Wissensakquisition und Wissensrepräsentation in Expertensystemen

Günter Bachelier

23 November 1995
(Verändert Jan. 2004)

Inhalt

1) Einführung in Expertensysteme

1.1) Expertensysteme als eine Ausprägung von Informationssystemen

Ziel der Fachkommunikation ist die „Vermittlung von Fachwissen zwischen Experten bzw. zwischen Experten und interessierten Laien“ (Zimmermann (1990: 1102)), wobei zur Unterstützung dieser Aufgabe Informationssysteme (IS) verwendet werden. Informationssystem ist ein Sammelbegriff für Systeme, mit den Hauptfunktionen Aufnahme, Verarbeitung, Wiedergabe von Informationen. Folgende IS-Typen werden in der Regel unterschieden: Datenbank-Systeme, Information-Retrieval-Systeme, natürlich-sprachliche Frage-Antwort-Systeme, wissensbasierte Systeme wie z.B. Experten-Systeme.

Diese IS-Typen können als alternative Ausprägungen eines gemeinsamen Modells aufgefaßt werden (siehe auch Panyr (1986: 22f)), welches sich als Quintupel IS = (A, W, I, V, O) beschreiben läßt. Die einzelnen Modellelemente sollen im folgenden am Beispiel der IS-Ausprägungen Information-Retrieval-System (IRS) und Expertensystem (XPS) dargestellt werden.

1) Erschließungsfunktion (A): Hierbei handelt es sich um Methoden, mit denen das Wissen in das IS gelangt, unabhängig von den Quellen und Repräsentationsformaten. Bei XPS entspricht dies die Phase der domänenspezifischen Wissensakquisition; bei IRS entspricht dies dem Indexierungsvorgang, dem Thesaurusaufbau und der automatischen (Dokument und/oder Term-)Klassifikation.

2) IS-Inhalt (W): Der Inhalt eines IS besteht aus Wissenseinheiten über bestimmte Aspekte von Objekten und ihren Beziehungen, die mit einem bestimmtem Detailliertheitsgrad und in einem bestimmten Repräsentationsformat abgelegt sind. Bei XPS ist dies die domänenspezifische Wissensbasis aus Fakten und Regeln; bei IRS ist dies der Dokumentenbestand, die Term- und Dokumentklassifikation und der Thesaurus.
3) Menge aller zugelassenen Inputs (I): Der erste Schritt der Nutzung des IS ist die Eingabe eines Inputs in einem bestimmten Repräsentationsformat. Bei XPS sind dies Problemformulierungen (Problemfall spezifisches Wissen); bei IRS sind dies Suchfragen nach Dokumenten oder Termen bzw. nach bestimmten Klassen von Dokumenten bzw. Termen.
4) Menge aller Verarbeitungsmechanismen (V): Der zweite Schritt der IS-Nutzung besteht in der Anwendung von intern ablaufenden Prozessen, die den Input auf einen Output abbilden. Bei XPS sind dies Problemlösungsschritte wie z.B. Inferenzmechanismen in der Logik oder der Anwendung von Regeln; bei IRS entspricht dies den Retrieval- und Adaptionsstrategien (z.B. Clusteraktualisierung).
5) Menge aller möglichen Outputs (O): Im letzten Schritt werden die Ergebnisse der Verarbeitungsmechanismen ausgegeben. Bei XPS sind dies Lösungsvorschläge des Problems; bei IRS sind dies nachgewiesene Dokumente oder z.B. eine erweiterte Termmenge (bei Queryexpansion).

Entsprechend den Bezeichnungen aus dem IS-Quintupel ist der Gegenstand dieser Arbeit die Erschließungsfunktion A und der IS-Inhalt W im Zusammenhang von Expertensystemen.

1.2) Eigenschaften Wissensbasierter Systeme

Wissensbasierte Systeme besitzen herausragende Eigenschaften im Vergleich zu den anderen Ausprägungen von Informationssystemen. Beispiele dieser Eigenschaften sollen im folgenden dargestellt werden, wobei IRS als Vergleich verwendet werden:

1) Informative Wissensrepräsentation: Die Repräsentation von Wissen in IS unterliegt, wie oben erwähnt, unterschiedlichen Sichtweisen und Abstraktionsgraden. In IRS (Referenz-IRS) werden Surrogate von Dokumenten erzeugt, in denen Wissen über den Inhalt kodiert ist, jedoch nicht der Inhalt selbst, wodurch man die Art der Repräsentation in Analogie zum Referieren als indikativ bezeichen könnte. Dies ergibt sich dadurch, daß Objekte (Terme) und je nach Modell zusätzlich ihre Bewertungen aus den Original-Dokumenten ermittelt werden, nicht jedoch die konkreten Beziehungen der Objekte zueinander. Die Textsurrogate repräsentieren somit ein Dokument auf einem geringeren Abstraktions- oder Detailliertheitsniveau als eine Repräsentation der Dokumentinhalte in einem WbS, bei dem Beziehungen der Objekte z.B. durch Semantische-Netz-Modelle (siehe 5.2) mitkodiert werden. Wird Wissen, das z.B. als natürlichsprachlicher Text (evtl. erweitert um Graphiken, Tabellen, Formeln, u.ä.) in Dokumentenform vorliegt, in einem wissensbasiertem System abgelegt, so kann dies als informative Repräsentation betrachtet werden, da das Wissen konkret vorhanden und maschinell verarbeitbar vorliegt.
   Der Output eines IRS kann je nach Modell eine Dokumentreferenz oder z.B. eine Textpassage (beim Passagen-Retrieval in Volltext-IRS) sein, wobei der Nutzer aus dieser Repräsentationsform das Wissen, das er zu seiner Problemlösung benötigt, selbst extrahieren muß. Bei WbS kann dem Nutzer das Wissen in einer Repräsentationsform präsentiert werden, die konkreter und seinem Problem adäquater ist.
   (Bsp.: Anfrage: Herstellung eines bestimmten chemischen Produktes. Antwort(IRS): Referenz von

Dokumenten in denen die Produktherstellung (möglicherweise) beschrieben wird. Antwort(WbS): Bestandteile (Edukte) und zugehörige Verfahren zur Erzeugung der Produkte (wenn dieses Wissen Bestandteil der WB ist).

2) Repräsentation der Suchfrage: Die Problemformulierungen bei wissensbasierten Systemen müssen ebenso wie Suchfragen bei IRS in einem Repräsentationsformat vorliegen, das mit dem Format von W kompatibel ist, um eine Suche oder Ableitung zu ermöglichen. Der Input in ein XPS kann jedoch wesentlich komplexer formuliert werden als die Angabe von Termen und ihren Gewichtungen, da die Beziehungen zwischen Objekten Teil der Suchfrage sein können.
   Verfügt der Nutzer, der eine Suchfrage stellt, über eine individuelle Wissensbasis bezüglich des betreffenden Fachgebietes, die kompatibel zu der Wissensbasis des IS ist, so kann bei der Bearbeitung der Suchfrage ein Abgleich der beiden Wissensbasen durchgeführt werden. Dieses Vorgehen besitzt große Vorteile, da z.B. ermittelt werden kann, was bezüglich der Suchfrage dem Nutzer schon bekannt ist, bzw. das IS kann den Nutzer auf Inkonsistenzen der beiden Wissensbasen aufmerksam machen.
3) Implizites Wissen: Der Input in ein IS bestimmt (zusammen mit den anderen Elementen der 5-Tupels) seinen Output in der Analogie zu einem Constraint. Bei IRS wird der Output ausschließlich durch Suchoperationen (Retrieval-Strategien) festgelegt. Bei WbS kommen neben Suchoperationen auch Inferenzmechanismen zur Anwendung, bei denen neue Wissensobjekte aus vorhandenen erzeugt werden, die vorher nicht vorhanden waren, und die die Anforderungen erfüllen sollen, die durch den Input determiniert wurden. Solche Mechanismen sind z.B. logische Ableitungen neuer Aussagen (siehe 5.3) oder die Bestimmung von Attributswerten innerhalb von Vererbungshierarchien (siehe 5.2).
   Ein vergleichbares Vorgehen ist bei IRS nicht zu finden, jedoch sind erste Anwendungen im Bereich der Datenbanken durchgeführt worden (Deduktive Datenbanken).
4) Nachprüfbarkeit der Ergebnisse: Die Ergebnisse, die ein IS aufgrund einer Suchfrage bzw. einer Problemstellung liefert, müßten durch den Benutzer auf ihre Richtigkeit und Vollständigkeit nachprüfbar sein. In IRS sind die Retrieval-Strategien ebenso wie der Inhalt des IS in der Regel nicht zugänglich (black box). Ansätze, das Systemverhalten und den Inhalt transparent zu machen, sind vor allem durch Visualisierungskomponenten versucht worden (Steinadler, KOAN, LYBERWORLD; PSS-Kohdex).
   In WbS und insbesondere bei XPS die Shells verwenden, wird die Transparenz direkt von einer Architekturkomponente (Erklärungskomponente, siehe 1.3) unterstützt.
5) Variable Darstellungsformen und Reorganisation: Man kann den Wissensbegriff in eine Vielzahl von Arten zerlegen (sicheres, unsicheres, heuristisches, vages, default, ... Wissen). Moderne wissensbasierte Systeme versuchen Repräsentationsstrukturen und Verarbeitungsmechanismen für eine möglichst große Anzahl dieser Wissensarten bereitzustellen. Hierbei kann das gleiche Wissen durch verschiedene Formalismen mit je eigenen Inferenzmechanismen repräsentiert werden, oder es werden verschiedene Aspekte mit je geeigneten Formalismen repräsentiert.
   Eine maximale Flexibilität wird erreicht, wenn Konvertierungsmöglichkeiten zwischen den einzelnen Repräsentationsstrukturen bestehen, da auf diese Weise die Vielzahl der Verarbeitungsmechanismen für das gesamte Wissen verwendet werden kann.
   In der Regel besitzt ein bestimmtes IRS im Gegensatz hierzu nur eine Repräsentationsstruktur (z.B. Binär-Vektoren beim Boolschen Ansatz) mit einem darauf spezialisierten Verarbeitungsmechanismus.
6) Ungenauer Input: Ungenauer Input wird von konventioneller Datenverarbeitung zurückgewiesen, da eine close-world-assumption vorliegt, d.h. Unbekanntes wird als falsch angenommen. Im Gegensatz hierzu

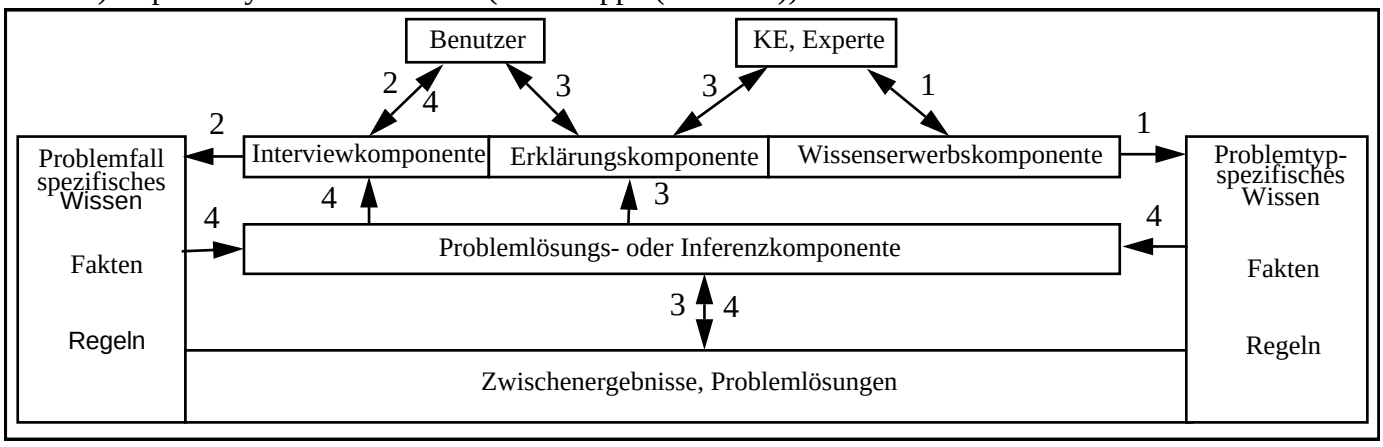
stehen viele Ansätze wissensbasierter Systeme, denen eine open-world-assumption zugrunde liegt, d.h. Unbekanntes wird als nicht bekannt behandelt und mit diesem Status kann weiter gearbeitet werden.
Ungenauer Input wird jedoch auch von den IRS-Modellen auf verschiedene Weise unterstützt, um einem ungenau formulierten Informationsbedürfnis entgegen zu wirken, was durch Trunkierung oder durch probabilistische Modelle geschieht.

7) Adaptive Erweiterbarkeit: Bei der adaptiven Erweiterbarkeit werden neue Elemente in eine gegebene Organisationsstruktur eingefügt, und verändern die Elemente ihrer Umgebung. Eine Entfernung würde dann irreversible Einflüsse besitzen, wenn während der Zeit, in der das Element eingefügt war, durch Inferenzen neue Elemente erzeugt wurden, wobei das Element bei der Inferenz beteiligt war. Die adaptive Erweiterbarkeit wird als ein wichtiges Element der menschlichen Intelligenz bezeichnet und findet in vielen WbS-Modellen (insbesondere in nichtmonotonen Logiken) ihre Anwendung.
Bei IRS ist diese Forderung erfüllt, bei denen das Vektorraummodell in Verbindung mit einer Clusterung der Repräsentationseinheiten verwendet wird, da das Einfügen eines neuen Elementes den Clusterrepräsentanten verändert, und bei komplexeren Modellen eventuell zu einer Aufspaltung oder Verschmelzung von Clustern führen kann.

1.3) Expertensystem-Architektur

Expertensysteme sind wissensbasierte-Systeme, die versuchen, Probleme innerhalb einer begrenzten Domäne zu lösen, indem die Vorgehensweise menschlicher Experten simuliert wird. Ihr allgemeiner Einsatz liegt dort, wo komplexe und ungenau formulierbare Probleme sich einer direkten algorithmischen Modellierung durch konventionelle DV entziehen.

Abb. 1) Expertensystem-Architektur (nach Puppe (1991: 13))



Erläuterungen zu Abb. 1:

1) Wissenserwerbskomponente (WEK): Der Wissens-Ingenieur (KE) oder Experte gibt als Wissensquellen bereichsspezifisches Domänenwissen in das XPS ein oder ändert vorhandenes Wissen. Hierzu wird eine WEK verwendet, die als einfacher Editor bis hin zu graphischen Werkzeugen realisiert sein kann, in denen Objekte und Relationen als Semantische Netze gezeichnet werden können.
2) Interview- bzw. Dialogkomponente (IK): Durch den Benutzer wird im Problemfall spezifisches Wissen eingegeben, das sein Problem beschreibt. Hierzu wird eine IK verwendet, die ähnlich gestaltet sein kann wie die WEK.

3) Erklärungskomponente (EK): Der Benutzer sowie KE und Experte können die EK für unterschiedliche Zwecke nutzen. Ihre Funktion besteht in beiden Fällen in der Verdeutlichung der Abläufe des XPS.
   Der Benutzer kann fragen, wie das System zu einer vorgestellten Lösung gekommen ist oder warum einzelne Lösungsschritte verwendet wurden.
   Der KE oder Experte kann die Debugging-Funktion der EK verwenden, indem er prüft, ob das System zu richtigen Lösungen, auch mit Hilfe eines als akzeptabel klassifizierten Weges gelangt, bzw. prüfen, wo das System bei falschen Lösungen während des Lösungsweges einen nicht akzeptablen Inferenzschritt getätigt hat. Die Erklärungskomponente verwendet dabei die abgelegten Zwischenergebnisse, um den Lösungsweg (Trace) nachzuvollziehen.
4) Problemlösungs- oder Inferenzkomponente (PLK): Nachdem die Beschreibung des zu lösenden Problemfalles eingegeben ist, kann der Benutzer das System zu einem Lösungsvorschlag auffordern. Hierzu wird die PLK aktiviert, die Wissen aus der Problemfall- und Problemtyp-Wissensbasis lädt und bearbeitet, wobei Zwischenergebnisse und die vorgeschlagene Problemlösung in einen eigenen Speicher abgelegt werden.

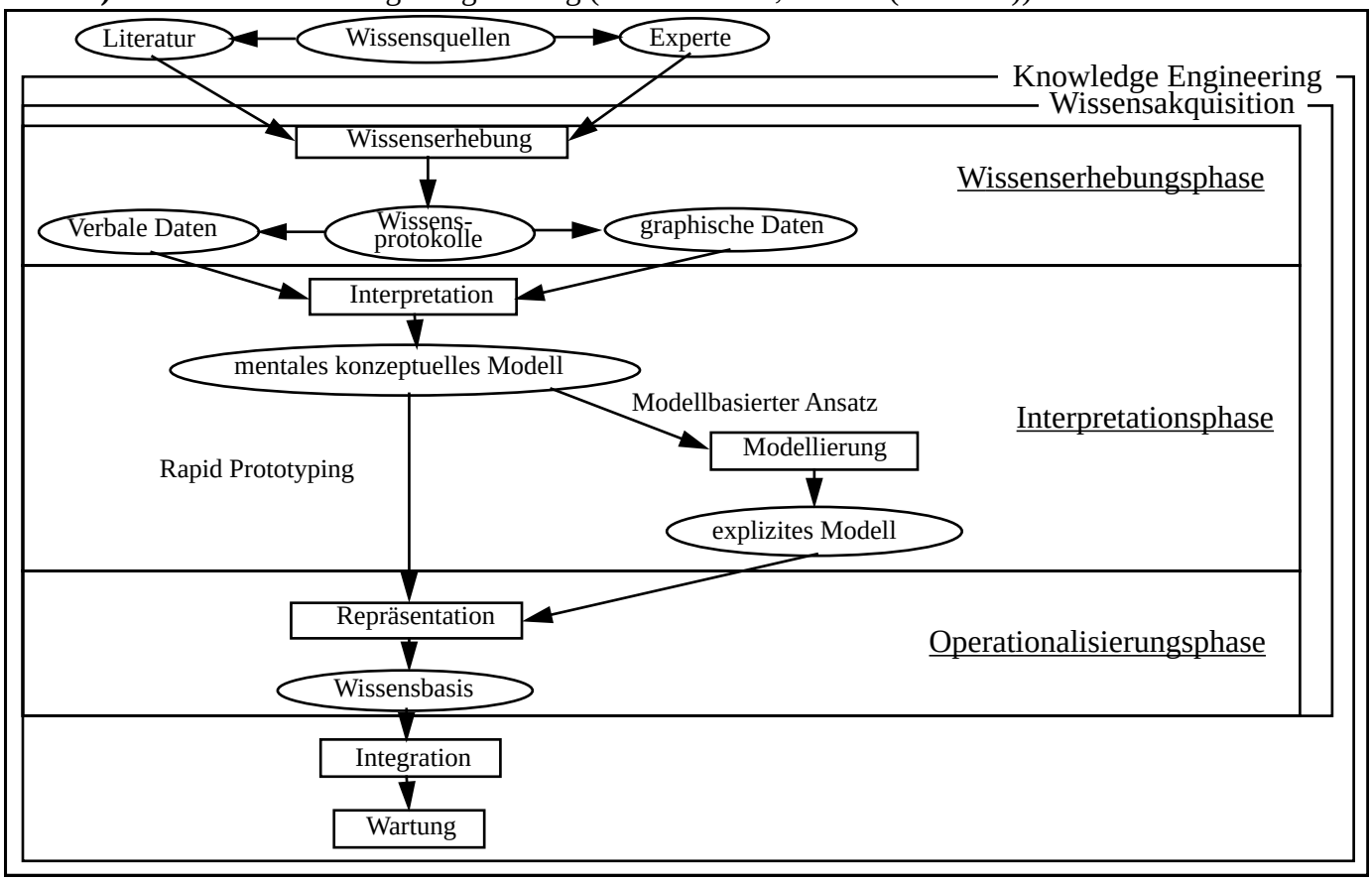
2) Knowledge-Engineering

Mit Knowledge Engineering wird der gesamte Erstellungs- und Wartungsprozeß eines wissensbasierten Systems beschrieben, von den Machbarkeitsstudien, der Toolauswahl, der Wissensakquisition, der Integration bis hin zur Wartung und Erweiterung. Dieser Prozeß wird ähnlich dem Software-Engineering in Phasen eingeteilt, die in Abb. 2 auf Seite 7 dargestellt sind.

1) Wissens-Akquisition: Erhebung von Wissen aus verschiedenen Wissensquellen und Umsetzung in eine operationale Wissensbasis, wobei die spätere Wartung der Wissensbasis erleichtert werden soll.
   Wissensakquisition wird von vielen Autoren als Flaschenhals der XPS-Entwicklung bezeichnet, was unterschiedliche Gründe besitzt:
   1) Quantitative Gründe
   1.1) Komplexe Anwendungsbereiche mit langen Einarbeitungszeiten.
   1.2) Große Menge von Wissen (Konzepte und Zusammenhänge) und Anwendungsfälle.
   2) Qualitative Gründe
   2.1) Keine exakte Expertise-Definition (was soll auf welcher Abstraktionsebene erfaßt werden).
   2.2) Keine eindeutigen Modelle, wie Expertise beim Experten organisiert ist.
   2.3) Keine formalen Methoden zur Überführung von implizitem W in explizite Strukturen.
   2.4) Keine geschlossene WR (nichtmonotone, räumliche, temporale, Common Sense-Inferenz).

   Wissensakquisition wurde in den ersten Ansätzen der XPS-Entwicklung als ein einfacher Transferprozeß verstanden, bei dem Wissen direkt in eine Wissensbasis umgesetzt werden kann, ähnlich wie in einem einfachen Sender-Empfänger-Kommunikationsmodell. Diese Sichtweise hat sich jedoch als nicht adäquat erwiesen, sodaß man in jüngeren Ansätzen dem Knowledge Engineer eine wesentlich aktivere Rolle zuspricht, indem er das erhobene Wissen interpretieren muß. Durch diese komplexe intellektuelle Tätigkeit besitzt er eine ähnliche Vermittlerrolle wie der Schreiber eines Abstractes im Abstracting-Prozeß, oder ein Übersetzer im Übersetzungs-Prozeß.

Abb. 2) Phasen des Knowledge-Engineering (nach Karbach, Linster (1990: 10))



2) Wissens-Erhebung: Dokumentation von Expertise mit Wissensprotokollen als Ergebnis, die verbal (schriftlich oder akustisch) sein kann, aber auch Zeichnungen oder Videoaufnahmen enthalten kann.
3) Interpretationsphase: Interpretation der Wissens-Protokolle durch den KE durch die Tätigkeiten Verstehen, Selektieren und Aggregieren, wodurch ein mentales konzeptuelles Modell der Expertise entwickelt wird. Dieses Modell kann implizit (siehe 4.1) oder explizit (siehe 4.2) sein. Es werden Strukturen des Faktenwissens und grundlegende Inferenzverfahren ermittelt, sowie Inkonsistenzen und Beziehungen zwischen den Protokollen identifiziert.
4) Operationalisierungsphase: Umsetzung der interpretierten Daten in operationale WR-Formalismen.
5) Integration: Einbettung in bestehende DV-Umgebungen (z.B. DB).
6) Wartung: Anpassung an sich verändernde Inhalte in Wissensquellen.

Gegenstand der weiteren Ausführungen sind die drei Phasen der Wissensaquisition: Erhebung (Kapitel 3), Interpretation (Kapitel 4) und Repräsentation (Kapitel 5).

3) Wissenserhebung

WE ist der erste Schritt bei der Umsetzung der Expertise, wobei das Wissen zunächst vom KE in einer möglichst uninterpretierten Form protokolliert werden soll. Eine Betrachtung der Herkunft der Techniken zeigt, daß viele aus der Experimental-Psychologie und der Wissensdiagnostik aus der Kognitiven Psychologie für die WE wiederentdeckt wurden. Es werden im folgenden 4 Klassen von WE-Techniken angesprochen:

1) Interviewtechniken: Explizite Fragen des KE und Introspektion des Experten.
2) Beobachtungen: Vorgehen des Experten bei konkreten Problemstellungen wird beobachtet.
3) Indirekte Techniken: Erhebung von Wissenselementen und deren Aggregation zu Strukturen, die der Experte zu Beginn nicht selbst verbalisieren konnte.
4) Automatische WE (Maschinelles Lernen): WbS gewinnt selbst aus Falldaten oder Fachliteratur Wissen.

3.1) Interviewtechniken am Beispiel strukturiertes Interview

Beim strukturierten Interview wird eine Agenda von Punkten besprochen, die sich aus unstrukturierten Interviews und aus Modellbasierten Ansätzen von Problemlöseprozessen (siehe auch Abschnitt 4.2)) ergeben. Vorbedingung ist die Kenntnis über die Terminologie der Domäne, das der Knowlege Engineer z.B. aus Handbüchern erworben haben muß.
Man kann folgendes 3-gliedriges Ablaufschema zugrunde legen:

1) Einführung mit der Vorstellung der Themen, des Ablaufes und des Ziels des Interviews.
2) Ansprechen von Punkten und eventuelles Nachfragen.
3) Zusammenfassung und die Auswertung gemeinsam mit dem Experten.

Es werden mit dem strukturierten Interview folgende Ziele verfolgt:

1) Erwerb von nicht dokumentiertem Faktenwissen und relevanten Problemtypen.
2) Aussagen über Objekte und Agenten in der Experten-Arbeitsumgebung.
3) Charakterisierung der späteren Benutzer.

3.2) Beobachtungen am Beispiel Protokollanalyse

Bei der Beobachtung wird das Vorgehen des Experten aufgezeichnet (mit Tonband, Video oder Interaktion mit einem Simulations-Modell), und später analysiert und interpretiert. Es läßt sich somit Wissen erheben, das sich der direkten Verbalisierung durch den Experten entzieht (siehe auch 3.4)).

Der Experte löst bei der Protokollanalyse einen Anwendungsfall und schildert sein Vorgehen, wobei eine gewisse Vertrautheit des Experten mit lautem Denken vorausgesetzt werden muß.
Es können 2 Ansätze der Protokollanalyse (PA) unterschieden werden:

1) Konkurrente PA: Gleichzeitiges Problemlösen und lautes Denken, sodaß keine Zeit zum Theoretisieren über Handlungen bleibt. Die Durchführung geschieht innerhalb der normalen Arbeitswelt des Experten, wobei ein korrekter, aber nicht vollständiger Lösungsweg dokumentiert wird, der daher durch ein strukturiertes Interview ergänzt werden muß.
2) Retrospektive PA: Expertenbefragung sofort nach der Problemlösung.

Es werden folgende Ziele verfolgt:

1) Ermittlung wann und wie spezifisches Wissen eingesetzt wird.
2) Erkennung von Problemlösungs- und Schlußfolgerungsstrategien.
3) Ausführung von Prozeduren und deren Auswahlkriterien.
4) Aufteilung der Hauptaufgabe in Unteraufgaben.

Bei der Protokollanalyse spielt die Auswahl der Fallbeispiele die entscheidende Rolle:
1) Typische Aufgaben: Sie werden am besten verstanden, kommen häufig vor und haben große Relevanz.
2) Limitierte Information: Aufgaben, bei denen weniger Informationen als im Praxisfall zur Verfügung stehen. Beobachtet wird, wie Defizite mit versteckten Heuristiken und Strategien begegnet werden.
3) Zeitbeschränkungen für Problemlösungsschritte: Hierbei soll auf Aspekte geschlossen werden, die für Experten wichtig sind bzw. sie schnell voranbringen.

3.3) Indirekte Techniken am Beispiel Konstruktgitter-Verfahren

Bei den indirekten Erhebungstechniken wird berücksichtigt, daß nicht alle Wissensbereiche einer Expertise durch den Experten direkt verbalisiert werden können. Es werden Wissenselemente erhoben, die sukzessive zu einer Gesamtstruktur verbunden werden, ohne daß der Experte zu Beginn in der Lage wäre, diese Struktur selbst zu formulieren.
Der Experte kann weitgehend selbständig mit diesen Techniken arbeiten, was zu weniger Einflußnahme und Interpretationsfehlern führen kann.
Das Konstruktgitter-Verfahren (KGV) wurde von George Kelly im Rahmen der Personal-Construct-Psychology entwickelt, und basiert auf der Annahme, daß persönliche Konstrukte durch dichotome (2-polige) Attribute aufgebaut sind.
Die Durchführung des KGV geschieht in 2 Phasen:
1) Erhebung
   1.1) Der Experte nennt wichtige Begriffe der zu modellierenden Domäne.
   1.2) Bildung von Begriffstripel.
   1.3) Der Experte nennt ein Attribut, das zwei der drei Begriffe gemeinsam hat, das dritte jedoch nicht.
   1.4) Einordnung der beiden Begriffe in eine Ratingskala bezüglich des Attributes aus 1.3).
2) Auswertung, z.B. durch eine hierarchische aggregative Clusteranalyse, bei der das Konstruktgitter in eine Ähnlichkeitsmatrix umgewandelt wird. Je nach der verwendeten CA können sich unterschiedliche Beziehungen zwischen den Konstrukten ergeben.

3.4) Automatische Wissenserhebung am Beispiel ID3

Automatische WE-Verfahren versuchen aus Beispielen, die aus einer Falldatenbank entnommen werden, eine WB aufzubauen oder eine bestehende Wissensbasis zu modifizieren.
Es lassen sich somit folgende Einsatzbereiche von Lernmethoden innerhalb XPS-Architekturen benennen:
1) Modifizierung der Bewertung von Wissenselementen (z.B. Evidenzwerte von Regeln).
2) Modifizierung der Wissensstruktur bei Erhaltung der Anzahl der Wissenselemente.
3) Generierung neuer Wissenselemente.

ID3 ist ein Lernalgorithmus der symbolischen Induktion, der aus Beispielmengen (Attribut-Wert-Listen mit negativen und positiven Beispielen eines zu lernenden Konzeptes) Klassifikationsregeln generiert, was ihn zu einem Bsp. der Kategorie 3 macht. Verwendet wird ein entropiebasierter Ansatz, d.h. es wird zuerst das Attribut betrachtet, mit dem man die Unordnung am besten reduzieren kann.
Als Voraussetzungen der Anwendung dieses Verfahrens dürfen keine Widersprüche in den Beispielen vorhanden sein, und es müssen genügend Beispiele vorhanden sein, da sonst Attribute wegfallen können.

Verfahrensablauf

1) Für jede Ausprägung a_{1k} eines Attributes A_1 wird seine Entropie E_{1k} berechnet:
   1.1) Bestimme die relative Anzahl der Bsp., bei denen a_{1k} auftritt: p_{ik}.
   1.2) Bestimme die relative Anzahl der Bsp., bei denen a_{1k} auftritt und das Bsp. positiv ist: q_{ik}.
   1.3) Berechne die Entropie der Attributsausprägung: $E_{1k} = -q_{ik}\, ld(q_{ik}) - (1-q_{ik})\, ld(r_{ik})$.
2) Berechne den Informationsgehalt des Attributes A_1: $IG(A_1) := \Sigma\, p_{ik}\, E_{1k}$.
3) Bestimme das Spaltungs-Attribut als das Attribut mit dem kleinsten Informationsgehalt.
4) Teile die Menge der Bsp. so auf, daß sich in den Teilmengen (Blätter) nur Bsp. befinden, die eine Ausprägung des Spaltungs-Attributes besitzen.
5) Befinden sich in den neuen Teilmengen Bsp. mit unterschiedlichen Erfolgswerten, so müssen diese Blätter durch eine weitere Iteration aufgespalten werden.

4) Wissensinterpretation

Bei der Wissensinterpretation durch den KE soll aus den Wissensprotokollen ein mentales konzeptuelles Modell der Expertise erzeugt werden. Dieses Modell kann implizit (siehe Rapid-Prototyping, Abschnitt 4.1) oder explizit (siehe Modellbasierte Ansätze, Abschnitt 4.2) sein.

4.1) Rapid Prototyping

Es können 3 Arten von Rapid Prototyping (RP) unterschieden werden:

1) Exploratives RP: Formulierung von Anforderungen an das System.
2) Experimentelles RP: Simulation von einzelnen Systemfunktionen.
3) Evolutionäres RP: Erste Systemversion, bei dem anhand einer kleinen Menge von erhobenen Fallbeispielen ein Prototyp konstruiert wird, der aufgrund weiterer Erhebungen sukzessive erweitert und verfeinert wird, bis er die gewünschten Anforderungen erfüllt. Kennzeichnend bei diesem Vorgehen ist die Vermischung der Teilaufgaben des Knowledge Engineerings.

Vorteile:

1) Hinweise auf fehlendes oder ungenügendes Wissen in der Wissensbasis durch den Experten.
2) Direkte Überprüfbarkeit von Wissenstransfer, Fakten, Inferenzstrategien und System-Verhalten.

Nachteile:

1) Repräsentation entspricht Implementierungs-Forderungen, aber nicht der Experten-Begriffswelt.
2) Phaseneinteilung und Meilensteine sind nicht vorhanden.
3) Revision von konzeptuellem Modell und System-Architektur ist schwierig und kostspielig.
4) Architektur des Systems wird ohne abschließende Beurteilung der Expertise festgelegt.
5) Explizites Modell des Schlußfolgerungsprozesses des Experten fehlt.
6) Überforderung des Wissens-Engineers durch gleichzeitige Interpretation und Repräsentation.
7) Überblick, welche Teile der Expertise schon erhoben und implementiert sind fehlt.

4.2) Modellbasierte Ansätze

Modellbasierte Ansätze versuchen die Expertise auf einer höheren Abstraktionsstufe mit Termini darzustellen, die für den Experten verständlich sind, was zu einer besseren Kommunikation zwischen Wissensengineer und Experten führen soll. Die so erstellten expliziten Modelle entsprechen Repräsentationen auf dem Knowledge-Level im Sinne A. Newells, auf dem das dargestellt wird, was repräsentiert werden soll, unabhängig davon, wie der Formalismus der Repräsentation konkret aussehen soll. Dies wird auf dem Symbol-Level entschieden, der sich mit der Operationalisierungsphase deckt.
Unter dieser Perspektive besitzen explizite Modelle eine analoge Rolle wie das Interlingua-Konzept in der Machinellen Übersetzung.
Mit den expliziten Modellen werden folgende Ziele verfolgt:

1) Schnellere Einordnung neuer Konzepte, wenn sie in das vorhandene Modell passen.
2) Gezielte weitere Suche nach fehlenden Konzepten.
3) Leichteres Finden von Ursachen von Inkonsistenzen und Unvollständigkeiten in Wissens-Protokollen.

Es haben sich zwei Arten modellbasierter Ansätze herausgebildet:

1) Generic Tasks: Es wird davon ausgegangen, daß alle Expertenaufgaben durch eine kleine Anzahl elementarer Problemlösungsverfahren beschrieben werden können. Durch die Wahl eines dieser Verfahren wird gleichzeitig die Art und der Umfang der Repräsentation des benötigten Faktenwissens festgelegt, woraus ein standardisiertes und effizientes Vorgehen resultiert, das jedoch geringere Flexibilität besitzt.
2) Integrierte Modelle wie z.B. KADS: Es wird davon ausgegangen, daß sich die Expertise auf hierarchischen Ebenen modellieren läßt, wobei nur die untereste Ebene domänenspezifisch ist. Die oberen Ebenen enthalten vordefinierte Module und Bibliotheken für die Darstellung von Inferenzen, Aufgaben und Strategien, die ausgewählt, kombiniert und gegebenenfalls modifiziert werden können.

Die vier Analyseebenen von KADS lassen sich wie folgt darstellen:

1) Die Anwendungsebene entspricht einem Datenmodell im konventionellen Software-Engineerung, und enthält Begriffe der Domäne (Objekte, Attribute, Werte), Relationen (is_a, part_of, has, causes) und Strukturen als Verknüpfung von Begriffen und Relationen.
2) Die Inferenzebene entspricht einem Funktionsmodell und enthält Lösungsschritte (Knowledge-Source), Funktionen und die Darstellung des Datenflusses. Konzepte aus der Anwendungsebene werden dabei zu sog. Metaclasses abstrahiert, die die Rolle von Konzepten innerhalb des Problemlösungsprozesses spezifizieren.
3) Die Aufgabenebene entspricht einem Kommunikationsmodell, in dem Ziele und Teilaufgaben spezifiziert werden, indem Kombinationen der Inferenzen zu einer Problemlösungsstrategie gebildet werden.
4) Die Strategieebene enthält strategische Informationen (Pläne und Metaregeln) wann welches Faktenwissen oder Inferenzstrategie anzuwenden ist, und wie auf eine Sackgasse im Problemlösungsprozeß reagiert werden soll.

Vorteile:

1) Terminologie des Experten wird verwendet.
2) Transparenz und Dokumentierbarkeit des Interpretationsprozesses.
3) Trennung von Analyse und Implementierung ermöglicht Einsatz von jeweiligen Spezialisten.

4) Spezifikation für die Implementierung.
5) Explizite Darstellung der Expertisenstruktur.
6) Generische Modelle können die Interpretation leiten.
7) Vollständige Beschreibung der Expertise nach der Analysephase.
8) Änderungsaufwand eines Analysemodells geringer als implementierte Wissensbasis.

Nachteile:
1) Vorteile des Rapid-Prototyping entfallen.
2) Ablauf des Modells kann nicht direkt überprüft werden.
3) Formale Semantik steht den Modellen nicht zur Verfügung.

4.3) Bewertung der beiden Ansätze zur Wissensinterpretation

Tabelle 1) Vorteile (+) und Nachteile (-) des Rapid Prototyping und der Modellbasierten Ansätze

	Rapid Prototyping	Modellbasierter Ansatz
WR verwendet	Anforderungen der Implementierung (-)	Expertenterminologie (+)
Verwendung von Phasen/meilensteinen	nein (-)	ja (+)
Anleitung der Implementierung	durch Wissensprotokolle (-)	durch explizite Modelle aus den (+)
Änderungsaufwand	groß (-)	klein (+)
Gefahr des Verlustes des Überblicks	groß (-)	gering (+)
Gefahr der Überforderung	groß (-)	gering (+)
Verwendung des expliziten Schlußfolgemodells	nein (-)	ja (+)
Dokumentierbarkeit der Expertise	nein (-)	ja (+)
Verwendbarkeit von generischen Modellen	nein (-)	ja (+)
Wiederverwendbarkeit von Elementen	gering (-)	groß (+)
Eignung für große Projekte	nein (-)	ja (+)
Hinweise auf Inkonsistenzen, fehlende Teile, ...	durch Experten (+ -)	durch Modellanalyse (+ -)
Direkte Anwendbarkeit und Überprüfbarkeit	ja (+)	nein (-)

Als Ergebnis der Darstellung der Eigenschaften der beiden Ansätze zur Wissensinterpretation läßt sich sagen, daß der modellbasierte Ansatz methodisch dem Rapid Prototyping weit überlegen ist, wenn er auch für ein Einzelprojekt möglicherweise mehr Aufwand bedeutet. Werden mehrere verwandte Projekte geplant, und ist eine längerfristige Weiterentwicklung und Wartung vorgesehen, so besitzt der modellbasierte Ansatz durch seine Eigenschaften Wiederverwendbarkeit und Dokumentation eindeutige Vorteile, die von besonderer Bedeutung sind, wenn man berücksichtigt, daß bei DV-Projekten die Wartung der größte Kostenfaktor darstellt. Das Rapid Prototyping ist nur dann sinnvoll, wenn ein einmaliges Projekt schnell durchgeführt werden soll.
Beide Ansätze können prinzipiell auch kombiniert werden, indem während einer der Interationsphasen des evolutionären RP ein explizites Modell erstellt wird, oder wenn ein explizites nicht lauffähiges Modell durch ein RP-Tool operationalisiert wird.

5) Klassische Formen der symbolischen Wissensrepräsentation

Es werden im folgenden die drei am weitesten verbreiteten symbolischen WR-Arten dargestellt (Produktionsregeln (PR), Objektorientierte Ansätze, (klassische) Logik), die in XPS Anwendung finden.
Neben der Verbreitung sind die Beziehungen der Repräsentationsformen ein Grund für ihre Auswahl:
1) Aussagen innerhalb der Logik beziehen sich auf Fakten, die als Objekte und Attribute dargestellt werden.
2) Methoden von Objekten können immer als PR formuliert werden.
3) Im Bedingungsteil von PR werden meist Attributswerte von Objekten abgefragt, und im Aktionsteil können Attributswerte modifiziert werden.

Die Vielzahl anderer symbolischer Ansätze (z.B. Constraints, Nichtmonotones Schließen, Unsicheres Wissen (z.B. Bayes-Theorem), Vages Wissen (Fuzzy Logic), Temporales Schließen) sollen im diesem Rahmen ebenso unberücksichtigt bleiben wie subsymbolische Ansätze (Konnektionistische Repräsentationen).

5.1) Regelbasierte Ansätze

Regelbasierte Systeme bestehen aus 3 Komponenten:
1) Wissensbasis: Menge von Fakten, z.B. Aussagen der Logik oder Objekt-Attribut-Tupel.
2) Regelbasis: Menge von Produktionsregeln (PR), die Wissen über allgemeingültige Zusammenhänge im Anwendungsgebiet repräsentieren. PR bestehen aus einem Bedingungsteil (Wenn) und einem Aktionsteil (Dann). Eine PR kann angewendet werden (Ausführen der Aktionen im Aktionsteil), wenn der gesamte Bedingungsteil erfüllt ist, wodurch Fakten in der Wissensbasis modifiziert werden.
3) Regelinterpreter (Inferenzmaschine): Mechanismen, welche die Bedingungsteile prüfen und die Aktionsteile ausführen (Recognize-Act-Zyklus). Es werden auch Strategien zur Konfliktlösung bereitgestellt, wenn mehrere PR angewendet werden könnten, jedoch nur eine angewendet werden darf.

Vorteile:
1) Natürliche Wissensbeschreibung durch Expertenregeln.
2) Rapid Prototyping.
3) Zeitlicher Ablauf wird durch Regelinterpreter dynamisch bis ins Detail festgelegt.
4) Transparenz und Modifizierbarkeit der Regeln.
5) Unvollständiges Wissen repräsentierbar, da nicht für jede Situation eine Regel zu spezifizieren ist.
6) Inferenzen sind in Metaregeln formulierbar.
7) Erklärungen sind durch Regelfolgen darstellbar.

Nachteile:
1) Recognize-Act-Zyklus ist bei großen (ungeclusterten) Wissensbasen ineffizient.
2) Steuerfluß kann undurchsichtig werden.
3) Fehler im Systemverhalten sind schwer zu beheben.
4) Strukturierung von Wissens- und Regelbasis fehlt im Grundmodell.
5) Vermischung von Meta- und Anwendungs-Wissen durch Verstreuung von Regeln.
6) Hinzufügen neuer Regeln ist schwer einschätzbar.
7) Konsistenz unsicher.
8) Algorithmisches Wissen muß verschlüsselt und auf verschiedene Regeln aufgeteilt werden.

5.2) Objektbasierte Ansätze

Objekte sind Zusammenfassungen von Attributen und Methoden (Operationen auf Attributen des Objektes). Sie sind eine Ausprägung des Konzeptes der Frames, die aus einem Framenamen und Slots bestehen, wobei in den Slots Attribute, Methoden oder andere Frames als sogenannte Filler liegen können.
Die Bestandteile von Objekten sind für andere Objekte von Außen nicht direkt zugreifbar (Information Hiding), sodaß Objekte nur durch Nachrichten kommunizieren und somit Dienste austauschen können.
Objekte besitzen den Status von unterscheidbaren Entitäten (Identitätsprinzip) und können während der Laufzeit erzeugt oder vernichtet werden.
Objekte gleicher Art werden zu Objekt-Klassen zusammengefaßt, wodurch es zu einer Strukturierung einer Faktenbasis kommt. Objekte und Klassen können Beziehungen besitzen, die als Graph darstellbar sind. Eine wichtige Modellierungsmöglichkeit sind Klassen-Hierarchien, wobei is_a-Hierarchien (Generalisierungen) und part_of-Hierarchien (Aggregation) am meisten verwendet werden.
Is_a-Hierarchien ermöglichen Vererbungs-Hierarchien, bei denen individuelle Attribute bei Objekten abgespeichert werden, während allgemeine Eigenschaften den Vorgängern in der Hierarchie zugeordnet werden. Auf diese Weise erhält man eine ökonomische Strukturierung der vorhandenen Objekte.
Eine besondere Art der Vererbung ist der Polymorphismus, bei dem die gleiche Methode je nach Empfänger-Objekt unterschiedliche Wirkungen haben kann.
Zur Flexibilitätssteigerung werden Vererbungs-Heterarchien verwendet, bei denen ein Objekt Eigenschaften mehrerer Vorgänger erbt, und in denen selektive Unterdrückung der Vererbung einzelner Eigenschaften möglich sind.

Bekanntester und ältester OO-Ansatz sind Semantische Netze, die in sehr unterschiedlichen Realisierungen existieren. Gemeinsam haben alle Ansätze, daß Wissenselemente in Knoten und Kanten abgelegt werden. Die Knoten repräsentieren physikalische Objekte, Aussagen oder Situationen, die Kanten verknüpfen diese miteinander, wobei unterschiedliche Kanten-Arten eingeführt werden können (z.B. IS_A, PART_OF, A_KIND_OF). Es läßt sich zeigen, daß ein Semantisches Netz mit diesen drei Kanten-Arten äquivalent zu der Ausdrucksmächtigkeit der Prädikatenlogik ist (siehe Abschnitt 5.3).

Vorteile:
1) Transparenz, Anschaulichkeit und natürliche Wissensbeschreibung.
2) Zuverlässigkeit durch Datenabstraktion, Klassenkonzept, Vererbung, Polymorphie.
3) Erweiterbarkeit und Modifizierbarkeit durch Modularität.
4) Call-by-Desire = Nachricht drückt Wunsch des Senders aus und nicht seine Erwartungshaltung.
5) Synchronisation und Prozeßkommunikation gut modellierbar.

Nachteile:
1) Algorithmische Beschreibung komplexer Probleme schwierig.
2) Laufzeitnachteile durch dynamisches Binden und Nachrichtensenden.
3) Objektorientierte Steuerungsstrukturen können ungewohnt sein.

5.3) Logikbasierte Ansätze am Beispiel Prädikatenlogik

Eine wissenschaftliche Theorie verwendet (idealerweise) ein System von Sätzen (Axiome), die sie als wahr anerkennt, zu dessen Formulierung die Logik verwendet wird.
Es existieren verschiedene Logiken mit unterschiedlich mächtigen Ausdrucksmitteln. Die Aussagenlogik untersucht die Gültigkeit von Aussagen bei der Verwendung bestimmter Verknüpfungen und besitzt die schwächsten Ausdrucksmittel. Die Prädikatenlogik umfaßt die Ausdrucksmittel der Aussagenlogik, und erweitert diese, indem Aussagen über Individuen im mathematischem Sinn unter Verwendung von Prädikaten, Funktionssymbolen, logischen Operatoren und Quantoren zugelassen werden. Logiken höherer Ordnung erlauben es, Prädikate als Argumente ihrer selbst, als Variablen und Quantifizierungen zu verwenden.

Die syntaktischen Elemente der Prädikatenlogik lassen sich zusammenfassen zu:

1) Subjekte: sind Namen für Individuen im mathematischem Sinn, wie z.B. Personen, Begriffe oder physikalische Objekte. Sie können als Konstanten und Variablen auftreten.
2) Variablen: Stehen als Platzhalter für eine Menge von Individuen, und besitzen ansonsten keine selbständige Bedeutung. Sie werden bei der Formulierung allgemeingültiger Gesetze verwendet.
3) Konstanten: Sie sind auf ein Individuum beschränkt, und haben eine genau definierte Bedeutung.
4) Terme: Substitution für Subjekt bzw. Funktion von Subjekten [z.B. x im Prädikat p(x)].
5) Prädikat: Auf Terme angewendete Relationen, denen ein binärer Wahrheitswert zugeordent wird [z.B. groß_sein(x), teil_von(x, y)].
6) Quantoren: 6.1) Allquantor: „$\forall x$: p(x)“ ist wahr, wenn der Wert von p(x) für alle x wahr ist.
   6.2) Existenzquantor: „$\exists x$: p(x)“ ist wahr, wenn p(x) für mindestens ein x wahr ist.
7) Formeln: umfassen Prädikate, Sätze und Quantifizierungen [z.B.: p(x), $p(x) \wedge q(y)$, $\exists(x)$: p(x)]
8) Regeln: Zuordnung einer Aussage B (Konklusion) zu einer Folge von Aussagen A_i (Prämissen), die eine logische Konsequenz der Aussagen A_i darstellt.
9) Tautologien: Gesetze, die durch Regeln oder Wahrheitstafeln allein durch ihre Syntax unabhängig vom Wahrheitsgehalt der Einzelaussagen beweisbar sind

Folgende Bewertungen können für die logikbasierten Ansätze aufgeführt werden:

Vorteile:

1) Eindeutige Syntax und Semantik.
2) Korrekter, vollständiger Inferenzmechanismus.
3) Theorembeweiser als Inferenzmaschinen (Resolutionsbeweiser) verfügbar.
4) Hohe Flexibilität und Modularität.

Nachteile:

1) Inferenzmechanismus ist sehr zeitaufwendig und rein formal.
2) Heuristiken lassen sich schwer in die Inferenzmechanismen integrieren.
3) Vages und unsicheres Wissen ist nicht darstellbar.
4) Algorithmisches Wissen ist nur schwer repräsentierbar.
5) Ursprüngliche Problemstruktur geht bei Resolution verloren.
6) Beschränkung auf 2-wertige Logik.
7) Keine Abduktion und Induktion.

6) Aussicht und Schlußbemerkung

Die historische Entwicklung wissensbasierter Systeme zeigt, daß man Zyklen identifizieren kann, d.h. es gibt Phasen mit hoher und geringer Entwicklungstätigkeit. Unabhängig, ob eine solche Zyklenentwicklung auch weiterhin zu beobachten sein wird, kann man feststellen, daß zum Ende der 80´er Jahre eine Phase hoher Entwicklungstätigkeit zu beobachten war, währenddessen die Aktivität zur Mitte der 90´er Jahre relativ gesehen, abgeflaut ist. Dies ist auch dadurch zu erklären, daß seit dem Ende der 80´er Jahre die Forschungsaktivität im Bereich der Neuronalen Netze nach einer fast 20 jährigen Phase geringerer Intensität wieder eine Renaissance erlebte, die bis heute anhält.

Das Verhältnis zwischen symbolisch orientierten wissensbasierten Systemen und Neuronalen Netzen ist in der Vergangenheit als unversöhnlicher Kampf zweier Paradigmen oder Ideologien gesehen worden, die sich gegenseitig ausschließen. Meiner Ansicht wird sich diese Sichtweise zugunsten einer Menge von hybriden Systemen verändern, in denen auf unterschiedliche Weise beide Ansätze enthalten sind. So werden zunehmend Problembereiche identifiziert, in denen eine der beiden Ansätze die effizientere Lösung bietet. Beispielsweise sind Verfahren der heuristischen Assoziation mit symbolischen Mitteln zu lösen (z.B. innerhalb des Systems KSS0), doch auf Grund des erheblichen Aufwandes einer manuellen Wissensakquisition ist dieser Problemtyp besser mit Neuronalen Netzen zu lösen, wenn die teilweise harten Vorbedingungen für Verfahren des maschinellen (symbolischen) Lernens nicht erfüllt sind (siehe ID3 in Abschnitt 3.4). Eine Ausprägung hybrider Systeme, verfügt über symbolische und subsymbolische Problemlösungsmethoden, und wird (symbolisches) Wissen darüber erwerben, welche Probleme mit welchen Lösungsansätzen am effizientesten bezüglich eines vorgegebenen Grades an Effektivität zu bewältigen sind.

Die Trennung dieser beiden Wissensbereiche wird auch dadurch verwischt, daß große Forschungsanstrengungen unternommen werden, um Verfahren der Regelextraktion aus Netzen zu entwickeln. Umgekehrt kann durch die Erzeugung von Beispielmengen aus Regeln, mit denen ein Netz trainiert werden kann, ein Wissenstransfer in die entgegengesetzte Richtung durchgeführt werden.

Architekturen für IS, die komplexe Informationsdienstleistungen erbringen sollen, werden eine Fülle von kommunizierenden Einzelkomponenten besitzen, die spezielle Unteraufgaben jeweils effizient lösen können. Je komplexer die Dienstleistungen werden, desto wichtiger wird der Faktor Lernfähigkeit des Systems werden. Hierbei werden symbolische wie subsymbolische Komponenten ihre spezifischen Vorteile ausspielen und sich nebeneinander behaupten können.

7) Literatur

1) Altenkrüger, Doris; Büttner, Winfried: Wissensbasierte Systeme, Wiesbaden, 1992.
2) Buder, Marianne, et al.: Grundlagen der praktischen Information und Dokumentation, München, 1990.
3) Hennings, Ralf-Dirk: Expertensysteme als neue Zugangssysteme zur Fachinformation. In: Buder et al., S. 247-263.
4) Karbach, Werner; Linster, Marc: Wissensakquisition für Expertensysteme, München, 1990.
5) Panyr, Jiri: Automatische Klassifikation und Information Retrieval, Tübingen, 1986.
6) Puppe, Frank: Einführung in Expertensysteme, Berlin, 1991.
7) Zimmermann, Harald H.: Informationswissenschaft an der Universität des Saarlandes („Saarbrücker Modell“). In: Buder et al., S. 1100-1107.