Kapitel 7

Expertensysteme, Planen und Problemlösen

Kapitelherausgeber: Ipke Wachsmuth

Mit diesem Kapitel wird in zwei Abschnitten ein vertiefter Einstieg in das Gebiet wissensbasiert der Informationsverarbeitung mit problemlösenden Systemen gegeben. Das Kapitel führt in das Gebiet der Expertensysteme ein, das die derzeit erfolgreichste Anwendungsgebiet der Künstlichen Intelligenz darstellen. Ein Expertensystem ist ein Computerprogramm, mit dem versucht wird, anhand von erhobenen Gedankengängen und Erfahrungen von Experten eines bestimmten Fachgebiets ein maschinelles System zu konstruieren, das Anwendern Aspekte einer Problemlösungskompetenz zur Verfügung zu stellt.


Der zweite Abschnitt über „Planen und Konfigurieren“ greift ein Anwendungsthema auf, in dem über die letzten Jahre viel Kompetenz und Erfahrung gewonnen wurde, das sich aber im Gegensatz zur verfügbaren Literatur über Diagnose-Systeme bislang kaum in dieser Geschlossenheit dargestellt findet. Planen bedeutet das Erstellen einer Struktur von Aktionen, deren Ausführung ein gegebenes Problem löst, und Konfigurieren das Erstellen einer Struktur aus Komponenten, die gegebenen Bedingungen genügt. Planen und Konfigurieren sind verwandte Aktivitäten, da sowohl die erzeugten Strukturen als auch die verwendeten Methoden einander ähnlich sind. Im vorliegenden Beitrag werden anhand

Die beiden Abschnitte dieses Kapitels gehen auf Kurse zurück, die auf den jährlich stattfindenden Frühjahrsschulen über Künstliche Intelligenz durchgeführt worden sind. Anregungen und Kritiken von Kursteilnehmern sind weitgehend aufgenommen worden, wodurch wir dem Ziel dieses Kapitels, bei beschränktem Seitenumfang einen tiefergehenden Einblick in das Gebiet zu geben, hoffentlich nahegekommen sind.

7.1 Expertensysteme und Wissensmodellierung

*Josef Meyer-Fujara, Frank Puppe und Ipke Wachsmuth*

7.1.1 Expertensysteme und Expertenwissen

7.1.1.1 Motivation

Mit Expertensystemen wird versucht, Gedankengänge und Erfahrungen von Experten bestimmter Fachgebiete auf eine Menge von formalisierten, maschinenverarbeitbaren Operationen abzubilden, um Aspekte einer Problemlösungskompetenz zu reproduzieren und Anwendern zur Verfügung zu stellen. Von der Formalisierung von Expertenwissen verspricht man sich:

- Explizierung und Überprüfbarkeit von Fachwissen
- Unterstützung und Ergänzung menschlicher Fachtätigkeit
- Begünstigung einer Wissensevolution
- Vorteile bei der fachlichen Ausbildung
- Möglichkeit einer Konservierung von Fachwissen
- technische Hilfsmittel für die Wissensverarbeitung und -nutzung

Eine schwierige Aufgabe ist die Modellierung von Wissen, welches Experten benutzen. Noch erschwert wird diese Aufgabe dadurch, daß es sich um eine Rekonstruktion von – überwiegend impliziten – Modellen, die Experten bei ihrer Fachtätigkeit benutzen, handelt (siehe Abb. 7.1). Nicht alle Vorgehensweisen der Wissensmodellierung machen das sogenannte mentale konzeptuelle Modell explizit; auf diesen Punkt richten sich neuere modellbasierte Vorgehensweisen, auf die später eingegangen werden wird.
Abbildung 7.1: Mentale konzeptuelle Modelle, über welche Experten für Problemlösungen in ihrer Anwendungswelt verfügen, sind in operationale Modelle für Expertensysteme zu überführen.
7.1.1.2 Einordnung und Abgrenzung

Wissensmodellierung ist zunächst eine Teilaufgabe der *Wissensrepräsentation*. Unter einer Wissensrepräsentation versteht man ein symbolisch dargestelltes Modell eines Wissensbereichs aus Objekten, Fakten und Regeln in operationaler Form für einen Handlungsträger mit symbolverarbeitender Kompetenz. Das Gebiet „Wissensrepräsentation“ umfaßt zwei erlei: (1) Die Entwicklung von Repräsentationsformalismen, die Beschreibungsmittel für die symbolische Darstellung des Wissens bestimmter Fachgebiete und Weltbereiche bereitstellen und die automatische aufgabengesteuerte Berechnung von Inferenzen über der Menge kodierten Wissens ermöglichen; (2) die Bereichsmodellierung (Modellierung von Wissensbereichen), bei der insbesondere folgende Fragen zu beantworten sind:

- *Welche* Kategorien sind zur Einordnung von Objekten und Ereignissen eines Wissensbereichs zu wählen?
- *Welche* Eigenschaften und Annahmen sind ihnen zuzuordnen?
- *Wie* werden sie untereinander in Beziehung gesetzt?
- *Welche* Folgerungen sollen im Kontext bestimmter Annahmen möglich sein?


**EXPERTEN**: Hierunter verstehen wir Personen, die berufliche Aufgaben bewältigen, für die man sowohl eine lange Fachausbildung als auch praktische Erfahrung benötigt. Zu den Fähigkeiten von Experten gehört, daß sie Probleme erkennen und verstehen, Probleme lösen, die Lösung erklären, die eigene Kompetenz einschätzen, Randgebiete ihres Fachs überschauen sowie Wissen erwerben und strukturieren können. Es ist möglich, daß Experten starke, aber fehlerhafte Annahmen darüber haben, wie sie zu Urteilen gelangen. Ihr Wissen kann unbewußt sein und nicht mit Worten ausdrückbar. Mit der Besonderheit von Experten und ihrem Wissen werden wir uns noch eingehender befassen.


**EXPERTENSYSTEME** sind eine Form wissensbasierteter Systeme (also Programme), die spezifisches Wissen und Schlußfolgerungsfähigkeiten qualifizierter Fachleute nachbilden. Ihr Anwendungsfeld ist beim Stand der Technik auf scharf abgegrenzte, schmale Gebiete beschränkt. Ihre Arbeitsweise beruht im Regelfall auf interaktiver oder automatischer Lösung von Problemen durch Symbolmanipulation in mehrstufigen Such- und Entscheidungsprozessen. Expertensysteme und grundlegende Techniken für ihre Realisierung sind Gegenstand von Teil 7.1.2 des Beitrags; die Entwicklungsmethodik wird in Teil 7.1.5 behandelt.

**KNOWLEDGE ENGINEERS** („Wissensingenieure“) sind die Leute, die Wissen von erfahrenen Fachleuten auf ein Computersystem übertragen, welches in der Lage ist, einen Bereich zu repräsentieren, darin Schlüsse durchzuführen und in diesem Bereich (Routine-) Probleme zu lösen. Wissensingenieure haben bei der Entwicklung von Expertensystemen etwa die Funktion von Systemanalytikern bei klassischen Software-Systemen. Die Spannweite ihrer Tätigkeiten ist enorm; sie umfaßt die Analyse und Bewertung der kognitiven Fähigkeiten von Experten, einblickhaftes Verständnis des jeweiligen Anwendungsbereichs, die Erstellung eines Wissensmodells und schließlich systemnahe Entscheidungen bei der Im-
plementierung der Wissensbasis. Bei entsprechender Schulung kann die Rolle des Knowledge Engineers auch vom Domänenexperten selbst übernommen werden. Mit Methodologien zur Wissensmodellierung werden wir uns in Teil 7.1.3 des Beitrags auseinandersetzen.

**EXPERTENSYSTEMWerkzeuge** sind Software-Werkzeuge zur Erstellung von Expertensystemen, die Verallgemeinerungen und Formalisierungen bewährter Konzepte von Expertensystemen bereitstellen. Insbesondere die Ansätze und Möglichkeiten verschiedener sog. hybrider Expertenverarbeitungswerkzeuge, die sich teils auf eigene Erfahrungen stützen, sind Gegenstand von Teil 7.1.5 des Beitrags; Spezialwerkzeuge für die Wissensakquisition werden in Teil 7.1.3 angesprochen.

### 7.1.1.3 Expertensysteme – Einsatzfelder und Arbeitsprinzip


1. einer Sammlung relativ unabhängiger „Module“ (z.B. Regeln, Frames, Constraints, Logik-Klauseln), die für passende aktuelle Daten aktiviert werden können,
2. einer dynamischen Datenstruktur, die durch diese Module inspiziert oder verändert werden kann und
3. einer zyklischen Steuerung für die Aktivierung der Module (Interpreter).

Den Kontrollfluß bestimmt der Interpreter in Abhängigkeit von den aktuellen Daten; der Experte legt nur fest, was in bestimmten Situationen zu tun ist.

Als Beispiele für solche Module führen wir hier zwei unterschiedlich geartete Regeln (mit Implikationen bzw. mit Handlungen) an:

1. **Wenn** 1. Nackensteife und  
               2. hohes Fieber und  
               3. Bewußtseinstrübung  
               *zusammentreffen,*  
               **dann** besteht Verdacht auf Meningitis.

2. **Wenn** Verdacht auf Meningitis besteht,  
               **dann** bestimme Liquorstatus.


### 7.1.1.4 Wissen – Fachwissen – Erfahrungswissen

Angesichts der häufig formulierte Kritik über die Adäquatheit der Wissensauffassung, die sich in realisierten Expertensystemen widerspiegelt, wird sicherlich mit einigem Recht behauptet, daß man sich in der Forschung über Künstliche Intelligenz in unzureichendem Maße mit natürlicher Intelligenz auseinandergesetzt habe. Dies gibt Anlaß, hier vor allen Anlässen zur Wissensmodellierung die Natur von Expertenwissen eingehender zu betrachten. Expertenstudien in der psychologischen Problemlösesforschung sind u.a. für die Gebiete Physik, der Medizin und auch der Wissensvermittlung (Lehrer) durchgeführt worden. Diese Studien sind gekennzeichnet durch die Verwendung von Problemen, zu deren Lösung Fach- und Hintergrundwissen über Realitätsbereiche erforderlich ist. Grob zusammengefaßt stellte sich heraus: Gegenüber weniger erfahrenen Fachleuten haben Experten qualitativ unterschiedliches Wissen; dies kann betreffen:

- Inhalte des Wissens (was für Wissen)
- Quantität (mehr Wissen)
- sachliche Richtigkeit und Angemessenheit
- Abstrakttheit der Begriffe

---

• Kohärenz des Wissens
• fall- oder aufgabenbezogene Organisation des Wissens
• Prozeduralisierung von Wissen
• Verknüpfung des Wissens über Sachverhalte mit dem Wissen über Lösungsschritte

Bei den Expertenstudien zeigte sich auch, daß zwischen dem in Büchern niedergeschriebenen Fachwissen und der Expertise, die sich als Folge langjähriger beruflicher Erfahrung ausbildet, kein einfacher Zusammenhang besteht: Gleiche Erfolge in der Bewältigung einer Problemlösung sind auch bei unterschiedlichem theoretischem Fachwissen möglich, andererseits sind deutliche Zusammenhänge zwischen Fachwissen und Problembearbeitung nachweisbar. Expertenwissen umfaßt einerseits weniger, andererseits mehr als das theoretische Kernwissen des Fachs. Und es ist oft anders strukturiert. Im Vergleich zu guten „Novizen“ zeigt sich eine andere Qualität des Fachwissens von Experten, was die Exaktheit, Detailliertheit und Differenziertheit anbelangt.


7.1.1.5 Besonderheit von Expertenwissen


### 7.1.1.6 Arten von Expertenwissen

Auch bei spezifischem Expertenwissen sind verschiedene Arten unterschieden worden. Dazu gehören assoziatives (heuristisches) Wissen, kausales (modellbasiertes) Wissen, statistisches und schließlich fallvergleichendes Wissen. Es ist wichtig einzusehen, daß die Unterscheidung solcher Wissensarten von vornherein nichts mit den zur Wissensmodellierung herangezogenen Repräsentationsformalismen zu tun hat (also nicht etwa: Regeln stehen für assoziatives Wissen etc.). „Gelbfärbung der Augen deutet auf Leberkrankheit“ ist eine empirisch gerechtfertigte Heuristik zur diagnostischen Hypothesenfindung, während die Aus-


**Abbildung 7.2: Veranschaulichung des Kliff-und-Plateau-Effektes bei Expertensystemen**

Eine Haupterkennnis aus den oben zitierten Expertenstudien ist allerdings folgende: Das
als „Expertenwissen“ bezeichnete Spezialwissen ist (mindestens teilweise) aufgaben-, fall- bzw. zielorientiert organisiert, hat also eine bestimmte „Ausrichtung“. Im Gegensatz dazu ist Allgemeinwissen, oder auch das Weltwissen für das Verstehen natürlicher Sprache, „unspezifisch relational“. Ähnliches gilt allerdings auch für das Hintergrundwissen von Experten (z.B. von Ärzten über die allgemeinen Lebensumstände von Patienten). Soweit sich „ausgerichtetes“, auf die Lösung bestimmter Problemklassen zugeschnittenes Expertenwissen identifizieren läßt, bestehen beim gegenwärtigen Stand der Kunst der Expertensystemtechnik Chancen, es für zweckgerichteten Einsatz zu modellieren.

\section*{7.1.2 Wissensrepräsentationen und Ableitungsstrategien}


\subsection*{7.1.2.1 Regeln}


### 7.1.2.2 Objekte und Frames


Abbildung 7.3: Beispiel eines Constraints für einen Widerstand mit Widerstandswert R

Prozedur.

Erwartungswerte (Defaultwerte) sind Vorbelegungen von Werten, die als Standardannahme meistens, aber nicht immer stimmen, z.B. „alle Vögel können fliegen“. Erwartungswerte gelten nur dann, wenn kein genaues Wissen verfügbar ist. Da sie gegebenfalls korrigiert werden müssen, wird bei Verwendung von Erwartungswerten ein System zum nicht-monotonen Schließen (s. Abschnitt 7.1.2.5) benötigt.


7.1.2.3 Constraints

Mit Constraints können beliebige Relationen zwischen Variablen repräsentiert werden. So beschränkt z.B. ein elektrischer Widerstand die Werte für Spannungs- und Stromvariablen \( U_1, U_2, I \) nach dem Ohmschen Gesetz (Abb. 7.3).

Im Unterschied zu Regeln können Constraints ungerichtet sein, d.h. wenn in dem Constraint in Abb. 7.3 zwei beliebige Variablen bekannt sind, kann die dritte Variable bestimmt werden. Viele Probleme lassen sich natürlich als ein Constraint-Netz beschreiben, d.h. als eine Menge von Constraints, die durch gemeinsame Variablen verbunden sind. Ein Constraint-Problem ist dann eine Anfangsbelegung einiger Variablen eines Constraint-Netzes, und seine Lösung besteht darin, möglichst eindeutige Werte für die übrigen Variablen zu finden. Ein Beispiel eines Constraint-Netzes für einen elektrischen Schaltkreis mit Widerständen, Verzweigungen, Dioden, Kondensatoren und Transistoren als Constraints und Spannungen und Strömen als Variablen findet sich in [Stallman und Sussman, 1977].

Ähnlich wie Regeln basieren Constraints auf einer allgemeinen Idee, für die es vielfältige Realisierungen gibt. So kann man Constraints als Tabellen, Funktionen oder als charakteristische Prädikate definieren und z.B. mit Datenbanken (für Tabellen), mit Regeln oder unmittelbar als Programmcode implementieren. Die Grundidee zur Lösung von Constraint-
\[
P(D_i/S_i \& \ldots \& S_m) = \frac{P(D_i) \times P(S_i/D_i) \times \ldots \times P(S_m/D_i)}{\sum_{j=1}^{n} P(D_j) \times P(S_i/D_j) \times \ldots \times P(S_m/D_j)}
\]

Abbildung 7.4: Die zur Bewertung von Alternativen hauptsächlich benutzte Form des Theorems von Bayes

Problemen besteht darin, daß, ausgehend von der Anfangsbelegung der Variablen, alle damit verbundenen Constraints aktiviert werden; bei deren Auswertung erhalten dann weitere Variablen einen Wert, was zur Aktivierung neuer Constraints führt, bis keine weitere Wertzuweisung an eine Variable mehr möglich ist. Dies Verfahren heißt Propagierung. Während im einfachsten Fall nur feste Werte für eine Variable propagiert werden können, sind leistungsfähigere Constraintsysteme auch in der Lage, Wertemengen, Intervalle oder symbolische Ausdrücke zu propagieren. Weiterhin kann die Funktionalität eines Constraintsystems dadurch gesteigert werden, daß es plausible Annahmen für einen Variablenwert macht und diesen im Falle eines Widerspruches (der Zuweisung der leeren Menge an eine Variable) wieder zurückziehen kann. Falls ein Constraint-Problem unlösbar ist, kann das System versuchen, durch Zurücknahme von Constraints (d.h. durch Aufgeben der entsprechenden Anforderung) eine Lösung zu finden. Eine formale Beschreibung von Constraints findet sich z.B. in [Voß und Voß, 1989], und ein Beispiel eines Constraint-Interpreters, der feste Werte und Wertemengen propagieren kann, in [Güsgen, 1989].

7.1.2.4 Probabilistisches Schließen

7.1. EXPERTENSYSTEME UND WISSENSMODELLIERUNG

1. starte mit der Apriori-Wahrscheinlichkeit der Alternativen

2. modifiziere bei jeder neuen Information die Wahrscheinlichkeit der Alternativen entsprechend den Evidenzwerten und dem Verkehrungsschema

3. wähle die am besten bewertete Alternative aus


Das Theorem von Bayes eignet sich dazu, aus den Apriori-Wahrscheinlichkeiten \( P(D_i) \) einer Menge von \( n \) Diagnosen und aus den bedingten Wahrscheinlichkeiten \( P(S_j / D_i) \), d.h. der statistischen Häufigkeit des Auftretens der Symptome bei gegebenen Diagnosen, die bedingte Wahrscheinlichkeit \( P(D_i / S_1 \& \ldots \& S_m) \) der Diagnosen bei gegebenen Symptomen gemäß der Formel in Abb. 7.4 zu berechnen.

Die Voraussetzungen zur Anwendung des Theorems von Bayes sind die Unabhängigkeit der Symptome untereinander, die Vollständigkeit und der wechselseitige Ausschluß der Diagnosen sowie gute und große Statistiken.

Das MYCIN- und INTERNIST-Schema unterscheiden sich vom Theorem von Bayes vor allem durch die getrennte Bewertung von positiver und negativer Evidenz (also von Fakten, die für bzw. gegen eine Diagnose sprechen), durch größere Bewertungskategorien, durch die Einteilung der Diagnosen in Gruppen konkurrierender Diagnosen (INTERNIST) und durch die Berücksichtigung von Symptomkombinationen (MYCIN).

7.1.2.5 Nicht-monotones Schließen

Im Alltag werden häufig Schlüsse gezogen, weil sie aufgrund zunächst vorhandener Informationen plausibel scheinen (z.B. "der Kilometerzähler zeigt 2.000 km, das Auto ist also kaum gefahren"). Sie können durch zusätzliche Information ("der Zähler wurde einmal ausgetauscht") hinfällig werden, sind aber für das Überleben unumgänglich, denn über volle Information verfügen wir nahezu nie. Weil bei dieser Art zu schließen aus mehr Information weniger geschlossen werden kann, nennt man sie nicht-monoton im Gegensatz zum monotonen klassischen logischen Schließen, wo aus mehr Information stets auch mindestens gleich viel geschlossen werden kann. (Dort gilt für alle Aussagemengen \( S_1, S_2, P \), daß wenn \( P \) aus \( S_1 \) herleitbar ist, dann auch aus der Vereinigung von \( S_1 \) und \( S_2 \). Während theoretische Aspekte des nicht-monotonen Schließens u.a. in [Brewka, 1989] und [Genesereth und Nilsson, 1987, Kap. 6] behandelt werden, konzentrieren wir uns hier auf Mechanismen, wie nicht-monotonen Schließen in Expertensystemen behandelt werden kann.

Beim nicht-monotonen Schließen müssen Schlüsse, die sich als ungültig herausstellen, mit allen Konsequenzen revidiert werden. Das einfachste Verfahren wäre die vollständige Neuberechnung aller Inferenzen aufgrund der neuen Menge an Ausgangsdaten. Dieser Weg

- direkte Begründungen: JTMS (Justification-based TMS, z.B. das System von Doyle [Doyle, 1979])
- Basisannahmen, die einer Begründung zugrunde liegen: ATMS (Assumption-based TMS, z.B. das System von de Kleer [de Kleer, 1986]).

Der Unterschied läßt sich an einem einfachen Beispiel mit zwei Regeln verdeutlichen: Regel: \( A \Rightarrow B \), Regel: \( B \Rightarrow C \). Ist \( A \) eine Basisannahme, so wird im JTMS für \( C \) die Begründung Regel2 abgespeichert, während im ATMS ein Kontext (d.i. eine Menge von Basisannahmen) generiert wird, der \( A \) enthält.

Der JTMS-Basis-Algorithmus ist einfach und effizient: eine Änderung eines Fakultums bewirkt, daß alle mit diesem Faktum assoziierten Begründungen überprüft werden. Falls eine Begründung ungültig ist, wird getestet, ob die zugehörige Schlussfolgerung noch weitere Begründungen hat. Falls nein, wird die Schlussfolgerung zurückgezogen und mit ihr als Input dieser Algorithmus rekursiv aufgerufen.

Das Hauptproblem dabei ist die Behandlung von Ableitungsschleifen; z.B. würde eine Schleife \( A \Rightarrow B \) und \( B \Rightarrow A \) im Basis-Algorithmus bewirken, daß \( A \) und \( B \) nicht mehr rücksetzbar sind. Eine Lösung dieses Problems ist das Abspeichern von nicht-zirkulären (well-founded) Begründungen für eine Schlussfolgerung.


7.1.2.6 Temporales Schließen

Die Voraussetzung zum temporalen Schließen ist die Erweiterung der Wissensrepräsentation von Fakten um Zeitangaben, z.B. „Startzeitpunkt der Brustschmerzen: vor drei Wochen“. Wichtige Aspekte der Zeitrepräsentation sind:
7.1. EXPERTENSYSTEME UND WISSENSMOdELLIERUNG

- punkt- oder intervallbasierte Basisrepräsentation
- exakte, qualitative oder quantitativ ungenaue Zeitangaben
- Bezug auf eine absolute Zeitskala oder auf Referenzereignisse
- Zeitangaben als Zahlen oder als Zahlen mit Zeiteinheiten

Die punkt- und intervallbasierte Repräsentation sind im Prinzip gleich mächtig, da man ein Intervall durch Anfangs- und Endpunkt bzw. einen Zeitpunkt durch ein beliebig kleines Intervall darstellen kann. Trotzdem kann die jeweilige Einfachheit und Eleganz der Handhabung für verschiedene Anwendungsbereiche sehr unterschiedlich sein.

Am einfachsten ist die Repräsentation exakter Zeitangaben. Wenn keine genauen Angaben verfügbar sind, können Ungenauigkeiten qualitativ (A begann vor B) oder quantitativ ungenau (A begann 3-4 Wochen vor B) angegeben werden.


Die Verwendung von Zeiteinheiten kann neben der besseren Lesbarkeit auch dazu dienen, Ungenauigkeiten auszudrücken: sog. „vor einem Jahr“ und „vor 365 Tagen“ nicht unbedingt dasselbe (siehe [Mittal et al., 1984]).

Die Komplexität der Zeitrepräsentation bestimmt, wie aufwendig und effizient zeitbezogene Fragen beantwortet werden können. Typische Fragen an Zeitdatenbanken sind:
- Ist ein Faktum während eines bestimmten Zeit-Intervalls gültig?
- Hat sich ein Wert oder der Anstieg eines Wertes während der letzten Zeit verändert?
- Wie ist die zeitliche Relation zwischen zwei Fakten?

7.1.3 Wissensmodellierung

7.1.3.1 Methodologien zur Wissensmodellierung


konstruieren, im Großen aber schwer vorherzusagen und zu warten. In vielen Fällen ergaben sich Vermischungen von Bereichswissen und Kontrollstrategie durch den Zwang, die Ausführung sämtlicher Aktionen des Systems über den Regelinterpreter abzuwickeln, so daß dieser Zugang oft ein falsch verstandenes Knowledge Engineering, nämlich "Hinprogrammieren" erforderte.


Methoden zur Wissensmodellierung müssen sich grob zusammengefaßt auf die folgenden Einzelaspekte richten:

- Techniken, um Daten – wie Videoaufzeichnungen und Protokolle – von Experten zu erheben
- Analyse und Interpretation dieser Daten, um herauszuschälen, worin das zugrundeliegende Wissen und der zugrundeliegende Schlüsselelement einer Expertenleistung besteht, was für Wissen dabei eingesetzt wird und in welcher Weise
- Benutzung der interpretierten Daten, um formale Beschreibungen der Expertenstrategien und des eingesetzten Domänenwissens zu gewinnen
- Entwurf und schließlich Implementierung eines operationalen Wissensmodells


7.1.3.2 Begriffsabgrenzungen

Wissensmodellierung als Teilaufgabe der Wissensrepräsentation (vgl. Teil 7.1.1.2) betrifft die inhaltliche Frage der Auswahl von Kategorien zur Darstellung des betrachteten Welt-
bereichs, Festlegung ihrer Zwischenbeziehungen und darauf bezogener Schlußmöglichkeiten etc. (vgl. Teil 7.1.1). Das heißt, es handelt sich in erster Linie um eine Design-Tätigkeit, die jedoch nicht ohne vorangehende Analyse der zu modellierenden Wissensdomäne vorgenommen werden kann. Diese Analyse ist neben der Erhebung des maßgeblichen Wissens eine Hauptaufgabe der Wissensakquisition.


**Abbildung 7.5: Die engeren Aufgaben des Knowledge Engineering**


Bei der Wissenserhebung (knowledge elicitation) geht es darum, Einzelheiten über die besonderen Expertenfähigkeiten „ans Licht zu bringen“ und in sog. Wissensprotokollen zu dokumentieren (zunächst als reine Datensammlung). Wissensanalyse und Wissensinter-
7.1. EXPERTENSYSTEME UND WISSENSMODELLIERUNG

Der Begriff "Unternehmen" dienen der Umsetzung solcher Wissensprotokolle in formale Repräsentationen: allerdings nicht als neutraler Transfer, sondern vermittelt durch den Knowledge Engineer, der sein Bild von der beobachteten Expertise in ein mentales konzeptuelles Modell entwickelt, welches sowohl den Problemlöseprozeß als auch Strukturen des Bereichswissens betrifft. Hier erfolgen die entscheidenden Schritte von der „weichen“ zur „harten“ Wissensbeschreibung, also der Formalisierung.

Analyse und Modellierung von Expertise wirken stark aufeinander ein: die Analyse gibt das zu Modellierende vor; das Modell bestimmt seinerseits die Kategorien der Analyse. Es muß gestatten, Wissen in „modulare“ Bestandteile zu zerlegen, die im Problemlösungsprozeß verwendet werden und die nach und nach inkrementell erhoben werden können. Hierfür gibt es stark unterschiedliche Ansätze, auf die später näher eingegangen wird; z.B. verfolgen Rapid-Prototyping-Ansätze die Idee, so früh wie möglich eine Minimalversion des in Angriff genommenen Systems zu bauen, das als Referenz für die weitere Entwicklung dient. Dabei beschränkt man sich zunächst auf das zur Behandlung weniger ausgewählter Fälle von Aufgaben notwendige Wissen. Das Wissensmodell ergibt sich hier am Ende der Entwicklung. In modellbasierten Ansätzen wird dagegen vor irgendwelchen Schritten zur Implementierung versucht, Modelle des Problemlöseprozesses zu explizieren; zunächst geht es um die Erstellung eines Modells (in vielen Fällen als Papierdokument) in einer dem Experten verständlichen Weise – „auf der Wissensebene“.


7.1.3.3 Erhebung von Expertenwissen

Bei der Wissenserhebung soll die Expertise, die Fähigkeiten eines Experten ausmacht, „herauspräpariert“ und dokumentiert werden. Eine besondere Rolle spielen dabei die Begriffe, die Experten in der Wahrnehmung ihrer Fachtätigkeit benutzen. Wie bereits erwähnt sind dies keineswegs nur Fachbegriffe, die sich in der jeweils einschlägigen Literatur finden, sondern gerade der von Experten selbst geprägte und verwendete Begriffsjargon stellt möglicherweise einen bedeutenden Faktor ihrer Expertise dar.

Konstruktgitterverfahren. Unter mehreren Verfahren, die sich auf die Abklärung der Begriffssstrukturen von Experten beziehen, soll eine psychologische, standardisierte Erhebungsmethode herausgegriffen werden, die auf Kellys Psychologie der persönlichen Kon-


Interviewtechniken. Um näheren Aufschluß über das Vorgehen von Experten bei der Bearbeitung von Aufgaben und der Lösung von Problemen zu erhalten, wird man nicht umhinkommen, sie eingehend zu beobachten und zu befragen. Neben dem Repertory-Grid-Test und ähnlichen Verfahren finden weitere individuenzentrierte Verfahren zur Erhebung
verbaler Daten Einsatz ([Huber und Mandl, 1982]), die sich u.a. darin unterscheiden, wie strukturiert die Anforderungen an eine Versuchsperson sind, über ihre Überlegungen Auskunft zu geben (z.B. ist der Repertory-Grid-Test oder ein Fragebogen stärker strukturiert als ein freies Interview oder die Methode des „lauten Denkens“, bei der die Versuchspersonen möglichst alle Überlegungen während einer Problembearbeitung verbalisieren sollen).


Das Frageverhalten des Interviewers wird hinsichtlich der Verwertbarkeit der erhaltenen Information für kritisch gehalten. Erfahrungsgemäß ist es günstig,

- nach einer gestellten Frage zureichend lange ohne weitere Intervention zu warten, daß die Versuchsperson zu antworten beginnt (die Bereitschaft zu antworten erhöht sich);
- der Versuchsperson zu signalisieren, daß ihre Antworten in jedem Fall begehrt und anerkannt sind (schon aufmerksames Zuhören kann dies leisten);
• Begründungen für eine Antwort möglichst umgehend zu erfragen (solange das zur Antwort führende Gedachte noch im Denken der Versuchsperson gegenwärtig ist – später erhöht sich die Wahrscheinlichkeit erfundener Begründungen).


Schwierigkeiten bei der Wissenserhebung können auch teils motivationaler Natur sein, teils haben sie mit der bereits erwähnten grundsätzlichen Kluft zwischen beobachtetem Können und dem zur Verfügung stehenden Beschreibungsmöglichkeiten zu tun. Experten sind möglicherweise nicht an der Explizierung ihrer Modelle interessiert. Sie artikulieren sich eher in erlebten Situationen und Fallbeispielen als in Abstraktionen, die etwa Regelbeschreibungen darstellen. Häufig werden alte Erfahrungen an eine neue Situation unter Berücksichtigung der Unterschiede angepaßt, um die Situation zu meistern. Als besonders schwierige Punkte erweisen sich die Kontextbezogenheit, Verallgemeinerbarkeit und Adaptierbarkeit des Expertenwissens: Fast zu jeder identifizierten Regel gibt es Ausnahmen. Erklärungen und Begründungen sind auch bei sensibler Interviewführung schwer zu erhalten, sie können konstruiert wirken und es sogar sein. Es kann hilfreich sein, mit mehreren Experten zu arbeiten, was als Vorteil hat, daß sich umfassendes Wissen erlangen läßt und überdies weniger leicht ein idiosynkratisch gestaltetes System entsteht (bessere Akzeptanz bei Benutzern); als Nachteil muß dabei jedoch größerer Zeitaufwand in Kauf genommen werden, und Konflikte aufgrund schwer vereinbarer Sichten der beteiligten Experten sind möglich.

7.1.3.4 Formalisierung von Expertenwissen am Beispiel KADS

Abbildung 7.6: Strukturierung des KADS-Prozesses: In KADS-I werden die vier Ebenen eines konzeptuellen Modells gleichzeitig bzw. zyklisch erarbeitet, wogegen die Erarbeitung der Schichten des Design-Modells sequentiell vorgesehen ist.


Die auf die Problemlösung bezogene Einteilung von Bereichskonzepten und -relationen auf der Inferenzebene ist ein wesentliches Moment der KADS-Vorgehensweise, da hier eine Interpretation der Begriffe im Hinblick auf ihren Gebrauch bei Problemlösungen vorgenommen wird; dies soll am Beispiel näher beleuchtet werden. Typische Metaklassen in der diagnostischen Problemlösung der Medizin sind etwa Symptome (z.B. „Brustschmerz“),
7.1. EXPERTENSYSTEME UND WISSENSMODELLIERUNG

Prädispositions faktoren (z.B. „Alter“), Verdachtsdiagnosen, Enddiagnosen, Therapien. Zu beachten ist, daß ein und derselbe Fachbegriff in verschiedene Metaklassen fallen kann; etwa kann der Begriff „Infektion“ für die Metaklassen Verdachtsdiagnosen, Enddiagnosen, aber auch Symptome relevant sein.


Die Bedeutung der konzeptuellen Modelle ergibt sich daraus, daß eine Bottom-Up-Analyse des Anwendungsbereichs schwer ohne Vorwissen durchführbar ist – jedoch ist das in KADS berücksichtigte Vorwissen abstrakt und auf die vergleichende Interpretation vieler konkreter Problemlösungsbereiche bezogen. Die Hauptleistung von KADS besteht in der Bereitstellung begrifflichen Repertoires für die Aufgabe der Knowledge Engineers, was man als KADS-spezifische Einschätzung des Tuns sehen sollte. Die Anwendung dieses Repertoires bleibt jedoch der Interpretations- und Urteilsfähigkeit der Knowledge Engineers vorbehalten.

KADS wurde bislang eher als ein Programm mit einer bestimmten Philosophie der Wissensmodellierung für Expertensysteme statt als ein operationalisierbares Verfahren bewertet.
Abbildung 7.8: KADS-Modellbibliothek: Interpretationsmodelle für bestimmte Aufgabenklassen (nach [Breuker et al., 1987])
Abbildung 7.9: Übersicht über Problemlösungsmethoden (nach [Puppe, 1990])


7.1.3.5 Indirekter, direkter und automatischer Wissenserwerb

Die in den bisherigen Ausführungen vorherrschende Betrachtung des Wissenserwerbs (Wissensakquisition) sah diesen als einen Prozeß, der starke Beteiligung eines Knowledge Engineers erfordert. Dies wird immer dann der Fall sein, wenn in einem Projekt die Natur des zu modellierenden Problemlöseprozesses erst noch geklärt werden muß (Problemcharakterisierung) oder wenn sich das Bereichswissen nur im Gespräch zwischen Knowledge Engineer und Experten abklären läßt. In gut verstandenen Bereichen wie der Diagnostik oder in einer Projektphase, die die Problemcharakterisierung bereits erfolgreich abgeschlossen hat, ist es aber auch ein verfolgtes Vorgehen, zunächst eine Shell (vgl. Abschnitt 7.1.5.4) zu entwickeln und das Bereichswissen „in der Breite“ direkt vom Experten eingeben zu lassen oder gar eine automatische Erstellung einer Wissensbasis zu erwägen. Entsprechend dieser Vorstellungen sind folgende Grundarten des Wissenserwerbs unterschieden worden [Puppe, 1991]:

- **Indirekter Wissenserwerb**: Ein Wissensingenieur befragt einen Experten und formalisiert die Ergebnisse für das Expertensystem; dieses Verfahren ist aufwendig und naturgemäß fehleranfällig, da das Wissen auf dem Umweg über den Knowledge Engineer akquiriert wird.

- **Direkter Wissenserwerb**: Der Experte formalisiert sein Wissen selbst. Dieses Vorgehen erfordert ein komfortables Wissenserwerbsystem, ist aber nur in gut verstandenen Anwendungsbereichen durchführbar.

- **Automatischer Wissenserwerb**: Das Expertensystem extrahiert sein Wissen selbständig aus Falldaten oder verfügbarer Literatur: Hierzu gibt es erste Ansätze, die bisher aber nicht praxisreif sind.


Beim direkten Wissenserwerb ist der Grundgedanke, daß eine „Programmierungsumgebung für Experten“ geschaffen wird, die Experten von den formalen Anforderungen der Wissensba-
sierungselektrow gelöst und es ihnen ermöglicht, ihr spezifisches Wissen selbst zu
formalisieren und ggf. zu warten. Solche *Wissenswerbsysteme* sind Komponenten einer
Expertensystem-Shell zum Aufbau und zur Wartung einer Wissensbasis durch Experten.
Sie sollen eine dem Experten bekannte Wissensrepräsentation verwenden und sollen kom-
fortabel sein hinsichtlich der Eingabeunterstützung, der sofortigen Überprüfbarkeit von
Änderungen (mit Hilfe einer Testfälle-Datenbank) und der Durchführung von Konsistenz-
tests. Die meisten leistungsfähigen Wissenswerbsysteme betreffen bisher die Diagnostik;
Beispiele findet man in [Puppe, 1991].

Im Hinblick auf einen *automatischen Wissenswerb* wird einerseits die Extraktion von
Sachverhaltswissen aus Falldaten und andererseits aus Texten (Protokollen oder verfügba-
rer Fachliteratur) diskutiert. Es geht im Grunde also um lernfähige Systeme, die bislang nur
in der KI-Grundlagenforschung behandelt werden, allerdings mit stark zunehmendem Inter-
esse. Auch hier beziehen sich für Expertensysteme relevante Entwicklungen fast ausschließ-
lich auf die Diagnostik, vgl. dazu [Puppe, 1991]. Am erfolgversprechendsten erscheinen zur
Zeit Ansätze, die aus großen Falldatenbanken auf Regularitäten schließen. Das System
KRITON [Diederich, 1989] verwendet Konstruktions- und Protokollanalyse-Techniken,
um – allerdings sehr einfache – Regeln direkt aus Interviewdaten zu generieren. Dieser
Ansatz ist jedoch wegen erkennbarer Grenzen nicht weiter verfolgt worden. Die Extrak-
tion von Sachverhaltswissen aus Texten ist Gegenstand jüngerer Projekte zum Verstehen
natürlicher Sprache (wie LILOG); die Anwendung für die automatische Wissensakquisition
ist zwar ein forschungsleitendes Ziel, gegenwärtig für den Bereich Expertensysteme aber
noch nicht von praktischer Bedeutung.

### 7.1.4 Problemlösungsmethoden in Expertensystemen

#### 7.1.4.1 Problemklassen und Problemlösungsmethoden

In diesem Abschnitt werden auf den Grundtechniken der Wissensrepräsentation aufbau-
ende Problemlösungsmethoden für die *Problemklassen* Klassifikation, Konstruktion und
Simulation skizziert. Vorausgeschickt sei, daß sich aus der Kenntnis der Problemklasse be-
reits grob auf die nötigen Grundtechniken der Wissensrepräsentation und -verarbeitung
schließen läßt. Eine Übersicht über diese Zuordnung zeigt die Abb. 7.10. In den angeführ-
ten Problemklassen läßt sich häufig sogar festlegen, wie bereichsspezifisches Wissen zur
Problemlösung verwendet wird. Einen Algorithmus, der dies leistet, bezeichnen wir als
*Problemlösungsmethode*. Es ist sinnvoll, zwischen schwachen und starken Methoden zu
unterscheiden. Schwache (oder *Basis*problemlösungsmethoden) sind solche, die offen für
eine große Vielfalt von Wissensrepräsentationen sind, wie z.B. „Regeln mit Vorwärtsver-
kettung“. Sie sind zwar breit anwendbar, stellen aber keine große Hilfe beim Wissens-
erwerb dar, da sie Repräsentation und Funktion bereichsspezifischen Wissens nicht fest-
legen. Eine weitgehende Festlegung in dieser Hinsicht charakterisiert gerade die starken
Problemlösungsmethoden, die dadurch zwar weniger flexibel sind, aber – sofern anwend-
bar – umfassende Unterstützung beim Wissenserwerb und beim Systementwurf leisten. Im günstigsten Fall kann so viel festgelegt sein, daß das Wissen einfach durch Instantiierung vorgegebener Objekttypen und Angabe der Beziehung zwischen Instanzen eingegeben werden kann. Im folgenden werden die wichtigsten starken Problemlösungsmethoden vorgestellt. Eine ausführliche Behandlung findet sich in [Puppe, 1990].


7.1.4.2 Klassifikation

Die Problemklasse Klassifikation (häufig auch Diagnose genannt) umfaßt alle Problembereiche, bei denen die Lösung aus einer vorgegebenen Menge von Alternativen ausgewählt wird, wie z.B. in der medizinischen Diagnostik, bei der Fehlersuche in technischen Geräten, bei der Qualitätskontrolle, bei der Prozeßüberwachung in der Fertigung, bei der Auswahl von Produkten oder der Überprüfung, welche juristischen Bestimmungen auf eine Situation anwendbar sind².

Historisch haben sich die Diagnostik-Expertensysteme aus dem medizinischen Bereich heraus entwickelt. Die praktischen Erfolge von Diagnostiksystemen liegen jedoch fast ausschließlich im technischen Bereich, während die medizinischen Systeme mit wenigen Ausnahmen Demonstrationsprototypen geblieben sind.

Das für die Diagnostik typische Zurückschließen von Beobachtungen auf Systemzustände bzw. Objekte, die die Beobachtungen hervorrufen, ist eine Form der Abspaltung: wenn eine Ursache $U$ ein Symptom $S$ verursacht, und das Symptom $S$ wird beobachtet, dann ist $U$ eine mögliche Erklärung für das Symptom $S$.

²Wenn nicht anders ausgewiesen, sind die Abbildungen in diesem Abschnitt [Puppe, 1990] entnommen.

Abbildung 7.11: Basisstruktur der Diagnostik mit Problemmerkmalen (Symptomen) und Problemlösungen (Diagnosen)

- Diagnosebewertung mit unsicherem Wissen,
- Diagnosebewertung mit unvollständigem Wissen,
- Plausibilitätskontrolle der Eingabedaten,
- Erkennen von Mehrfachdiagnosen,
- adäquate Behandlung von Widersprüchen,
- kosteneffektive Symptomherbeugung,
- Auswertung von Folgesitzungen.
Standardtechniken zur Bewältigung von Diagnoseproblemen, die auch in psychologischen Untersuchungen bestätigt wurden [Elstein et al., 1978], sind die Verwendung eines diagnostischen Mittelbaus, die hypothetisch-deduktive Vorgehensweise und die Differentialdiagnostik. Der diagnostische Mittelbau beschreibt die Verdichtung der Rohdaten zu den Enddiagnosen über einfache Symptominterpretationen und Grobdiagnosen [Clancey, 1984; 1985], z.B.:

| Rohdaten: | Puls = 100, Blutdruck = 80 | → |
| Einfache Symptominterpretation: | hoher Schockindex | → |
| Grobdiagnose: | Kreislaufshock | → |
| Feindiagnose: | Ursache des Kreislaufshocks | |


7.1.4.3 Konstruktion


Konstruktionsprobleme zeichnen sich gewöhnlich durch einen sehr großen Suchraum aus, z.B. bei einem Zuordnungsproblem mit zwei Objektgruppen von je n Elementen in der Größenordnung von \(n!\) oder bei einem Planungsproblem mit einer durchschnittlichen Operatorendauer der Länge \(n\) und durchschnittlich \(m\) Alternativen bei der Operatorauswahl in der Größenordnung von \(m^n\). Die Grundlage für Konstruktionsmethoden sind daher häufig Suchtechniken. Jedoch scheitert eine einfache Breiten- oder Tiefensuche meistens an dem zu hohen Aufwand. Durch Einbezug von heuristischem Wissen läßt sich der Suchaufwand

oft erheblich reduzieren. Dafür eignet sich insbesondere die Hill-Climbing-Strategie, bei der an jeder Verzweigung Wissen verfügbar sein muß, um die lokal beste Alternative auszuwählen. Da Sackgassen nicht ausgeschlossen werden können, sind zum Backtracking außerdem Rücksetztechniken erforderlich, die effizient mit einem TMS (Abschnitt 7.1.2.5) und gegebenenfalls mit Zusatzwissen realisiert werden können.


Aus diesen Grundideen lassen sich einige effiziente Problemlösungsmethoden ableiten, die bei ausreichendem Wissen den Suchraum so handhabbar machen, daß nur noch sehr wenige Alternativen ausprobiert werden müssen:


3. **Least-Commitment** (Problemumformulierung + Constraint-Propagierung): Zunächst
7.1. EXPERTENSYSTEME UND WISSENSMODELLIERUNG

wird das Problem strukturell umformuliert, und das einfachere Problem kann dann häufig mit Constraint-Propagierung oder Basissuchstrategien gelöst werden.


Diese heuristischen Techniken funktionieren jedoch nur, wenn man viel Erfahrungswissen über den Anwendungsbereich hat. Für Planungsprobleme mit wenig Erfahrungswissen eignet sich die Differenzmethode (means-ends-analysis), bei der zunächst die Differenz zwischen Ausgangs- und Zielzustand festgestellt und dann ein Operator gesucht wird, der diese Differenz maximal verringert. Falls der ausgewählte Operator nicht direkt anwendbar ist oder seine Aktion den Zielzustand nicht vollständig herleitet, wird für die neue Differenz rekursiv ein Unterziel generiert, bis ein vollständiger Plan vorliegt. Im allgemeinen wird auch dabei zunächst mit Groboperatoren geplant, die später verfeinert werden.

Bei der Differenzmethode kann die obengenannte Schwierigkeit von Interaktionen zwischen unabhängig voneinander verfeinerten Groboperatoren besonders leicht auftreten, da in die Grobplanformulierung (im Gegensatz zum Skelett-Konstruieren) kein Erfahrungswissen eingeht. Für die Behandlung von Interaktionen gibt es verschiedene Techniken, die wir am Beispiel einer „negativen“ Interaktion im Problem „Streiche die Leiter und die

Decke skizzieren (beginnt ein ungeschickter Plan mit dem Streichen der Leiter, so ist das Streichen der Decke wegen der nun unbrauchbaren Leiter für eine Weile blockiert):

- Der Plan wird umgestellt, so daß der blockierte Operator vor dem ihn blockierenden Operator ausgeführt wird (lineares Planen mit Verschieben, wie etwa in [Waldinger, 1977]).


- Wenn es nicht anders möglich ist, wird ein zusätzlicher Planschritt generiert, der die verletzte Voraussetzung des Operators wiederherstellt. In diesem Fall würde man sich zum Streichen der Decke eine neue Leiter besorgen oder warten, bis die alte getrocknet ist.

Eine vereinheitlichende Darstellung der diesen Verfahren unterliegenden abstrakten Struktur findet man in [Hertzberg und Horz, 1989]. Wenn die Planerstellung selber kompliziert wird, weil z.B. verschiedene Methoden miteinander kombiniert werden müssen, dann kann auch die Anwendung von Planungs-techniken zur Planerstellung sinnvoll sein (Meta-Planen,

3Das kann sowohl beim linearen wie beim nichtlinearen Planen erforderlich sein

7.1.4.4 Simulation


Abbildung 7.15: Ergebnis einer qualitativen Simulation eines Thermostat-Heizungssystems; nach [Charniak und McDermott, 1985]

Der Übergang zwischen Intra- und Interstate-Analyse wird durch die globale Analyse gesteuert, die für das Erkennen von Schleifen, Gleichgewichtszuständen und Vereinigungen von Verzweigungen zuständig ist. Ein Beispiel für das Ergebnis einer qualitativen Simulation zeigt Abb. 7.15.


Eine ausführliche Darstellung der qualitativen Simulation findet sich in [Iwasaki, 1989]. Es darf jedoch nicht übersehen werden, daß die derzeitigen Systeme zumeist noch sehr einfache Fragestellungen behandeln und kaum praxisreif sind.
7.1. EXPERTENSYSTEME UND WISSENSMODELLIERUNG

7.1.5 Entwicklung von Expertensystemen

7.1.5.1 Methodiken der Expertensystementwicklung


Problemdarstellung. Bevor man ein System in Angriff nimmt, muß man die Ziele des Systems definieren, d.h. die zu lösende Probleme abgrenzen und beschreiben. Die Unterprobleme, die zur Verfügung stehenden Daten und die vorgesehenen Benutzergruppen sind zu bestimmen; es ist festzulegen, was eine Lösung ausmacht (reicht etwa eine Diagnose oder werden auch Therapievorschläge gebraucht?). Die Beschreibung der Anforderungen muß auf jeden Fall Kompetenzgrenzen und Verantwortlichkeiten für eventuelle Fehler festlegen. Von vornherein sollten auch die zukünftigen Benutzer in die Planung des Systems einbezogen werden (partizipative Entwicklung), um nicht am Bedarf vorbeizuentwickeln.


⁴ auch, wenn diese Spezifikation ggf. mit Hilfe eines Prototypen gewonnen wird


Im Unterschied zu klassischen Systemen kann bei Expertensystemen der zentrale Teil, die Wissensbasis, in kleinen Schritten inkrementell implementiert werden. Dabei sollten immer wieder die Anmerkungen des beteiligten Experten und auch der zukünftigen Benutzer eingeholt und durch Korrigieren und Modifizieren berücksichtigt werden. Beim Ra-
71. EXPERTENSYSTEME UND WISSENSMODELLIERUNG

Rapid Prototyping wird in der Implementierungsphase der Hauptteil des Wissens akquiriert; Wissenselemente, die ausreichend präzise bekannt sind, werden sofort in den Repräsentationsformalismus überführt.

Wenn sich auch die Wissensbasisentwicklung klassischen Vorgehensweisen des Software-Entwurfs entzieht, sollte man doch stets versuchen, Teile des Gesamtsystems zu identifizieren, die sich klasisch entwickeln lassen (wie häufig die Benutzeroberfläche), und hier auch entsprechende Verfahren verwenden [Partridge, 1989].


Die Wartung, vor allem die in der Regel unvermeidbare laufende Aktualisierung des in Expertensystemen verwendeten Wissens, ist besonders wichtig und aufwendig; etliche Projekte sind schließlich hieran gescheitert [Mertens et al., 1990]. Beim Rapid Prototyping wird die Wartung dadurch erschwert, daß die Dokumentation notgedrungen unvollständig ist, besonders was die abstrakte Ebene (wie Entwurf, Problemlösungsstrategie) angeht. Um hier den Überblick zu behalten, ist es von zentraler Wichtigkeit, übersichtliche Darstellungen der Wissensbasis bekommen zu können. Das Wartungsproblem wird durch die deklarative Darstellung des Wissens erleichtert, weil so auch entsprechend geschulte Nicht-programmierer (erfahrene Benutzer) Wartungstätigkeiten übernehmen können. Der Personalaufwand, der pro Jahr für die Wartung eines Expertensystems erforderlich ist, wird in [Schild, 1987] auf ein Drittel des Potentials geschätzt, das für Entwicklungsarbeiten benötigt wird.

richtig lösen, kann man darauf vertrauen, daß sie generell keine groben Fehler machen werden – bei einem technischen System läßt sich das nicht ausschließen.

Die Systemnützlichkeit wird u.a. danach beurteilt, ob die Lösungen Benutzern signifikant helfen, ob die Ergebnisse angemessen organisiert und detailliert sind und ob eine bedienungsfreundliche Benutzerschnittstelle einen akzeptablen Umgang mit dem System erlaubt. Wegen der häufig erforderlichen Wartung ist schließlich die Änderungsfreundlichkeit des System von entscheidender Bedeutung.

7.1.5.2 Rapid Prototyping vs. modellbasierter Ansatz


Als Nachteil des Rapid Prototyping wird gesehen, daß die Repräsentation der Expertise tendenziell durch Implementierungsformalismen bestimmt wird. Daß die Systemarchitektur vor der abschließenden Analyse der Wissensprotokolle festgelegt werden muß, beengt die Sicht auf die Expertise; spätere Änderungen sind teuer. Die Planung ist unübersichtlich: selbst während der Implementierung ist nie recht klar, wieviel noch zu tun ist (z.B. akquiriert werden muß). Es ist ferner ungewiß, ob die Struktur des kleinen Systems tatsächlich im Großen standhält („scaling up“). Vor allen Dingen wird aber keine explizite Entscheidung über das Wissensmodell getroffen; es ergibt sich im Verlauf der Entwicklung.

Im modellbasierten Ansatz wird dagegen versucht, Modelle des Problemlöseprozesses explizit zu machen: das bedeutet die Offenlegung des mentalen konzeptionellen Modells, das sich der Knowledge Engineer von der beobachteten Expertise macht, unabhängig von ir-
7.1. EXPERTENSYSTEME UND WISSENSMODELLIERUNG

Abbildung 7.16: Erarbeiten der Schichten des Design-Modells, vgl. auch Abb. 7.5 und 7.6


Der wohl bekannteste modellbasierte Entwicklungsansatz ist KADS. Das dort festgelegte Vorgehen im Rahmen der Wissensakquisition und bei der Erstellung des konzeptuellen Modells ist in Abschnitt 7.1.3.5 behandelt worden. Wichtig für die Entwicklungsmethodik ist die Tatsache, daß hier mehrere Zwischenschritte zwischen dem Erwerb von Expertenwissen und der Implementation eingelegt werden, was die zu leistende Aufgabe in Abschnitt zwei geringerer Komplexität aufteilt. Die Bedeutung des ersten Abschnitts, der Erstellung des konzeptuellen Modells, liegt darin, daß das auf der Inferenzebene, Aufgabebene und Strategieebene formulierte Interpretationsmodell den Knowledge Engineer bei der Wissensakquisition und auch bei der Auswahl von Werkzeugen für die Systementwicklung leiten kann.

Auf dem konzeptuellen Modell baut das sogenannte Design-Modell auf. Geht es bei dem


7.1.5.3 Komponenten von Expertensystemen


Die Erklärungskomponente stellt für den Endbenutzer das Lösungsergebnis und den Lösungsweg dar und begründet evtl. die gefundene Lösung. Hier ist besonders darauf hinzuweisen, daß sinnvolle Erklärungen nur dann möglich sind, wenn man bei der Syste-
7.1. EXPERTENSYSTEME UND WISSENSMODELLIERUNG


Die Benutzerschnittstelle insgesamt muß wegen der großen Komplexität der zu lösenden Aufgaben sehr komfortabel sein; sie muß den Benutzer einerseits ausreichend führen, damit er sich nicht verirrt und nichts vergißt, ihm aber andererseits genügend Freiheit lassen, alles relevante Wissen in der für ihn natürlichen Reihenfolge einzugeben und irrtümliche Eingaben nachträglich zu korrigieren. Eine moderne Dialogtechnik (Masken, Menüs, Grafik) ist nötig, die es erlaubt, mehrere Fragen im Zusammenhang zu beantworten. Wichtig ist dabei eine ausreichende Reaktionsgeschwindigkeit des Systems und ein großer Bildschirm, auf dem mehrere Fenster gleichzeitig sichtbar gehalten werden können.
7.1.5.4 Expertensystemwerkzeuge

Da bei gegenwärtigen Expertensystemprojekten in der Regel nicht mehr „from the scratch“, sondern mit Unterstützung von Expertensystemwerkzeugen gearbeitet wird, wollen wir im folgenden die Rolle solcher Werkzeuge diskutieren. Im Anschluß stellen wir dann einige hybride Expertensystemwerkzeuge vergleichend gegenüber.

Arten von Werkzeugen. Für die Entwicklung eines Expertensystems kann man auf drei verschiedenen Ebenen aufsetzen: (K1-)Programmiersprache, Expertensystemwerkzeug, Expertensystemshell.


Nexpert Object. Hinsichtlich Hardware-Umgebung und soweit nichts speziell anderes bemerkt, kann man sagen, daß die grafikorientierten Entwicklungsumgebungen vorwiegend für Unix-basierte Workstations, die eine hohe Rechenleistung und genügend Hauptspeicher und Swapkapazität pro Arbeitsplatz zur Verfügung stellen können, konzipiert sind und eine solche Umgebung auch erfordern. Bei allen Werkzeugen muß man die zugrundeliegende Programmiersprache beherrschen, wenn man objektorientiert programmieren möchte.


Man vermißt bei diesen drei Werkzeugen die Unterstützung von Regressionstests; Schnittstellen zu Datenbanken sind nicht vorhanden oder müssen zusätzlich gekauft werden.


Die hauptsächliche Unterstützung, die solche hybriden Expertensystemwerkzeuge leisten, liegt darin, dem Knowledge Engineer komfortable Ausdrucksmöglichkeiten für Wissen

**Expertensystemshells.** Expertensystemshells sind problemtypspezifische Werkzeuge, die eine bereits implementierte Problemlösungsstrategie für einen speziellen Problemlösungstyp zur Verfügung stellen, dem Verwender aber den eigenständigen Aufbau einer dazu passenden Wissensbasis mit bereichsspezifischem Problemlöswissen überlassen. Bekannte Diagnoseshells sind D3 (Puppe), TEST bzw. TESTBENCH (Carnegie Group), ESS (Coherent Thought); ein Konfigurierungssystem ist PLAKON [Cunis et al., 1991], s. auch S. 786. Wir erläutern beispielhaft D3 [Bamberger et al., 1991], das nach dem Prinzip der heuristischen Klassifikation arbeitet. Neben der Symptom- und Diagnosehierarchie ist Wissen über Symptome einer Diagnose, über die Prädispositionen von Diagnosen und über probabilistische Korrelationen zwischen Symptomen und Hypothesen erforderlich. Es wird durch Auswahl aus angebotenen Menüs und durch Eintrag in Tabellen direkt vom Experten eingegeben. Wie bei allen Shells muß der Experte dazu vorher in die Grundstrukturen des Werkzeugs, die Arten benötigten Wissens und Kriterien für sinnvolle Hierarchisierung eingeführt werden, was jedoch durch die grafische Entwicklungsoberfläche vereinfacht wird. Durch Anhängen von geeigneten Fragegruppen kann der Experte außerdem deklarativ die Benutzerschnittstelle schaffen. Das System ist stets unmittelbar ablauffähig, kann also jederzeit durch die Expertin oder den Experten getestet werden.

Man sollte sich darüber im klaren sein, daß eine Diagnosehull notwendigerweise für alle Diagnoseaufgaben geeignet ist, z.B. wenn die Lösung einer Aufgabe über die hier dargestellten Wissensarten Symptome, Diagnosen und deren statistische Bezüge hinaus weitere Wissensarten benötigt. Entsprechendes gilt für Shells für andere Problemlösungstypen.

### 7.1.6 Resümee

**Erfolg und Grenzen der Expertensysteme** Expertensysteme haben begonnen, aus den Forschungslabors in die Anwendung vorzudringen. Es hat sich herausgestellt, daß wirklich im Wissen Macht liegt (und Problemlösen nicht einfach durch Means-Ends-Analysis oder schlaue Beweiser möglich ist). Gleichzeitig ist jedoch offenbar geworden, wie umfassend und divers menschliches Wissen ist. Deshalb sind die Anwendungen bislang stärker


Aber: Es könnte qualifizierte Tätigkeiten geben, die in dem bisher bekannten Rahmen noch nicht verstanden werden können; es gibt Leute, die solchen Typisierungen insgesamt skeptisch gegenüberstehen, da sich die Komplexität bestimmter qualifizierter Tätigkeiten einer Typisierung verschließt. Und das „unspezifisch-relationale“ Wissen ist bisher nicht Gegenstand der Betrachtung (auch bei KADS nicht).


Knowledge Engineering ist wesentlich eine Design-Aufgabe. Modelle, die zunächst Abbild von Expertise und dann Vorbild für Expertensysteme sind, scheinen ein gutes Leitbild für das Knowledge Engineering zu liefern. Während man früher dazu tendierte, das Wissen gleich bei der Erfassung in die Repräsentationsform zu „gießen“ (mittlerweile als „Kübelmethode“ verrufen), hat die Erfahrung gezeigt, daß man meistens besser beraten ist, zuerst ein konzeptionelles Modell zu erstellen, das die Expertise interpretiert und das direkt oder über den Zwischenschritt eines weiteren Modells als Vorlage für die Implementierung des Expertensystems dient. Die Schwierigkeit dabei: Das konzeptionelle Modell ist maßgeblich für die Analyse des Expertenwissens und Ausgangspunkt für das Design eines wissens-

Ein Problem ist die Bewertung der Gültigkeit der erstellten Modelle und ihre Abhängigkeit von zeitlichen Veränderungen des Aufgabenbereichs oder auch des Wissens darüber. Ferner muß der Aspekt der Strukturierung von bereichsspezifischem Wissen noch weiter geklärt werden, zum einen auf der technischen Seite (wie läßt sich das effizient und komfortabel realisieren?), zum andern empirisch auf der kognitiven Seite (bildet Expertenwissen ein in sich kohärentes System, oder gibt es eher modulare Wissenseinheiten?). Besonders wichtig wäre dies für die Realisierung von Systemen mit größeren Kompetenzbereichen, denn für die notwendigen großen Wissensmengen scheint eine auf der Wissensebene vorgenommene Modularisierung unabdingbar.


werden. Kann man solche Systeme, die sich aufgrund ihrer Erfahrungen vielleicht von Tag zu Tag ändern, für wichtige Entscheidungen einsetzen?