierten Metakomponente des wissensbasierten Systems ebenso unterstützt wie die Nutzung des Wissens in herkömmlicher Weise durch die Anwendungskomponente.

## 1. Wissenserwerb

Ursprünglich nur auf menschliche Lernvorgänge angewandt, wird heute der Begriff Wissenserwerb auf den Bereich maschineller Intelligenz übertragen. Im heutigen Sprachgebrauch wird unter dem Begriff *Wissenserwerb* oder *Wissensakquisition* sowohl die *Aneignung von Wissen durch Menschen* wie auch die *Übertragung von Wissen auf Computersysteme* verstanden.

In diesem Kapitel ist ganz allgemein der Erwerb des Wissens gemeint, das zum Lösen von Problemen in einem Aufgabenbereich erforderlich ist. In [Michalski, Carbonell, Mitchell 83] wird Wissenserwerb definiert als das Lernen neuer symbolischer Information zusammen mit der Fähigkeit, diese Information auf effektive Weise anzuwenden.

Wissenserwerb ist jedoch nicht mit Lernen an sich gleichzusetzen. Lernen umfaßt außer dem Erwerb von Wissen auch das Einüben von Fähigkeiten. Während das eine ein Prozeß der Bewußtmachung von Fakten und Prinzipien ist, stellt das andere die Aneignung von eher unbewußten Handlungsabläufen dar. Dieser Aspekt des Lernens, der sehr anschaulich in [Fischer et al. 78] beschrieben ist, soll in diesem Kapitel nicht betrachtet

werden. Ebenfalls außer Betracht bleiben soll die maschinelle Unterstützung menschlicher Wissenserwerbsvorgänge durch rechnerunterstütztes Lernen [Gunzenhäuser 84].

Wir werden maschinelle und menschliche Lernvorgänge gleichermaßen untersuchen, jedoch mit dem Ziel, Wissenserwerbstechniken für automatische Wissensverarbeitungssysteme abzuleiten. Der Zusammenhang zwischen menschlichen und maschinellen Wissenserwerbsvorgängen wird besonders deutlich am Beispiel von sogenannten *Expertensystemen*. Diese Systeme übernehmen auf einem definierten Anwendungsgebiet die Aufgaben eines menschlichen Experten; es wird daher erforderlich, Expertenwissen, das bisher nur Menschen zu eigen war, auf Computersysteme zu übertragen.

Bei Wissenserwerbsvorgängen lassen sich verschiedene Strategien unterscheiden: auf der einen Seite eher passive, bei denen eine Lehrperson den Lehrstoff aufbereitet und dem Lernenden vermittelt und im Extremfall regelrecht "eintrichtert", auf der anderen Seite die aktiven Strategien, bei denen der Lernende mit Hilfe von Analogieschlüssen oder anderen Schlußtechniken das Wissen selbständig aus vorgelegten Beispielen oder gar eigenen Beobachtungen ableitet. Die hier vorgestellten Wissenserwerbskomponenten von Softwaresystemen haben eher passiven Charakter, da zur Wissensübertragung ein Gebietsexperte als "Lehrer" erforderlich ist; sie sind aber auch aktiv in dem Sinne, daß sie ihren menschlichen Bediener beim Einbringen des Wissens aktiv unterstützen und das erworbene Wissen selbständig in eine Wissensbasis einbetten und konsistent halten.

## 2. Psychologische Aspekte

Es ist von Nutzen, Wissenserwerbsmechanismen des Computers und Lernvorgänge des Menschen parallel zu betrachten. Für Psychologen bedeutet dies die Möglichkeit, Modelle menschlicher Wissenserwerbsvorgänge in der Computersimulation auf ihre Schlüssigkeit zu überprüfen. Für Computerwissenschaftler sind menschliche Lernvorgänge Vorbilder bei der Gestaltung wissensverarbeitender Systeme.

Piaget unterscheidet in seiner *Psychologie der Intelligenz* [Piaget 72] zwei Arten der Anpassung an die Umwelt: Assimilation und Akkommodation. Im biologischen Sinn ist *Assimilation* ein Stoffwechselfvorgang, bei dem körperfremde Substanzen in körpereigene umgewandelt und in den eigenen Organismus eingebaut werden. *Akkommodation* bezeichnet den Vorgang der Anpassung an die Umwelt, indem die eigenen Assimilationsmechanismen verändert werden.

Piaget wendet diese Begriffe auf die geistige Auseinandersetzung des Menschen mit seiner Umwelt an. Für J. Moore und A. Newell, die die Frage aus der Sicht der Künstlichen-Intelligenz-Forschung betrachten, ist die Fähigkeit zu Akkommodation und Assimilation eine notwendige Eigenschaft eines Programmes, das Probleme verstehen können soll (*How can Merlin understand?* [Moore, Newell 74]).

Auf die Vorgänge der Wissensverarbeitung bezogen, ist Assimilation ein Wissenserwerbsvorgang, bei dem Informationen, die von außen stammen, den eigenen Wissensstrukturen eingepaßt werden. Akkommodation hingegen ist die Anpassung der eigenen Wissenserwerbsmechanismen an neue Erfordernisse. Bei beiden Vorgängen spielt Wissen eine zweifache Rolle: Zum einen haben sie Wissen zum Gegenstand, zum andern laufen sie wissensgesteuert ab. Es ergibt sich die Möglichkeit, Wissen danach zu klassifizieren, welche Rolle es bei Assimilation und Akkommodation spielt.

## 2.1 Formen von Wissen

Das beim Wissenserwerb bedeutsame Wissen läßt sich in die folgenden drei Kategorien einteilen:

1. *Sachwissen*: Dies ist Wissen um konkrete Sachverhalte aus einem Problemraum. Es zeigt sich in der Kenntnis von Objekten, deren Eigenschaften und wechselseitigen Beziehungen. Sachwissen ist diejenige Form von Wissen, die sich am schnellsten verändert und häufig nur befristete Gültigkeit besitzt.
2. *Konzeptuelles Wissen*: Dies ist Wissen über den Umgang mit Sachwissen. Im konzeptuellen Wissen liegen die Schemata, die erforderlich sind, um Sachwissen zu erschließen, einzuordnen, zu memorieren und zu nutzen. Nach außen hin ist diese Art von Wissen erkennbar als die Fähigkeit, Problemstellungen aus einem Sachgebiet aufzunehmen und zu lösen.
3. *Metawissen*: Dies ist Wissen über den Umgang mit konzeptuellem Wissen. Metawissen bildet den begrifflichen Rahmen, der erforderlich ist, um konzeptuelles Wissen zu erwerben, anzuwenden und zu erklären. Nach außen hin drückt sich Metawissen in der Fähigkeit aus, neue Problemräume zu erfassen und das eigene Verhalten beim Lösen von Aufgaben zu reflektieren und zu begründen.

## 2.2 Assimilation

Der Erwerb von Sachwissen ist ein Assimilationsvorgang. Er wird gesteuert durch das vorhandene konzeptuelle Wissen des Lernenden. Der Lernende erkennt in den Oberflächenstrukturen der Außenwelt Ausprägungen ihm bekannter Schemata und kann daher

das Ergebnis seiner Wahrnehmungen in die interne Struktur seines Sachwissens einordnen. Dieses Sachwissen kann beim Lösen von Problemen genutzt werden und bestimmt so das künftige Verhalten des Wissensträgers.

Ein Beispiel hierfür ist die Tätigkeit eines Sachbearbeiters in einer Versicherung. Das konzeptuelle Wissen des Sachbearbeiters besteht in der Kenntnis der verschiedenen Klassen von Geschäftsvorgängen. Jeder Vorfall, beispielsweise die Schadensmeldung eines Kunden, wird in ein geeignetes Schema eingepaßt, in diesem Fall vielleicht in den Geschäftsvorfall "Kraftschadenregulierung". Damit lassen sich die Angaben des Kunden in die gebräuchliche Vorgangsstruktur einordnen, und die Bearbeitung des Falls kann nach Sachlage erfolgen.

Beim Assimilationsvorgang werden also von außen kommende Elemente für eine bestehende interne Struktur passend gemacht und in diese eingebaut. Das Schema, nach dem die interne Struktur aufgebaut ist, bleibt unverändert. Auf Softwaresysteme übertragen bedeutet dies, daß neue Datenobjekte erzeugt werden, die aber bekannten abstrakten Datentypen angehören. Die abstrakten Datentypen selbst und die Methoden für deren Verarbeitung bleiben unberührt.

## 2.3 Akkommodation

Der Erwerb vom konzeptuellem Wissen ist ein Akkommodationsvorgang. Er ist erforderlich, wenn die Assimilationsmechanismen nicht mehr ausreichen, um die Erscheinungen der Umwelt zu erfassen. Es werden die Schemata des konzeptuellen Wissens verändert und bereichert, damit der Umgang mit neuen Formen von Sachwissen möglich wird. Die menschlichen Fähigkeiten zur Akkommodation kommen zum Ausdruck, wenn es darum geht, neue Aufgabenfelder zu beherrschen. Alles Lernen, das mehr ist als bloßes Auswendiglernen von Einzelfakten, ist stets ein Akkommodationsvorgang.

In Softwaresystemen bedeutet Akkommodation, daß sich das Verhalten des Programmes beim Lösen von Problemen qualitativ verändert. Auf die Implementierung bezogen heißt dies, daß neue Datentypen und neue Verarbeitungsmethoden geschaffen werden müssen. Dazu sind herkömmliche Softwaresysteme selbständig noch nicht in der Lage. Die Akkommodationsaufgaben müssen von Systemanalytikern und Programmierern vorgenommen werden.

Für den Akkommodationsvorgang wird konzeptuelles Wissen einer höheren Stufe benötigt: Der Erwerb von konzeptuellem Wissen wird gesteuert durch Metawissen. Metawissen ist Wissen über den Umgang mit konzeptuellem Wissen, d.h. im Metawissen liegen die Schemata, nach denen neue Schemata erworben werden können. Am Beispiel einer Versicherungsgesellschaft läßt sich dies wiederum verdeutlichen. Wenn eine Versicherung neue Dienste anbietet, zum Beispiel eine Haftpflichtversicherung für Privatflugzeuge, so ist es erforderlich neue Schemata für Geschäftsvorfälle in diesem Bereich zu entwickeln, es ist also ein Akkommodationsproblem zu lösen. Eine andere Betrachtungsweise führt dies auf eine Assimilationsaufgabe zurück: Es gibt ein Metaschema, das für alle Versicherungen zutrifft, und das Konzept der neuen Dienstleistung muß anhand dieses Metaschemas klassifiziert werden.

### 3. Wissensverarbeitung

Jeder Problemlösevorgang ist zugleich auch ein Vorgang der *Wissensverarbeitung*, ganz gleich ob er von Menschen oder von Computerprogrammen ausgeführt wird: Gegeben ist ein Ausgangszustand, ein Ziel und eine Menge von Wissensquellen. Die Aufgabe besteht darin, die geeigneten Wissensquellen aufzufinden und aus diesen Aktionsfolgen abzuleiten, die vom Ausgangszustand zum Zielzustand führen. Von entscheidender Bedeutung ist dabei, in welcher Form diese Wissensquellen dargestellt, wie sie kodiert sind und wie sie verwendet werden.

Bei der *Wissensnutzung* steht die Frage im Vordergrund, "wie" im konkreten Fall ein Problem gelöst werden soll. Unter dem Aspekt der Nutzung ist daher eine "compilierte Darstellung" des Wissens wünschenswert: Dabei tritt das gesamte Wissen nur noch *implizit* in Erscheinung, und zwar in Form von vorgefertigten Anweisungsfolgen, die im gegebenen Fall nur noch befolgt werden müssen. Diese Art der Notation von Wissen setzt nur geringe Intelligenz bei der Instanz voraus, die das Wissen anwenden soll; sie ist infolgedessen optimal für eine maschinelle Verarbeitung. Eine compilierte Wissensdarstellung ermöglicht schnelle und effiziente Problemlösungen, sie ist für Standardaufgaben gut geeignet, kann aber in Ausnahmefällen versagen, da es nicht möglich ist, alle Situationen vorherzusehen.

Diese Anforderungen an die Wissensdarstellung kollidieren mit denen, die sich beim Vorgang des *Wissenserwerbs* stellen. Wissensvermittlung, die sich auf die Weitergabe von

bloßen Rezepten beschränkt, wird stets zu unbefriedigenden Ergebnissen führen, weil dabei kein tieferes Verständnis der zugrundeliegenden Sachverhalte entsteht. Einfacher, präziser und natürlicher ist es, nicht die Lösungsverfahren zu beschreiben, sondern das Problemgebiet selbst mit all seinen Eigenschaften, Gesetzmäßigkeiten und Abhängigkeiten. Dann erst wird klar, "was" die Eigenheiten des Problemraums sind, weil das Wissen darüber *explizit* vorliegt. Bei der Lösung eines konkreten Problems besteht aber dann die Notwendigkeit, das vorhandene Wissen zu interpretieren und in Methoden, Pläne und Aktionen umzusetzen.

### 3.1 Wissensdarstellung

Die Wissensdarstellung in herkömmlichen Softwaresystemen ergibt folgendes Bild: Explizit dargestellt ist in aller Regel das Sachwissen, und zwar als Inhalt von Dateien oder hochentwickelten Datenbanksystemen. In den meisten Computerprogrammen läßt sich das konzeptuelle Wissen jedoch nicht mehr definitiv lokalisieren, da es zu Kontrollstrukturen und Anweisungsfolgen transformiert ist. Das Metawissen schließlich führt eine vom Programm abgekoppelte Existenz im Kopf des Systementwicklers oder bestenfalls in der Programmdokumentation.

Die Darstellung konzeptuellen Wissens in herkömmlichen Systemen ist also völlig auf die Belange der Wissensnutzung abgestimmt. Die prozeduralen Aspekte des konzeptuellen Wissens stehen im Vordergrund. Diese Aspekte sind durchaus wichtig für die Performance des Programmsystems. Sie tragen aber nichts für den Wissenserwerb bei.

Es ist also notwendig, das konzeptuelle Wissen unter multiplen Perspektiven betrachten zu können: einmal in *deskriptiver* Form für die Zwecke des Wissenserwerbs, zum andern in *prozeduraler* Form zum Zweck der Wissensnutzung. Dies wird möglich durch eine objektorientierte Darstellung aller drei Formen von Wissen (Sachwissen, konzeptuelles Wissen und Metawissen) in einer *Wissensbasis*.<sup>1</sup>

### 3.2 Objektorientierte Wissensrepräsentation

In unseren Systemen ist das Wissen in objektorientierter Form dargestellt; die zugrundeliegenden Objekte sind Klassen und Instanzen, die mit Hilfe der Wissensrepräsentations-

---

<sup>1</sup>Zum Begriff der Wissensbasis siehe Kapitel II.

sprache ObjTalk<sup>2</sup> definiert werden. ObjTalk vereinigt die deskriptiven und die prozeduralen Aspekte des konzeptuellen Wissens:

1. *Objekte besitzen Merkmale und Merkmalswerte*, die in Form von sogenannten Slots dargestellt sind. Da Slots Referenzen auf andere Objekte enthalten können, läßt sich Wissen in Form von semantischen Netzen [Quillian 68] darstellen. Merkmale lassen sich beschreiben und lesen. Objekte lassen sich erzeugen und löschen. Dadurch kann das Wissen der Wissensbasis zu jedem Zeitpunkt manipuliert werden. Die Belegung von Slots mit Merkmalswerten berücksichtigt den deskriptiven Aspekt der Wissensdarstellung.
2. *Objekte verfügen über Methoden*. Durch Botschaftenaustausch zwischen Objekten werden diese Methoden aktiviert. In den Methoden ist festgelegt, wie sich ein Objekt auf eine eintreffende Botschaft hin verhält. Methoden lassen sich ansehen als Programmstücke, bei deren Ausführung entweder Merkmale des betreffenden Objektes verändert werden oder aber neuerlich Botschaften an andere Objekte versandt werden. Daher kommt in den Methoden der prozedurale Aspekt des Wissens zum Ausdruck.

Was bleibt, ist das Problem, wie die prozeduralen und die deskriptiven Elemente der Objekte ineinander überführt und miteinander konsistent gehalten werden können. ObjTalk bietet hierzu Hilfen durch seine Constraint-Mechanismen und die Möglichkeit, Regeln zu formulieren. Eine völlig anwendungsunabhängige Lösung erscheint aber prinzipiell nicht möglich. Die Transformation der beiden Aspekte des konzeptuellen Wissens muß daher in jedem Anwendungssystem von neuem gelöst werden. Das zur Lösung dieser Transformationsaufgabe erforderliche Wissen - das Metawissen - läßt sich aber wiederum in der objektorientierten Wissensbasis darstellen.

### 3.3 Interaktionsformen

Die Benutzerschnittstelle für die Wissenserwerbskomponente eines Softwaresystems sollte so gestaltet sein, daß sie das Einbringen des Wissens in einer deskriptiven Darstellungsform auf einfache Weise gestattet. Es sind hierfür prinzipiell drei Arten von Interaktionsformen denkbar: die Verwendung (eventuell vereinfachter) natürlicher Sprache, die Verwendung einer formalen Sprache (z.B. aussagenlogische Ausdrücke) oder die direkte Manipulation von graphischen Objekten auf dem Bildschirm.

In unseren interaktiven Systemen bevorzugen wir die *direkte Manipulation* von graphischen Objekten. Dabei zeigt der Bildschirm einen Ausschnitt aus der Wissensbasis, wo-

<sup>2</sup>Siehe Kapitel III.



bei die Objekte der Wissensbasis in Form von Piktogrammen und Formularen des Fenstersystems<sup>3</sup> dargestellt sind. Der Benutzer erhält eine graphische Veranschaulichung des Wissens über die Welt des Anwendungsgebiets, so wie sie dem System bekannt ist. Diese objektorientierte externe Darstellungsform des Wissens hat zum einen den Vorteil, daß sie den modellhaften Vorstellungen des Benutzers sehr nahekomm, da dieser die Objekte der realen Welt in ihren Visualisierungen wiedererkennen kann, zum andern läßt sich so eine enge Kopplung mit der ebenfalls objektorientierten internen Repräsentation des Wissens herstellen.

Der Vorgang des Wissenserwerbs geschieht nun dadurch, daß neue graphische Objekte auf dem Bildschirm erzeugt bzw. bereits existierende verändert werden. Mit Hilfe eines Zeigeelements, beispielsweise einer Maus, kann der Benutzer über sensitive Bildschirmbereiche Funktionen auslösen und die zu manipulierenden Objekte bezeichnen. Funktionen sind visualisiert durch Menüs und Piktogramme<sup>4</sup>. Objekte werden entweder durch unmittelbare textuelle Ersetzung von Formularinhalten oder durch Auswahl von Funktionen aus kontextabhängigen Menüs modifiziert.

Gegenüber den anderen oben erwähnten Interaktionsformen hat die direkte Manipulation eine Reihe von Vorzügen:

- *Das sogenannte WYSIWYG-Prinzip<sup>5</sup> ist erfüllt.* Modifikationen der Wissensbasis werden direkt an den auf dem Bildschirm dargestellten Objekten vorgenommen, und die Auswirkungen dieser Aktionen werden auf dem Bildschirm unmittelbar sichtbar.
- *Das Wissen kann in deskriptiver Form in das System eingebracht werden.* Die Verwendung von Formularen erlaubt eine einfache Spezifikation von Attributen und Merkmalen. Syntaktische Konventionen, die bei der Verwendung von formalen Sprachen auftreten, müssen nicht beachtet werden.
- *Das aktive Erinnerungsvermögen des Menschen wird entlastet.* Die Auswahl aus Menüs und das Ausfüllen von Formularen ist einfacher als die eindeutige sprachliche Formulierung von Sachverhalten, da der Benutzer im Angebot der Manipulationsmöglichkeiten lediglich die wiedererkennen muß, die für ihn von Bedeutung sind.

---

<sup>3</sup>Siehe Kapitel V.

<sup>4</sup>Siehe Kapitel IV.

<sup>5</sup>WYSIWYG = "What you see is what you get."

#### 4. Das System D&I als Wissenseditor

Digester und Informant (D&I)<sup>6</sup> ist ein interaktives System, das einen Sachbearbeiter in einem Dokumentations- und Informationsinstitut beim Sammeln und Verwalten von Fachinformationen unterstützt. Die Begriffe *Digester*<sup>7</sup> und *Informant* stehen für die beiden Hauptaufgaben des Systems, den Erwerb und die Nutzung von Fachwissen. In der vorliegenden Implementation des D&I beschränken wir uns auf die Verarbeitung von Berichten aus der Computerindustrie; die betrachtete Wissenswelt umfaßt die Beschreibung von Sachverhalten und Ereignissen, die Firmen, Personen, Produkte und Projekte aus der Computerszene zum Gegenstand haben.

Die Wissenserwerbskomponente des Systems, der Digester, bietet eine Unterstützung bei zwei Arten von Vorgängen an:

1. Beim *Erwerb von Sachwissen* liegen die zu verarbeitenden Fachinformationen in unstrukturierter Form als Inhalt von Zeitungsartikeln und anderen Veröffentlichungen vor. Mit Hilfe des D&I kann ein Sachbearbeiter die eintreffende Fachinformation in eine strukturierte Form überführen und in eine Wissensbasis einbetten.
2. Bei der Anpassung des Systems an neue Anwendungsgebiete stellt der Digester ein Werkzeug zum *Erwerb des erforderlichen konzeptuellen Wissens* dar. Ein Organisationsexperte kann das System für eine Verarbeitung von Sachwissen aus anderen Fachgebieten umrüsten.

Ziel der Forschung war es, die Wissenserwerbskomponente des Systems so zu gestalten, daß der Benutzer beide Aufgaben lösen kann, ohne DV-Spezialist sein zu müssen.

Die Wissensbasis des D&I ist objektorientiert aufgebaut. Sachwissen, konzeptuelles Wissen und Metawissen sind durch Eigenschaften von Objekten repräsentiert. Der Wissenserwerb geschieht generell dadurch, daß die Eigenschaften vorhandener Objekte der Wissensbasis modifiziert werden oder neue derartige Objekte erzeugt werden [Csima, Riekert 83b]. Die Benutzerschnittstelle des Systems D&I [Csima 83] erlaubt die direkte Manipulation von Objekten mit Hilfe von Bildschirmformularen, die einen Aspekt der Wissensbasis zeigen. Die Interpretation von Ereignissen und Sachverhalten sowie die *Navigation*

---

<sup>6</sup>Das System D&I wurde im Rahmen eines Förderprojekts des Ostasien-Instituts (Bonn) von Feodora Csima und dem Autor dieses Kapitels entwickelt.

<sup>7</sup>To digest (engl.) = verdauen.

[Fischer 83b] in der Wissensbasis werden unterstützt durch eine Dialogführung, die auf sogenannten Skriptmenüs beruht. Der Prototyp des Systems D&I läuft auf einer VAX 11/780 und ist mit Hilfe von FranzLisp als Programmiersprache und ObjTalk als Wissensrepräsentationssprache implementiert.

## 5. Erwerb von Sachwissen mit dem System D&I

Am Beispiel eines einfachen Zeitungsartikels soll der Vorgang des Erwerbs von Sachwissen mit Hilfe des Systems D&I deutlich gemacht werden. Der betreffende Artikel ist am 7. Januar 1982 in der Computerzeitung erschienen:

*Dipl.-Ing. J. Benno Durst, 42, übernahm die Leitung des Werkes für Medizinelektronik der Hewlett-Packard GmbH, Böblingen. Nach HP-Angaben wurde sein Vorgänger, Dr. Otto Brand, zum Leiter des Werkes für Tischcomputer berufen.<sup>8</sup>*

Zeitungsartikel wie dieser beschreiben Objekte aus einem bestimmten Sachgebiet und Ereignisse, die diese Objekte betreffen [Rosenberg 77]. Der Begriff "Objekt" ist dabei in einem allgemeinen Sinn zu verstehen, der alle Objekte unseres Denkens umfaßt, die unverwechselbare Eigenschaften tragen; der Leser mag ihn nach Belieben durch die Worte Gegenstand, Wesenheit, Gebilde oder begriffliche Einheit ersetzen, wann immer eines von diesen treffender erscheint.

### 5.1 Die Wissensbasis

Die in diesem Zeitungsartikel erwähnten *Objekte* sind die Herren Durst und Brand, deren frühere und jetzige Tätigkeiten sowie die Hewlett-Packard GmbH. Jedes dieser Objekte läßt sich einem bestimmten *Typ* zuordnen. In der Zeitungsmeldung ist insgesamt von drei Typen die Rede: PERSON, TÄTIGKEIT und FIRMA. Zwei *Ereignisse* werden beschrieben: die Neubesetzung einer Stelle und ein Stellenwechsel. Ziel des Wissenserwerbs mit dem D&I ist es, diese Objekte und die Implikationen der beschriebenen Ereignisse in der Wissensbasis des Systems D&I zu repräsentieren.

Wir wollen nun einmal annehmen, daß die bisherige Tätigkeit von Dr. Brand als Leiter des Werks für Medizinelektronik der Firma HP dem System bereits aufgrund einer früher verarbeiteten Zeitungsmeldung bekannt ist. Das heißt, in der Wissensbasis sind bereits

---

<sup>8</sup>Die Namen der beiden Personen wurden vom Verfasser geändert.

drei Objekte angelegt, die PERSON "Brand", die TÄTIGKEIT "Leitung Medizinelektronik" und die FIRMA "HP". Diese Objekte tragen *Merkmale*, die mit *Merkmalswerten* belegt sind. Merkmale einer Person sind beispielsweise ihr Titel, ihre Adresse oder ihre berufliche Tätigkeit. Die Merkmalswerte sind entweder Texte, wie der Titel "Dr.", oder aber Bezeichnungen anderer Objekte. Auf die letztere Art lassen sich Beziehungen zwischen Objekten darstellen, wie etwa die Tatsache, daß Dr. Brand leitender Mitarbeiter der Firma HP ist. Die Objekte bilden so ein Netz. Abbildung IX-1 zeigt dieses Netz für unseren Beispielfall.

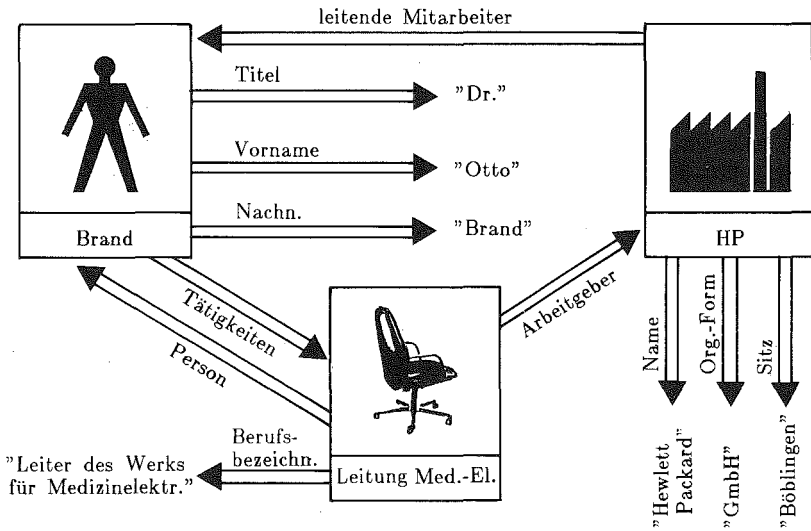


Abbildung IX-1: Zustand der Wissensbasis vor der Verarbeitung

Die Verarbeitung der im Zeitungsbericht beschriebenen Ereignisse mit Hilfe des Digesters soll nun zwei Arten von Änderungen an der Wissensbasis bewirken:

1. Es werden neue Objekte mit ihren Merkmalen in der Wissensbasis angelegt und mit den vorhandenen Objekten verknüpft.
2. Die Objekte in der Wissensbasis erhalten zusätzliche Merkmale beziehungsweise bereits vorhandene Merkmale erhalten neue Werte.

## 5.2 Die Benutzerschnittstelle

Abbildung IX-2 zeigt den Bildschirmaufbau des Systems D&I bei einer derartigen Verarbeitung. Der Bildschirm ist aufgeteilt in vier Fenster:

1. Das *Arbeitsfenster* enthält ein Formular, das einen Aspekt, das heißt einen Teil der gesamten Merkmale und Merkmalswerte des gerade betrachteten Objekts in einzelnen Formularfeldern zeigt.
2. Das *Kopffenster* enthält Statusinformationen, nämlich den Namen des aktuell bearbeiteten Objekts, seinen Typ (z.B. Person, Firma etc.) und die Bezeichnung des gewählten Aspekts. Außerdem werden dort kurze Abfragen abgewickelt.
3. Das *Skriptfenster* enthält ein Menü, dessen Inhalt vom gerade gezeigten Aspekt abhängig ist. Das Menü enthält die Namen von sogenannten Skripts; dabei handelt es sich um Funktionen, die der Benutzer auswählt, um die zu verarbeitenden Ereignisse zu interpretieren.
4. Das *Kommandomenü* bietet kontextunabhängige Funktionen zur Navigation in der Wissensbasis, zum Rückgängigmachen fehlerhafter Eingaben (Undo) und zum Abrufen von Hilfe und allgemeinen Informationen an.

*** D&I *** Bearbeitetes Objekt: Leitung des Werks Medizinelektronik vom Typ TAETIGKEIT unter dem Aspekt TAETIGKEIT			
FELDNAME:	FELDKONTENT:	SKRIPTS	KOMMANDO
Person:	Durst_Benno	-Personenbericht	BEARBEITE
Arbeitgeber:	Hewlett-Packard	-Zeige Arbeitgeber	Objekt
Berufs- Bezeichnung:	Leitung des Werks Medizinelektronik		Aspekt
Aufgabengebiet:			Zurueck
			INFORMIERE
			?Typen
			?Protokoll
			VERWALTUNG
Vorgaenger:	Brand_Otto		Vergessen
Nachfolger:	<Nachname_Vorname>...		Retten
			Neubeginn
			Ende
Begonnen:	<TT-MM-JJ>		SYSTEM
Beendet:	<TT-MM-JJ>		Eval

Abbildung IX-2: Bildschirmaufbau des Systems D&I

Zu jedem Zeitpunkt kann über eine *Hilfetaste* ein kontextabhängiger Hilfetext abgerufen werden, der den Bildschirm teilweise überlagert; falls das System die Eingabe von Werten erwartet, erscheint zusätzlich ein Menü von Eingabeparametern, unter denen der Benutzer auswählen kann. Mittels einer *Abbruchtaste* kann jeder eingeleitete Verarbeitungsschritt wieder abgebrochen werden, ohne daß inkonsistente Systemzustände entstehen.

### 5.3 Identifikation von Objekten

Im Grundzustand enthält das Skriptfenster ein Menü aller verarbeitbaren Arten von Berichten. Der Sachbearbeiter wählt das Skript "Personenbericht" aus. Im sog. Skriptdialog, der daraufhin im Kopffenster abläuft, wird der Name einer beteiligten Person erfragt. Der Sachbearbeiter gibt den Namen "Durst Benno" ein. Das System meldet, daß die genannte Person ihm noch nicht bekannt ist, und bittet den Sachbearbeiter zu bestätigen, daß ein entsprechendes Objekt in der Wissensbasis neu erzeugt werden soll.

Dies ist ein ganz typischer Vorgang beim Erwerb von Wissen: Es liegt ein Objekt vor, von dem bestimmte Eigenschaften bekannt sind. Zunächst muß das Objekt nach seinem Typ *klassifiziert* werden, im gegebenen Fall als Person. Dieser Vorgang wird vom Digester durch das angebotene Menü unterstützt. Anschließend wird das Objekt *identifiziert*. Dies geschieht, indem seine Eigenschaften mit denen der bereits bekannten Objekte verglichen werden. Im gegebenen Fall sucht der Digester nach einer Person mit gleichem oder ähnlichem Namen. Wenn die Suche fehlschlägt, wird ein neues Objekt in der Wissensbasis angelegt.

### 5.4 Navigation zu Aspekten

Nach dem Identifikationsvorgang wechseln die Bildschirmfenster ihren Inhalt und beziehen sich auf die Person Benno Durst als Privatmann. Im Arbeitsfenster wird nur ein Objektausschnitt, der *Aspekt* PRIVAT der Person dargestellt. Aspekte haben unter anderem die Funktion von Filtern. Sie gewährleisten, daß nur diejenigen Informationen sichtbar werden, die in der gegenwärtigen Dialogphase relevant sind.

Die Art der gerade bearbeiteten Information ist auch ein Kriterium dafür, welche Verarbeitungsschritte voraussichtlich als nächste vom Benutzer vorgenommen werden. Im System D&I bestimmt daher der aktuelle Aspekt den Inhalt des Skriptmenüs. Das Skriptmenü enthält ein Angebot von Funktionen, die in der Regel zu diesem Zeitpunkt der Verarbeitung benötigt werden.

Es lassen sich zwei Arten von Skripts unterscheiden, die auch gemischt auftreten können. Zum einen können Skripts zur Interpretation von Sachverhalten dienen, indem sie eine entsprechende Modifikation der Wissensbasis bewirken. Diese Skripts bezeichnen wir als *Interpretationsskripts*. Zum andern können Skripts einen Kontextwechsel bewirken, indem sie auf einen neuen Aspekt umschalten. Sie tragen dann den Namen *Navigations-skript*. Das oben beschriebene Skript "Personenbericht" gehört zu beiden Kategorien. Es führt die Identifikation eines Objekts durch und bewirkt einen Wechsel des Aspekts.

## 5.5 Eintrag von Merkmalen

Wir wollen nun ein paar Schritte bei der Verarbeitung unseres Artikels überspringen. Mit Hilfe eines Navigations- und eines Interpretationskripts ist der Sachbearbeiter bei einem Aspekt angelangt, der die neuangetretene Tätigkeit von Herrn Durst zeigt (siehe Abbildung IX-2). Die Berufsbezeichnung "Leitung des Werks Medizinelektronik" und der Arbeitgeber "HP" wurden bereits bei der Identifikation der Tätigkeit vom System erfragt und sind schon in das Formular im Arbeitsfenster eingetragen. Der Arbeitnehmer in dieser Tätigkeit ist Herr Durst selbst. Das entsprechende Feld wurde vom System automatisch gefüllt.

Die übrigen Formularfelder sind nicht mit Informationen gefüllt und können vom Benutzer beschrieben werden. Das Formularfeld "Vorgänger:" ist mit dem Syntaxvorschlag <Nachname\_\_Vorname> vorbelegt. Der Sachbearbeiter gibt den Namen Brand\_\_Otto ein. Diese Eintragung wird vom System ohne Rückfrage akzeptiert, da es die Person ohne weiteres identifizieren kann. Die neu in die Wissensbasis übernommenen Objekte sind nun mit den bereits vorhandenen verknüpft. Das Beschreiben des Formularfelds bewirkte die unmittelbare Veränderung des zugehörigen Merkmalswertes.

## 5.6 Inferenzen des Systems

Der Eintrag des Vorgängers bewirkte jedoch noch mehr als nur die Veränderung des zunächst betroffenen Objekts, also der Tätigkeit. Aus der Eigenschaft, daß Herr Brand Vorgänger von Herrn Durst ist, kann geschlossen werden, daß eine gleichartige Tätigkeit von Herrn Brand existieren muß. Diese wird vom Digester auch identifiziert, da Berufsbezeichnung und Arbeitgeber übereinstimmen. Wäre die Tätigkeit nicht bekannt gewesen, so hätte das System eine solche neu angelegt. In dieser Tätigkeit kann nun automatisch als Nachfolger Herr Durst eingetragen werden. Da nun der Nachfolger feststeht, wechselt die Tätigkeit ihre Rolle als jetzige Tätigkeit von Herrn Brand und wird zu einer früheren Tätigkeit.

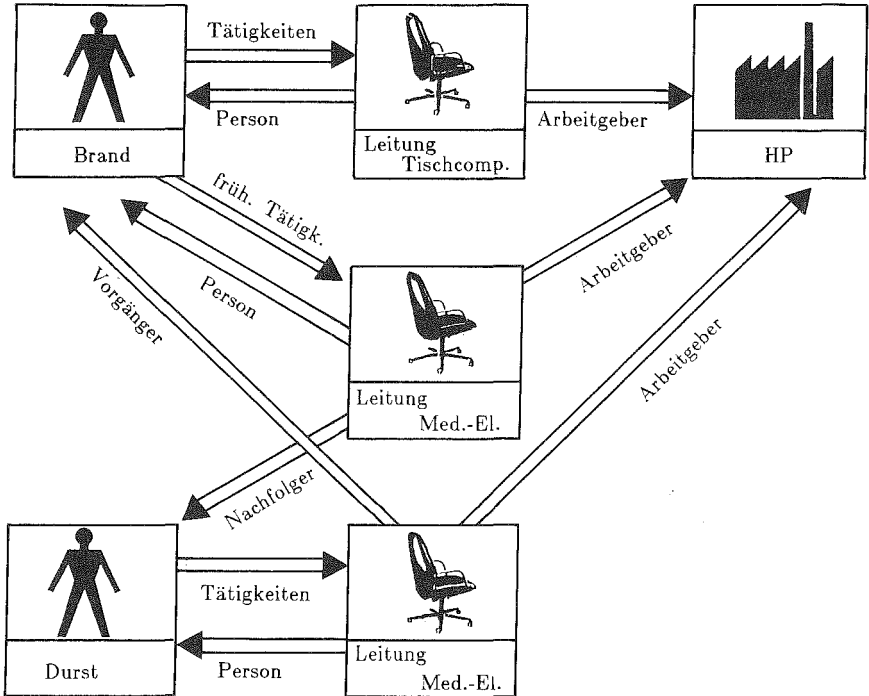


Abbildung IX-3: Zustand der Wissensbasis nach der Verarbeitung

Wenn nun der Sachbearbeiter mittels eines Navigationskripts die Informationen über Herrn Brand abrufen, erkennt er, daß die bisherige Tätigkeit bereits richtig eingeordnet ist, und muß nur noch die neue Tätigkeit des Managers im Werk für Tischcomputer eingeben. Dann ist der Inhalt des Artikels verarbeitet. Das entstandene Wissensnetz ist in der Abbildung IX-3 dargestellt.

Diese Art des Systems, Schlüsse zu ziehen oder *Inferenzen* zu bilden, ist notwendig für eine sinnvolle Wissensverarbeitung: Zum einen erleichtert sie die Aufgabe des Wissenserwerbs an sich, zum anderen enthält die Wissensbasis durch Mehrfachvernetzung von Objekten gewollte Redundanzen, die konsistent erhalten werden müssen.



## 6. Erwerb von Konzepten mit dem System D&I

Im letzten Abschnitt haben wir gesehen, wie Sachwissen, also Wissen um konkrete Sachverhalte aus dem Anwendungsgebiet, mit dem Digester verarbeitet und in Gestalt von Objekten und deren Merkmalen dargestellt werden kann. Diese Fähigkeit des Systems, mit Sachwissen aus einem bestimmten Anwendungsgebiet umzugehen, beruht selbstverständlich wiederum auf Wissen. Im Gegensatz zum Sachwissen ist dieses Wissen von einer abstrakteren Art. Es handelt sich dabei um das dem System innewohnende *konzeptuelle Wissen*.

Konzeptuelles Wissen kommt im System D&I an zwei Stellen zum Ausdruck. Zum einen bestimmt es darüber, welche Arten von Sachverhalten in der Wissensbasis dargestellt und konsistent gehalten werden können. Zum andern ergibt sich aus dem konzeptuellen Wissen die Funktionalität des Systems, die dem Benutzer bei der Verarbeitung von Informationen zur Verfügung steht.

### 6.1 Konzepte

Der Begriff "konzeptuelles Wissen" ist zunächst nur eine Wortschöpfung, die sich auf jedes Programmsystem anwenden läßt, um einen bestimmten Aspekt desselben noch gänzlich unabhängig von der Art seiner Implementierung benennen zu können. Die Funktionsweise jedes nichttrivialen Programms beruht auf dem konzeptuellen Wissen seines Programmierers, und die Realisierung des Programms ist eine Ausprägung dieses konzeptuellen Wissens.

Im Gegensatz zu fast allen herkömmlichen Softwaresystemen ist aber das dem System D&I zugrundeliegende konzeptuelle Wissen nun nicht etwa im Code eines Programmes fest verdrahtet, vielmehr ist es ebenso wie das Sachwissen in Form von Objekten der Wissensbasis repräsentiert. Zum Zweck der Unterscheidung von den "gewöhnlichen" Objekten der Wissensbasis werden wir die Objekte, die das konzeptuelle Wissen repräsentieren, als *Konzepte* bezeichnen. Abbildung IX-4 zeigt, wie solche Konzepte in der Wissensbasis des Systems D&I dargestellt sind.

Wir haben im Beispiel der Verarbeitung des Personenberichts bereits einige Konzepte kennengelernt, ohne daß wir sie schon als solche bezeichnet haben. Einige davon sollen hier nochmals genannt werden, wobei jedes der aufgeführten Konzepte stellvertretend für eine ganze Gattung von Konzepten steht. Im Beispiel traten in Erscheinung: das *Skript*

"Personenbericht", der *Typ* PERSON, die *Aspekte* PRIVAT und BERUF sowie das *Merkmal* "Tätigkeiten:".

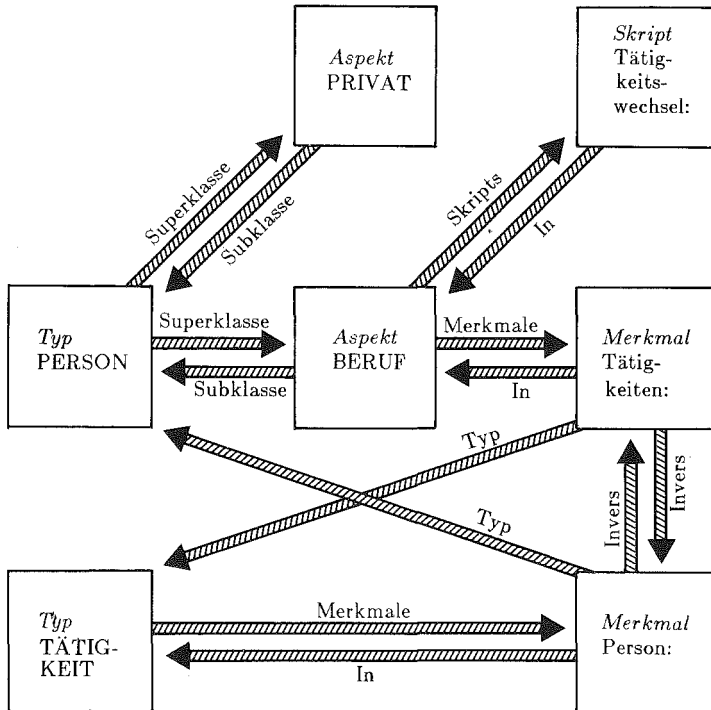


Abbildung IX-4: Konzepte der D&I-Wissensbasis

## 6.2 Das Metasystem des Systems D&I

Typen, Aspekte, Skripts und Merkmale sind also die grundlegenden Arten von Konzepten, mit deren Hilfe das gebietsspezifische Wissen des Systems D&I dargestellt ist. Gegenwärtig enthält die Wissensbasis des Systems D&I Konzepte aus der Wissenswelt der Computerindustrie, daher läßt sich das System in diesem Fachgebiet einsetzen. Ersetzt man die vorhandenen Konzepte durch solche aus einem anderen Anwendungsgebiet, so ist das System D&I bereit, Zeitungsartikel mit einer neuen Thematik zu verarbeiten.

Der Zugang zum konzeptuellen Wissen des Systems wird ermöglicht durch eine sogenannte *Metakomponente*. Die Metakomponente des Systems D&I stellt eine Benutzerschnittstelle zum Erwerb von konzeptuellem Wissen bereit. Mit Hilfe der Metakomponente ist der Benutzer des Systems imstande, die Konzepte der Wissensbasis zu inspizieren, zu erzeugen und zu modifizieren. Das System kann so in seiner Funktionalität erweitert oder auch auf gänzlich neue Anwendungsgebiete umgestellt werden.

Das System D&I erlaubt also zwei Arten der Benutzung zum Zweck des Wissenserwerbs. Als Anwendungssystem dient es zur Assimilation von Ereignissen und Sachverhalten des Anwendungsgebiets. Als Metasystem unterstützt es die Akkommodation auf veränderte Einsatzgebiete.

### 6.3 Uniforme Benutzerschnittstelle

Konzepte sind zwar eine besondere Art von Objekten der Wissensbasis, dennoch sind sie im Grunde genauso aufgebaut wie alle anderen Objekte. Das heißt, sie können identifiziert und erzeugt werden, und sie besitzen Merkmale, aus denen sich ihre charakteristischen Eigenschaften ergeben. Daher kann beim Erwerb von Konzepten dieselbe Benutzerschnittstelle Verwendung finden wie beim Erwerb von Sachwissen.

Der Bildschirmaufbau des Metasystems ist derselbe wie der des Anwendersystems (Abbildung IX-5). Das Arbeitsfenster enthält ein Formular, das die Eigenschaften eines Konzeptes darstellt. Das Skriptfenster enthält ein Menü, das auf die Aufgabe abgestimmt ist, das konzeptuelle Wissen in ein Netz von Konzepten abzubilden und in diesem Netz zu navigieren.

Auch im Metasystem ist die Auswahl der gezeigten Formularfelder im Arbeitsfenster und das Angebot der Funktionen im Skriptmenü abhängig vom aktuellen Aspekt. Die Anzahl der erforderlichen Aspekte zur Darstellung der verschiedenen Arten von Konzepten ist beschränkt, da es nur endlich viele Arten von Konzepten im System D&I gibt, nämlich im wesentlichen nur die oben genannten vier Arten: Typen, Aspekte, Skripts und Merkmale. Im Gegensatz zum Anwendersystem, das wegen der sich ständig wandelnden Anforderungen des Benutzers prinzipiell unvollständig ist, ist es so möglich, die Funktionalität des Metasystems auf einen weitgehend vollständigen Standard zu bringen.

*** D&I *** Bearbeitetes Objekt: BERUF vom Typ ASPEKT unter dem Apekt ASPEKT			
FELDDNAME:	FELDDINHALT:	SKRIPTS	KOMMANDO
Merkmale:	taetigkeiten:	-Zeige Superklasse	BEARBEITE
	fruehere_taehtigkeiten:	-Zeige Subklasse	Objekt
	berufsausbildung:	-Zeige Slot	Aspekt
	besondere_verdienste:	-Zeige Skript	Zurueck
	veroeffentlichungen:	-Zeige Methode	
	komentar:		INFORMIERE
Superklassen:	instance		?Typen
Subklassen:	PERSON		?Protokoll
Skripts:	(Beginn einer Taetigkeit)		VERWALTUNG
	(Beenden einer Taetigkeit)		Vergessen
	(Taetigkeitswechsel) (wird abgeloes) (loest ab)		Retten
	(naehere Angaben zu Veroeffentlichung) (Private Daten)		Neubeginn
			Ende
			SYSTEM
			Eval

Abbildung EX-5: Benutzerschnittstelle des Metasystems

## 6.4 Typen enthalten schematisches Wissen

Welche Art von Sachwissen in der Wissensbasis des Systems D&I dargestellt werden kann, ist zunächst einmal davon abhängig, welche *Typen* von Objekten existieren. Weil in der Wissensbasis der Typ PERSON existiert, ist es möglich, die Eigenschaften einer Person, zum Beispiel die der Person Durst\_Benno, in der Wissensbasis darzustellen.

Im Typ PERSON liegt also das Wissen darüber, welche Merkmale eine konkrete Person tragen kann. Ein solches Merkmal ist beispielsweise der "Vorname:" der Person; das Merkmal "Organisationsform:" hingegen läßt sich nicht auf Personen anwenden, es ist dem Typ FIRMA und nicht dem Typ PERSON zugeordnet. Der Typ eines Objektes ist also ausschlaggebend dafür, nach welchem *Schema* das Objekt aufgebaut ist.

## 6.5 Typen sind mit Hilfe von Aspekten aufgebaut

Solche Schemata sind häufig sehr umfangreich und komplex. Daher sind die Typen im System D&I keine elementaren Objekte, sondern sind mit Hilfe anderer Konzepte aufgebaut. Die Bausteine zur Konstruktion von Typen sind die *Aspekte*. Wenn wir also mit dem System D&I beispielsweise den Typ PERSON inspizieren, sehen wir im Arbeitsfenster unter der Benennung "Superklassen:" eine Auflistung der Aspekte, aus denen der

Typ PERSON gebildet ist, in diesem Fall die beiden Namen PRIVAT und BERUF für die beiden Aspekte einer Person.<sup>9</sup>

Wenn wir auch noch die Verwandtschaftsbeziehungen von Personen darstellen wollten (was das System derzeit nicht tut), müßten wir in der Liste einen weiteren Aspekt, sagen wir VERWANDTSCHAFT, eintragen. Das System würde versuchen, diesen Aspekt zu identifizieren, und da er noch nicht existiert, würde es einen solchen Aspekt in der Wissensbasis erzeugen. Mit Hilfe des Navigationsskripts "Zeige Superklasse" könnten wir uns dann diesen Aspekt auf dem Bildschirm zeigen lassen.

## 6.6 In den Aspekten sind Merkmale festgelegt

In den Aspekten sind die Merkmale festgelegt, die bei der Betrachtung eines Objekts unter diesem Aspekt von Bedeutung sind. Als wir die Person Durst\_Benno unter dem Aspekt BERUF betrachteten, war im Arbeitsfenster ein Formularfeld mit der Bezeichnung "Tätigkeiten:" vorgesehen. Daß dies so war, liegt in den Eigenschaften des Aspekts BERUF. Eine wesentliche Funktion der Aspekte ist es also, daß sie als *Filter* für die Darstellung von Objekten dienen.

Lassen wir uns den Aspekt BERUF vom System D&I zeigen (Abbildung IX-5). Das Formularfenster zeigt uns vier Felder, von denen für uns im Moment nur dasjenige von Interesse ist, das die Benennung "Merkmale:" trägt. Dieses Feld enthält die Liste aller Merkmale, die zur Charakterisierung der beruflichen Betätigung einer Person erforderlich sind, unter anderem die Merkmale "Tätigkeiten:", "Frühere Tätigkeiten:" und "Berufsausbildung:".

Ebenso können wir für den oben eingeführten Aspekt VERWANDTSCHAFT die erforderlichen Merkmale spezifizieren. Wir brauchen dort nur im Formularfeld "Merkmale:" die entsprechenden Benennungen, beispielsweise "Eltern:", "Kinder:" und "Ehegatte:" einzutragen, und das Metasystem sieht diese zusätzlichen Merkmale für alle vorhandenen und künftigen Objekte vom Typ PERSON vor.

---

<sup>9</sup> Typen und Aspekte sind realisiert in Form von ObjTalk-Klassen. Wie in Kapitel III beschrieben, bilden diese Klassen eine Hierarchie. In dieser Hierarchie sind die Aspekte BERUF und PRIVAT Superklassen von PERSON. Daher "erbt" PERSON alle in den Aspekten BERUF und PRIVAT definierten Merkmale. Durst\_Benno ist eine *Instanz* von PERSON und ist gekennzeichnet durch die Ausprägungen dieser Merkmale in Form von Merkmalswerten.

## 6.7 Aus Merkmalen leiten sich Inferenzen ab

Wenn wir ein Merkmal betrachten, müssen wir unterscheiden zwischen dem *Merkmal* selbst als abstraktem Begriff und seinen konkreten Ausprägungen, also den *Merkmalswerten*, die sich in den einzelnen Objekten manifestieren. Ein Merkmalswert, beispielsweise der Merkmalswert "Leiter des Werks für Medizinelektronik", ist Bestandteil eines konkreten Objektes der Wissensbasis (hier Durst\_Benno). Ein Merkmal (hier das Merkmal "Tätigkeiten:" des Aspekts BERUF) ist definierender Bestandteil eines abstrakten Konzepts (hier des Typs PERSON) und selbst wiederum ein Konzept.

Am Beispiel des oben angeführten Merkmals "Ehegatte:" des Aspekts VERWANDTSCHAFT des Typs PERSON soll gezeigt werden, wie konzeptuelles Wissen spezifiziert werden kann, das an Merkmale gebunden ist. Mit Hilfe des Navigationsskripts "Zeige Merkmal" gelangen wir von der Bildschirmdarstellung des Aspekts VERWANDTSCHAFT zu der des Merkmals "Ehegatte:". Im Arbeitsfenster müssen wir zwei Formularfelder ausfüllen, das mit der Benennung "Typ:" und das mit der Benennung "Invers:". Weitere Formularfelder sind in diesem Fall nicht von Belang und brauchen nicht ausgefüllt werden oder sind bereits vom System mit Standardwerten vorbelegt worden (siehe Abbildung IX-6).

*** D&I *** Bearbeitetes Objekt: Ehegatte: vom Typ MERKMAL unter dem Aspekt MERKMAL			
FELDNAMEN:	FELDHINHALT:	SKRIPTS	KOMMANDO
Name:	Ehegatte:	-Zugehoeriger Aspekt	BEARBEITEN
Typ:	PERSON	-Editiere Funktion	Objekt
Format:	Einzelwert		Aspekt
Invers:	Ehegatte:		Zurueck
In-Aspekt:	VERWANDTSCHAFT		
Feldhoehe:	1		INFORMIEREN
Syntax:			?Typen
Hilfe:			?Protokoll
Argumente:			VERWALTUNG
If-Filled:	<function(eingabe)>		Vergessen
If-Printed:	<function(wert)>		Retten
			Neubeginn
If-Needed:	<function(<>>		Ende
If-Added:	<function(element)>		
If-Removed:	<function(element)>		SYSTEM
Link:	<function(element)>		Evaluation

Abbildung IX-6: Definition eines Merkmals

Als "Typ:" tragen wir "PERSON" ein, da als Wert des Merkmals "Ehegatte:" nur ein Objekt vom Typ PERSON sinnvoll ist. Dies hat Auswirkungen auf die Dialoggestaltung beim Editieren des betreffenden Merkmalswerts als Formularfeld des Arbeitsfensters. Die Hilfekomponente des Systems D&I kann aus einer solchen Angabe Eingabemuster, Eingabevorschläge und Erklärungstexte generieren. Des weiteren steuert die Angabe des Typs den Identifikationsprozeß nach dem Eintragen eines Ehegatten, weil dann in der Wissensbasis nach einer Person gesucht wird und dort gegebenenfalls eine solche neu angelegt wird.

Wenn die Person x Ehegatte der Person y ist, so gilt auch umgekehrt, daß y Ehegatte von x ist. Das Merkmal "Ehegatte:" ist also zu sich selbst invers. Daher tragen wir im Formularfeld "Invers:" die Bezeichnung "Ehegatte:" ein. Damit haben wir die Semantik des Merkmals festgelegt und befähigen den Digester zur Bildung von Inferenzen. Wenn wir in Zukunft den Ehegatten x einer konkreten Person y angeben, wird das System automatisch auch die Person y als Ehegatten von x eintragen. Ähnlich können wir die Merkmale "Eltern:" und "Kinder:" spezifizieren. Beide nehmen Werte vom Typ PERSON an und stehen in einer Inversbeziehung zueinander.

Es gibt noch eine Reihe weiterer Beschreibungselemente, mit denen die Bedeutung von Merkmalen spezifiziert werden kann, darüber hinaus noch solche, die das Druckbild von Merkmalen bei der Darstellung als Formularfelder beeinflussen. Wesentlich für alle diese Beschreibungselemente ist jedoch, daß sie deskriptives, konzeptuelles Wissen darstellen, aus dem das System D&I für den Benutzer unsichtbares, prozedurales Wissen ableitet, das beim Dialog mit dem System zur Anwendung kommt.

## 6.8 Skripts gestalten den Dialog

Durch die Anzahl und Zusammenstellung der im Menü angebotenen Skripts ergeben sich die Möglichkeiten, die dem Sachbearbeiter beim Dialog mit dem Anwendersystem zur Verfügung stehen. Der Benutzer kann die Form des Dialogs selbst bestimmen und umgestalten, indem er diese Skripts mit Hilfe des Metasystems definiert und verändert.

Ebenso wie die Merkmale sind auch die Skripts einem Aspekt zugeordnet. Wenn wir den Aspekt BERUF auf dem Bildschirm betrachten, erscheint im Formularfeld "Skripts:" eine Auflistung aller Skripts, die für die Verarbeitung von Ereignissen vorgesehen sind, in deren Zusammenhang eine beruflich tätige Person steht. In diese Liste können weitere Skripts aufgenommen werden, indem die Liste ergänzt wird.

Ein Skript selbst, beispielsweise das Skript "Tätigkeitswechsel", kann wie jedes Konzept auf dem Bildschirm dargestellt und durch das Ausfüllen eines Formulars definiert werden. Zwei Formularfelder haben hierbei zentrale Bedeutung: Das eine enthält eine Liste der "Argumente:", die nach der Auswahl des Skripts im Kopfenster erfragt werden. Das andere enthält die "Aktion:", die nach der Abfrage der Argumente ausgeführt wird.

Das Skript Tätigkeitswechsel besitzt vier solche Argumente: die Bezeichnungen des früheren und des jetzigen Arbeitgebers sowie die frühere und die jetzige Berufsbezeichnung. Die Argumente selbst sind in einer ähnlichen Form repräsentiert wie die oben beschriebenen Merkmale. Dies führt dazu, daß beim Eingeben von Argumenten ähnliche Inferenzen ablaufen können wie beim Ausfüllen von Formularfeldern.

Skriptaktionen sind im System D&I noch mangelhaft repräsentiert. Im Feld "Aktion:" steht in der Regel ein kurzes Stück Programmcode. Sofern es sich um eine Navigationsaktion handelt, was der häufigste Fall ist, ist die Aktion auch einem DV-unkundigen Benutzer verständlich. Soll das Skript aber komplexere Aktionen bewirken, so sind Programmierkenntnisse unumgänglich. Bei der Erprobung des Systems D&I hat sich jedoch gezeigt, daß die wesentlichen Aufgaben der Skripts in der Navigation zu Aspekten und der Identifikation von Objekten bestehen, so daß eine deskriptive Repräsentation von Skriptaktionen in der Zukunft möglich erscheint.

## 6.9 Metawissen

Die Funktionsweise der Metakomponente des Systems D&I beruht auf *Metawissen*. Auch das Metawissen ist in Form von Konzepten dargestellt. Diese Konzepte bezeichnen wir als *Metakonzepte*. Metakonzepte sind die Konzepte, nach denen Konzepte aufgebaut sind. Einige Namen von Metakonzepten sind bereits gefallen: Es sind dies die Konzepttypen TYP, ASPEKT, MERKMAL und SKRIPT, deren Skripts "Zeige Merkmal", "Zeige Superklasse" sowie deren Merkmale "Superklassen:", "Merkmale:", "Typ:" usw. Abbildung IX-7 zeigt einen Ausschnitt aus dem Netz der Metakonzepte des Systems D&I.

Die Metakomponente des Systems läßt sich daher ebenfalls modifizieren, da sie auf sich selbst angewandt werden kann. Der Erwerb neuen Metawissens unterscheidet sich dabei nicht wesentlich vom Erwerb neuen konzeptuellen Wissens. Das System D&I ist in sich selbst enthalten, wie etwa ein Pascal-Compiler, der selbst in Pascal geschrieben ist.



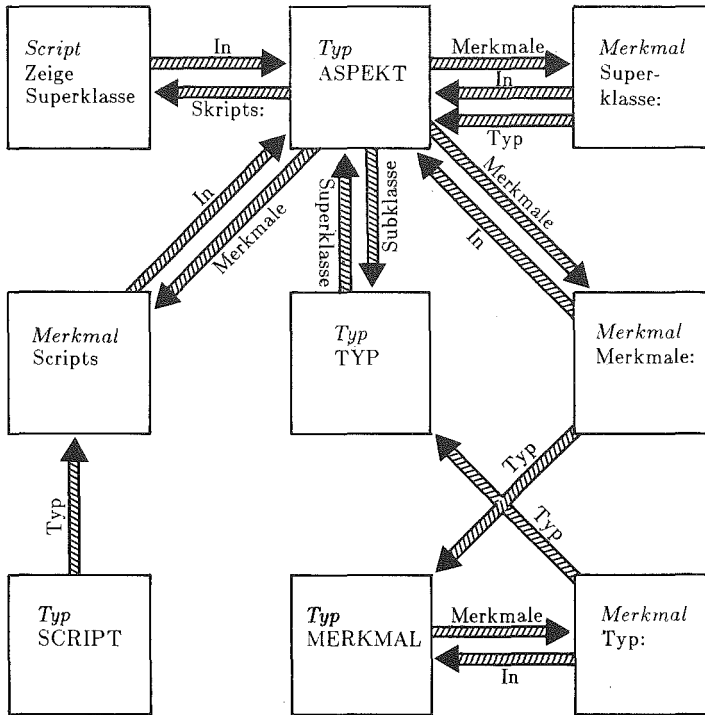


Abbildung IX-7: Metakonzepte der D&amp;I-Wissensbasis

## 7. Zusammenfassung und Ausblick

Wissensbasierte Systeme ermöglichen die Darstellung konzeptuellen Wissens in einer Wissensbasis. Mit Hilfe eines Metasystems ist es möglich, die Funktionalität eines wissensbasierten Systems zu ändern und zu erweitern. Das Metasystem stellt den Kontakt her zwischen dem Benutzer und den Konzepten der Wissensbasis. Dadurch kann der Benutzer das bestehende System nach seinen Vorstellungen umgestalten. Das Metasystem trägt damit zur Konvivialität des Systems bei.

Die bisher bekannten Metasysteme verfolgen meist das Ziel, konzeptuelles Wissen in Form von Regeln darzustellen. Mit den Regeln wird versucht, die Komplexität herkömmlicher Programmsysteme zu vermeiden, indem das prozedurale Wissen in kleinere, voneinander unabhängige Einheiten, die Regeln, zerlegt wird. Dennoch bleibt das Pro-

blem, daß Regeln nur beschreiben können, *wie* ein Problem zu lösen ist, nicht aber *was* die zugrundeliegenden Eigenschaften des Problemraums sind [Davis 82].

Der im System D&I eingeschlagene Weg, das konzeptuelle Wissen deskriptiv in Form von Konzepten darzustellen und aus den Konzepten automatisch prozedurales Wissen abzuleiten, führt weg von der Frage nach dem *Wie*. An die Stelle von Programmiervorgängen tritt der Erwerb konzeptuellen Wissens. Dadurch wird es dem Benutzer möglich, das Verhalten des Systems von Grund auf zu verstehen und zu verändern.

Dennoch zeigt das System D&I eher Ziele auf als Lösungen. Es ist durchaus noch nicht klar, wie sich die Prinzipien des Systems auf gänzlich andere Anwendungsgebiete als das der inhaltlichen Verarbeitung von Zeitungsartikeln übertragen lassen. Sicher ist jedoch, daß ein ungehinderter und natürlicher Zugang zu dem Wissen, das die Arbeitsweise eines Computersystems steuert, dem Benutzer des Systems ein tieferes Verständnis und eine höhere Kontrollierbarkeit der Vorgänge ermöglicht, die er bei der Arbeit mit dem System auslöst. Fortschritte auf dem Gebiet der Darstellung und des Erwerbs von Wissen bei computerunterstützten Prozessen der Problemlösung sind daher zugleich auch Fortschritte auf dem Weg zu benutzergerechten Computersystemen.

<b>VI Universelle Editoren für die Mensch-Computer-Kommunikation</b>	121
1. Klassifikation von Editoren	122
2. Generelle Designprinzipien für Editoren	123
3. Implementationsaspekte	139
4. Der Editor BISO	145
5. Abschließende Bemerkungen	149
<b>VII Visualisierungstechniken</b>	151
1. Formen der Mensch-Computer-Kommunikation	152
2. Psychologische Grundlagen: das visuelle System des Menschen	154
3. Visualisierungskomponenten in Programmierumgebungen	155
4. Visualisierung in Anwendungsprogrammen	174
5. Abschließende Bemerkungen	175
<b>VIII Software-Dokumentationssysteme</b>	177
1. Problemstellung	178
2. Der Softwareproduktionsprozeß	178
3. Ein Kommunikationsmodell für Softwaredesign	181
4. Programmdokumentation im Designprozeß	182
5. Programm-Dokumentationssysteme	187
6. Abschließende Bemerkungen	201
<b>IX Systemkomponenten zum Wissenserwerb</b>	203
1. Wissenserwerb	204
2. Psychologische Aspekte	205
3. Wissensverarbeitung	208
4. Das System D&I als Wissenseditor	212
5. Erwerb von Sachwissen mit dem System D&I	213
6. Erwerb von Konzepten mit dem System D&I	219
7. Zusammenfassung und Ausblick	227
<b>X Computerunterstützte Planungsprozesse</b>	229
1. Charakterisierung von Planungen	230
2. Planungsprozesse in der Psychologie und der Künstlichen Intelligenz	234
3. Computereinsatz in Planungsaufgaben	236
4. PLANER - ein System zur computerunterstützten Studienplanung	240
5. Abschließende Bemerkungen	248
<b>XI Computerunterstützte Büroarbeit - ein Erfahrungsbericht</b>	249
1. Zur Situation	249
2. Fortschritt und Benutzerfreundlichkeit	251
3. Unterstützung und Veränderungen im Tätigkeitsprofil	253
<b>Literaturverzeichnis</b>	261
<b>Index</b>	277



---

# Walter de Gruyter

## Berlin · New York

---

**A. Schulz**

### **Methoden des Software-Entwurfs und Strukturierte Programmierung**

2., bearbeitete und erweiterte Auflage.

15,5 x 23 cm. 181 Seiten. 1982. Kartoniert DM 49,50  
ISBN 3 11 008895 9 (de Gruyter Lehrbuch)

### **Modellierungs-Software**

#### **Konzeption und Anwendung**

Herausgegeben von Hans-Dieter Heike  
17 x 24 cm. VIII, 398 Seiten. 1981.

Gebunden DM 120,- ISBN 3 11 008301 9

**P. Schnupp**

### **Rechnernetze**

#### **Entwurf und Realisierung**

2., gründlich überarbeitete Auflage.

15,5 x 23 cm. 266 Seiten mit 107 Abbildungen  
und 9 Tabellen. 1982. Gebunden DM 76,-  
ISBN 3 11 008951 3 (de Gruyter Lehrbuch)

**L. Peat**

### **Practical Guide to DBMS Selection**

15,5 x 23 cm. 340 pages. With 74 illustrations. 1982.  
Cloth DM 128,- ISBN 3 11 008167 9

**B. E. Meyer**  
**H.-J. Schneider**  
**G. Stübel**

### **Computergestützte Unternehmensplanung**

#### **Eine Planungsmethodologie mit Planungs- instrumentarium für das Management**

15,5 x 23 cm. 537 Seiten. Zahlreiche Abbildungen.  
1983. Gebunden DM 98,- ISBN 3 11 006915 6

**G. Niemeyer**

### **Einführung in das Programmieren in ASSEMBLER**

#### **Systeme IBM, Siemens, Univac, Interdata, IBM-PC/370**

5., bearbeitete und erweiterte Auflage. 15,5 x 23 cm.  
332 Seiten. 1984. Kartoniert DM 44,-  
ISBN 3 11 010128 9 (de Gruyter Lehrbuch)

## Literaturverzeichnis

- [Csima 83] F. Csima: *"Gestaltung der Benutzerschnittstelle eines Informationssystems zur inhaltlichen Verarbeitung von Texten"*. Dissertation, Universität Stuttgart, Februar, 1983.
- [Csima, Riekert 83a] F. Csima, W.-F. Riekert: *"Die Benutzerschnittstelle des Expertensystems D&I"*. In H. Balzert (editor), *Software-Ergonomie*. Teubner, Stuttgart, April, 1983. Vortrag bei der Software-Ergonomie-Tagung I des German Chapter of the ACM.
- [Csima, Riekert 83b] F. Csima, W.-F. Riekert: *"D&I - Ein computerunterstütztes System zum Wissenserwerb"*. *Office Management*, pp 53-55, Sonderheft, 1983. Vortrag bei der Arbeitstagung Mensch-Maschine-Kommunikation der GMD, Bad Honnef, November 1982.
- [Davis 82] R. Davis: *"Expert Systems: Where Are We? And Where Do We Go From Here?"*. *AI Magazine*, pp 3-22, Spring, 1982.
- [Fischer 81c] G. Fischer: *"Computer als konviviale Werkzeuge"*. In *Proceedings der Jahrestagung der Gesellschaft für Informatik*, pp 407-417. Gesellschaft für Informatik, Springer Verlag, München, 1981.
- [Fischer 83b] G. Fischer: *"Navigationswerkzeuge in wissensbasierten Systemen"*. *Office Management* 31 (Sonderheft), pp 49-52, 1983.
- [Fischer et al. 78] G. Fischer, R. Burton, J.S. Brown: *"Analysis of Skiing as a Success Model of Instruction: Manipulating the learning environment to enhance skill acquisition"*. In *Proceedings of the Second National Conference of the Canadian Society for Computational Studies of Intelligence*. Conference of the Canadian Society for Computational Studies of Intelligence, 1978.
- [Gunzenhäuser 84] R. Gunzenhäuser: *"Lernen als Dimension der Mensch-Maschine-Kommunikation"*. In H.Schauer, M.J.Tauber (editor), *Psychologie der Computerbenutzung*, pp 226-252. Oldenbourg Verlag, Wien - München, 1984. Schriftenreihe der Österreichischen Computer Gesellschaft, Band 22.
- [Illich 73] I. Illich: *"Tools for Conviviality"*. Harper and Row, New York, 1973.
- [Michalski, Carbonell, Mitchell 83] R.S. Michalski, J.G. Carbonell, T.M. Mitchell (editors). *"Machine Learning"*. Tioga Publishing Company, P.O.Box 98, Palo Alto, CA 94302, 1983.
- [Moore, Newell 74] J. Moore, A. Newell: *"How can MERLIN understand?"*. In L.W. Gregg (editor), *Knowledge and Cognition*. Erlbaum, Potomac, 1974.
- [Newell 69] A. Newell: *"Heuristic programming; ill structured problems"*. In Aronofsky (editor), *Progress in Operations Research, Volume 3*, pp 362-414. Wiley, New York, 1969.
- [Piaget 72] Piaget: *"The Psychology of Intelligence"*. Littlefield, Adams & Co., Inc., Totowa, N.J., USA, 1972.
- [Quillian 68] M.R. Quillian: *"Semantic Memory"*. In M. Minsky (editor), *Semantic Information Processing*, chapter 4, pp 227-270. The MIT Press, Cambridge, Ma., 1968.
- [Rosenberg 77] S.T. Rosenberg: *"Frame based Text Processing"*. AI-Memo 431, MIT, 1977.