

# Problemklassen in Expertensystemen

Einordnung und Diskussion des GWAI-90 – Workshops

Martin Bauer\*      Benno Stein†      Jürgen Weiner‡

FB 11 – Praktische Informatik  
Universität Duisburg  
Postfach 10 15 03  
D-4100 Duisburg 1

---

\*E-Mail: [hn279ba@unidui.uucp](mailto:hn279ba@unidui.uucp)

†E-Mail: [hn295st@unidui.uucp](mailto:hn295st@unidui.uucp)

‡E-Mail: [hn293we@unidui.uucp](mailto:hn293we@unidui.uucp)

## 1 Einleitung

Die Beschäftigung mit Problemklassen resultiert aus dem Wunsch von Anwendern und Entwicklern, eine Basis für den Austausch von Problemlösungswissen zu erhalten. Dieser Wunsch ist getragen von der Vorstellung, daß für eine neue Problemstellung bereits entwickelte Verfahren und Ansätze ganz oder teilweise wiederverwendet werden können. Zumindest verspricht man sich eine Aussage über die prinzipielle Lösbarkeit einer aktuellen Problemstellung. Beabsichtigt ist sowohl eine rückblickende Einteilung existierender Systeme auf der Basis der bearbeiteten Problemstellungen als auch eine Einteilung der jeweils verwendeten Methoden (Problemlösungsmethoden).

Eine solche Einteilung ist im Rahmen eines strukturierten Expertensystementwurfs sinnvoll und gewinnt zunehmend an Bedeutung. Hierfür liegen mittlerweile umfangreiche Erfahrungen aus der Entwicklung einzelner Systeme vor, auf die zurückgegriffen werden kann. Die Aktualität dieser Fragestellung zeigte sich auch in der Veranstaltung eines Workshops auf der GWAI-90 zu diesem Thema.

Ziel dieses Textes ist die Darstellung des Diskussionsstandes und eine Aufbereitung des Workshops. Zu diesem Zweck führt der folgende Text zunächst in die allgemeine Problematik der Problemklassen ein. Daran schließt sich eine Beschreibung des Workshopverlaufs und der erzielten Ergebnisse an. Der letzte Abschnitt, der als Anregung für weitergehende Diskussionen dienen soll, spiegelt in erster Linie Ideen und Interpretationen der Autoren wider.

## 2 „Klassische“ Ansätze

Im folgenden stellen wir bisherige Einteilungen vor, die auf den zentralen Begriffen der Problemklasse und der Problemlösungsmethode basieren. Dazu sollen diese Begriffe zunächst erklärt werden:

Problemklassen stellen ein Schema zur Einteilung von Problemstellungen dar. Sie dienen im Bereich der Expertensysteme der anwendungsbezogenen Einteilung von Aufgaben auf der Basis von Domänenkriterien. Problemlösungsmethoden sind strukturierte Vorgehensweisen, um eine abgrenzbare Aufgabenstellung zu bearbeiten und werden in Expertensystemen abgebildet. Hierbei ist entscheidend, daß diese Vorgehensweisen einen gewissen Grad der Allgemeingültigkeit bzw. Wiederverwendbarkeit besitzen.

### Die Einteilung von Hayes-Roth et al.

Die Einteilung von Hayes-Roth et al. [3] ist einer der ersten Versuche, eine Übersicht der existierenden Expertensysteme zu schaffen. Dabei wurden die Systeme aus einer rein anwendungsorientierten Sicht in einzelne Kategorien eingeteilt und gegeneinander abgegrenzt. In [3] wird die folgende flache Auflistung nebeneinander angeordneter Problemstellungen bzw. Anwendungsgebiete verwendet: Interpretation, Vorhersage, Diagnose, Design, Planung, Überwachung, Debugging, Reparatur, „Tutoring“ und Steuerung.

### Die Hierarchie von Clancey

Clancey kritisiert in [2] die Einteilung von Hayes-Roth et al. als zu unpräzise und mehrdeutig. Seinem Verbesserungsvorschlag liegt ein abstrakter Systembegriff zugrunde. Seine Einteilung basiert auf der Unterscheidung „... what we can do to or with a system“. Die Klassen von Hayes-Roth zugrundeliegend, schlägt er eine Strukturierung der Problemklassen in zwei Hierarchien vor, die in nachfolgender Abbildung dargestellt sind. Das Prinzip bei

dieser Zuordnung von Aufgabenstellungen zu den Hierarchien beschreibt er so: „We group operations in terms of those that construct a system and those that interpret a system, ...“.

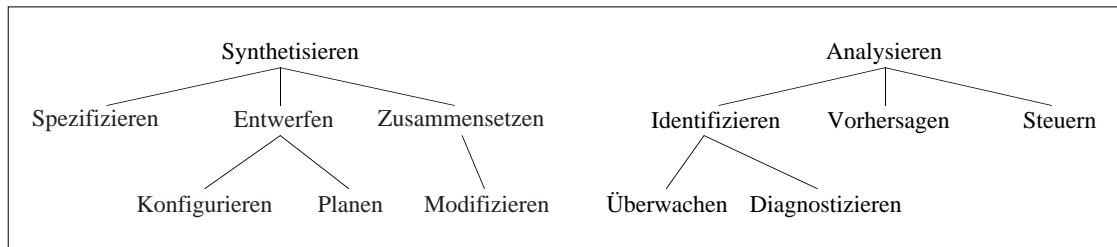


Abbildung 1: Problemklassenhierarchien von Clancey.

Neben der Einteilung in Problemklassen untersuchte Clancey die verwendeten Problemlösungsmethoden. Dabei identifizierte er die Methode der heuristischen Klassifikation als ein wiederverwendbares Verarbeitungsprinzip.

### Der Generic Task-Ansatz von Chandrasekaran

Chandrasekaran [1] befaßt sich mehr mit der Struktur und dem Aufbau von Problemlösungsmethoden als der Klassifizierung von Problemklassen. Dabei leitet er wiederverwendbare Elementaraufgaben (Generic Tasks) als Grundbausteine wissensbasierter Systeme ab. Ein Generic Task repräsentiert einen Baustein, bei dem spezialisierte Wissensrepräsentationen und Verarbeitungsmechanismen zur Durchführung einer bestimmten (Teil-)Aufgabe im Rahmen wissensbasierter Systeme kombiniert sind.

Bei der Identifikation der Generic Tasks läßt sich Chandrasekaran von der bisherigen Problemklasseneinteilung leiten, indem er für einige Kategorien Tasks bzw. Subtasks konzipiert. Ein „Nebeneffekt“ dieses Vorgehens ist die Analyse der Problemklassen aus Sicht der Entwickler. Daraus leitet er Anforderungen bezüglich der Kombination und des Einsatzes seiner Tasks als geeignete Problemlösungsmethoden ab.

Chandrasekaran erhebt mit seinen bisher identifizierten Tasks keinen Anspruch auf Vollständigkeit. In [1] werden folgende Generic Tasks beschrieben: Hierarchical Classification, Abductive Assembly, Structured Matching, Database Inference, Object Synthesis using Plan Selection and Refinement und State Abstraction.

### Die Idee der Role-Limiting-Methods von McDermott

McDermott [7] liefert dadurch einen Beitrag zu der Problemklassendiskussion, daß er sehr spezialisierte Problemlösungsmethoden (Role-Limiting-Methods) für bestimmte Aufgabenstellungen entwickelte. Sein Ziel ist nicht der Entwurf generischer Methoden, sondern die Entwicklung von Problemlösern, die den Anwender in einem hohen Maße unterstützen (Wissensakquisition, Erklärungsfähigkeit). Die entwickelten Methoden verlangen von dem Anwender ausschließlich die Eingabe von Domänenwissen; die notwendige Kontroll- und Steuerinformation ist fest vorgegeben. Zu den bekanntesten Systemen zählen MORE, MOLE, SALT und KNACK, die in [6] ausführlich beschrieben sind.

### Die Einteilung nach Puppe

Anstelle von Problemklassen unterscheidet Puppe in [9] Problemtypen und Problemlösungstypen. Unter einem Problemtyp versteht er eine Menge von Problemgebieten, die aus Anwendersicht ähnlich sind. Ein Problemlösungstyp dagegen beschreibt Problemgebiete, die mit denselben Problemlösungsmethoden gelöst werden können. Puppe beschäftigt sich in erster Linie mit der Identifikation und Zuordnung von Problemlösungsmethoden zu den von ihm vorgeschlagenen Hauptproblemlösungstypen Klassifikation, Konstruktion und Simulation.

### Die KADS-Methodologie von Wielinga, Breuker et al.

Die KADS-Methodologie [11] stellt eine allgemeine Methode zur strukturierten Wissensakquisition dar, die unabhängig von einem Implementierungsformalismus ist. Dabei beruht diese Methodologie besonders auf dem Prinzip: Wissensanalyse vor Wissensimplementierung. Es wird versucht, das Problemlösungswissen auf der Verständnisebene des Experten darzustellen, um so sein mentales, konzeptuelles Modell zu erfassen. Darüberhinaus kann durch eine Abstrahierung von dem domänenspezifischen Wissen versucht werden, aus einem konzeptuellen Modell wiederverwendbare Interpretationsmodelle zu gewinnen.

Solche Interpretationsmodelle können als abstrakte Beschreibungen von Problemlösungsmethoden aufgefaßt werden. Damit wird eine Zuordnung von anwendungsorientierten Problemklassen und geeigneten (abstrakten) Problemlösungsmethoden möglich. Die Operationalisierung der Interpretationsmodelle ist zur Zeit Gegenstand der Forschung.

## 3 Workshop

Auf der GWAI-90 wurde vom Fachausschuß „Expertensysteme“ ein Workshop initiiert, der sich mit Problemklassen in Expertensystemen beschäftigte. Hierfür hatten 22 Teilnehmer Positionspapiere eingereicht, aus denen ein Programm für den Verlauf des Workshops erstellt wurde.

Die wichtigsten Ergebnisse wurden dann in einer Abschlußdiskussion den Teilnehmern der GWAI-90 vorgetragen. Diese Diskussion sowie der positive Workshopverlauf zeigen, daß die Problemstellung von allgemeinem Interesse ist und diskutiert werden sollte.

Das auf Basis der eingereichten Papiere zusammengestellte Programm war in vier Blöcke eingeteilt, deren Ergebnisse wir im folgenden darstellen. Zusätzlich wurden Anmerkungen der sich anschließenden Mail-Diskussion von uns eingearbeitet. Wir unternehmen in diesem Abschnitt nicht den Versuch, eine globale Synthese zu erstellen, sondern lassen den Fragestellungen und Ergebnissen der einzelnen Sektionen Freiraum zur Interpretation.

### 1. Sektion: „Wozu sind Problemklassen da?“

In dieser Sektion sollte als Grundlage für die folgenden Sektionen auf die Frage nach dem Sinn der Einteilung in Problemklassen und nach Kriterien zur Bildung dieser Klassen Antworten gefunden werden. Dabei herrschte ein gewisser Konsens bei dem mit einer Problemklasseneinteilung verfolgten Ziel: Eine Einteilung soll Unterstützung in den Bereichen Werkzeugauswahl, Methodenauswahl, Implementierung und Wissensakquisition bieten. Hier stand für viele Teilnehmer eine Zuordnung von Problemklassen zu Problemlösungsmethoden im Vordergrund.

Als Kriterien zur Bildung bzw. Differenzierung solcher Problemklassen wurden dann verschiedene Ansätze diskutiert. Denkbar erschien eine Unterscheidung nach:

- Art der eingesetzten Problemlösungsmethode
- Eigenschaften der Lösung und des Suchraums
- Anwendungsdomänen wie z.B. Technik, Medizin und Prozeßmodellierung
- anwendungstypischen Problem- und Aufgabencharakteristika

Die Meinungen über eine Eignung dieser Kriterien für die Bildung von Problemklassen divergierten unter den Teilnehmern stark. Das lag vor allem daran, daß keine Einigkeit über die Zielgruppe einer Problemklasseneinteilung erreicht werden konnte.

## 2. Sektion: „Problemklassen“

Aufgabe der zweiten Sektion war die Untersuchung, ob die Differenzierung existierender Problemklassen ausreichend ist und inwieweit bestimmte (teilweise oben genannte) Kriterien dabei eine Rolle spielen. Zur Auswahl standen zwei Klassifizierungsansätze:

Der erste Ansatz unterscheidet zwischen analytischen Problemen, bei denen die Lösungsmenge explizit gegeben ist, und synthetischen Problemen, wo die Lösungsmenge konstruiert werden muß. Der zweite Ansatz stellt eine Systembeschreibung in den Mittelpunkt, die im analytischen Fall ergänzt, bei der Synthese konstruiert und bei transformativen Problemen geändert wird.

Als Arbeitsgrundlage wurde die Problemklasseneinteilung von Puppe [8], [9] verwendet, die sich stark an der Einteilung von Clancey [2] orientiert und dem ersten Ansatz entspricht. Es interessierten in diesem Zusammenhang vor allem die folgenden Fragen:

- i) Sind die Problemklassen Analyse und Synthese ausreichend?
- ii) Handelt es sich hierbei um Problemklassen – oder sind es eventuell Problemlösungsmethoden?

In der Diskussion wurde klar, daß eine eindeutige Zuordnung realer Aufgaben zu einer Problemklasse nicht immer möglich ist. Es existieren Aufgaben, die sowohl analytische als auch synthetische Aspekte vereinen. Die Zuordnung ist in solchen Fällen eine Frage der Interpretation: So ist eine Betrachtung der (analytischen) Diagnoseaufgabe als Synthese von Erklärungen zulässig (Die Symptome beschreiben die Anforderung an eine Lösung, während die Diagnosen diese Anforderungen erfüllen). Umgekehrt ist eine Betrachtung der Konfiguration als Diagnoseaufgabe möglich (Die Anforderungsspezifikationen entsprechen den Symptomen, die durch die Zuordnung entsprechender Komponenten als Diagnosen erklärt werden).

Generell zeigte die Diskussion, daß Kriterien zur Unterscheidung von Analyse und Synthese bestehen; man konnte sich jedoch nicht auf konkrete Charakteristika einigen. Durch diesen Mangel an Kriterien wurde eine weitere Aufgliederung von Aufgabenstellungen der Analyse und Synthese zum Scheitern verurteilt.

Die gleiche Problematik trat auch in dem sich anschließenden Streitgespräch: „Sind Planung und Konfigurierung dasselbe?“ offen zutage. Ein teilweiser Konsens ließ sich hier mit der Unterscheidung von Anwender- und Entwicklersicht erzielen: Ein Unterschied besteht auf jeden Fall aus Anwendersicht; vom Entwicklerstandpunkt aus betrachtet tritt diese Unterscheidung jedoch in den Hintergrund. Diese Diskussion wurde später via E-Mail weitergeführt. Dabei wurde erneut betont, daß gerade die Betrachtungsebene (Knowledge Level, Methodenebene, Implementierungslevel) für die unterschiedliche Beantwortung der Frage verantwortlich ist. Es wurde jedoch die Meinung vertreten, daß eine Unterscheidung von Planen und Konfigurieren schon allein bei der Wissensakquisition aus epistemologischer Sicht notwendig ist.

In der Diskussion, welche Problemklassen auf oberster Ebene ausreichend sind, wurde jedoch dafür plädiert, die – von anderen Autoren als eigenständige Problemklasse aufgefaßte – Simulation in die Gruppe der Problemlösungsmethoden einzureihen. Nach diesem anstrengenden Nachmittag widmeten wir (teilweise auch frustriert) unsere Aufmerksamkeit den Problemlösungsmethoden.

## 3. Sektion: „Problemlösungsmethoden“

Die Kernfragen dieser Sektion waren:

- i) Wie sollte man Problemlösungsmethoden beschreiben?
- ii) Welche Problemlösungsmethoden existieren bzw. sind also als solche identifizierbar?

Zur ersten Frage wurde die notwendige Granularität eines Beschreibungsvokabulars untersucht. Hier wurden drei Möglichkeiten diskutiert: Eine Black-box-artige, abstrakte Beschreibung, eine konzeptionelle Beschreibung auf dem Knowledge Level – ähnlich den Interpretationsmodellen der KADS-Methodologie – oder eine Beschreibung auf der Ebene operationalisierter Bausteine.

Favorisiert wurde eine Black-box-artige Darstellung von Problemlösungsmethoden, die folgende Spezifikationen beinhalten soll: Input- und Outputverhalten, Charakterisierung des anwendungsspezifischen Wissens und zusätzliche Eigenschaften wie z.B. Echtzeit- oder Erklärungsfähigkeit.

An dieser Stelle wurde deutlich, daß zwischen den hier genannten Begriffen zur Beschreibung von Problemlösungsmethoden und einem in Sektion 1 diskutierten Ansatz zur Charakterisierung von Problemklassen (Eigenschaften der Lösung und des Suchraums) große Ähnlichkeit besteht.

Die sich anschließende Frage, was Problemlösungsmethoden sind, führte auf die Unterscheidung in starke und schwache Methoden. Puppe definiert in [9] diese Begriffe in Anlehnung an McDermott [7]:

Eine schwache Problemlösungsmethode ist offen für eine Vielzahl bereichsspezifischer Wissensrepräsentationen und -funktionen und daher breit einsetzbar. Sie bietet aber nur wenig Hilfe beim Wissenserwerb. Eine starke Problemlösungsmethode legt die Repräsentation und Funktion des bereichsspezifischen Wissens weitgehend fest und ist daher nur begrenzt einsetzbar, bietet aber umfassende Hilfe beim Wissenserwerb.

In diesem Zusammenhang waren die Workshopteilnehmer einig, daß schwache Methoden wie z.B. Suchverfahren oder Deduktion eine zu niedrige Beschreibungsebene repräsentieren. Die „richtige Ebene“ wird wahrscheinlich durch stärkere Methoden wie z.B. Propose-and-Revise identifiziert. Dabei ist auch eine Beschreibung und Realisierung von Problemlösungsmethoden durch Komposition aus Modulen – vergleichbar den Generic Tasks – denkbar. Weiterhin bieten starke Problemlösungsmethoden die Möglichkeit, zu einer Shell ausgebaut zu werden.

#### 4. Sektion: Zuordnung

In dieser Sektion sollte versucht werden, Problemklassen und Problemlösungsmethoden einander zuzuordnen. Dieses Ziel erschien aufgrund der Ergebnisse der vorausgegangenen Sektionen nicht erreichbar – u.a. deshalb, weil man sich nicht auf eine Problemklasseneinteilung einigen konnte.

Als Ausweg aus dieser Situation bot sich folgendes Vorgehen an:

1. Außerachtlassen der Problemklassen
2. Aufstellen charakteristischer Merkmale für Problemlösungsmethoden
3. Charakterisierung einzelner Problemlösungsmethoden mit diesen Merkmalen
4. Beschreiben individueller Aufgabenstellungen mithilfe derselben Merkmale
5. Zuordnung von Problemlösungsmethoden zu einer Aufgabenstellung über diese Merkmale

Durch dieses Vorgehen wird nicht ausgeschlossen, eventuell später auch Problemklassen mithilfe dieser Merkmale zu identifizieren<sup>1</sup>.

---

<sup>1</sup>Anm. d. Autoren: Gemeint ist hiermit eine Identifikation der Klassen durch eine Abstraktion individueller Aufgaben unter Zuhilfenahme der Merkmale.

		Entscheidungs- bäume	Entscheidungs- tabellen	Vorschlägen & Verbessern	assoziative Konfigurierung	
Lösungsraum	Struktur	unstrukturiert	x	x		
		hierarchisch	(x)	(x)	x	
Definition v. Lösung	Größenordnung	kombinatorisch m. Abh.		x	x	
		kombinatorisch o. Abh.		x	x	
		Eindeutigkeit	klein	klein	groß	mittel
		Mehrdeutigkeit	x	x		
Art der Lösung	Korrektheitskriterium ex.	Korrekt	x	x	x	
		optimal			x	
		vollständig	x	x		x
		approximativ			x	
Wissensquellen	Art	kausal			x	
		assoziativ	x	x	x	
		strategisches Wissen ex.		x		
		kategorisch	x	x		x
Schlußweise	Kategorisch	unsicher				
		unvollständig			x	
		widersprüchlich				
		kategorisch	x	x	x	x
Input	unsicher	unvollständig				
		widersprüchlich				
		Realzeitfähigkeit			?	
		Erklärungsfähigkeit			evtl.	x
Sonderanforderungen	Lernfähigkeit					

Abbildung 2: Charakterisierung von Problemlösungsmethoden.

Die in Abbildung 2 angeführten Merkmale umfassen Attribute, die in erster Linie Aspekte aus Entwicklersicht repräsentieren. Als Ausgangspunkt für die Auswahl sinnvoller Attribute dienten hier Kriterien, die bereits in Sektion 3 zur Beschreibung von Problemlösungsmethoden vorgeschlagen wurden. Eine erste Überprüfung dieser Merkmale orientierte sich an den Problemlösungsmethoden: Entscheidungsbäume, Entscheidungstabellen, Vorschlägen-und-Verbessern sowie assoziative Konfigurierung. Eine Beschreibung der ersten drei Methoden befindet sich in [9], die Methode der assoziativen Konfigurierung wird in [4] vorgestellt. Die sich anschließende Tabelle charakterisiert diese Methoden anhand der ausgewählten Merkmale.

### Podiumsdiskussion

Die Ergebnisse des Workshops wurden auf einer anschließenden Podiumsdiskussion vorgestellt. Auch in dieser Diskussion wurde betont, daß die oben angeführten Merkmale noch weiter spezifiziert werden müssen. Dazu zählt u.a. ihre Hinterfragung bezüglich Vollständigkeit und Motivation. Als mindestens genauso wichtig wurde eine ausführliche Beschreibung weiterer Problemlösungsmethoden herausgestellt. Daher werden hiermit alle interessierten Leser aufgefordert, sich mit ihren Vorstellungen bzgl. einer ausführlicheren Menge von Merkmalen und der Beschreibung weiterer Methoden aktiv zu beteiligen.

Die zentrale Fragestellung, aus wessen Blickwinkel die Problemklasseneinteilung bzw. die Beschreibung von Problemlösungsmethoden erfolgen soll, wurde im Verlauf der Podiumsdiskussion erneut aufgeworfen. In diesem Zusammenhang spielen vor allem Aspekte aus der Anwendungs- und Implementierungssicht wie auch der Wissensakquisition eine Rolle.

Unbeantwortet blieb die Frage nach den Auswirkungen dieser Ergebnisse auf die Fachgebietsexperten. Eventuell besteht durch eine Problemklasseneinteilung die Gefahr, daß ein zu enges Raster für das Expertenvorgehen vorgegeben wird.

Von den Workshopteilnehmern wurde die Teilnahmebedingung, ein Positionspapier einzureichen, positiv bewertet. So war sichergestellt, daß die Teilnehmer mit der Thematik vertraut waren und sich ein sehr engagierter Workshopverlauf einstellte.

## 4 Diskussion und Ausblick

Eine wesentliche Motivation für die Veranstaltung des Workshops war der Wunsch, mithilfe einer Problemklassendiskussion Entwurfskriterien für Expertensysteme zu finden. Sicher gab es noch weitere Zielsetzungen und Vorstellungen einiger Teilnehmer, die jedoch im Workshopverlauf keine so entscheidende Rolle spielten.

### **Problemklassendiskussion – Ziele**

Die folgenden Überlegungen beziehen sich in erster Linie auf die Entwurfsprinzipien – genauer: auf die Unterstützung von Expertensystementwickler durch die Identifizierung und Charakterisierung von Problemlösungsmethoden für bestimmte Klassen von Problemstellungen. Steels bringt diesen Wunsch in [10] deutlich zum Ausdruck: „At some point, we should end up with a knowledge engineering handbook, similar to handbooks for other engineering fields, that relates task features with expert system solutions“.

Um diese Idee zu realisieren, wurde auf dem Workshop die Identifizierung von Problemmerkmalen (task features) als erster Schritt in Angriff genommen. Als nächster Schritt wurde mit diesen Merkmalen eine Charakterisierung von Problemlösungsmethoden versucht, um bei der Beschreibung einer konkreten Problemstellung eine Zuordnung von Problemlösungsmethoden zu diesem Problem zu ermöglichen. Ein weiteres Ziel – die Abstraktion konkreter Problemstellungen zu Problemklassen mithilfe obiger Merkmale – sollte danach ins Auge gefaßt werden.

Wir vertreten in diesem Zusammenhang die Auffassung, daß das Wissen um die Zugehörigkeit der betrachteten Problemstellung zu einer Problemklasse für die Auswahl von Problemlösungsmethoden weitgehend unerheblich ist. Vielmehr ist für den Entwickler eine Beschreibung der Problemstellung im Sinne derjenigen Merkmale von Nutzen, die zu der Auswahl der Problemlösungsmethode führen. Mit anderen Worten: Wichtiger als Merkmale für die Klassenbildung sind Merkmale, welche die konkrete Problemlösung (Realisierung) bestimmen.

Unserer Meinung nach erscheint eine Zuordnung von Problemstellungen zu Problemklassen eigentlich nur unter folgenden zwei Aspekten sinnvoll:

1. Einschränkung der Anzahl und Festlegung des Wertebereichs von zu untersuchenden Merkmalen.
2. Diskussion und Kommunikation (z.B. mit Anwendern) auf einer vom Entwicklervokabular losgelösten Verständnisebene.

Möchte man diese Aspekte realisieren, muß also doch eine Zuordnung im obigen Sinne stattfinden. Einleuchtend ist, daß die hierfür notwendigen Problemklassen durch Abstraktion von konkreten Problemstellungen gewonnen werden können. Der für die gewünschte Zuordnung



anfänglich als aussichtsreich empfundene Weg<sup>2</sup> scheint uns jedoch nicht gangbar. Ein Ausweg bietet sich durch die Verwendung unterschiedlicher Merkmalsmengen für die Identifikation von Problemlösungsmethoden bzw. Problemklassen an. Abbildung 3 veranschaulicht die geschilderten Zusammenhänge.

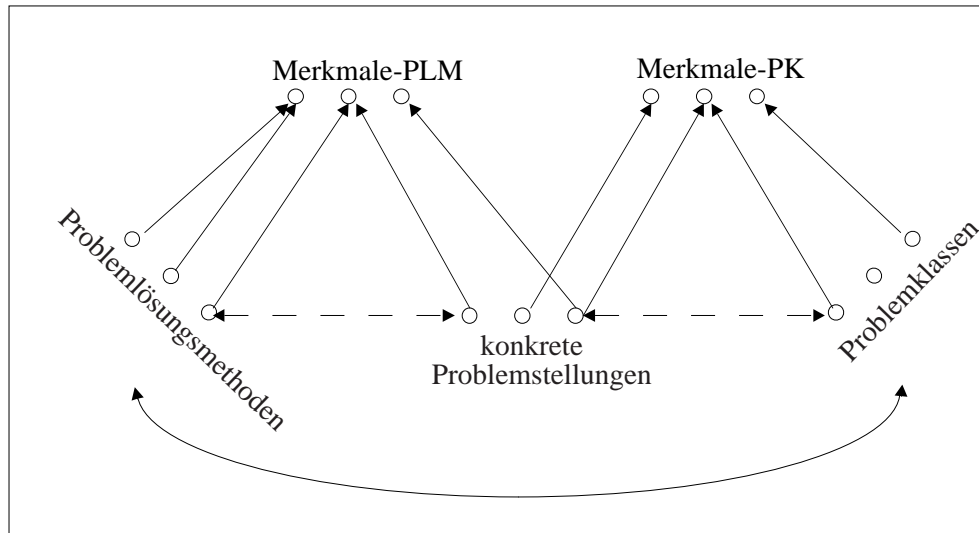


Abbildung 3: Zusammenhang zwischen Problemklassen und Problemlösungsmethoden.

Bei den Merkmalen-PLM aus Abbildung 3 handelt es sich um dieselben Merkmale, wie sie in Abbildung 2 vorgeschlagen wurde. Die Merkmale-PK dagegen enthält Attribute oder Situationsbeschreibungen, mit denen Problemklassen auf domänenspezifischer Ebene identifiziert werden. Die hierbei verwendeten Begriffe sind für die jeweiligen Anwender intuitiv bzw. mit geringer Erklärung verständlich und sind häufig Bestandteil der Umgangssprache. Typische Begriffe könnten z.B. sein: Fehlersuche, mechanisch, Beratungsaufgabe, entscheidungsunterstützend, prognostizierend.

Dieses Vorgehen halten wir für sinnvoll – leider führt es nicht unbedingt zu einer eindeutigen Zuordnung von Problemen. Der Grund hierfür liegt darin, daß die individuelle Gewichtung von Teilaspekten bei der Zuordnung von Problemstellungen eine Rolle spielen kann.

Als Einstieg für die Arbeit mit dem obigen Schema dienen konkrete Problemstellungen: Zur Auswahl geeigneter Problemlösungsmethoden wird die aktuelle Problemstellung durch die Merkmale-PLM, für die Einordnung in eine Problemklasse mithilfe der Merkmale-PK beschrieben. Analysiert man viele konkrete Problemstellungen, lassen sich eventuell problemübergreifende Zusammenhänge zwischen Problemlösungsmethoden und Problemklassen erkennen.

### Problemlösungsmethoden

Der Begriff der Problemlösungsmethode ist ein zentraler Begriff in der gesamten Problemklassendiskussion. Bei der Betrachtung der Methoden fällt auf, daß diese auf sehr unterschiedlichen Ebenen angesiedelt sind: Sie reichen von sehr allgemein gehaltenen Strategien bis hin zu genau spezifizierten Vorgehensweisen. Eine Einteilung der Methoden in Klassen gleichen Abstraktionsgrades ist sowohl für eine Diskussion als auch für die Methodencharakterisierung wünschenswert. Die in Ansätzen durchgeführte Unterscheidung in starke und schwache Methoden ist dafür nicht in allen Aspekten ausreichend.

Unserer Meinung nach lassen sich mindestens drei Kategorien von Methoden identifizieren:

<sup>2</sup>Zuordnung von Problemstellungen zu Problemklassen mithilfe der Merkmale, die auch zur Charakterisierung von Problemlösungsmethoden dienen.

Manche Methoden beschreiben eine Verarbeitungsphilosophie (Heuristische Klassifikation, Hypothesize and Test), andere stellen implementierungsnahe Grundtechniken dar (Frames mit Vererbung, Regeln mit Vorwärtsverkettung), wieder andere sind soweit spezifiziert, daß sie vom Anwender ausschließlich Domänenwissen verlangen (Role-Limiting-Methods). In diesem Zusammenhang interessieren auch operationalisierte Bausteine, aus denen Methoden zusammengesetzt werden können. Hier erhebt sich u.a. die Frage nach der Universalität des Generic Task-Ansatzes. Unabhängig von dieser Diskussion stellt die Identifikation, Beschreibung und Einordnung von Problemlösungsmethoden eine wichtige Aufgabe dar.

### Modellierungsaspekte

Ein Aspekt des Workshops war, durch eine Problemklassendiskussion Erkenntnisse für Entwurfsprinzipien von Expertensystemen zu gewinnen. Die hierbei diskutierte Methodenauswahl ist dabei nicht der einzige Aspekt; genauso wichtig ist der Aufbau eines adäquaten Domänenmodells. Beide Aspekte können nicht unabhängig voneinander betrachtet werden. Über die Wechselwirkung von Modell und Methode existieren zwei unterschiedliche Ansätze: Eine Seite favorisiert die Meinung, daß zuerst eine Methode ausgewählt werden muß, welche dann die zu modellierenden Domänenaspekte erzwingt. Gegensätzlich dazu ist der Ansatz, bei dem ein unabhängiges und möglichst vollständiges Modell der Domäne im Mittelpunkt steht, auf das einzelne Methoden dann zugreifen können. In diesem Zusammenhang sagt Steels [10]: „These two viewpoints are, of course, complementary and which approach is followed depends on what is available“. Wir halten die Einbettung eines Modellbegriffs in die Problemklassenproblematik für so sinnvoll, daß sie als Anhaltspunkt für weitere Diskussionen dienen kann.

## Literatur

- [1] B. Chandrasekaran: *Towards a Functional Architecture for Intelligence Based on Generic Information Processing Tasks*. Proc. IJCAI, S. 1183-1192, 1987.
- [2] W. Clancey: *Heuristic Classification*. Artificial Intelligence 27, S. 289-350, 1985.
- [3] F. Hayes-Roth, D. A. Waterman, D. B. Lenat: *Building Expert Systems*. Addison Wesley, 1983.
- [4] M. Hein, W. Tank: *Assoziative Konfigurierung*. in H. Marburger (ed.): Proc. GWAI-90, Springer Verlag, 1990.
- [5] A. Newell: *The knowledge level*. Artificial Intelligence 18, S. 87-127, 1982
- [6] S. Marcus: *Automating Knowledge Acquisition for Expert Systems*. Kluwer Academic Publishers, 1988.
- [7] J. McDermott: *Preliminary Steps Toward a Taxonomy of Problem-Solving Methods*. In [6]
- [8] F. Puppe: *Einführung in Expertensystemen*. Springer Verlag, 1988.
- [9] F. Puppe: *Problemlösungsmethoden in Expertensystemen*. Springer Verlag, 1990.
- [10] L. Steels: *Components of Expertise*. AI Magazin, S.29-49, Summer 1990.
- [11] B. J. Wielinga, J. A. Breuker: *Models of Expertise*. In B. DuBoulay, D. Hogg, L. Steels (eds.): *Advances in Artificial Intelligence - II (ECAI-86)*, North-Holland, 1987.