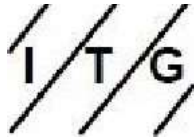


Einsatz wissensbasierter Systeme zur  
Modellierung und Darstellung von  
gartenbautechnischem Fachwissen am  
Beispiel des hybriden Expertensystems  
HORTEX

Thomas Rath

Dissertation, Universität Hannover, 1992



INSTITUT FÜR TECHNIK IN GARTENBAU  
UND LANDWIRTSCHAFT  
UNIVERSITÄT HANNOVER

Thomas Rath

Einsatz wissensbasierter Systeme zur Modellierung  
und Darstellung von gartenbautechnischem Fachwissen  
am Beispiel des hybriden Expertensystems HORTEX

Dissertation Universität Hannover

Gartenbautechnische Informationen

Heft 34

1992

# INHALTSVERZEICHNIS

Seite

Verzeichnis der Formelzeichen und Abkürzungen	
Verzeichnis der Indizes	
1 Einleitung .....	1
2 Grundlagen.....	3
2.1 Expertensysteme und wissensbasierte Systeme .....	3
2.1.1 Definition und Abgrenzung.....	3
2.1.2 Bestandteile .....	5
2.1.3 Projekte und Anwendungen .....	6
2.2 Computergestützte Modellierung und Darstellung von Fachwissen.....	9
2.2.1 Allgemeines.....	9
2.2.2 Einsatz wissensbasierter Systeme.....	11
2.2.2.1 Und-Oder-Graphen .....	11
2.2.2.1.1 Definition und Darstellung.....	11
2.2.2.1.2 Bearbeitung und Auswertung .....	13
2.2.2.1.3 Mehrwertige Und-Oder-Graphen.....	18
2.2.2.2 Wissensrepräsentationsformen.....	23
2.2.2.2.1 Allgemeines.....	23
2.2.2.2.2 Logik.....	23
2.2.2.2.3 Produktionsregeln.....	28
2.2.2.2.4 Frames .....	29
2.2.2.2.5 Prozeduren.....	31
2.2.2.2.6 Sonstige.....	32
2.2.2.3 Wissensakquisition.....	33
2.2.2.4 Werkzeuge.....	34
2.2.3 Einsatz arithmetischer Gleichungen.....	36
2.2.4 Einsatz von Datenbankkonzeptionen.....	38
2.2.5 Kombination verschiedener Ansätze.....	40
2.2.5.1 Allgemeines.....	40
2.2.5.2 Wissensbasierte Systeme und arithmetische Modelle.....	40
2.2.5.3 Wissensbasierte Systeme und Datenbanken.....	42
3 Das hybride System HORTEX.....	44
3.1 Allgemeines.....	44
3.1.1 Motivation, Zielsetzung, Anforderungen.....	44
3.1.2 Bisherige Arbeiten und einsetzbare Systeme .....	46

3.1.3 Wissensakquisition.....	48
3.2 Beschreibung des Systems.....	51
3.2.1 Modularisierung der Aufgabenstellung.....	51
3.2.2 Konzepte der Wissensrepräsentation.....	52
3.2.2.1 Ansatz orientiert an Produktionsregeln.....	52
3.2.2.2 Ansatz orientiert am Framekonzept.....	56
3.2.2.3 Datenbankorientierter Ansatz.....	60
3.2.3 Modellierung und Darstellung des Wissens .....	62
3.2.3.1 ... im Bereich Betriebsdateneingabe.....	62
3.2.3.2 ... im Bereich Wärmebedarfsrechnung.....	66
3.2.3.2.1 Klima.....	66
3.2.3.2.2 Wärmeverbrauch.....	68
3.2.3.2.3 Auslegungsleistung.....	77
3.2.3.3 ... im Bereich Planung der Heizenergiebereitstellung.....	78
3.2.3.3.1 Modularisierung.....	78
3.2.3.3.2 Beschreibung.....	78
3.2.3.3.3 Übersicht.....	91
3.2.3.4 ... im Bereich Planung der Heizenergieausbringung.....	92
3.2.3.4.1 Modularisierung.....	92
3.2.3.4.2 Beschreibung.....	92
3.2.3.4.3 Übersicht.....	103
3.3 Technische Realisierung.....	105
3.4 Anwendungsbeispiele.....	106
3.4.1 Wärmeverbrauchsrechnung.....	106
3.4.2 Planung der Heizenergiebereitstellung.....	108
3.4.3 Planung der Heizenergieausbringung.....	112
4 Diskussion.....	115
4.1 Modellierung und Darstellung von Fachwissen mit wissens- basierten Systemen.....	115
4.2 Darstellung von unscharfem Wissen.....	121
4.3 Möglichkeiten und Grenzen beschriebener Wissensrepräsen- tationsformen.....	123
4.4 Einsatz von Expertensystemen in der Gartenbautechnik.....	127
4.5 Bewertung des beschriebenen Systems.....	129
4.5.1 Validierung und Einsatzfähigkeit.....	129
4.5.2 Anwendung.....	134
4.6 Übertragbarkeit der Konzeption.....	137
4.7 Zukünftige Arbeiten.....	138
5 Zusammenfassung.....	140

6 Literatur.....	144
7 Eingesetzte Software.....	162
Anhang.....	163
A Regelinterpreter.....	163
B Objektinterpreter.....	169
C Implementierungsbeispiele.....	174
C.1 Wissensbasiertes Modell (Prolog) im Bereich Betriebsdaten- eingabe.....	174
C.2 Wissensbasiertes Modell (Produktionsregelansatz) im Bereich Heizenergiebereitstellung.....	180
C.3 Wissensbasiertes und arithmetisches Modell (Frameansatz) im Bereich Heizenergieausbringung.....	193
C.4 Arithmetisches Modell (Simulation) im Bereich Wärmever- brauchsrechnung.....	209
C.5 Datenbankorientiertes Modell (Standardvorgaben und Struktur) im Bereich Heizenergiebereitstellung.....	234

## Verzeichnis der Formelzeichen und Abkürzungen

<u>Zeichen</u>	<u>Bedeutung</u>	<u>Einheit</u>
A	Fläche	[m <sup>2</sup> ]
AI	Artificial Intelligence	
An	Anteil	[-]
Ant	Benutzerantwort	
b	Breite	[m]
Be	Bedingung zur Erfüllung einer Regel bei Benutzereingaben	
Bes	Beschreibung	
BF	Berücksichtigungsfaktor	[-]
BHKW	Blockheizkraftwerk	
BImSchV	Bundesimmissionsschutzverordnung	
CF	Certainty-Factor, Sicherheitsmaß	
D	Durchlässigkeit (Licht)	[-]
DISF	DICOTU Standard Interface Format	
Du	Durchmesser	[mm]
DW	Dachneigungswinkel	[Grad]
E, È	Knoten eines Graphen, Faktum, Beleg, Aussage, Ereignisse, Wert	
e	Wert eines Tupels	
EE	Energieeinsparung	[-]
Ei	Einheit	
EK	Kante eines Graphen	
EKO	Energiekosten	[DM/Jahr]
EP	Energiepreis	[DM/Bezugsgröße]
ES	Energieschirm	
EZ	Zielknoten einer Kante eines Graphen	
Fa	Facette (Beschreibung der Art eines Slots)	
Fr	Frage	
f	Funktion	
G	In einem Modell angesprochene Zustandsgeneratoren	
GIPAR	Umwandlungsfaktor Globalstrahlung in PAR	[W(PAR)/W]
H	Hypothese	
h	Höhe	[m]
HF	Hüllfläche (innen, außen)	
HW	Heizwert	[kWh/Liefereinheit]
In	Inhalt eines Slots	
ITG	Institut für Technik in Gartenbau und Landwirtschaft der Universität Hannover	
IV	Investitionsvolumen	[DM]
k'	Wärmeverbrauchsbeiwert	[W/(m <sup>2</sup> .K)]

<u>Zeichen</u>	<u>Bedeutung</u>	<u>Einheit</u>
KF	Korrekturfaktor, Korrekturglied	(im Text)
KI	'Künstliche Intelligenz'	
KS	Anzahl Knotenschichten im Ableitungsbaum	
l	Länge	[ m ]
Li	Lichtangebot	[ Wh(PAR)/m <sup>2</sup> ]
Lö	Lösung	
LWK	Landwirtschaftskammer	
Mat	Material	
max	Maximum	
min	Minimum	
N-WAB	Normwärmeabgabe (90 °C Vorl., 70 °C Rückl., $\vartheta_i = 20 \text{ °C}$ )	[ W/Einheit ]
P, $\dot{P}$	Leistung	[ W ], [ kW ]
PAR	Photosynthetisch aktive Strahlung	
Pf	Suchpfad	
Pr	Prädikat	
$\dot{Q}$	Wärmestrom	[ W ], [ kW ]
Q	Wärmemenge	[ Wh ], [ kWh/Jahr ]
$\dot{q}$	Wärmestromdichte	[ W/m <sup>2</sup> ]
R	Relation	
r	Tupel der Relation R	
SA	Strahlungsausbeute (Kunstlicht)	[ W(PAR)/W ]
Sl	Slot (Eigenschaft, Variable)	
SP	Schaltpunkt	[ W(PAR)/m <sup>2</sup> ]
SW	Anzahl möglicher Suchwege im Ableitungsbaum	
T	Teilungsverhältnis Heizflächen im Dachraum/Kulturraum	[ - ]
t	Zeit	[ h ]
TA	Technische Anleitung	
TRY	Testreferenzjahr	
v	Geschwindigkeit	[ m/s ]
VF	Verzweigungsfaktor der Knotenschicht	
W	Wahrscheinlichkeit	
WF1,2	Wichtungsfaktoren für die Aufteilung der Heizlasten	[ - ]
x,X	Variable	
Z	Modellzustand	
ZEB	Zusatzenergiebedarf	[ % ]
$\eta$	Wirkungsgrad	[ - ]
$\mu$	Zugehörigkeitsfunktion	
$\vartheta$	Temperatur	[ °C ]
$\Delta$	Differenz	
$\Sigma$	Summe	
$\neg$	Negation (nicht ...)	
$\wedge$	Konjunktion (... und ...)	
$\vee$	Disjunktion (... oder ...) (einschließendes Oder)	
$\dot{\vee}$	Alternative (entweder ... oder ...) (ausschließendes Oder)	

<u>Zeichen</u>	<u>Bedeutung</u>	<u>Einheit</u>
$\rightarrow$	Subjunktion (wenn ... dann...)	
$\langle \rightarrow$	Bijunktion (... genau dann, wenn ...)	
$\Rightarrow$	Implikation (aus ... folgt logisch .... )	
$\langle \Rightarrow$	logische Äquivalenz	
$\forall$	Allquantor (für alle ...)	
$\exists$	Existenzquantor (es gibt ein ...)	
$\varepsilon$	Element	
$\sim$	unscharfe Menge	
$\cap$	Durchschnitt	
$\cup$	Vereinigung	
$\times$	Kreuzprodukt	
$\subseteq$	Teilmenge	

### **Verzeichnis der Indizes**

<u>Zeichen</u>	<u>Bedeutung</u>
a	außen, Außenflächen
AB	Bereitstellung (absolut)
Akt	aktuelles, zu berechnendes Gewächshaus
an	angrenzend
Ausl	Auslegungsfall
BA	Berechnungsabschnitt
Ch	Charakteristika
d	Jahrestag
Da	Dach, Dachraum
DB	Datenbank
EA	Energieträgeranschluß
EB	Energiebereitstellung
EL	elektrische Energie
ES	Energieschirm
F	Fläche
Ges	Gesamt
GH	Gewächshaus
Gi	Giebel
GS	Globalstrahlung
ht	hoch temperiert
i	Laufindex
i	innen
IW	Innenwand
JB	Jahresbetrieb
j	Laufindex
K	Kulturbereich
k	Anzahl möglicher Hypothesen
k	Anzahl eingesetzter Energieträger



Zeichen Bedeutung

Kap	Gewächshauskappe
KL	Kunstlicht
L	Liste
L	Sollwert Lüftung
LB	Bereitstellung (leistungsbezogen)
LE	Liefereinheit
Li	Link, Verbindung
l	Anzahl verwendeter Altanlagen
m	Laufindex
m	aktuelle Simulationsstunde
m	Anzahl
max	Maximum
min	Minimum
n	Laufindex
n	Anzahl
n	Stunde des Jahres
Na	Nachbargewächshaus
nt	niedrig temperiert
O	Objekt
oH	ohne Heizung (unbeheizt)
pot	potentielle
r	Laufindex
r	rechnerisch
S	Sollwert innen
s	Anzahl Zusatzeinrichtungen
Seg	Gewächshaussegment, -abteilung
SG	Stehwand-Giebelwandbereich
Si	Simulationsschritt
Sp	Tag-Nacht-Speicherwirkung
St	Standard
Ste	Stehwand
Sum	Sollwert Summenregelung
Te	Text
TF	Teilfläche
th	theoretisch
Tr	Trapez
Um	Umrüstung
W	Wind
WA	Wärmeausbringung
z	Laufindex
ZS	Zusatzeinrichtung

## 1 Einleitung

Der Gartenbau heutiger Zeit ist geprägt durch die Herausforderungen gesättigter Märkte, durch erschwerte Produktionsbedingungen infolge des gewachsenen Umweltbewußtseins der Bevölkerung und durch eine zunehmende Technisierung der Produktion. Der gartenbauliche Betriebsleiter steht bei vielen seiner Entscheidungen im Spannungsfeld zwischen Kultur, Technik, Ökonomie und Ökologie. Gezielte Informationen und das Wissen über Handlungsalternativen gewinnen bei gärtnerischen Entscheidungen zunehmend an Bedeutung. Hierbei kommt es nicht darauf an, möglichst viele Informationen zu gewinnen, sondern entscheidend ist, gezielte und auf das konkrete Problem abgestimmte Erkenntnisse zu erhalten.

Seit einigen Jahren wird zur Unterstützung eines gezielten Wissenstransfers der Einsatz von sogenannten Expertensystemen propagiert. In Verbindung mit leistungsfähigen Rechenanlagen soll Expertenwissen situationsspezifisch einem breiten Nutzerkreis zugänglich gemacht werden. Durch den rapiden Preisverfall im EDV-Bereich - oder besser gesagt, durch die steigenden Leistungskapazitäten gewöhnlicher Personal-Computer - können solche Systeme heute auch im Agrarbereich eingesetzt werden. GÖHLER et al. (1989) beschreiben das mögliche Anwendungsspektrum im Gartenbau, welches von dem Bereich des Pflanzenschutzes über die Düngung und Bewässerung, die Anbauberatung, die Gartenbautechnik bis hin zur Ökonomie reicht.

Die Entwicklung solcher Expertensysteme ist eng mit dem jeweiligen Fachgebiet, für das es konzipiert wird, verbunden. Gleichzeitig spielt aber auch die Computerwissenschaft, speziell die Forschung auf dem Gebiet der "künstlichen Intelligenz" (KI), eine wichtige Rolle, da das vorhandene Wissen in eine codierte und vom Rechenwerk des Computers interpretierbare Form gebracht werden muß.

Ziel der vorliegenden Arbeit ist es, die Umsetzung von Wissen in codierte Information näher zu beleuchten und wissenschaftlich zu hinterfragen. Im einzelnen soll die Frage geklärt werden, inwieweit Methoden zur Entwicklung von Expertensystemen zur Modellierung und Darstellung von gartenbautechnischem Fachwissen eingesetzt werden können, inwieweit sie angepaßt werden müssen, oder inwieweit sie zumindest eine Erweiterung konventioneller Ansätze erlauben. Schwerpunkt der Betrachtungen ist dabei die Wissensrepräsentation. Da die Gartenbauwissenschaft jedoch eine angewandte Wissenschaft ist, kann dieser Methodenaspekt nicht ohne konkreten Bezug zum Anwendungsfall aus dem Fachgebiet existieren. Speziell im Bereich der Expertensysteme ist dieser

Anwendungsbezug enorm wichtig, da die von Seiten der Informatik entwickelten Ansätze vielfach nur mit trivialen Anwendungsbeispielen hinterlegt werden, die auch nicht aussagekräftiger werden, indem man sie ständig wiederholt.

Die vorliegende Arbeit stellt von daher einen Versuch dar, sowohl methodische als auch inhaltliche, gartenbautechnische Fragen der Entwicklung von Expertensystemen zu bearbeiten. Daß die Ausführungen dabei zum Teil sehr stark in den Bereich der Informatik hineinreichen, läßt sich aufgrund der engen Verflechtung zwischen Wissensrepräsentation und Programmierung nicht vermeiden. Denn nur durch die wechselseitige Betrachtung ist dem wissenschaftlichen Anspruch des Themas gerecht zu werden. Eine Trennung von Computer und Wissen würde für den Bereich der Expertensysteme eine Abkehr von den gesteckten Zielen bedeuten.

Die Arbeit setzt sich aus drei Teilen zusammen. Im ersten Teil wird auf die für die weiteren Betrachtungen innerhalb der Arbeit notwendigen Grundlagen eingegangen. Besonders großer Wert wird dabei auf die Darstellung der mathematischen Hintergründe von wichtigen, häufig als Schlagwörter eingesetzten Begriffen wie Inferenz, Logik, Unschärfe, Vorwärtsverkettung, Rückwärtsverkettung etc. gelegt.

Im zweiten Teil der Arbeit wird die inhaltliche Übertragung und Anpassung der Expertensystemtechnologie anhand eines konkreten Beispiels aus dem Bereich der Gartenbautechnik dargestellt. Dabei wird versucht, alle wichtigen Systemelemente möglichst genau zu beschreiben. Denn gerade im Detail zeigen viele Expertensysteme erst ihre wahre Struktur.

Da neben der Beschreibung von Einzelheiten auch der Gesamtaufbau dargestellt wird, müßte es dem Leser gelingen, eine Vorstellung von der Wirkungsweise und den methodischen Ansätzen des Systems zu erhalten. Das vollständige Verständnis setzt allerdings eine gewisse Kenntnis der Programmiersprache Prolog voraus. Hier kann nur auf die zitierte Literatur verwiesen werden, da nähere Ausführungen dem Rahmen der Arbeit nicht angemessen wären.

Im abschließenden Kapitel werden die eingesetzten Methoden und die Anwendung des erstellten Systems diskutiert. Auf eine Diskussion der in dem hybriden System enthaltenen konventionellen Wissensdarstellung wird aufgrund des thematischen Schwerpunktes der Arbeit verzichtet.

## 2 Grundlagen

### 2.1 Expertensysteme und wissensbasierte Systeme

#### 2.1.1 Definition und Abgrenzung

Eine Definition und Abgrenzung von Expertensystemen ist schwierig, da unterschiedliche Vorstellungen und Assoziationen aus den verschiedensten Blickwinkeln beim Gebrauch des Begriffes Expertensystem vorliegen. Im folgenden wird dennoch versucht, ein Begriffsverständnis zu entwickeln.

Feigenbaum, einer der führenden Wissenschaftler auf dem Gebiet der Expertensysteme, beschreibt ein Expertensystem als "... ein intelligentes Computerprogramm, das Wissen und Inferenzverfahren benutzt, um Probleme zu lösen, die immerhin so schwierig sind, daß ihre Lösung ein beträchtliches menschliches Fachwissen erfordert. Das auf diesem Niveau benötigte Wissen in Verbindung mit den verwendeten Inferenzverfahren kann als Modell für das Expertenwissen der versiertesten Praktiker des jeweiligen Fachgebietes angesehen werden" (HARMON und KING 1989, S. 3).

Der Duden 'Informatik' (ENGESSER 1988, S. 222) schreibt über Expertensysteme: "...Programmsystem, das 'Wissen' über ein spezielles Gebiet speichert und ansammelt, aus dem Wissen Schlußfolgerungen zieht und zu konkreten Problemen des Gebietes Lösungen anbietet".

SAVORY (1988, S. 7), ein Expertensystemspezialist der Nixdorf AG, bezeichnet Expertensysteme als "...rechnerunterstützte Entscheidungshilfsmittel, die mit Hilfe der Methoden der Künstlichen Intelligenz die fachliche Kompetenz von Experten (in Form von Sach- und Erfahrungswissen) speichern, um diese Kompetenzen dann vervielfacht und dezentralisiert ihrem Benutzer zur Verfügung zu stellen".

Die Liste der Definitionen ließe sich weiter fortsetzen, wobei die genannten Beispiele schon zwei grundlegende Aspekte verdeutlichen. Im allgemeinen Verständnis haben Expertensysteme einen qualitativen Charakter (sie beinhalten Wissen) und wenden zur Darstellung und Auswertung dieses Wissens bestimmte Methoden an.

Eine Abgrenzung von Expertensystemen gegenüber herkömmlichen Computerprogrammen setzt demnach eine Spezifikation des Wissensbegriffes und des Methodenaspektes voraus.

Zum ersten Punkt ist anzumerken, daß in der gesamten KI keine konsensfähige und anerkannte Definition des Wissensbegriffes existiert (BREWKA 1986). Da

letztendlich jedes Computerprogramm Wissen in irgendeiner Form darstellt, ist folglich eine Abgrenzung aufgrund dieses Aspektes nur schwer möglich (vgl. FURBACH 1988). STOYAN (1988) arbeitet in einem lesenswerten Artikel dieses Problem deutlich heraus. Er kommt zu dem Schluß, daß die Expertensysteme von heute nichts wissen, sondern lediglich modellbasierte Programme darstellen. Das Vorhandensein von fachspezifischem Wissen ist demnach für ein Expertensystem notwendig, aber nicht ausreichend, um das Programm von anderen Systemen abzugrenzen (vgl. RATH 1991a).

Eine Unterscheidung zwischen Expertensystemen und konventionellen Programmen aufgrund unterschiedlicher Wissensdarstellungsmethoden ist schon eher gegeben. Expertensysteme präsentieren Wissen in Form von logischen Aussagen und Verknüpfungen. Konventionelle Systeme speichern Wissen in der Regel überwiegend als eine Folge prozeduraler Anweisungen ab. Darüberhinaus ist ein wesentliches Unterscheidungsmerkmal, daß Expertensysteme im Gegensatz zu 'herkömmlichen' Systemen auf einer strikten Trennung zwischen dem gespeicherten Wissen (Wissensbasis) und einem Wissensverarbeitungsmechanismus aufbauen (vgl. GRAHAM und JONES 1988, S.19 und die drei Definitionen zu Beginn des Kapitels).

Neben Expertensystemen wird vielfach auch von wissensbasierten Systemen gesprochen. Die Zuordnung beider Begriffe zueinander ist innerhalb der Literatur unterschiedlich. NEUMANN (1986) bezeichnet beispielsweise Expertensysteme als eine besondere Form der wissensbasierten Systeme. HARMON und KING (1989, S. 4f) schreiben, daß frühe Systeme in der Regel als Expertensysteme bezeichnet werden, heute aber von wissensbasierten Systemen oder Wissenssystemen gesprochen wird. Sie selber bezeichnen kleine Systeme als Wissenssysteme und große Systeme als Expertensysteme. Zusammenfassend läßt sich sagen, daß im überwiegenden Teil der Literatur Expertensysteme und wissensbasierte Systeme als synonyme Bezeichnungen benutzt werden, wobei der Ausdruck wissensbasiert mehr im Kontext einer Systemeigenschaft Verwendung findet und der Begriff Expertensystem eine Systembezeichnung darstellt (vgl. hierzu die Aufsätze zum Computerkongreß "Wissensbasierte Systeme in der Landwirtschaft" (DEUTSCHE LANDWIRTSCHAFTS-GESELLSCHAFT 1988)).

Ausgehend von den obigen Ausführungen und in Anlehnung an TANIMOTO (1990, S. 523) wird für die vorliegende Arbeit folgendes Begriffsverständnis zugrunde gelegt:

Ein Expertensystem ist ein Computerprogramm, in dem bestimmte Formen der Wissensmodellierung und -repräsentation eingesetzt werden. Im wesentlichen lassen sich diese Formen auf den dominierenden Einsatz logischer Aussagen und

Verknüpfungen sowie auf die Trennung zwischen abgespeichertem Expertenwissen und einem Abarbeitungsmechanismus zurückführen. Die Begriffe "Wissensbasiertes System" und "Expertensystem" werden als äquivalent betrachtet. Eine wertende Abgrenzung (in Bezug auf das implementierte Wissen) zwischen Expertensystemen und konventionellen Computerprogrammen wird per Definition nicht gegeben.

### 2.1.2 Bestandteile

Abbildung 2.1-1 gibt die wesentlichen Bestandteile eines Expertensystems wieder.

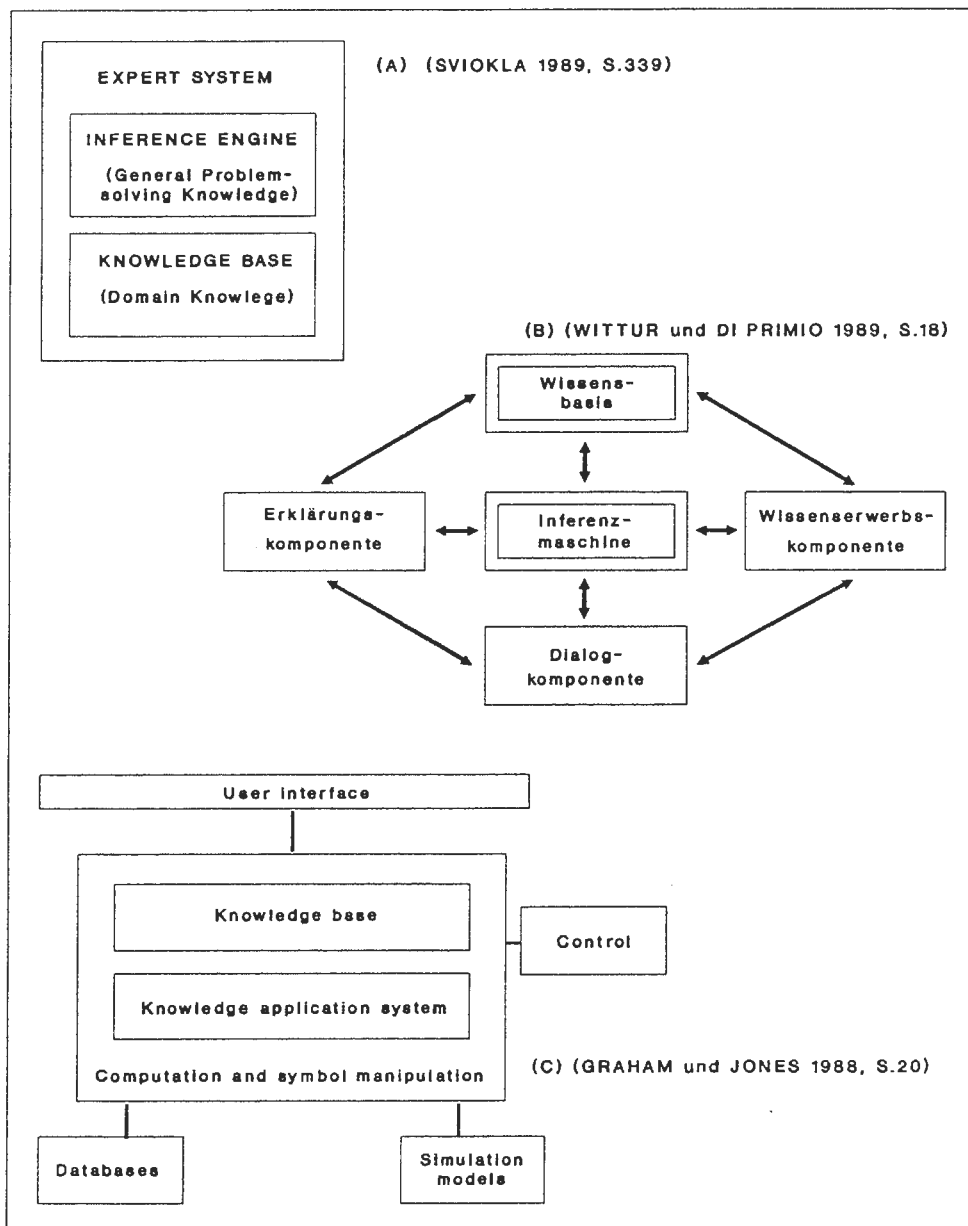


Abbildung 2.1-1: Bestandteile von Expertensystemen nach verschiedenen Autoren

Je nach Sichtweise und Systemansatz ergeben sich unterschiedliche Systembausteine. Der Kern des Systems liegt in der Wissensbasis und dem Auswertesystem (A). In einem einsetzbaren System kommt in der Regel noch die

- Erklärungskomponente
- Wissenserwerbskomponente und die
- Dialogkomponente hinzu (B). Erweiterte Systeme beziehen auch noch Simulationsanwendungen und Datenbanken mit ein (C). Die am häufigsten anzutreffende Komponentenbeschreibung entspricht dem Fall B.

### 2.1.3 Projekte und Anwendungen

MERTENS (1987) gibt einen umfassenden Überblick über den Einsatz von Expertensystemen in betrieblichen Anwendungen. Von den 694 in einer Datenbank der Universität Erlangen-Nürnberg erfaßten Anwendungen kann der überwiegende Teil dem industriellen Bereich zugeordnet werden (591 Expertensysteme). Wesentliche Einsatzbereiche sind Produktion, Forschung, Entwicklung und Verwaltung in den Aufgabenklassen Diagnose, Konfiguration, Beratung und Planung. Die Anzahl laufender Systeme (wirkliche Anwendungen) im deutschsprachigen Raum wird mit 28 angegeben. RIMEK (1991) bestätigt die Angaben von Mertens. Er schreibt, daß es sich bei den in einer Datenbank der Universität Dortmund gespeicherten 1664 Expertensystemen (Stand: Juli 1989) zum großen Teil um Prototypen aus der Industrie handelt.

Ähnlich sieht es auch auf dem Gebiet der Heizungs- und Klimatechnik aus. Aufgeführte Systeme scheinen noch nicht über den Prototypstatus hinausgekommen zu sein. BAER und DIEMER (1990) beschreiben ein Projekt, bei dem ein Expertensystem zur Vorplanung von Heizungsanlagen in der Gebäudetechnik entwickelt wurde. HÄLKER et al. (1986) stellen Möglichkeiten dar, die im Bereich der Heiztechnik wichtige Umweltgesetzgebung (in diesem Fall die 4. BImSchV) in ein wissensbasiertes Regelwerk umzusetzen. CAMEJO und HITTLE (1989) beschreiben einen sehr interessanten Expertensystemprototyp zur Planung von Heizungs-, Lüftungs- und Klimaanlage von Gebäuden. Das System ist streng in einzelne Wissensmodule aufgeteilt, die folgende Aufgaben bearbeiten: 1. Grobe Vorplanung der Anlage, 2. Überprüfung aller planungsrelevanten Voraussetzungen, 3. technische Auslegung der Anlage und 4. Auswahl einzelner Bauteile aus Firmendatenbanken. HABERL und CLARIDGE (1987) berichten über einen Expertensystemprototyp zur Diagnose und Darstellung von Energieeinsparmöglichkeiten in klimatechnischen Anlagen. CASE et al. (1990) beschreiben ein

klimatechnisches System, bei dem unterschiedliche Wissensrepräsentationsformen mit einer Blackboard-Architektur verknüpft werden. Weitere Systeme, die inhaltlich stark in den Bereich der Heizungsregelungstechnik hineinragen, sind bei ANDERSON et al. (1989) und bei KALER (1990) zu finden.

Im Bereich der Agrarwirtschaft ergibt sich folgendes Bild: Einer Studie von LINK (1991) zufolge, die 81 Expertensystemprojekte (25 deutsche) erfaßt, werden lediglich drei Projekte in alleiniger privatwirtschaftlicher Regie durchgeführt. Die übrigen Systeme werden bzw. wurden überwiegend an Hochschulen oder anderen Forschungseinrichtungen entwickelt. Die Systeme werden den Bereichen Pflanzenproduktion (52 %), gesamtbetriebliche Leitung (22 %), Tierproduktion (17 %) und Agrartechnik (7 %) zugeordnet. Meistens befinden sie sich im Prototyp- oder Projektionsstadium. Nach Link wird bei keinem deutschen Expertensystem von einem Routineeinsatz berichtet.

Im Bereich der Gartenbautechnik wurde und wird an sehr unterschiedlichen Expertensystemen gearbeitet. Der thematische Schwerpunkt liegt eindeutig im Bereich der Bewässerungstechnik. Mit Ausnahme des im folgenden zuerst genannten Systems handelt es sich um ausländische Entwicklungen.

RATH und VON ELSNER (1989) beschreiben ein System, mit dessen Hilfe Drehstrahlregnerbewässerungsanlagen konzipiert werden können. Im wesentlichen werden technische Algorithmen mit Benutzereingaben, ökonomischen Kennzahlen und Simulationsberechnungen gekoppelt. GRANIER (1991) stellt ein ähnliches System vor, welches jedoch um wissensbasierte Auswahlregeln erweitert wurde. Ein Expertensystem von BELCHER und MERVA (1988) behandelt speziell die Auslegung von Unterflurbewässerungssystemen. BRADLEY et al. (1988) und HART et al. (1989) entwickelten allgemein gehaltene Entscheidungs- und Auswahlssysteme für Freiland-Bewässerungsanlagen. THOMSON und PEART (1986) zeigen die Möglichkeiten auf, mit einem Expertensystemansatz die Bewässerungssteuerung/Regelung zu unterstützen. FREELAND und HOWARD (1990) erstellten ein sehr spezielles System, welches die Auswahl von Sprühdüsen bei der Pflanzenschutzmittelausbringung unterstützt.

Auf die prinzipielle Anwendung von wissensbasierten Techniken im Bereich der Klimaregelung von Gewächshäusern wiesen schon HOSHI und KOZAI (1984) hin. KOZAI (1985) konkretisierte diese Ansätze und zeigte Wege auf, wie diese in eine Klimaregelung integriert werden können. JACOBSON et al. (1989) stellen in einem praktischen Versuch dar, wie die Klimaregelung durch ein Expertensy-



stem mit Faustregeln unterstützt werden kann.

Den Bereich des innerbetrieblichen Transportes in einem Gewächshaus bearbeiteten GIACOMELLI et al. (1987). TULUCA et al. (1989) unterstützen mit einem Expertensystem die Auswahl von Versuchsgewächshäusern und deren Inneneinrichtungen.

Weitere Systeme sind nicht direkt dem Gartenbau zuzuordnen. Ein Diagnosesystem unterstützt die Fehlersuche in hydraulischen Systemen bei Traktoren (GAULTNEY et al. 1987). Den Expertensystemeinsatz bei der Konfigurierung und Auslegung spezieller hydraulischer Systeme beschreiben LI et al. (1990). KLINE et al. (1988) stellen ein System vor, welches durch die Kombination von linearer Optimierung, Simulationsberechnung und Expertensystem das Maschinenmanagement eines Betriebes verbessern kann. NEGAHBAN et al. (1988) entwickelten ein System zur Unterstützung der Entscheidungsfindung bei der Auswahl eines geeigneten Schleppers. Andere Systeme finden sich in den Bereichen Schlepperwartung (KAVKA et al. 1988) und Getreidetransport, -trocknung und -lagerung (PEART 1989, WATSON und BROOK 1990, LOEWER et al. 1990). Frühere Systeme ab 1970 sind bei SCHNEIDER (1987) verzeichnet.

## 2.2 Computergestützte Modellierung und Darstellung von Fachwissen

### 2.2.1 Allgemeines

Der erste Schritt bei der Entwicklung eines fachspezifischen Softwaresystems besteht in der Formulierung einer Abbildung der zu repräsentierenden realen Welt (SOMMERVILLE 1990, S. 65). Im kybernetischen Sinn muß ein System analysiert und in ein abstraktes Modell überführt werden. Modelle sind dabei nach NIEMEYER (1977, S. 57) materielle oder immaterielle (geistige, formale) Systeme, die andere Systeme so darstellen, daß eine experimentelle Manipulation der abgebildeten Strukturen und Zustände möglich ist. Die wesentlichen Unterscheidungsmerkmale zwischen Modell und realem System sind daher notwendigerweise Vereinfachung und Abstraktion (vgl. auch LENTZ (1987); SCHMIDT (1985); NIEMEYER (1977); ROTHENBERG (1989)).

Es ist üblich, die so entstehenden Modelle in unterschiedliche Klassen und Hierarchien einzuordnen. Obwohl in der Literatur die Begriffsdefinitionen und -verwendungen zum Teil recht unterschiedlich sind, soll im folgenden ein kurzer Überblick über die gebräuchlichsten Modelltypen gegeben werden. Die Einteilungsklassen sind im wesentlichen den Arbeiten von FUTO und GERGELY (1990, S. 29ff) und NIEMEYER (1977, S. 58ff) entnommen.

Gruppiert man Modelle nach der Art ihres Abbildungsmediums, so können materielle, verbale, formale, graphische, mathematische, logische, arithmetische und numerische Modelle voneinander unterschieden werden. Unterteilt man Modelle nach Art ihrer Zeitrepräsentation, werden diskrete, kontinuierliche, statische und dynamische Modelle unterschieden. Betrachtet man bei der Modelleinteilung besonders die Abhängigkeiten der Modellvariablen, so werden die Modellattribute stochastisch und deterministisch verwendet. Bei besonderer Beachtung der Modellentstehungsgeschichte finden die Attribute experimentell oder theoretisch Verwendung. Der Darstellungslevel und die Sichtweise des Modellbauers und des Modellanwenders kommen besonders in den Modelleinteilungen Mikro-, Makro-, Black-Box-, White-Box- oder Metamodell zum Ausdruck. Eine Einteilung nach Verwendungszweck unterscheidet häufig Steuerungs-, Erklärungs-, Prognose-, Planungs-, Simulations- oder Entscheidungsmodelle.

Selbstverständlich kann ein Modell zu mehreren Typenklassen gehören, genauso wie die Übergänge von einem Modelltyp zu einem anderen fast immer fließend und vom persönlichen Blickwinkel des Betrachters abhängig sind.

Die Konstruktion eines abstrakten Modells umfaßt nach SCHMIDT (1985, S. 17ff) folgende vier Schritte:

1. Abgrenzung gegen die Umwelt:

Das offene reale System wird durch ein geschlossenes System ersetzt. Für Einflüsse, die von außen in das theoretisch geschlossene System hineinragen, müssen Ersatzdarstellungen gefunden werden.

2. Bestimmung der Modellobjekte und Attribute:

Es wird bestimmt, welche Objekte im Modell vorkommen sollen und welche Eigenschaften und Attribute sie haben.

3. Definition der Modellstruktur:

Es muß festgelegt werden, in welcher Weise die einzelnen Modellobjekte miteinander in Verbindung stehen sollen und wie sie sich dabei gegenseitig beeinflussen.

4. Validierung des Systems:

Da das abstrakte Modell immer weniger Einflußgrößen enthält als das reale System, aus dem es hervorgegangen ist, wird es grundsätzlich nicht möglich sein, eine vollständige Übereinstimmung zwischen Modell und realem System zu erzielen. Eine Überprüfung der Gültigkeit des Modells ist daher wichtig. Möglichkeiten sind die Verifikation an Einzelbeispielen und die Falsifikation durch Negativfälle. Eine weitergehende Analyse der einzelnen Modellparameter bietet die Sensitivitätsanalyse (vgl. auch MCLEOD 1982, S. 116ff).

Die Übertragung eines Wissensmodells in ein Computersystem kann mit unterschiedlichen Ansätzen erfolgen. SOMMERVILLE (1990, S. 6ff) unterscheidet drei zur Zeit praxisrelevante Verfahren. Am häufigsten wird das sogenannte 'Wasserfallmodell' der Softwareentwicklung eingesetzt. Im Idealfall folgt nach einer Bedarfsanalyse und Anforderungsspezifikation die Modellierung des Systems und anschließend die Implementierung einzelner Programmeinheiten und deren Überprüfung. In einem vierten Schritt wird das erforderliche System zusammengestellt, getestet, korrigiert und anschließend für den Einsatz freigegeben. Neuere Ansätze versuchen diesen Prozeß zu verkürzen, indem sie die explorative Softwareentwicklung einsetzen. Hierbei wird auf einen genauen Anforderungskatalog vor Projektbeginn verzichtet. Das entwickelte Softwaresystem wird relativ schnell zum Einsatz freigegeben und solange verbessert, bis eine ausreichende Funktionalität erreicht ist. Besonders im Bereich der KI wird vielfach nach diesem Prinzip gearbeitet. Einen ähnlichen Ansatz stellt das sogenannte 'rapid prototyping' dar. Das Ziel dieses Verfahrens ist jedoch nicht die Erstel-

lung eines hochwertigen Programms, sondern es soll durch die rasche Entwicklung eines Prototypen die prinzipielle Machbarkeit eines Projektes überprüft werden. Gleichzeitig erfolgt die Spezifikation der Systemanforderungen für das neu zu implementierende Endsystem.

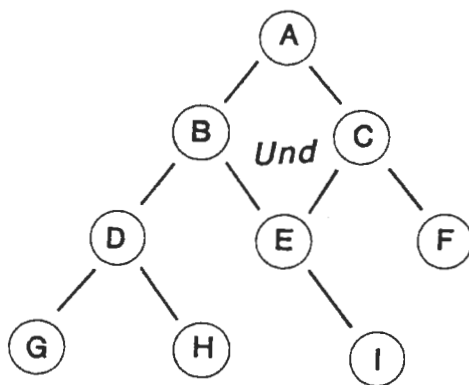
## 2.2.2 Einsatz wissensbasierter Systeme

### 2.2.2.1 Und-Oder-Graphen

#### 2.2.2.1.1 Definition und Darstellung

Bei der Entwicklung von Wissensmodellen hat sich im Bereich der "Künstlichen Intelligenz" die Darstellung von Such- bzw. Problemräumen mit Und-Oder-Graphen (auch Inferenz-, Ableitungs- oder Objektbaum genannt) durchgesetzt (SAVORY 1988, S. 52; ALTENKRÜGER 1987, S. 20). Ein Graph besteht dabei aus einer Menge  $E_i \{i = 1, 2, \dots n\}$  Knoten und einer Menge  $EK_z \{z = 1, 2, \dots m\}$  Kanten, wobei jede Kante durch ein ungeordnetes Paar von Knoten, die Elemente der Menge  $E_i$  sind, beschrieben werden kann (z.B.  $EK_1 = (E_1, E_2)$ ). Werden die Kanten mit einer Verknüpfungsrichtung (z.B. Pfeilspitzen) versehen, bezeichnet man den Graphen als Digraph (STRUNZ 1977, S. 202).

### Und-Oder-Graph



### Entscheidungstabelle

Bedingung								
A	X	X						
B			X	X				
C			X					X
D					X	X		
E							X	
Aktion								
B	X							
C		X						
D				X				
E			X					
F								X
G					X			
H						X		
I							X	

Abbildung 2.2-1: Beispiel eines Und-Oder-Graphen mit Entscheidungstabelle

Abbildung 2.2-1 zeigt ein Beispiel eines Und-Oder-Graphen mit den wichtigsten Verzweigungsarten:

- Und-Verzweigung
- Oder-Verzweigung

Die dargestellten Knoten und Kanten können sowohl ein Entscheidungsregelmodell repräsentieren, genauso könnte der abgebildete Graph aber auch hierarchische Strukturen (Klassen, Unterklassen) der Objekte A bis I modellieren. Durch die weitere Vernetzung der Knoten untereinander (z.B. durch Kanten von H nach I oder von I nach F) erhält der beschriebene Graph eine komplexere Struktur. Werden dabei unterschiedliche Kantentypen (Aggregation, Klassifikation) eingesetzt, bezeichnet man solche Modelle häufig als semantische Netze (ALTENKRÜGER 1987, S. 15; BÖCK 1991).

Es ist anzumerken, daß das Verfahren der Ergebnisdeduktion durch Aufbau eines Ableitungsbaumes in keiner Weise eine neuzeitliche Entwicklung der KI ist. Vielmehr wurde schon in römischen Schulen das Prinzip des "Baum des Prophyrius" gelehrt (SCHNUPP und NGUYEN HUU 1987, S. 63).

Und-Oder-Graphen lassen sich mit dem Instrumentarium der Graphentheorie in unterschiedliche Matrixformen überführen (und umgekehrt). Wichtige Transformationsmatrizen sind die Nachbarmatrix (bei Digraphen als Präzedenzmatrix bezeichnet), mit der beschrieben wird, ob zwei Knoten durch eine Kante verbunden sind, und die Erreichbarkeitsmatrix, die beschreibt, ob ein Knoten von einem anderen Knoten durch eine Kantenfolge erreichbar ist (zur näheren Beschreibung siehe STRUNZ 1977, S. 201ff; HARARY 1974, S. 208ff).

Eine häufig als Darstellungsmittel benutzte Matrix ist die Entscheidungstabelle, in der Aktionen bestimmten Bedingungen zugeordnet werden. Der Übersichtlichkeit halber kann eine Entscheidungstabelle in mehrere Tabellen untergliedert werden, die dann einen Entscheidungstabellenverband darstellen. Der rechte Teil der Abbildung 2.2-1 gibt den dargestellten Und-Oder-Graph als Entscheidungstabelle wieder.

Die Beschreibung von Ableitungsbäumen in der Informatik geschieht durch den Einsatz von verknüpften Programmsequenzen. Verkettet man mehrere gleichartige Objekte zu einem neuen Objekt, so spricht man von Listen. Lineare Listen sind eine besonders einfach strukturierte Datenform, bei der jedes Listenelement genau einen Nachfolger hat (bis auf das letzte). Erweitert man eine Liste in der Art, daß jedes Element beliebig viele Nachfolger hat, entsteht eine Baumstruktur. Besteht der Baum aus Knoten, die jeweils genau zwei Nachfolger

haben, spricht man von Binärbäumen. Die Darstellung von Bäumen kann durch rekursive Datentypen erfolgen, so daß jeder Teilbaum eines Baumes wieder einen eigenen Baum darstellt. Will man das Konzept der Baumdarstellung auch auf Graphen erweitern, bei denen zwischen den Teilbäumen eines Knotens Kanten existieren, gibt es die Möglichkeit, die sich ergebende Präzedenzmatrix zu speichern oder aber das Konzept der Teilbäume auf Graphen mit Knoten und Kanten zu erweitern (LIPSCHUTZ 1987, S. 123ff, ALTENKRÜGER 1987, S. 42ff).

Obwohl heute bei der Beschreibung von Zustandsräumen innerhalb der Expertensystementwicklung fast immer auf "höhere" Programmiersprachen oder Entwicklungsumgebungen ausgewichen wird und somit eine maschinennähere Darstellung der Baumstrukturen nicht notwendig ist, bleibt festzuhalten, daß letztlich Ableitungs- oder Objektbäume aus gleich strukturierten verketteten Objekten bzw. Listen bestehen (vgl. WINSTON 1987, S. 177ff) und somit ein konventionelles Programmierkonzept darstellen. TANIMOTO (1990, S. 79f) schreibt dazu: "Wir sehen also, daß Produktionssysteme an sich nichts Besonderes darstellen, aber Produktionssysteme können bei geeignetem Einsatz genauso wie die Techniken des 'strukturierten Programmierens' die Strukturierung großer Systeme unterstützen". (zu Produktionssystemen siehe Kapitel 2.2.2.2.3)

#### **2.2.2.1.2 Bearbeitung und Auswertung**

Um aus einem Und-Oder-Graphen Informationen zu gewinnen, wird der Graph mit Hilfe von prozeduralen Algorithmen durchsucht. Suchen kann dabei verstanden werden als das Durchlaufen des gerichteten Graphen (RICH 1988, S. 59). Auf die Beschreibung der Suche als 'Ableiten im Kalkül' (vgl. HERTZBERG 1986, S. 6) wird in Kapitel 2.2.2.2.2 näher eingegangen.

Bei der Suche können drei beteiligte Prozesse unterschieden werden:

1. Das Verfolgen einer Suchrichtung unter Einbeziehung von Rückziehverfahren.
2. Die Überprüfung, ob ein Knoten bei der Suche als Suchschritt zur Verfügung steht.
3. Die Auswahl des nächsten Knotens, wenn mehrere zur Verfügung stehen.

ad 1.: Ein Und-Oder-Graph kann in verschiedene Richtungen durchsucht, verkettet oder auch analysiert werden. Zum einen wird versucht, eine Lösung rückwärts durch Bestätigung oder Widerlegung von notwendigen oder hinreichenden Bedingungen herzuleiten. Die zweite Möglichkeit besteht darin, durch

gegebene Bedingungen neue Knoten zu erreichen (vorwärts), bis ein Zielknoten gefunden ist oder keine weiteren Suchschritte mehr möglich sind. Vielfach werden auch beide Suchrichtungen miteinander kombiniert oder es findet eine bidirektionale Suche statt, d.h. gleichzeitiges Suchen in beide Richtungen, bis die Suchfronten aufeinandertreffen.

Obwohl letztendlich formal kein Unterschied besteht, in welche Richtung gesucht wird, können aufgrund inhaltlicher und topologischer Aspekte des Suchraumes unterschiedliche Suchrichtungen sinnvoll sein. Nach RICH (1988, S. 62) gibt es drei wichtige Kriterien, die diese Tatsache unterstreichen und Handlungsempfehlungen notwendig machen:

- Unterschiedliche Anzahl an Startzuständen gegenüber Zielzuständen. Empfehlung: von der kleinen zur größeren Zustandsmenge bewegen.
- Unterschiedliche Verzweigungshäufigkeiten. Empfehlung: vom niedrigen zum hohen Verzweigungsfaktor bewegen.
- Notwendigkeit der Programmtransparenz. Empfehlung: Suchrichtung wählen, die dem Gedankenmodell des Benutzers am nächsten kommt.

Wird die Suche in aussichtslosen Situationen abgebrochen und werden schrittweise alternative Zielwege ausprobiert, bezeichnet man diese Verfahren als Backtracking (CLAUS und SCHWILL 1989, S. 66ff). WIRTH (1975, S. 191) sieht in ihnen eine allgemeine Problemlösung, bei der Algorithmen bestimmt werden, die zum Finden von Lösungen 'Versuch und Nachprüfen' verwenden. Backtracking stellt somit eine wichtige, aus der konventionellen Programmieretechnik kommende Erweiterung der Expertensystemtechnologie dar. In der Regel wird ein chronologisches Rückziehen durchgeführt. Jeder Pfad, der beim Durchlaufen des Suchraumes nicht eingeschlagen wird, wird auf einem Stack zwischengespeichert. Durch rückwärtiges Abarbeiten dieses Stacks können bei Bedarf Alternativpfade ausprobiert werden. Nicht chronologisches Rückziehen (abhängigkeitsgesteuertes Rückziehen) beschreibt WINSTON (1987, S. 100ff).

ad 2.: Dieser Teilschritt der Suche in einem Ableitungsbaum wird in der Literatur allgemein als 'matching' oder auch 'pattern-matching' bezeichnet. Dabei gilt es, herauszufinden, ob ein Objekt die von einem Muster geforderte Form hat oder aus den von einem Muster vorgegebenen Elementen besteht (TANIMOTO 1990, S. 75). Die Übersetzer von Tanimotos Buch schreiben in einer Fußnote, wie unglücklich der allgemein gebräuchliche Ausdruck Mustervergleich ist, da beim Matching nicht nur Vergleiche sondern auch Wertzuweisungen an Variablen stattfinden. Letztendlich besteht somit das Matching aus einer Art Stringvergleich mit gekoppelter Variablenzuweisung.

Häufig stellen eingegebene, direkt gespeicherte oder hergeleitete Fakten die Muster und die in der Wissensbasis abgelegten Regelprämissen, Regelkonklusionen oder sonstigen Objekte die Vergleichsausdrücke dar.

Je mehr Informationen über Objekte vorliegen, desto eher wird man dabei in der Lage sein, Lösungen einzig und allein durch Matching zu identifizieren. Liegen jedoch nur wenige Informationen vor, wird Matching nur selten zum Erfolg führen. Vielmehr müssen Lösungsansätze geschlußfolgert oder heuristisch gesucht werden (vgl. HANUS 1989, S. 43).

Der Einsatz von Matching-Verfahren in Expertensystemen ist heute breit gefächert. Er reicht von der Unifikation im Bereich der Prädikatenlogik bis hin zur Datenbankabfrage. Wird ein Produktionsregelsystem sehr stark auf Matching-Prinzipien abgestimmt, bezeichnet man es als 'discrimination net' (TANIMOTO 1990, S. 80). Ein häufig in Matching-Systemen eingesetzter Algorithmus ist der RETE-Algorithmus (FORGY und SHEPARD 1987). Er ermöglicht ein schnelles und präzises Matching, indem er alte Vergleichsmuster abspeichert und in neuen Situationen benutzt. Die Beschreibung einer vereinfachten RETE-Version in Prolog findet sich bei MERRITT (1989, S. 138ff).

ad 3.: Stehen in einem Graphen mehrere Suchpfade zur Verfügung, müssen Verfahren angewendet werden, die Pfadentscheidungen ermöglichen. Die in der Expertensystemtechnologie eingesetzten Suchstrategien weisen z.T. große Ähnlichkeit mit einigen im Bereich des Operation Research eingesetzten Optimierungsverfahren (vgl. LENTZ 1987, S. 110ff) auf. Dieses ist nicht verwunderlich, da sowohl Ableitungsbäume als auch die numerischen Modelle des Operation Research Zustandsräume beschreiben, in denen auf möglichst effektive Art und Weise 'gute Lösungen' gefunden werden sollen.

Die Einteilung der verschiedenen Suchstrategien im Bereich der KI wird unterschiedlich vorgenommen. KAINDL (1989, S. 52) schlägt eine Unterteilung der Strategien danach vor, ob unwiderrufliche Pfadentscheidungen oder zurücknehmbare Pfadentscheidungen getroffen werden (Irrevocable- bzw. Tentative-Strategien). WINSTON (1987, S. 106) unterteilt die Suchstrategien in die Suche nach "irgendeinem Pfad" und die Suche nach dem "optimalen Pfad". TANIMOTO (1990, S. 182, S. 188) unterscheidet "elementare Suchtechniken" und "heuristische Suchmethoden".

Im folgenden werden daher die wichtigsten Suchverfahren ohne eine nähere Einteilung vorgestellt. Die Ausführungen entstammen im wesentlichen den Arbeiten von KAINDL (1989).



Tiefensuche (depth-first search): Bei der Tiefensuche wird ausgehend vom Anfangsknoten unmittelbar versucht, möglichst tief in den Suchbaum vorzudringen, um so eine Lösung zu finden. Um bei Fehlversuchen nicht zu scheitern, werden Backtrackingmechanismen eingesetzt.

Breitensuche (breadth-first search): Bei der Breitensuche wird der Ableitungsbaum in seiner gesamten Breite nach einer Lösung durchsucht (das Verfahren schreitet in Schichten gleicher Tiefe voran). Die gefundene Lösung stellt eine Lösung mit minimaler (Ableitungs-)Länge dar.

Erzeugen und Prüfen (generate and test): Bei diesem Verfahren wird der gesamte Suchbaum durchwandert, um alle möglichen Lösungen zu erzeugen. Die Suche erfolgt mit Tiefensuche oder Breitensuche, die jedoch nicht endet, wenn eine Lösung gefunden worden ist. Da sich die Anzahl möglicher Suchwege aus der Beziehung  $SW = VF^{KS}$  mit  $SW =$  Anzahl möglicher Suchwege,  $VF =$  Verzweigungsfaktor,  $KS =$  Anzahl der Knotenschichten (Baumtiefe) ergibt, sind mit diesem Verfahren nur kleine und überschaubare Suchprobleme zu lösen.

Bergsteigen (hill-climbing): An jeder Verzweigungsstelle des Baumes werden sämtliche Nachfolgeknoten mit einer Bewertungsfunktion bewertet. Es wird der Pfad ausgewertet, der den besten Wert ergibt. Vorkehrungen zur Rücknahme der Entscheidungen werden nicht getroffen. Dieses Verfahren kann bei ungünstiger Topologie des Suchraumes erhebliche Probleme aufwerfen (lokale Maxima, lokale Minima, Plateaus etc.). Im Extremfall wird keine Lösung gefunden, auch wenn der Lösungsraum welche anbietet (Sackgasse).

Bestensuche (best-first search): Bei der Bestensuche wird die Suche vom jeweils besten offenen Knoten des Ableitungsbaumes fortgesetzt, egal, wo sich dieser Knoten im Ableitungsbaum befindet. Im Unterschied zum Bergsteigen setzt dieses Verfahren das Abspeichern bisheriger Suchergebnisse voraus. Werden spezielle heuristische Bewertungsfunktionen eingesetzt, die sowohl bisherige Aufwendungen als auch eine Schätzung der Aufwendungen bis zum Zielknoten berücksichtigen, bezeichnet man die Verfahren als A\*-Algorithmen.

Abbildung 2.2-2 gibt die prinzipielle Vorgehensweise der einzelnen Verfahren an einem Oder-Baum schematisch wieder.

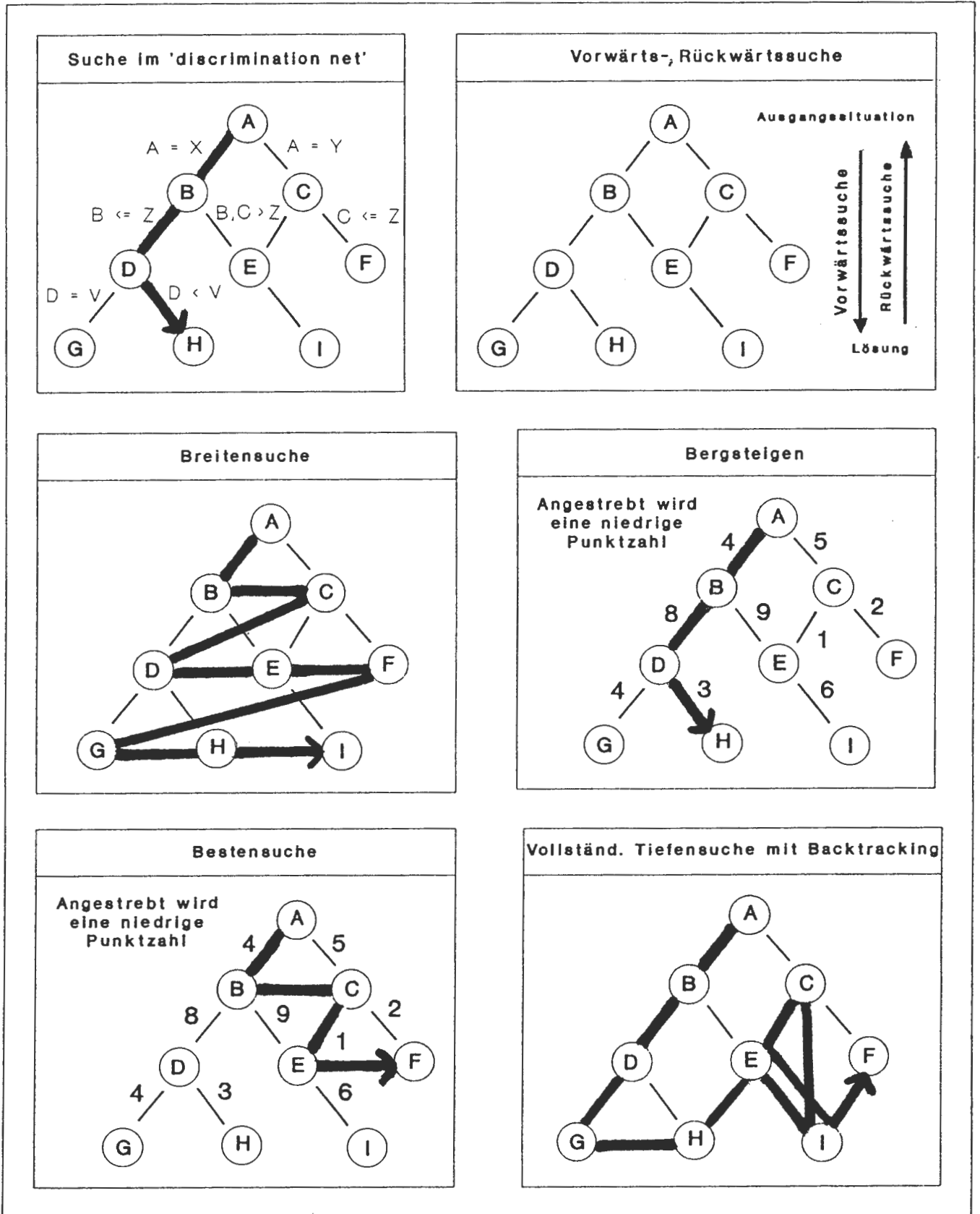


Abbildung 2.2-2: Suchverfahren in Expertensystemen

### 2.2.2.1.3 Mehrwertige Und-Oder-Graphen

Ein allgemeines Problem bei der Entwicklung von wissensbasierten Modellen besteht darin, daß unsicheres oder auch vages Wissen abgebildet werden muß. Als wesentliche Ursachen für die Unsicherheit gibt PINTO (1989, S. 318) folgende Gründe an:

- die vorhandene Information ist nicht zuverlässig,
- die heuristischen Regeln sind nicht sicher,
- die Information ist unvollständig,
- die Information resultiert aus der Aggregation verschiedener Wissensquellen.

Nehmen wir an, daß der in Abbildung 2.2-1 dargestellte Und-Oder-Graph mit Unsicherheiten behaftet ist, so stellt sich die Frage: Wie sicher sind die aus dem Graphen ableitbaren Lösungen? Da der Graph nur zwei Zustandswerte (wahr, falsch) 'kennt', kann mit ihm diese Frage nicht beantwortet werden. Es sind Modellansätze notwendig, die sowohl unsichere Knoten und Kanten als auch die Fortpflanzung (Propagation) von Unsicherheiten in Ableitungsbäumen ermöglichen.

In den letzten Jahren sind zahlreiche mathematische Ansätze zur Lösung dieses Problems entwickelt worden. Sie werden von PRADE (1985) in einem umfassenden Artikel beschrieben und analysiert. Die verschiedenen Ansätze lassen sich grob unterscheiden in statistisch orientierte Modellansätze, Modellansätze mit heuristischen Konfidenzfaktoren und Modellansätze basierend auf der Beschreibung unscharfer Mengen (vgl. KARRAS et al. 1987, S. 47ff; HEATWOLE und ZHANG 1988; BUCHANAN und DUDA 1983, S. 190ff).

Die statistische Abbildung von Unsicherheit basiert im wesentlichen auf dem Theorem von Bayes, mit dem bedingte Wahrscheinlichkeiten von Ereignissen hergeleitet werden können. Sei

- $W(H_i|E)$  = die Wahrscheinlichkeit, daß die Hypothese  $H_i$  auf der Grundlage des Belegs  $E$  wahr ist
- $W(E|H_i)$  = die Wahrscheinlichkeit, daß wir den Beleg  $E$  erhalten, wenn die Hypothese  $H_i$  wahr ist
- $W(H_i)$  = die a-priori-Wahrscheinlichkeit, daß die Hypothese  $H_i$  bei Abwesenheit irgendwelcher spezieller Belege wahr ist
- $k$  = Anzahl der möglichen Hypothesen

so kann das Bayessche Theorem nach RICH (1988, S. 201) definiert werden als

$$W(H_i|E) = \frac{W(E|H_i) \cdot W(H_i)}{\sum_{n=1}^k (W(E|H_n) \cdot W(H_n))} \quad (2.2-1)$$

Der direkte Einsatz dieses Modells in Expertensystemen ist bisher selten. Häufig wird das Bayessche Theorem mit Hilfe von sogenannten "odds" und "likelihood-ratios" dargestellt, die auf dem Verhältnis von  $W(H)/(1-W(H))$  aufbauen (DUDA et al. 1976; PINTO 1989, S. 319; HEATWOLE und ZHANG 1988).

Die Unsicherheit von Belegen (z.B. durch unsichere Benutzereingaben) kann mit  $W(E|\tilde{E})$  beschrieben werden, wobei mit  $\tilde{E}$  alle aktuell verfügbaren, relevanten Beobachtungen gemeint sind. Ist  $W(E|\tilde{E}) = W(E)$ , das heißt, es existieren keine zusätzlichen aktuellen Informationen, muß in einem konsistenten Inferenznetz  $W(H|\tilde{E}) = W(H)$  sein. Wenn wir die a-priori Wahrscheinlichkeiten von E für die Berechnung der aktualisierten Wahrscheinlichkeit von H verwenden, muß somit das Ergebnis mit der a-priori Wahrscheinlichkeit von H übereinstimmen. Diese Bedingung ist im praktischen Einsatz nur sehr schwer zu realisieren, da bei der Erstellung des Unschärfemodells die a-priori-Wahrscheinlichkeiten aufeinander abgestimmt werden müßten.

Als Ausweg schlagen DUDA et al. (1976) die Teilung der Interpolationsgeraden vor, um die beschriebene Konsistenzbedingung zu erfüllen (Abbildung 2.2-3).

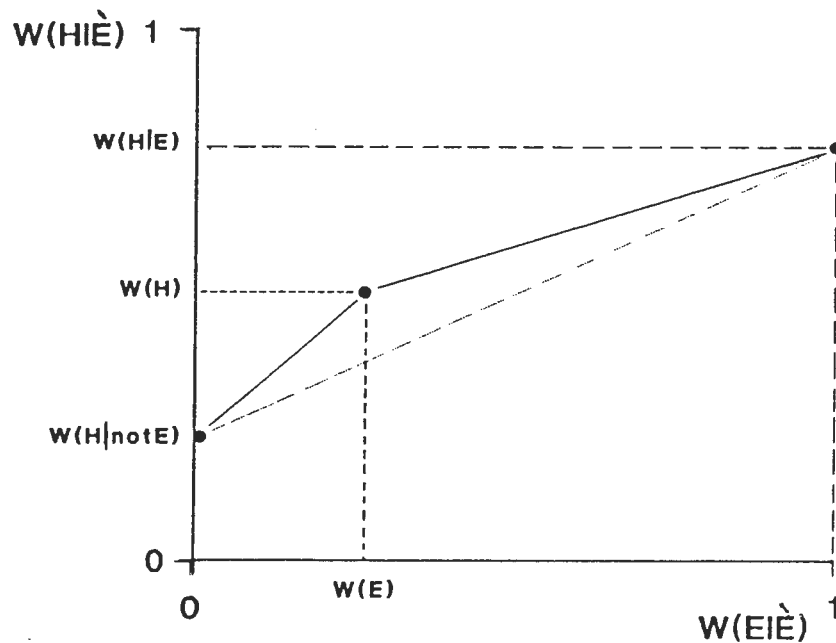


Abbildung 2.2-3: Interpolationsgerade zur Berechnung von  $W(H|\tilde{E})$

(nach DUDA et al. 1976)

Die Definition der Geraden lautet:

$$W(H|\dot{E}) = \begin{cases} W(H|\text{not}E) + \frac{W(E|\dot{E})}{W(E)} \cdot (W(H) - W(H|\text{not}E)) & \text{falls } 0 \leq W(E|\dot{E}) \leq W(E) \\ \frac{W(H) - W(H|E) \cdot W(E)}{1 - W(E)} + W(E|\dot{E}) \cdot \frac{W(H|E) - W(H)}{1 - W(E)} & \text{falls } W(E) \leq W(E|\dot{E}) \leq 1 \end{cases} \quad (2.2-2)$$

Weitere Möglichkeiten zur Anpassung der Interpolationsgeraden und der Verknüpfung von mehreren Ereignissen beschreiben DUDA et al. (1976) und HEATWOLE und ZHANG (1988).

Der Vorteil der oben beschriebenen wahrscheinlichkeitstheoretischen Modellsätze liegt in ihrer "klaren mathematischen Basis" (PINTO 1989, S.319). Auf der anderen Seite sprechen jedoch einige Gründe gegen diese Verfahren (vgl. RICH 1988, S. 209; KARRAS et al. 1987, S. 52): In der Regel ist ein Experte nicht in der Lage, Punktschätzungen für Wahrscheinlichkeiten und sogar bedingte Wahrscheinlichkeiten anzugeben. Die a-priori-Wahrscheinlichkeiten sind meistens unbekannt. Das Bayessche Theorem ist nur gültig, wenn die Belege E statistisch unabhängig voneinander sind. Desweiteren müssen alle möglichen Ergebnisse disjunkt sein (somit dürfen niemals zwei Ergebnisse gleichzeitig auftreten). Die Wahrscheinlichkeiten aller möglichen Ergebnisse müssen sich zu 1 summieren. Änderungen oder Hinzunahmen neuer Aspekte verursachen, daß das komplette Wahrscheinlichkeitsmodell geändert werden muß.

Aufgrund der genannten Kritikpunkte werden Bayessche Unschärfemodelle im Bereich der Expertensystementwicklung selten angewandt (KARRAS et al. 1987, S. 52).

Eine wesentlich häufigere Methode ist die Beschreibung unsicheren Wissens mit Hilfe von heuristischen Sicherheitsfaktoren. Hierbei bilden geschätzte Sicherheitswerte die Basis für die Modellbildung. Es existieren unterschiedliche Berechnungsverfahren. HEATWOLE und ZHANG (1988) beschreiben die folgende Unschärferepräsentationsform:

Jeder Knoten, der im System auftritt, wird mit einem Sicherheitswert CF zwischen -1 und 1 belegt. Bei einem CF-Wert von 1 ist der Knoten auf jeden Fall wahr. Bei CF = -1 ist der Knoten auf jeden Fall falsch, bei CF = 0 liegen keine Informationen vor. Auf die gleiche Art und Weise werden Sicherheitswerte für Kanten definiert. Für die Verknüpfung mehrerer Knoten  $E_i \{ i = 1, 2, \dots, n \}$  gilt

für den resultierenden Sicherheitswert  $CF(E)$  bei logischem Und

$$CF(E) = CF(E_1 \wedge \dots \wedge E_n) = \min(CF(E_1), \dots, CF(E_n)) \quad (2.2-3)$$

bei logischem Oder

$$CF(E) = CF(E_1 \vee \dots \vee E_n) = \max(CF(E_1), \dots, CF(E_n)) \quad (2.2-4)$$

bei unabhängiger Verknüpfung (Kombinationsverknüpfung)

$$CF(E) = \begin{cases} CF(E_1) + CF(E_2) - CF(E_1) \cdot CF(E_2) & \text{falls } CF(E_1) \text{ und } CF(E_2) \geq 0 \\ CF(E_1) + CF(E_2) + CF(E_1) \cdot CF(E_2) & \text{falls } CF(E_1) \text{ und } CF(E_2) < 0 \\ CF(E_1) + CF(E_2) & \text{sonst} \end{cases} \quad (2.2-5)$$

und bei der Schlußfolgerung

$$CF(E) = CF(E_1 \rightarrow E) \cdot \max(0, CF(E_1)) \quad (2.2-6)$$

Die beschriebenen Berechnungsverfahren gehen auf die Ansätze im Expertensystem MYCIN (SHORTLIFFE und BUCHANAN 1975) zurück. Übersichtliche Rechen- und Anwendungsbeispiele finden sich bei MARCELLUS (1989, S. 83ff), HEATWOLE und ZHANG (1988) oder JANSON (1989, S. 236ff).

Nach KARRAS et al. (1987, S. 51) stellen Konfidenzfaktoren eine gute Möglichkeit dar, unsicheres Wissen auf pragmatische Weise zu repräsentieren.

Ein weiteres Verfahren, welches in letzter Zeit in den Blickpunkt öffentlicher Interessen gerückt ist, ist die Darstellung unsicheren Wissens mit unscharfen Mengen, den sogenannten fuzzy sets (vgl. BILLERBECK et al. 1991). Diese auf ZADEH (1965) zurückgehende Theorie versucht, durch eine Verallgemeinerung der Mengenlehre Unschärfe zu repräsentieren.

Ist  $X$  eine Menge von Elementen, die hinsichtlich einer unscharfen Aussage zu bewerten sind, so heißt  $\tilde{E} = \{x, \mu_{\tilde{E}}(x)\}$ ,  $x \in X$  eine unscharfe Menge auf  $X$ . Hierbei wird  $\mu_{\tilde{E}}(x)$  als Zugehörigkeitsfunktion bezeichnet (ZIMMERMANN 1988). Ein fuzzy set besteht somit aus einer Grundmenge, die alle für das Objekt in Frage kommenden Zustände mit Hilfe einer Zugehörigkeitsfunktion beschreibt. Im allgemeinen gilt dabei, daß für  $\mu_{\tilde{E}}(x) = 0$  keine Zugehörigkeit und für  $\mu_{\tilde{E}}(x) = 1$  die volle Zugehörigkeit zur Grundmenge besteht.

Vielfach werden in Fuzzy-Systemen sprachliche (linguistische) Bezeichnungen

mit Hilfe der oben definierten Zugehörigkeitsfunktionen umschrieben (z.B. groß, klein etc.) (vgl. ZADEH 1989). Genauso werden auch unscharfe Zahlen definiert. Steigerungen oder Abschwächungen können durch Potenzieren der entsprechenden Zugehörigkeitsfunktion erzielt werden (z.B.  $\mu_{\text{sehr groß}}(x) = \mu_{\text{groß}}^2(x)$ ). Ausführliche Beschreibungen möglicher Zugehörigkeitsfunktionen liefern KAUFMANN (1975, S. 168ff) und NEGOITA (1985, S. 50ff),

Um in einem Inferenznetz unscharfe Mengen verarbeiten zu können, müssen Berechnungsvorschriften für die Verknüpfung mehrerer Mengen (sets) definiert werden. Für die Verbindung zweier unscharfer Mengen  $\tilde{E}_1$  und  $\tilde{E}_2$  mit ihren Zugehörigkeitswerten  $x_{E_1}$  und  $x_{E_2}$  können u.a. folgende Verarbeitungsvorschriften angegeben werden (vgl. ZIMMERMANN 1988; GRAHAM und JONES 1988, S. 150f, PINTO 1989, S. 312; KANDEL 1986, S. 123ff; KAUFMANN und GUPTA 1988, S. 10):

1. Logisches Und:

$$\mu(\tilde{E}_1 \cap \tilde{E}_2) = x_{E_1} \cdot x_{E_2} \quad (2.2-7)$$

$$\mu(\tilde{E}_1 \cap \tilde{E}_2) = \max(0, x_{E_1} + x_{E_2} - 1) \quad (2.2-8)$$

2. Logisches Oder:

$$\mu(\tilde{E}_1 \cup \tilde{E}_2) = x_{E_1} + x_{E_2} - x_{E_1} \cdot x_{E_2} \quad (2.2-9)$$

$$\mu(\tilde{E}_1 \cup \tilde{E}_2) = \min(1, x_{E_1} + x_{E_2}) \quad (2.2-10)$$

3. Schlußfolgerung:

$$\mu(\tilde{E}_1) = \min(x_{E_1}, \mu(\tilde{E}_1 \rightarrow \tilde{E}_2)) \quad (2.2-11)$$

Die Verarbeitung unscharfer Mengen in einem Regelwerk erfolgt durch die Verknüpfung der aus den Eingangsgrößen und den Regeln resultierenden Mengen mit den jeweils ausgewählten Berechnungsverfahren. Die neu entstehende unscharfe Menge stellt die Beschreibung einer unscharfen Lösung dar. Soll diese unscharfe Lösungsmenge jedoch nur eine konkrete Entscheidung bedingen, muß eine sogenannte 'defuzzification' durchgeführt werden. Nach GRAHAM und JONES (1988, S. 146ff) werden dabei besonders zwei Auswahlverfahren eingesetzt: 1. die Auswahl der Lösung mit dem höchsten Zugehörigkeitsgrad (maximum method), 2. die Auswahl der Lösung unter Verrechnung aller Zugehörigkeitsgrade (moment method).

Der Einsatz von Fuzzy-Sets in praxisorientierten Systemen außerhalb der Regelungstechnik ist bisher selten. Erfahrungsberichte aus dem naturwissenschaftlichen Bereich liefern SALSKI und KANDZIA (1991) und KAFKA et al. (1991).

## **2.2.2.2 Wissensrepräsentationsformen**

### **2.2.2.2.1 Allgemeines**

Zentrale Aufgabe seit Bestehen der KI ist es, Formalismen und Methoden zur Darstellung von Wissen in Expertensystemen zu entwickeln und zu diskutieren (vgl. FREKSA (1988)). Der Begriff der Wissensrepräsentation spielt dabei eine dominierende Rolle und wird unterschiedlich definiert und interpretiert. Zwei Aussagen mögen dieses verdeutlichen. BARR und FEIGENBAUM (1981, S. 143) schreiben: "In AI, a representation of knowledge is a combination of data structures and interpretive procedures that, if used in the right way in a program, will lead to 'knowledgeable' behavior". STOYAN (1988) hingegen beschreibt die Wissensrepräsentation einfach als "Programmierung mit bestimmten Programmierarten(stilen)". Für ihn steht die Sicht des Programmierens eindeutig im Vordergrund. Er begründet seine Aussage folgendermaßen: "... wegen der weithin akzeptierten funktionalen Programmierung muß man zum einen ohnehin den Programmierbegriff verallgemeinern und darf nicht Programme mit Folgen von Anweisungen ("Prozeduren") identifizieren, sondern hat in ihnen formale Strukturen zu sehen, für die der Rechner Interpretationsregeln hat, um aus ihnen ein beabsichtigtes Resultat zu erzeugen...".

Trotz dieser zum Teil doch recht kontrovers geführten Diskussion wird die elementare Bedeutung der Auswahl geeigneter Wissensrepräsentationsformen bei der Entwicklung von Expertensystemen immer wieder hervorgehoben. WINSTON (1987, S. 266) schreibt dazu: "Die Erfahrung hat gezeigt, daß eine gute Repräsentation oft der Schlüssel dazu ist, schwierige Probleme in einfache umzuwandeln."

Im folgenden werden die wichtigsten Wissensrepräsentationsformen in Expertensystemen kurz vorgestellt und erläutert. Eine Abgrenzung zum Kapitel 2.2.2.1 wird zwar angestrebt, sie ist aber aufgrund der sehr engen inhaltlichen Verflechtungen nicht immer zu erreichen.

#### **2.2.2.2.2 Logik**

Die älteste und am meisten untersuchte Wissensrepräsentationsform ist die Logik. Nach BREWKA (1986) kann sie, wenn man die Ursprünge in der Antike und dem Mittelalter unbeachtet läßt, auf eine über hundertjährige Geschichte zurückgreifen. Heutige Logik-Repräsentationsformalismen sind geprägt von der Aussagenlogik und einer erweiterten Aussagenlogik, der Prädikatenlogik.



Die Aussagenlogik ist ein allgemeines, logisches System, welches sich mit der Formalisierung und Gültigkeit von atomaren, zusammengesetzten und komplexen Aussagen beschäftigt. Aussagen können entweder wahr oder falsch sein. Abbildung 2.2-4 gibt die bei zwei Aussagen durch Verknüpfung entstehenden 16 Möglichkeiten der Wahrheitswertefunktionen wieder.

E <sub>1</sub>	E <sub>2</sub>	f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>5</sub>	f <sub>6</sub>	f <sub>7</sub>	f <sub>8</sub>	f <sub>9</sub>	f <sub>10</sub>	f <sub>11</sub>	f <sub>12</sub>	f <sub>13</sub>	f <sub>14</sub>	f <sub>15</sub>	f <sub>16</sub>
W	W	W	W	W	W	F	W	W	W	F	F	F	W	F	F	F	F
W	F	W	W	W	F	W	W	F	F	W	W	F	F	W	F	F	F
F	W	W	W	F	W	W	F	W	F	W	F	W	F	F	W	F	F
F	F	W	F	W	W	W	F	F	W	F	W	W	F	F	F	W	F

E<sub>1</sub> und E<sub>2</sub> = Aussagen, f<sub>1...16</sub> = Verknüpfungsfunktionen,  
W = wahr, F = falsch.

Abbildung 2.2-4 : Wahrheitswertetafel für die Verknüpfung zweier Aussagen

GERSTER (1972, S. 30) schlägt folgende Interpretation der Wahrheitswertefunktionen vor:  $f_1 = W$ ,  $f_2 = E_1 \vee E_2$ ,  $f_3 = E_2 \rightarrow E_1$ ,  $f_4 = E_1 \rightarrow E_2$ ,  $f_5 = \neg (E_1 \wedge E_2)$ ,  $f_6 = E_1$ ,  $f_7 = E_2$ ,  $f_8 = E_1 \leftrightarrow E_2$ ,  $f_9 = E_1 \vee \neg E_2$ ,  $f_{10} = \neg E_2$ ,  $f_{11} = \neg E_1$ ,  $f_{12} = E_1 \wedge E_2$ ,  $f_{13} = \neg (E_1 \rightarrow E_2)$ ,  $f_{14} = \neg (E_2 \rightarrow E_1)$ ,  $f_{15} = \neg (E_1 \vee E_2)$ ,  $f_{16} = F$ .

Allgemein gilt, daß bei n Aussagen  $2^{(2^n)}$  verschiedene Wahrheitswertefunktionen existieren. Jede Wahrheitswertefunktion kann durch beliebig verschachtelte logische Ausdrücke repräsentiert werden, die sich alle auf Wahrheitswertefunktionen zurückführen lassen, die mit Hilfe der Junktoren  $\vee$ ,  $\wedge$ ,  $\neg$  gebildet werden können (konjunktive bzw. disjunktive Normalform).

Abbildung 2.2-5 gibt die wichtigsten aussagenlogischen Gesetze für die Verknüpfung mehrerer Aussagen wieder. Ferner gilt das Kommutativ-, Assoziativ- und Distributivgesetz. Mit den dargestellten Gesetzen kann von einer oder mehreren Prämissen auf eine Konklusion geschlossen werden. Sie stellen daher die Basis der in der KI vielfach zitierten Inferenzmechanismen dar. Besonders hervorzuheben sind in diesem Zusammenhang das Gesetz zum modus ponens und die Gesetze zur Subjunktion.

Gesetz von de Morgan:	$\neg(E_1 \wedge E_2) \Leftrightarrow \neg E_1 \vee \neg E_2$ $\neg(E_1 \vee E_2) \Leftrightarrow \neg E_1 \wedge \neg E_2$
Gesetze zur Subjunktion:	$E_1 \rightarrow E_2 \Leftrightarrow \neg E_1 \vee E_2$ $E_1 \rightarrow E_2 \Leftrightarrow \neg(E_1 \wedge \neg E_2)$ $E_1 \rightarrow E_2 \Leftrightarrow \neg E_2 \rightarrow \neg E_1$ $\neg(E_1 \rightarrow E_2) \Leftrightarrow E_1 \wedge \neg E_2$
Gesetze zur Bijunktion:	$E_1 \leftrightarrow E_2 \Leftrightarrow (E_1 \wedge E_2) \vee (\neg E_1 \wedge \neg E_2)$ $E_1 \leftrightarrow E_2 \Leftrightarrow (\neg E_1 \vee E_2) \wedge (E_1 \vee \neg E_2)$ $\neg(E_1 \leftrightarrow E_2) \Leftrightarrow (E_1 \wedge \neg E_2) \vee (\neg E_1 \wedge E_2)$
Gesetze zur Alternative:	$E_1 \vee E_2 \Leftrightarrow (E_1 \wedge \neg E_2) \vee (\neg E_1 \wedge E_2)$ $E_1 \vee E_2 \Leftrightarrow (E_1 \vee E_2) \wedge \neg(E_1 \wedge E_2)$
Gesetz des Duns Scotus:	$\neg E_1 \Rightarrow E_1 \rightarrow E_2$
Gesetz zum modus ponens:	$E_1 \wedge (E_1 \rightarrow E_2) \Rightarrow E_2$
Gesetz zum modus tollens:	$(E_1 \rightarrow E_2) \wedge \neg E_2 \Rightarrow \neg E_1$
Gesetz zum modus barbara:	$(E_1 \rightarrow E_2) \wedge (E_2 \rightarrow E_3) \Rightarrow E_1 \rightarrow E_3$

Abbildung 2.2-5: Wichtige Gesetze der Aussagenlogik

(nach GERSTER 1972, S. 44ff, verändert)

Die Grenzen der Aussagenlogik als direkte Wissensrepräsentationsform macht folgendes Beispiel deutlich: Zwei inhaltlich eng verknüpfte Aussagen (z.B. Chrysanthemen sind Schnittblumen; Rosen sind Schnittblumen) bedeuten in der Aussagenlogik völlig verschiedene Tatsachen, da sie jeweils eine ganz spezielle Aussage bedingen. Es ist nicht möglich, zwischen den beiden Aussagen Ähnlichkeiten darzustellen. Ebenso kann keine Verallgemeinerung dargestellt werden, da die Aussagenlogik keine Variablen erlaubt.

Es ist daher eine Sprache notwendig, die eine größere Ausdrucksmöglichkeit bietet und die die inneren Strukturen der Aussage berücksichtigt (vgl. TANIMOTO 1990, S. 140). Die Prädikatenlogik ist in der Lage, diese Bedingungen zumindest zum Teil zu erfüllen. Sie kann folgendermaßen beschrieben werden (BREWKA 1986):

- In der Prädikatenlogik werden Sachverhalte durch Formeln dargestellt, die aus Konstanten, Variablen, Funktionssymbolen, Prädikatensymbolen, logischen Junktoren (s. Aussagenlogik) und Quantoren ( $\forall, \exists$ ) aufgebaut sind.
- Durch Anwendung eines Prädikatensymbols auf Terme entstehen atomare Formeln.
- Atomare Formeln lassen sich durch Junktoren und Quantoren zu beliebig komplexen Formeln zusammensetzen.

Der Quantor  $\forall$  wird als universeller Quantor bezeichnet (Allquantor). Er drückt aus, daß bei  $\forall x [Pr(x)]$  im Definitionsbereich für alle  $x$  das Prädikat  $Pr(x)$  wahr ist. Der Quantor  $\exists$  drückt aus, daß bei  $\exists x [Pr(x)]$  mindestens ein Element existiert, das  $Pr(x)$  wahr macht (Existenzquantor). In der Prädikatenlogik 1. Stufe dürfen Variablen nur für Terme benutzt werden. Die Prädikatenlogik 2. Stufe erlaubt auch den Einsatz von Variablen für Prädikate (vgl. WINSTON 1987, S. 229).

Ein kleines Beispiel soll die Syntax und Semantik der Prädikatenlogik verdeutlichen:

Die Aussage "Alle Kreise der Abbildung 2.2-1 beinhalten Buchstaben" soll in eine Formel der Prädikatenlogik überführt werden. Hierzu wird zunächst die tiefere Struktur der Aussage mit den beteiligten Objekten näher betrachtet: Die Abbildung 2.2-1 enthält verschiedene Symbole (Kreise, Linien etc.). Eine Gruppe von Symbolen sind Kreise. Die Kreissymbole der Abbildung zeichnen sich dadurch aus, daß sie alle Buchstaben beinhalten. Wir können somit die Prädikate Kreis, Buchstabe, Element, Beinhaltet definieren und sie mit Termen zu atomaren Formeln erweitern:

<i>Kreis (Runde_Flaeche)</i>	Eine runde Fläche ist ein Kreis
<i>Buchstabe (A)</i>	Das A ist ein Buchstabe
<i>Element (Symbol, Abbildung2.2-1)</i>	Ein Symbol ist Element der Abb. 2.2-1
<i>Beinhaltet (Oberster_Kreis, A)</i>	Der oberste Kreis beinhaltet ein A

Durch den Einsatz von Junktoren, Quantoren und Variablen können wir die wohldefinierte Formel

$$\forall x [Element(x, Abbildung2.2-1) \wedge Kreis(x) \Rightarrow \exists y [Buchstabe(y) \wedge Beinhaltet(x,y)]] \quad (2.2-12)$$

erstellen, die den Anforderungen der Prädikatenlogik 1. Stufe entspricht.

Die Erstellung von wohldefinierten Formeln in der Prädikatenlogik ist kein Selbstzweck, sondern durch sie wird logisches Schlußfolgern erst ermöglicht.

Um zu beweisen, daß eine Formel aus einer Formelmenge logisch folgt, kann auf das von ROBINSON (1965) entwickelte Resolutionsverfahren zurückgegriffen werden. Hierfür ist es notwendig, alle Formeln so umzuwandeln, daß nur noch Oder-Verknüpfungen von atomaren und negierten atomaren Formeln auftauchen (Klauselform). Implikatoren ( $\rightarrow$ ) werden ersetzt durch  $\neg E_1 \vee E_2$ . Existenzquantoren werden eliminiert, indem Hilfsfunktionen (Skolemfunktionen) gebildet werden, die die Existenzquantoren ersetzen. Werden alle Variablen als allgemeingültig angenommen, kann auf Allquantoren verzichtet werden. Die Auflösung der Konjunktion kann durch die separate Notation einzelner Konjunktionsklauseln erfolgen (zur genaueren Beschreibung einzelner Umformungsschritte siehe WINSTON (1987, S. 235ff).

Der entscheidende Schritt im Rahmen des Resolutionsbeweises besteht darin, aus der verfügbaren Klauselmenge neue Klauseln (Resolventen) zu bilden, um zu beweisen, daß das Gegenteil der zu beweisenden Behauptung falsch ist (also einen Widerspruch in der Klauselmenge zu finden). Die Resolventenbildung geschieht dadurch, daß aus den Klauseln  $\neg E_1 \vee E_2$  und  $\neg E_2 \vee E_3$  die neue Klausel  $\neg E_1 \vee E_3$  gebildet wird. Da es sich bei der Prädikatenlogik 1. Stufe um ein semientscheidbares System handelt, kann es keinen Beweis geben, der beweist, daß eine Klausel aufgrund einer Klauselmenge nicht logisch ableitbar ist: Unter bestimmten Voraussetzungen kann diese Tatsache dazu führen, daß die Resolventenbildung immer weiter fortgesetzt werden kann und das Beweisverfahren "endlos läuft" (BREWKA 1986).

Anhand eines kleinen Beispiels soll die prinzipielle Wirkungsweise des Resolutionsprinzips verdeutlicht werden:

Die drei Knoten A, B, D in Abbildung 2.2-1 sollen die Aussagen  $A \rightarrow B$  und  $B \rightarrow D$  repräsentieren. Ferner wird davon ausgegangen, daß A wahr ist. Es soll bewiesen werden, daß D wahr ist (in diesem Fall sehr leicht mit modus ponens zu folgern).

Vorgehensweise:

1. Nach Eliminierung des Junktors  $\rightarrow$  ergibt sich die Klauselmenge  $\{\neg A \vee B, \neg B \vee D\}$ .
2. Durch Negation der zu beweisenden Aussage wird  $\neg D$  der Klauselmenge zugefügt.
3. Durch Resolventenbildung wird  $\neg A \vee D$  der Klauselmenge zugefügt.
4. Schlußfolgerung, daß D wahr ist, da in der Klauselmenge der Widerspruch  $A \wedge (\neg A \vee D) \Rightarrow D \wedge \neg D$  auftaucht.

Der Widerspruch innerhalb der Klauselmenge läßt sich bei der Aussagenlogik leicht erkennen. Da aber in der Prädikatenlogik Aussagen durch atomare Formeln dargestellt werden und sie demnach Prädikate, Variablen, Konstanten etc. enthalten, müssen besondere Verfahren angewendet werden, die überprüfen, ob zwei atomare Formeln äquivalent sind oder nicht. Im einzelnen erfolgt dieser Test durch die sogenannte Unifikation, die mittels Substitution und sukzessiven Überprüfungsalgorithmen auf Äquivalenz testet (vgl. matching). Eine genaue Beschreibung möglicher Algorithmen geben RICH (1988, S. 168ff), TANIMOTO (1990, S. 241ff) und BERGMANN und NOLL (1977, S. 182ff).

Eine direkte Darstellung von Wissen in Form der Prädikatenlogik ist selten. In der Regel wird die Programmiersprache Prolog eingesetzt, die auf der Prädikatenlogik und dem Resolutionsprinzip aufbaut. Aus Effizienzgründen werden vereinfachte Beweisverfahren genutzt, eingeschränkte prädikatenlogische Formeln in Form von Hornklauseln sowie nicht logische Elemente wie 'cut', Listenverarbeitung und Backtracking zur Verfügung gestellt. Auf eine weitere Beschreibung der logischen Programmiersprache Prolog wird an dieser Stelle bewußt verzichtet. Es wird auf die angegebene Literatur verwiesen (GESKE 1988; WALKER 1987; LLOYD 1987; GOLTZ und HERRE 1990; CLOCKSIN und MELLISH 1981).

### **2.2.2.2.3 Produktionsregeln**

Produktionsregelsysteme haben im Bereich der Expertensysteme weite Verbreitung gefunden und sind die 'klassische' Wissensrepräsentationsform in Expertensystemen (vgl. SHORTLIFFE 1976). Sie zeichnen sich aus durch eine einfache Struktur, gute Transparenz und eine strikte Trennung zwischen Wissensbasis und Kontrollmechanismus (PHAM und PHAM 1988, S. 6f). Es existieren sehr unterschiedliche Regelsysteme mit unterschiedlichen Organisationsformen. Nach BREWKA (1986) und DILGER (1991) bestehen Produktionsregelsysteme aus den Komponenten

- dynamische Datenbasis,
- Regelbasis,
- Kontrollsystem.

Die dynamische Datenbasis enthält Fakten, die a-priori bekannt sind, Benutzerangaben und Fakten, die das System herleitet.

Die Regelbasis enthält Regeln der Gestalt: Wenn <Bedingung> Dann <Aktion>. Nach PUPPE (1987, S. 17) kann unterschieden werden in einfache Regeln (s.o.),

konjugierte Regeln (Wenn  $\langle \text{Bedingung} \wedge \text{Bedingung} \rangle$  Dann  $\langle \text{Aktion} \rangle$ ) und strukturierte Regeln (Wenn  $\langle \text{Bedingung}, \text{Ausnahme} \rangle$  Dann  $\langle \text{Aktion} \rangle$ ). HARMON und KING (1989, S. 50) beschreiben dazu noch variable Regeln, bei denen Teile der Bedingung und/oder der Aktion Variablen enthalten. SCHNUPP und NGUYEN HUU (1987, S. 19) stellen heraus, daß Regeln sowohl logisch (Aus  $\langle \text{Bedingung} \rangle$  folgt logisch  $\langle \text{Ergebnis} \rangle$ ) als auch prozedural (Wenn  $\langle \text{Situation} \rangle$  Dann  $\langle \text{Aktion} \rangle$ ) verstanden und eingesetzt werden können.

Das Kontrollsystem beinhaltet den sogenannten Inferenzmechanismus (auch Regelinterpretierer genannt). Er stellt eine Art repeat- until-Schleife dar, die solange mit Hilfe der Regelbasis die Datenbasis manipuliert (d.h. die Regeln anwendet), bis ein Zielzustand erreicht ist oder keine Regel mehr anwendbar ist. Die Auswertung der Regeln kann dabei entweder vorwärts verkettend oder rückwärts verkettend erfolgen. Es kommen sowohl Tiefensuchverfahren als auch Breitensuche zum Einsatz.

Ein großes Problem bei der Auswertung von Regelsystemen ist die Auswahl einer Regel im Inferenzprozeß, wenn mehrere Regeln theoretisch in Frage kommen. Auf mögliche Auswahlstrategien wurde bereits in Kapitel 2.2.2.1.2 eingegangen. Bei Produktionsregelsystemen werden darüberhinaus noch vielfach Metaregeln eingesetzt, die eine bevorzugte Auswahl der Regeln nach

- Anordnung im System (Anordnung in der Wissensbasis),
- Spezialisierung in ihren Bedingungen (i.d.R. nach Anzahl ihrer Bedingungen),
- speziell definierten Prioritäten (Rangnummern),
- Auftreten im System (z.B. Regeln, die lange nicht mehr benutzt wurden)
- und Einteilung in Gruppen und deren Aktivierung (Kontextbeschränkung)

ermöglichen (vgl. WINSTON 1987, S. 188; BARR und FEIGENBAUM 1981, S. 197).

Verschiedene Arten von Regelsystemen mit Anwendungsbeispielen sind bei THIEME und WHITTAKER (1986), WINSTON (1987, S. 184ff) und PUPPE (1987, S. 16ff) zu finden.

#### **2.2.2.2.4 Frames**

Ein wichtiger Wissensrepräsentationsansatz innerhalb der Expertensysteme ist die Darstellung von Wissen mit Hilfe sogenannter Frames (Rahmen). Der Ansatz geht auf Arbeiten von MINSKY (1975) zurück, wurde in den letzten Jahren weiterentwickelt und zählt heute zu den bedeutendsten Darstellungs- und

Strukturierungsformen in Expertensystemen (DILGER 1991).

PHAM und PHAM (1988, S. 7) beschreiben Frames als "a record-like data structure, a form, for encoding information on a stereotyped situation, a class of objects, a general concept or a specific instance of any of these".

Im Sinne des Frameansatzes enthalten daher die in Abbildung 2.2-1 dargestellten Knoten des Ableitungsbaumes keine Regeln, sondern Beschreibungen von Objekten. Die Verbindungen zwischen den einzelnen Knoten definieren Hierarchiestrukturen, so daß Klassen-Unterklassen-Relationen beschrieben werden können. Die Frames bilden das Grundgerüst für diese Wissensdarstellung. Sie stellen den "Erwartungsrahmen zur Aufnahme und Abspeicherung von Wissen" dar (PUPPE 1987, S. 18). Andere Bezeichnungen für frameorientierte Wissensrepräsentationen sind "Class", "Unit", "Schema", "Flavor", "Concept" (BREWKA 1986; PHAM und PHAM 1988, S. 7).

Analog zu Produktionsregelsystemen ist auch bei framebasierten Expertensystemen die Wissensbasis (Objektbasis) streng vom Bearbeitungsmechanismus getrennt. Die wesentlichen Bestandteile eines Frame-Systems sind somit die Objektbasis und ein Objektinterpret (als Pendant zum Regelinterpret) (DILGER 1991).

Die Objektbasis besteht aus der Sammlung aller für die Lösung eines Problems relevanten Objekte mit den assoziierten Informationen in einer einheitlichen Struktur. Die Informationen werden in sogenannten Slots abgelegt. Neben dem eigentlichen Informationsinhalt enthält ein Slot zusätzlich noch Angaben darüber, welcher Art die gespeicherten Informationen sind (Slotart = Facet). Eine Informationseinheit in einem framebasierten Expertensystem besteht aus

- dem Objektnamen, welcher einen Rahmen identifiziert,
- der Slotbezeichnung in diesem Rahmen,
- dem Facet, welches die Slotauswertung beschreibt,
- den im Slot gespeicherten Daten.

Die wichtigsten Slotfacetten heutiger Framesysteme sind (vgl. BREWKA 1986; SAVORY 1988, S. 143ff):

- Vererbungs-Slots: Durch speziell gekennzeichnete Slots (Is-a-Slot, A-kind-of-Slot, Supers-slot) werden hierarchische Beziehungen zwischen Objekten definiert. Der Wert in diesem Slot gibt die Oberklassen/Unterklassen an, die mit dem Objekt in Beziehung stehen. Ein Objekt kann alle Eigenschaften von Oberklassen übernehmen (erben) oder seine Eigenschaften an Un-

terklassen weitergeben (vererben). Steht ein Objekt mit zwei Oberklassen in direkter Verbindung, so kann Mehrfachvererbung (multiple inheritance) stattfinden.

- Wert-Slots: In diesen Slots sind die für das Objekt zutreffenden Informationen abgespeichert (numerische Werte oder Zeichenketten).
- Prozedur-Slots: In diesen Slots werden Objektinformationen nicht direkt gespeichert, sondern es wird hinterlegt, mit welchen prozeduralen Vorschriften (Berechnungsalgorithmen, Formeln etc.) der Slot mit Informationen gefüllt werden kann.
- Default-Slots: Diese Slotfacette beinhaltet Standardinformationen, die genutzt werden können, wenn keine weiteren Informationen zur Verfügung stehen.

Die Bearbeitung und Auswertung eines Objektbaumes erfolgt mit Hilfe des Objektinterpreters. Im einzelnen muß das Bereitstellen und Verändern der Slotwerte, die Kontrolle der Vererbungsmechanismen und die Auswertung der Prozedur-Slots erfolgen (DILGER 1991).

Die Verarbeitung der Objektbasis beruht bei heutigen Systemen auf dem Pattern-Matching und auf speziell auf den Vererbungsmechanismus ausgerichteten Algorithmen (SHADBOLT 1989, S. 148). SAVORY (1988, S. 157ff), SCHNUPP und NGUYEN HUU (1987, S. 128ff) beschreiben Einzelheiten verschiedener Objektinterpreter mit deren Zugriffsmechanismen (Zugriffsprimitiven).

Framesysteme stellen Wissen auf der Basis von Objektbeschreibungen dar. Sie zählen daher zu den objektorientierten Wissensrepräsentationssystemen (TELLO 1989, S. 26) und haben zumindest vom strukturellen Ansatz her große Ähnlichkeit mit dem heute vielfach propagierten objektorientierten Programmierparadigma (vgl. MOSER 1991; VORWERK 1991). KARRAS et al. (1987, S. 46) stellen jedoch fest, daß beide Entwicklungen unabhängig voneinander verliefen und erst heute in modernen Darstellungswerkzeugen vereint werden.

#### **2.2.2.2.5. Prozeduren**

BARR und FEIGENBAUM (1981, S. 155) definieren die prozedurale Wissensrepräsentationsform mit: "In a procedural representation, knowledge about the world is contained in procedures - small programs that know how to do specific things, how to proceed in well-specified situations". Die prozedurale Wissensdarstellung beschäftigt sich daher besonders mit Algorithmen, numeri-



schen Funktionen, Differentialgleichungen etc., d.h. mit einem Wissen, welches beschreibt, wie Dinge auszuführen sind.

Der rein prozedurale Darstellungsansatz steht damit im krassen Gegensatz zu deklarativen Verfahren (Aussageverfahren), in denen das meiste Wissen als eine statische Sammlung von Tatsachen dargestellt wird, welche mit einer geringen Menge an prozeduralen Verfahren (z.B. Inferenzmechanismen) ausgewertet werden (RICH 1988, S. 218). Ohne auf die seit langem, besonders in der KI geführte Diskussion zur deklarativen/prozeduralen Kontroverse näher einzugehen (vgl. hierzu FREKSA 1988), ist jedoch festzustellen, daß eine allgemeine Übereinstimmung darin besteht, daß in den meisten Gebieten zumindest zum Teil prozedurale Wissensdarstellungsarten benötigt werden (RICH 1988, S. 218).

In wissensbasierten Systemen wird folgerichtig häufig von einer prozeduralen Komponente (SCHNUPP und NGUYEN HUU 1987, S. 153) gesprochen. Diese bezieht sich dann entweder auf die verschiedenen Arten der Schlußfolgerungsmechanismen oder auf die im vorigen Abschnitt beschriebenen "prozeduralen Anhängsel" der Frames.

Eine Wissensdarstellung mit Hilfe numerischer Modelle wird zwar als wichtiger Bestandteil eines Expertensystems betrachtet, gleichzeitig aber wird betont, daß die Entwicklung solcher Modelle nicht zum Gebiet der Expertensysteme gehört (TANIMOTO 1990, S. 526). Auf die nutzbringende Kopplung zwischen Simulation und Expertensystem wird allerdings besonders in neuerer Zeit verstärkt hingewiesen. In Kapitel 2.2.5.2 wird auf diesen Aspekt näher eingegangen.

#### **2.2.2.2.6 Sonstige**

Weitere Wissensrepräsentationsformen lassen sich auf die oben beschriebenen Ansätze zurückführen bzw. mit ihnen umschreiben. TANIMOTO (1990, S. 118ff) definiert Konzepthierarchien (is-a-Hierarchien), die ohne weiteres als ein Bestandteil des Frame-Ansatzes angesehen werden können. Ebenso können scripts (Drehbücher) als eine auf Ereignisse spezialisierte Art der Frames bezeichnet werden (vgl. RICH 1988, S. 252). Constraints, die das Wissen durch Auswertung von Bedingungen repräsentieren, stellen eine besondere Form von objektartigen Beschreibungen in Kombination mit Prozeduren dar (vgl. SCHNUPP und NGUYEN HUU 1987, S. 171ff; TANIMOTO 1990, S. 153f).

Wiederum andere Ansätze erweitern die Wissensrepräsentation in Bereiche, die normalerweise nicht mehr zum direkten Aufgabengebiet wissensbasierter Systeme gehören. Die angesprochenen Prozeduren sind ein Beispiel hierfür. Ein weiteres

Beispiel sind relationale Datenbasen, die von Seiten der Datenbanktechnik als Wissensrepräsentationsschema angeboten werden. BARR und FEIGENBAUM (1981, S. 200ff) beschreiben noch die direct representation, die eine Klasse von Repräsentationsschemata darstellt, welche Wissen mit Hilfe von Diagrammen, Karten, Modellen o.ä. auf besonders natürliche Art und Weise repräsentiert.

### 2.2.23 Wissensakquisition

Liegen noch keine strukturierten Wissensbeschreibungen vor, muß das Expertenwissen möglichst unbeeinflußt von Befragungs- und Beobachtungsmethoden erfaßt werden. Hierbei werden Methoden aus der Kognitionswissenschaft eingesetzt (oder sollten eingesetzt werden), die folgendermaßen klassifiziert werden können (zur näheren Beschreibung siehe DIEDERICH 1987, S. 16ff; RADIG 1987, S. 74ff; GREENWELL 1988, S. 44ff):

- Interviewtechniken
- Protokollanalysen von Dialog- oder Lösungsprotokollen (lautes Denken)
- textanalytische Verfahren
- Skalierungstechniken
- Strukturlege-Techniken, Graphengenerierung

Neuere Entwicklungen versuchen, diese Verfahren zu automatisieren. In dem hybriden Wissensakquisitionssystem KRITON wird beispielsweise mit Hilfe des Konstruktgitterverfahrens die Wissensakquisition automatisch durchgeführt (DIEDERICH 1987, S. 16ff): Ein Experte muß aus einem Objekttripel eine Eigenschaft benennen, die zwei dieser Tripel teilen, das dritte Element aber nicht besitzt. Durch ständige Variation der Tripelzusammensetzung können Objektbäume mit ihren taxonomischen Relationen erzeugt werden, die nach weiterer Bearbeitung zur Generierung von Regeln einer Wissensbasis eingesetzt werden.

Andere Systeme legen den Schwerpunkt auf die Visualisierung der Beziehungen zwischen einzelnen Wissensobjekten, d.h. eine EDV-gestützte Entscheidungsbaumentwicklung. Wiederum andere beschäftigen sich mehr mit der Überprüfung und Weiterverarbeitung vorhandener Wissensmodelle. Induktive Systeme leiten aus Fallbeispielen oder Entscheidungstabellen optimale Regelbäume ab. Wichtige Algorithmen zur Entscheidungsbaumgenerierung sind AQ, CN2 oder ID3 (vgl. KAEMENA 1990; MARCUS 1988; STENDER 1988). Einen guten und umfassenden Überblick über die entwickelten EDV-gestützten Wissensakquisitionshilfsmittel gibt BOOSE (1988).

Eine über die automatisierte Wissensakquisition hinausgehende Erhebung und Formalisierung von Expertenwissen wird schon seit einigen Jahren im Bereich des maschinellen Lernens diskutiert (vgl. HABEL und ROLLINGER 1985, S. 249f). Sie scheint sich jedoch noch nicht wesentlich auf die Entwicklung wissensbasierter Modelle ausgewirkt zu haben (COY und BONSIEPEN 1989, S. 60). Lediglich bei der Analyse großer Datenmengen sind mit Hilfe konnektionistischer Modelle (Modelle der Parallelverarbeitung in der KI) und neuronaler Netze eventuell lernende Systeme zu entwickeln. Lernen bedeutet hier aber mehr Hilfestellung zur Strukturierung und Interpretation von Massendaten (Ergänzungstechnologie) (ULTSCH et al. 1991; DZAEBEL 1990).

#### **2.2.2.4 Werkzeuge**

Zur Erstellung von wissensbasierten Systemen wurden in den letzten Jahren zahlreiche spezielle Softwarewerkzeuge (Tools) entwickelt. Während frühere Werkzeuge spezielle Rechnerarchitekturen benötigten, kommen heute zunehmend Tools zum Einsatz, die auf Personal-Computern oder Workstations lauffähig sind. Es lassen sich zwei Richtungen erkennen.

- Der Einsatz von höheren Programmiersprachen
- Der Einsatz von Expertensystemshells

Welches Tool bei welcher Anwendung eingesetzt wird, hängt von der jeweiligen spezifischen Situation ab. Nach DILGER (1991) sind die wichtigsten Auswahlkriterien: Maximale Größe der Wissensbasis, Offenheit der Architektur, Einarbeitungszeit, Unterstützung der Oberflächengestaltung und kommerzielle Gesichtspunkte.

Nach ALTENKRÜGER (1987, S. 23ff) kommen auf der Ebene der Programmiersprachen hauptsächlich die Sprachen Lisp, Prolog und Smalltalk zum Einsatz. Lisp ist die 'Muttersprache' der KI. Sie ist ein typischer Vertreter der funktionalen Programmiersprachen. Zentrale Wissensrepräsentationsform ist die Liste. Es lassen sich leicht komplexe Listenstrukturen (Bäume, Graphen) programmieren. Prolog baut auf dem Prädikatenkalkül erster Ordnung auf. Es zählt zu der Gruppe der logischen Programmiersprachen. Der besondere Vorteil von Prolog liegt darin, daß ein in der Sprache implementierter Such- und Inferenzmechanismus ausgenutzt werden kann. Hinzu kommt, daß Operationen wie Backtracking und Cut (Beschneiden des Suchraumes) standardmäßig zur Verfügung gestellt werden. Smalltalk ist ein typischer Vertreter objekt-orientierter Sprachen. Es können direkt Objekte, Klassen, und Hierarchien angegeben werden. Verer-

bungsstrategien und Objektkapselung wird auf allen Ebenen konsequent angewandt. Eine hervorragende Grafikoberfläche steht zur Verfügung.

Shells stellen demgegenüber mächtigere Entwicklungswerkzeuge dar. Durch vorgefertigte Benutzerschnittstellen und Inferenzmechanismen liegt praktisch ein leeres Expertensystem vor. Im Gegensatz zu Programmiersprachen ist für jedes herausgegebene System eine run-time-Gebühr zu entrichten. DILGER (1991) unterscheidet 4 Gruppen:

1. Allgemeine Shells mit eingeschränktem Leistungsumfang. Es wird nur eine Wissensrepräsentationsform zur Verfügung gestellt (meistens Regeln). Beispiele: EMYCIN, XiPlus, Rulemaster, OPSS, Level 5.
2. Allgemeine hybride Werkzeuge: Sie stellen mehrere Wissensrepräsentationsformalismen zur Verfügung. Beispiele: KEE, ART, Knowledge Craft, Nexpert Objekt, Babylon.
3. Problemspezifische Werkzeuge: Sie stellen problemspezifische Lösungskomponenten zur Verfügung (d.h., es existiert eine auf einen Problemtyp abgestimmte Repräsentationsform). Beispiel: MED2.
4. Aufgabenspezifische Werkzeuge: Sie enthalten neben den problemspezifischen Lösungskomponenten auch noch aufgabenspezifisches Basiswissen in der Wissensbasis. DILGER (1991) nennt als Beispiel SYNTEL und Financial Advisor.

Weitere Ausführungen zu den angeführten Werkzeugen, insbesondere der Vergleich einzelner Tools und Sprachen, sind bei KARRAS et al. (1987), HARMON und KING (1989) und ALTENKRÜGER (1987) zu finden. BOSCH (1988) veröffentlichte die Ergebnisse einer Umfrage bei Anwendern von KI-Programmiersprachen und Shells in Industrie und Wirtschaft über Vor- und Nachteile der verwendeten Tools. Einen umfassenden, wenn auch nicht ganz aktuellen, Überblick über verfügbare Tools im Bereich der Expertensysteme gibt BUNDY (1986).

### 2.2.3 Einsatz arithmetischer Gleichungen

Durch die Beschreibung formalisierten Wissens mit Hilfe von arithmetischen Gleichungen und Ungleichungen werden traditionell mathematische Modelle gebildet. Im Bereich des Gartenbaus werden häufig Polynome, Exponential-, Potenz- und Wurzelfunktionen, sowie logarithmische und trigonometrische Funktionssterme zur Modellbildung eingesetzt (z.B. BAZLEN 1985; VON ELSNER 1982). Besonders verbreitet ist die Darstellung von Änderungen innerhalb eines bestimmten Zeitinkrementes mit Gleichungen, die neben einer oder mehreren Variablen auch Funktionen dieser Variablen und deren Ableitungen enthalten (Differentialgleichungen). Durch die Verknüpfung mehrerer Gleichungen entstehen Gleichungssysteme, die mit Hilfe der Verfahren der Analysis und der numerischen Mathematik gelöst werden.

Durch Simulations- und Optimierungsrechnungen werden aus den arithmetischen Modellen Informationen gewonnen. Hierzu werden in der Regel deterministische Algorithmen formuliert, bei denen zu jedem Zeitpunkt der Ausführung höchstens eine Fortsetzungsmöglichkeit besteht (ENGESSER 1988, S. 169). Es lassen sich zwei Simulationstypen unterscheiden (vgl. NIEMEYER 1977, S. 67ff; ROTHENBERG 1989, S. 83f; WIDMAN und LOPARO 1989, S. 16) :

a. Stromorientierte Simulationsmodelle (continuous simulation), bei denen Systeme durch zeitbezogene Differential- bzw. Differenzenquotienten dargestellt werden. Eine allgemeine Beschreibung auf der Basis von Differenzenquotienten gibt NIEMEYER (1977, S. 67f) mit

$$\frac{\Delta x_i}{\Delta t} = f_i(t_1, x_1, \dots, x_i, \dots, x_{n_i}), \quad (i = 1, \dots, m) \quad (2.2-13)$$

an. Den Wert der Zustandsvariablen  $x_i$  zum Zeitpunkt  $t$  erhält man mit

$$x_i(t) = x_i(t_0) + \sum_{j=t_0}^t \frac{\Delta x_i(j)}{\Delta t} \cdot \Delta t \quad (i = 1, \dots, m) \quad (2.2-14)$$

Es gilt:

- $x_i(t)$  = Wert der Zustandsvariable im Zeitpunkt  $t$
- $f_i$  = arithmetische oder logische Funktion
- $t_1$  = Zeitpunkt  $\leq t$
- $n_i$  = Anzahl der unabhängigen Variablen des Systems und der Umwelt (in Bezug auf  $x_i$ )
- $m$  = Anzahl der abhängigen Variablen des Systems. Dabei gilt  $m \leq n_i$  für alle  $i = 1, \dots, m$ .

b. Ereignisorientierte Simulationsmodelle, welche entweder durch einen diskreten Zeittakt (time based) oder durch Ereignisse gesteuert werden (event based). Allgemein gilt (NIEMEYER 1977, S. 68)

$$Z(t) = f(Z(t_1), G(t_1), E(t_1)) \quad (2.2-15)$$

Darin bedeuten:

$Z(t)$  = Zustand (Vektor der Ausprägungen der Zustandsvariablen) des Modells im Zeitpunkt  $t$

$Z(t_1)$  = Zustand des Modells im Zeitpunkt  $t_1 \leq t$

$G(t_1)$  = Menge der im Zeitpunkt  $t_1$  angesprochenen Zustandsgeneratoren

$E(t_1)$  = Menge der im Zeitpunkt  $t_1$  eintretenden Ereignisse (Anstöße der Zustandsgeneratoren).

Optimierungsrechnungen sind gekennzeichnet durch die Suche nach Extremwerten in einem Problemlösungsraum. Handelt es sich bei der Problembeschreibung um differenzierbare Funktionen, können mit Hilfe der Differentialrechnung Maximal- und Minimalwerte bestimmt werden. Werden für eine lineare Funktion mit endlich vielen linearen Nebenbedingungen Minima oder Maxima gesucht, werden die Verfahren der linearen Optimierung (Simplexalgorithmen) eingesetzt (BRONSTEIN und SEMENDJAJEW 1987, S. 237ff, S. 695ff). Für komplexere arithmetische Modelle sind häufig keine analytischen Optimierungsverfahren vorhanden. In diesen Fällen muß durch wiederholtes Durchrechnen der Modelle mit unterschiedlichen Eingangswerten versucht werden, optimale oder annähernd optimale Lösungen zu finden (numerische Optimierung).

Für die Repräsentation, Entwicklung und Darstellung der in diesem Kapitel beschriebenen Modellansätze stehen unterschiedliche Softwarewerkzeuge zur Verfügung. Vielfach werden imperative Programmiersprachen der dritten Generation wie Fortran, Cobol, Algol, C, Simula, Pascal, Basic, Ada oder Modula eingesetzt. Sie stellen die klassischen "Allzweckprogrammiersprachen" dar, die durch Strukturierung und Kontrollanweisungen geprägt sind (MOSER 1991).

Leistungsfähige Tools verbessern die Einsatzmöglichkeiten dieser Werkzeuge beträchtlich. Neuere Entwicklungen bieten objektorientierte Erweiterungen an (z.B. C  $\rightarrow$  C++, Turbo Pascal  $\rightarrow$  Turbo Pascal 5.5). Sprachen, die speziell die an der Modellbildung beteiligten Objekte in den Vordergrund stellen (z.B. Eiffel, Smalltalk), scheinen vermehrt Zuspruch zu finden (SAGER 1991).

Neben den direkten Programmiersprachen werden komplette Programmpakete zur Entwicklung und Darstellung mathematischer Modelle eingesetzt. Im we-

sentlichen sind dies Tabellenkalkulationsprogramme (LOTUS 123, QUATRO-PRO,...), statistische Programmpakete (SPSS, NCSS...), spezielle Simulatoren (GPSS, CSMP, SLAM ...) und Optimierungssysteme (EXTREM, EVOL...). Nähere Hinweise finden sich z.B. bei O'KEEFE (1989), SCHMIDT (1985), SAGER (1991), MOSER (1991), LENTZ (1987) und TELLO (1989).

#### **2.2.4 Einsatz von Datenbankkonzeptionen**

Die Modellierung eines Realitätsausschnittes setzt vielfach die Sammlung, Auswertung und Manipulation von Informationen voraus, die durch komplex strukturierte Daten repräsentiert werden (z.B. Datensammlungen, Katalogeinträge etc.). Besonders bei der Verwendung von Massendaten kommen dabei EDV-gestützte Ansätze aus dem Datenbankbereich zum Einsatz. Zahlreiche Beispiele aus dem Agrarbereich geben MAINKA und MANGSTL (1991). Heutige Konzepte bauen vielfach auf einem Entity-Relationship-Modell auf (SCHLAGETER 1983). Die Modellbildung besteht darin, daß die zu beschreibenden Dinge der Welt als Objekte (Entities) angesehen werden, die Elemente bestimmter Mengen (Entity-Typen) sind (Beispiel: Entity-Typ = Pflanze, Entity = Rose). Durch Beschreibung von Beziehungen werden einzelne Entities in Verbindung zueinander gebracht. Attribute der Entity-Typen erweitern die Entities zu identifizierbaren Objekten. Man erkennt, daß diese Art der Wissensdarstellung eine detaillierte Modellierung aller Objekte des zu beschreibenden Realitätsausschnittes voraussetzt. Nach WINTER (1990) tritt dabei immer mehr die "Abstraktion" als Dreh- und Angelpunkt hervor. Daten werden als abstraktes Modell der Wirklichkeit betrachtet.

Die Umsetzung konzeptueller Schemata in konkrete Datenbankanwendungen erfolgt mit bestimmten Datenbankformen. Sie können als eine sehr direkte Art der Wissensrepräsentation verstanden werden. Die wichtigsten sind (vgl. SPRAGUE und CARLSON 1982, S. 227ff; DIEMER 1989, S. 8):

- Records,
- Hierarchien,
- Netze,
- Relationen.

Das älteste Datenmodell ist die Abspeicherung von Datensätzen in Recordform. Eine Verknüpfung einzelner Records erfolgt in der Regel nicht. In einer hierarchischen Datenbank weist demgegenüber eine Objektklasse Verbindungen zu untergeordneten Objektklassen auf (Baumstruktur). Netzorientierte Datenmodel-

le erlauben, daß ein Objekt mehrere übergeordnete Objektklassen beinhalten kann. Somit sind mehrere Suchpfade zu einem Objekt vorhanden. Das modernste und z. Zt. eindeutig favorisierte Datenbankkonzept ist die Darstellung mit relationalen Modellen.

Eine Relation wird mathematisch folgendermaßen definiert (nach SCHLAGETER 1983, S. 3):

Sind  $E_1, E_2, \dots, E_n$  Mengen von Werten, so ist  $R \subseteq E_1 \times E_2 \times \dots \times E_n$  eine  $n$ -stellige Relation über den Mengen  $E_1, E_2, \dots, E_n$ . Ein Element  $r = (e_1, e_2, \dots, e_n) \in R$  ( $e_i \in E_i, i = 1, \dots, n$ ) ist ein Tupel der Relation  $R$  ( $n$ -Tupel).

Eine Relation kann man somit als eine Tabelle betrachten, in der jede Zeile einem Tupel der Relation entspricht.

Wichtige Aspekte bei der Entwicklung eines relationalen Datenmodells sind verschiedene Formen der Tabellennormalisierung und der Schlüsselzuweisung. Unter Normalisierung versteht man den Prozeß, aus Relationen Tabellen zu erstellen, wo kein Datensatz sich wiederholende Gruppen enthält und jedes Datenfeld von einem einzelnen Schlüssel abhängt.

Informationen oder auch Wissen erhält man aus einem relationalen Datenmodell, indem grundlegende relationale Operationen auf die verschiedenen Tabellen ausgeführt werden. Die wichtigsten sind (DIEMER 1989, S. 21):

- Selektion: Erzeugt eine Untermenge aller Zeilen in eine Tabelle
- Projektion: Erzeugt eine Untermenge aller Spalten in eine Tabelle
- Verbindung: Verknüpft zwei Tabellen zu einer neuen Tabelle.

Es bleibt anzumerken, daß bei heutigen relationalen Datenbanken eine strikte Trennung zwischen dem Wissen (Datenbankrelation) und den Auswerte- und Abarbeitungsmechanismen besteht. Nicht zuletzt auch aus diesem Grund werden relationale Datenbasen auch als Wissensrepräsentationsform in der KI bezeichnet. TANIMOTO (1990, S. 156) schreibt dazu: "Wir würden weder der KI noch dem Gebiet der Datenbanksysteme gerecht, wenn wir relationale Datenbasen bei einer ernsthaften Untersuchung der Methoden zur Wissensdarstellung übergehen würden". MÜLLER (1989) zeigt die Entwicklung einer Wissensbasis mit einem relationalen Datenmodell an einem konkreten Beispiel auf.

Zunehmend werden auch nicht datenorientierte Informationselemente als Modellierungsobjekte innerhalb der EDV betrachtet. So beschreibt RADIG (1987, S. 9) Datenbanksysteme nur als eine Komponente der Informationsverarbeitung. Besondere Beachtung finden in letzter Zeit Hypertextsysteme, bei denen Dokumente durch maschinenunterstützte Verbindungen (links) eine logische Verar-



beitung ermöglichen (NASTANSKY und SEIDENSTICKER 1990). Ein erfolgreiches Beispiel zeigen RÖHRICH und TAEGER (1991). Durch zunehmende Rechnerkapazitäten können solche Konzepte zu Hypermedia (Multimedia)-Systemen erweitert werden, die Graphik, Sprache, Bilder, Musik, Videos und Animation einbeziehen (ZAHN 1991). Da solche Konzepte von der Anwenderseite gefordert werden (siehe GÖHLER 1990, S. 54), stehen wir hier mit Sicherheit erst am Anfang einer innovativen Entwicklung. Erste Implementierungen, wie z.B. das multimediale Expertensystem MUTEX (ZIEGLER und KOLLER 1991), zeigen die prinzipielle Machbarkeit und die immensen Möglichkeiten solcher Systeme.

## **2.2.5 Kombination verschiedener Ansätze**

### **2.2.5.1 Allgemeines**

Die letzten Kapitel haben gezeigt, wie unterschiedlich (manchmal auch wie ähnlich) Wissensrepräsentation aus den verschiedenen Blickwinkeln heraus sein kann. Vielfach wird gefordert, mehrere Ansätze zu nutzen und zu einem effektiven Gesamtkonzept zusammenzuführen (RADERMACHER 1988). Im folgenden werden die Möglichkeiten aufgezeigt, die in der Literatur beschrieben werden, um gekoppelte bzw. hybride Systeme zu erhalten. Auf die Beschreibung der Zusammenführung der drei beschriebenen Ansätze zu einem System wird verzichtet. Hier sei auf die methodisch ausgerichtete Arbeit von KARAGIANNIS (1987) verwiesen. Eine mehr anwendungsorientierte Diskussion findet man bei TURBAN und WATKINS (1986). Anwendungsbeispiele werden bei JONES et al. (1987) und MEYER M. (1989) gegeben.

Ebenso wird auf die Kopplung zwischen Datenbanksystemen und prozeduralen Auswertelgorithmen aufgrund des thematischen Schwerpunktes der Arbeit nicht eingegangen. Umfangreiche Literatur zu diesem Aspekt existiert im Bereich der decision support systems (vgl. SPRAGUE und WATSON 1989).

### **2.2.5.2 Wissensbasierte Systeme und arithmetische Modelle**

Die Verknüpfung von wissensbasierten Systemen mit arithmetischen Modellen kann auf unterschiedliche Art und Weise und aus unterschiedlichen Gründen erfolgen. Es lassen sich folgende Kopplungsvarianten unterscheiden (FUTO und GERGELY 1990, S. 222; O'KEEFE 1989; WIDMAN und LOPARO 1989; FLITMAN

und HURRION 1987; BECK und JONES 1989; REDDY et al. 1986) :

- Das wissensbasierte Modell benötigt arithmetische Modelle, um fehlende Informationen zu erhalten. Dazu werden die arithmetischen Modelle direkt in das wissensbasierte Konzept integriert (z.B. Prozeduren innerhalb des Frame-Konzeptes), oder aber das wissensbasierte System muß eine komplette Simulationsberechnung anstoßen. Im Extremfall laufen beide Prozesse nacheinander ab.
- Das arithmetische Modell (in der Regel Simulationsmodell) benötigt von Zeit zu Zeit die Information eines wissensbasierten Modells, um arbeiten zu können. Der Einsatz des wissensbasierten Modells ist in bisherigen Entwicklungen sehr unterschiedlich. Er reicht von der logischen Modellierung der Verbindung zweier an der Simulation beteiligter Objekte bis hin zur wissensbasierten Steuerung der Simulation. Im allgemeinen wird diese Art der Kopplung als 'knowledge based simulation' bezeichnet.
- Das wissensbasierte Modell dient als 'intelligent front end'. Dabei werden konventionelle Programmkonzeptionen durch eine wissensbasierte Dialoggestaltung, durch eine Anpassung des Systems an die Benutzeranforderungen, durch eine wissensbasierte Ermittlung gewünschter Eingabedaten und durch die Interpretation und Selektion der Ausgaben verbessert.

Die technische Realisierung der beschriebenen Kopplungsvarianten ist zum Teil mit erheblichen Problemen verbunden, da häufig zwei unterschiedliche Programmierparadigmen (imperativ, logisch) verknüpft werden müssen. Es können drei grundsätzliche Möglichkeiten voneinander unterschieden werden (PINTO 1989, S. 413):

1. Das zu repräsentierende Wissen wird mit unterschiedlichen Programmen dargestellt. Die Kopplung der Systeme erfolgt durch Betriebssystemaufruf. Der Datentransfer geschieht mit Hilfe von Übergabedateien.
2. Einsatz einer hybriden Expertensystemshell, die beide Wissensrepräsentationsformen unterstützt.
3. Erweiterung des Programmierkonzeptes oder Darstellungsrahmens um die jeweils fehlende Komponente. Daraus resultiert die Darstellung des wissensbasierten Modells mit einer imperativen Programmiersprache und die Darstellung von prozedural-arithmetischen Modellen mit wissensbasierten Techniken. Besonders die letzte Variante kann leicht zu einer verfehlten Wissensreprä-

sensation führen, da versucht wird, das Wissen mit einem vom Prinzip her nicht für das Problem konzipierten Instrumentarium darzustellen (RADERMACHER 1988; als Beispiel MEYER und SCHMIDT 1991).

### 2.2.5.3 Wissensbasierte Systeme und Datenbanken

In vielen Fällen ist es sinnvoll, wissensbasierte Systeme und Datenbankkonzeptionen zu einem mächtigen Hybridsystem zusammenzuführen. Dabei ist zu bedenken, daß beide Konzepte von ihrem prinzipiellen Ansatz her nahe verwandt sind. Abbildung 2.2-6 verdeutlicht dieses an einem Vergleich zwischen relationalen Datenbanken und der Programmiersprache Prolog. Wesentliche Unterschiede ergeben sich jedoch aus den für beide Modellkonzeptionen typischen Anwendungen und aus den unterschiedlichen mathematischen Grundlagen (Prädikatenlogik, relationale Algebra).

Relationale Datenbank	Wissensbasiertes System (Prolog)
<ul style="list-style-type: none"><li>- Trennung zwischen Datenbasis und Datenbankmanager</li><li>- relational-algebr. Ausdrücke</li><li>- Relationentupel</li><li>- Relation</li><li>- Parameterübergabe</li><li>- Verbindung, Selektion</li><li>- Vereinigung</li></ul>	<ul style="list-style-type: none"><li>- Trennung zwischen Wissensbasis und Schlußfolgerungsmechanismus</li><li>- Horn-Klauseln</li><li>- Fakten (Prädikate mit Argumenten)</li><li>- Liste</li><li>- Unifikation, Matching</li><li>- Konjunktion</li><li>- Disjunktion</li></ul>

Abbildung 2.2-6: Analogien zwischen einem wissensbasierten System in Prolog und einer relationalen Datenbank (nach PREIß 1986, S. 5).

Durch die Kopplung beider Ansätze können die Vorteile jedes einzelnen Konzeptes genutzt werden. Denkbare programmiertechnische Ansätze zeigen CERI et al. (1987).

Allgemein ergeben sich bei der Verbindung von Datenbanken und wissensbasierten Systemen folgende Einsatzmöglichkeiten (REUTER 1987; PREIß und STUCKY 1988):

Das wissensbasierte System kann als intelligentes Front-End-System agieren,

welches dem Benutzer der Datenbank Hilfen bei der Formulierung bzw. beim Verständnis schwieriger Anfragen liefert.

Dem Datenbankersteller kann bei der Modellierung der Objektstrukturen (Spezifikation, Definition) durch ein wissensbasiertes System geholfen werden.

Als dritte Möglichkeit kann das Datenbankmanagement direkt unterstützt werden, indem die Überprüfung der Integritätsbedingungen, die kontrollierte Duldung einzelner Integritätsverletzungen, der Umgang mit unscharfen Daten oder Nullwerten und die direkte Problemlösung durch Inferenzprozesse mit wissensbasierten Systemen unterstützt wird.

Beispiele für bisherige Entwicklungen sind noch relativ selten. TISCHENDORF (1989) beschreibt ein System, bei dem mit Hilfe eines Regelwerkes die in einem Datenbanksystem abzuspeichernden Daten (Meßwerte) auf Plausibilität hin überprüft werden. FISCHER und MENGES (1989) benutzen ein Expertensystem um Datenbankabfragen zu formulieren. GIBBERT (1991) nutzt einen Frameansatz, um Trends in einer On-Line-Datenbank schneller erkennen zu können.

Die Erweiterung von wissensbasierten Systemen um Datenbankkonzepte ermöglicht die leichte Veränderung und Erweiterung der im System benötigten Daten sowohl vom Systemersteller als auch vom Systemanwender. Ferner können die von der Datenbanktechnik zur Verfügung gestellten Mechanismen zur Massendatenhaltung (Datensicherheit, Datenzugriffsmöglichkeiten etc.) optimal und ökonomisch genutzt werden. Es stehen Erfahrungen aus vielen Projekten zur Verfügung. Beispiele aus dem Personal-Computer-Bereich liefern GÖHLER (1990) und PIGORSCH (1991).

Die technische Realisierung der beschriebenen Kopplungsvarianten erfolgt heute hauptsächlich nach drei Verfahren (REUTER 1987):

- Der heterogene Ansatz: Expertensystem und Datenbank sind separate Komponenten, die über explizite Dienstanfragen miteinander verkehren.
- Der homogene Ansatz: Das Expertensystem wird mit den erforderlichen Datenverwaltungsfähigkeiten ausgestattet.
- Der integrierte Ansatz: Eine Datenbank wird um die Expertensystemtechnologie erweitert (Expert Database System). Nach TISCHENDORF (1989) ist mit allgemein verfügbaren Systemen dieser Art jedoch erst ab Mitte der 90er Jahre zu rechnen.

### 3 Das hybride System HORTEX

#### 3.1 Allgemeines

##### 3.1.1 Motivation, Zielsetzung, Anforderungen

Das Expertensystem HORTEX (horticultural technical expert system) wurde in den Jahren 1989 - 1991 im Rahmen eines vom Bundesministerium für Ernährung, Landwirtschaft und Forsten geförderten Forschungsvorhabens am Institut für Technik in Gartenbau und Landwirtschaft der Universität Hannover entwickelt. Es ist ein Beratungssystem, welches sich mit der Planung der Heizenergieversorgung von Gewächshäusern beschäftigt.

Die Auswahl des thematischen Schwerpunktes erfolgte aufgrund der Tatsache, daß am durchführenden Institut seit vielen Jahren an dem Themenbereich Gewächshausenergieversorgung und Gewächshaus-Heizungstechnik gearbeitet wird und eine Menge Informationen in personifizierter und literarischer Form vorliegt. Ferner ergaben Umfragen unter den bundesdeutschen Gartenbautechnikberatern, daß fachspezifische Software fehlt und ein großer Bedarf an entscheidungsunterstützenden Systemen im Bereich der Gewächshausheizung besteht (Abbildung 3.1-1).

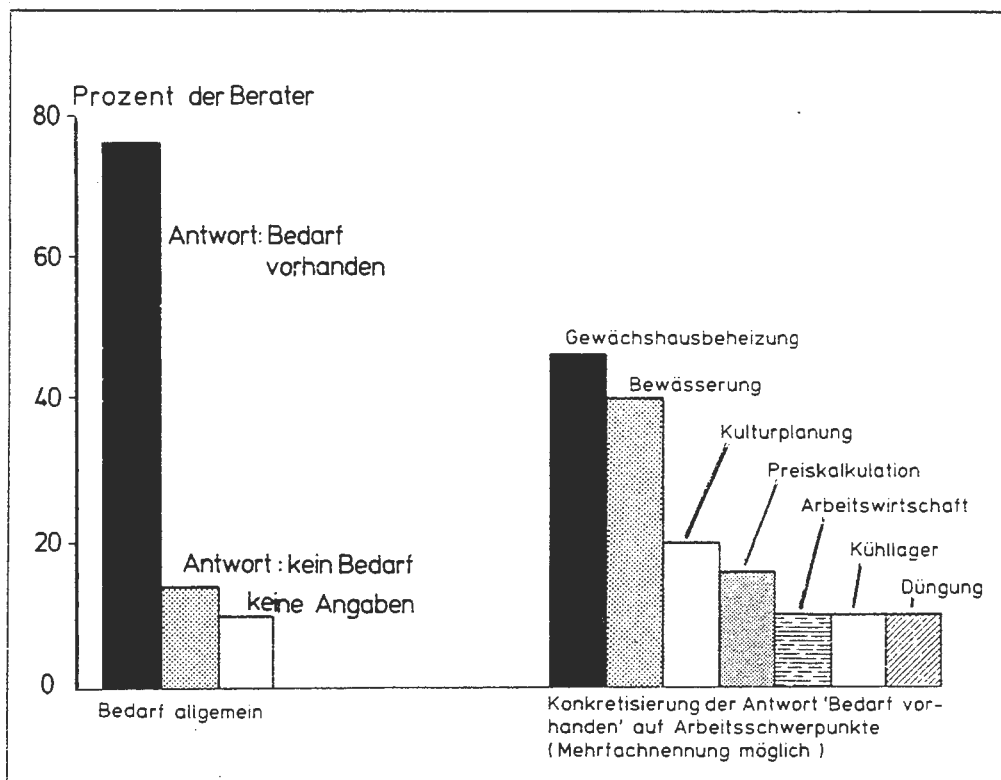


Abbildung 3.1-1: Bedarf an entscheidungsunterstützenden EDV-Systemen in der bundesdeutschen Gartenbautechnikberatung (RATH 1990).

Mit der Programmentwicklung wurden hauptsächlich drei Ziele verfolgt:

- Die inhaltlichen Anforderungen der Aufgabenstellung sollten mit einem Computerprogramm gelöst werden. Insbesondere sollte eine adäquate Modellierung und Darstellung des vorhandenen Wissens erfolgen.
- Der Einsatz der Expertensystemtechnologie im gartenbautechnischen Aufgabenbereich sollte untersucht und eventuell auf die konkrete Anwendung hin modifiziert oder adaptiert werden.
- Es sollte eine Software konzipiert werden, die auch im finanzschwachen (im Vergleich zu anderen Industriesparten) Gartenbau eingesetzt werden kann.

Eine Spezifizierung des Kreises der Endbenutzer erfolgte nicht. Im allgemeinen wurde von Nicht-Computer-Spezialisten ausgegangen, die eine gewisse Fachkenntnis besitzen. Besondere Bedeutung wurde der Gruppe der Gartenbautechnikberater zugewiesen.

Um die speziellen Anforderungen (z.B. Hardware, Software etc.) der gartenbautechnischen Praxis und Beratung zu erfassen, wurde zu einem frühen Projektzeitpunkt eine Umfrage unter den deutschen Gartenbautechnikberatern durchgeführt (RATH 1990). Aufgrund der Ergebnisse dieser Befragung wurde bei der Entwicklung und Konzeption des Systems HORTEX von folgenden Anforderungen ausgegangen:

Das System soll

- entscheidungsunterstützenden Charakter haben. Das heißt, es soll in einer konkreten Aufgabensituation Hinweise zur Lösung eines Problems geben. Hierbei sollen auch bewußt vom Nutzer nicht unmittelbar gewünschte Handlungsalternativen aufgezeigt werden.
- modular aufgebaut sein, damit eine Weiterentwicklung und -nutzung sowie eine methodische Übertragung auf ähnlich gelagerte Aufgaben möglich ist.
- eine gute und einfache Benutzerführung haben, damit auch Personen, die nur einen punktuellen EDV-Einsatz haben, das Programm nutzen können. Trotzdem sollte eine Anpassung des Systems an situationsspezifische Gegebenheiten möglich sein.
- mit möglichst geringen run-time-Gebühren (Lizenzgebühren von Seiten Dritter) behaftet sein.
- auf einem konventionellen Personalcomputer unter dem Betriebssystem DOS lauffähig sein.

### 3.1.2 Bisherige Arbeiten und einsetzbare Systeme

Wissensbasierte Systeme, die für den Bereich der Gewächshausheizung bzw. -energieversorgung konzipiert wurden, existieren zum gegenwärtigen Zeitpunkt noch nicht. Die in Kapitel 2.1.3 aufgeführten klimatechnischen Expertensysteme lassen sich nicht oder nur sehr begrenzt auf den Gewächshausbereich übertragen, da insbesondere in Bezug auf die Wärmeausbringung nur eine sehr geringe Parallelität zwischen Gewächshaus und Wohngebäude besteht. Ähnlich sieht es im Bereich der konventionellen Energieberatungssoftware aus. Durch die speziellen Anforderungen des Gartenbaues hinsichtlich der Gewächshausklimatisierung (Lüftung, Energieschirm, Belichtung, Regelstrategie etc.) ist der Einsatz vorhandener, auf die Gebäudetechnik zugeschnittener Software kaum möglich. (HERBERHOLZ (1991) schreibt, daß selbst in der Gebäudeheiztechnik die Anwendung verfügbarer Energie-Tools sehr schleppend verläuft).

Konventionelle Software, die für die Auslegung einzelner Heizungsanlagenkomponenten entwickelt wurde und in fast allen größeren Herstellerfirmen vorhanden ist (z.B. Pumpenauswahl, Kesselberechnungen, Konvektorauslegung o.ä.), kann zum Teil ohne weiteres auf heiztechnische Planungsfragen im Gartenbau übertragen werden. Auf eine Duplizierung dieser Systeme im Rahmen der hier vorgestellten Forschungsarbeiten wurde verzichtet.

Abbildung 3.1-2 gibt einen Einblick in die zur Zeit auf dem Personal-Computerssektor kommerziell verfügbaren heiztechnischen Planungs- und Beratungsprogramme. Die häufig von der Heizungstechnikindustrie in eigener Regie und auf ihre eigenen Produkte abgestimmten Systeme sind nicht aufgelistet. Es bleibt anzumerken, daß es sich bei den in der Abbildung 3.1-2 dargestellten Programmen um konventionelle Auslegungssoftware handelt, die keine Expertensystemtechnologie beinhaltet (zweifelsohne aber Fachwissen repräsentiert).

Programmbezeichnung Vertrieb Aufgabenbereich	Energieberatung, Wärmeschutz	Wärmebedarfsberechnung	BHKW-Berechnung	Rohrnetzberechnung	k-Wert Berechnung	Heizflächenberechnung	Kühllastberechnung	Schornsteinberechnung	Fußbodenheizungsberechnung	Pumpenberechnung	Wärmepelcherberechnung	Isolierungsberechnung	Armaturenberechnung	Ventilauslegung	Blendenberechnung
IBM PC Haustechn. MW Software	X	X		X	X	X			X	X	X	X	X		
Thermosoft ALBO-DATA	X	X		X	X	X									
label Gerald Box	X	X		X	X	X			X				X		
LI MW Software							X								
DISIG-X PDS	X	X		X	X	X			X						
ETU ETU-Datentechnik		X		X	X	X	X		X						
EDS Saß							X								
HATE Sykapan	X	X			X	X			X						
CEB-V Solar Computer	X	X	X		X										
Progr. - Paket RESCH	X	X	X	X	X			X	X					X	
P4705 Stegemann								X							

Abbildung 3.1-2: Kommerziell verfügbare Software für die Heiz- und Klimatechnik (Angaben zu den Firmen in Kapitel 6)



### 3.1.3 Wissensakquisition

Das Expertensystem HORTEX wurde auf der Grundlage von Expertenbefragungen und Literaturstudien erstellt. Es wurden sechs gezielte Expertenbefragungen durchgeführt, die jeweils zwischen zwei und fünf Stunden dauerten. Fünf der sechs Befragungen bezogen sich auf die Energiebereitstellung - sie dienten z.T. aber auch als Einstieg in die Gesamthematik. Eine Befragung bezog sich vornehmlich auf die Energieausbringung (Heizungssysteme). Folgende Personen wurden befragt:

- Prof. Dr. H.-J. Tantau
  - Prof. Dr.-Ing. C. von Zabeltitz
  - Dr.-Ing. B. von Elsner
  - Dipl.-Ing. O. Domke,
  - Dipl.-Ing. W. Gabloffsky,
- } Mitarbeiter des Institutes für Technik in  
Gartenbau und Landwirtschaft, Universität  
Hannover
- Technikberater Landwirtschaftskammer Rheinland  
Technikberater Landwirtschaftskammer Hannover

Sämtliche Befragungen wurden auf Band mitgeschnitten und anschließend analysiert. Die Befragung erfolgte (möglichst) nach fest definierten Vorgaben. Im einzelnen wurden folgende Erhebungen durchgeführt (zu den Oberbegriffen vgl. Kap. 2.2.2.3 und RADIG 1987, S. 74ff) :

#### Horizontale und vertikale Wissenserhebung durch Interview

- A: Beschreibung wichtiger Kriterien für die Auswahl eines Energieträgers (keine weiteren Vorgaben).
- B: Bewertung einzelner Energieträger für die Gewächshausbeheizung. Vorgabe: Öl, Gas, Kohle, Fernwärme, Umweltwärme, Holz, Stroh, Biogas, Solarenergie, BHKW.
- C: Abgrenzung und Bedeutung einzelner Energieträgerbewertungskriterien. Vorgabe: Energieträgerangebot, Lieferbedingungen, Pflanzenbau, Betriebssituation, Wartung, Arbeitsaufwand, rechtliche Auflagen, technische Kriterien.
- D: Planungsrichtlinien bei der Planung der Energieversorgung von Gewächshäusern (keine weiteren Vorgaben).
- E: Mögliche Energieträgerkombinationen und deren Leistungsaufteilung (keine weiteren Vorgaben) sowie die Einsatzmöglichkeit eines Wärmespeichers.
- F: Bewertung und Einsatzmöglichkeit von Energieträgerkombinationen anhand einer Kombinationsmatrix.
- G: Charakteristische Eigenschaften von Heizungssystemen. Vorgabe: Wärmeausbringung, pflanzenbauliche Verwendung, technische Realisierbarkeit, Anbindung an die Heizenergiebereitstellung.
- H: Kombination mehrerer Heizungssysteme und Auslegungsrichtlinien. Vorgabe:

Kombinationen, die sich ausschließen, Standardkombinationen, Ausnahmekombinationen.

#### Strukturerhebung durch Interview

I: Auflistung der wesentlichen technischen und ökonomisch relevanten Bauteile einer Energieversorgungsanlage. Vorgabe: Die unter B aufgeführten Energieträger.

J: Einteilung verschiedener Heizungssysteme in Klassenhierarchien.

#### Lösungsprotokolle

K: Angabe mehrerer Planungs-/Beratungssituationen im Bereich Energieträgerauswahl. Lösen der Aufgabe/Planung durch "lautes Denken".

L: Lösung mehrerer vorgegebener Planungsfälle im Bereich Heizungssysteme und Wärmeausbringung.

#### Skalierung

M: Bewertung von Energieträgern und Energieträgerkombinationen mit einer numerischen Bewertungsskala jeweils bezogen auf die Kriterien Umweltschutz, Versorgungssicherheit, Verfügbarkeit, technische Realisierbarkeit und Betriebssicherheit (mit Begründung).

#### Ord nende Rückschau

N: Bewertung und Veränderung vorgelegter, aus Literatur und Firmenangaben resultierender Investitionskostentabellen.

O: Überprüfung wesentlicher Planungsregeln durch Bejahung oder Verneinung vorgestellter Aussagen im Bereich Energieträgerauswahl.

Abbildung 3.1-3 gibt den Befragungsumfang der einzelnen Interviews wieder. Die bei der Umsetzung der Befragungsinhalte in das Computerprogramm auftretenden Wissenslücken, Interpretationsmöglichkeiten und Begriffsunklarheiten wurden jeweils bei Auftreten direkt mit Experten des Institutes abgeklärt. Diese 'punktuellen Befragungen' sind in Abbildung 3.1-3 nicht enthalten. Sie stellen aber mit Sicherheit eine entscheidende 'Wissensquelle' des Programms dar.

Neben den Expertenbefragungen wurde auch auf Literaturangaben zurückgegriffen. Hauptquellen waren hierbei die Fachbücher "Heizungsanlagen im Gartenbau" (TANTAU 1983), "Gewächshäuser" (VON ZABELTITZ 1986), "Taschenbuch für Heizung und Klimatechnik" (HÖNMANN und SPRENGER 1985) sowie Fachartikel, DIN-Normen und VDI-Richtlinien. Tauchten Widersprüche zwischen Ex-

pertenbefragungen und Literaturrecherchen auf, kam den punktuellen Expertenbefragungen die höchste Priorität zu. Da keinem der befragten Experten eine fachliche Priorität zugeordnet wurde, mußten widersprüchliche oder differierende Angaben durch weitere Diskussionen (wahrscheinlich aber auch durch nicht beabsichtigte, unterbewußte persönliche Bewertung) aufgelöst werden.

Befragungsschwerpunkt	v. Elsner 14.7.1989	Tantau 19.7.1989	v. Zabeltitz 28.7.1989	Gablofsky 4.5.1990	Domke 18.5.1990	Tantau 22.11.1990
A	X	X				
B	X	X	X	X	X	
C	X		X			
D	X	X	X			
E	X			X	X	
F		X				
G						X
H						X
I	X					
J						X
K				X	X	
L						X
M				X	X	
N				X	X	
O			X			

Abbildung 3.1-3: Expertenbefragungen für das Expertensystem HORTEX

## 3.2 Beschreibung des Systems

### 3.2.1 Modularisierung der Aufgabenstellung

Die unter 3.1.1 beschriebene Aufgabenstellung wurde in sechs Einzelaufgaben unterteilt. Jede einzelne Aufgabe wurde mit einem eigenständigen Programm abgebildet. Die Kommunikation erfolgt über definierte Schnittstellen. Die Kopplung aller Module ist durch ein übergeordnetes Menüprogramm möglich, aber nicht zwingend erforderlich. Die einzelnen Module können wie folgt beschrieben werden.

Betriebsdateneingabe: Hilfsprogramm, welches die Eingabe wichtiger betriebsspezifischer Daten unterstützt (Front-End-System). Hierzu zählen: Geometrische Abmessungen des vorhandenen oder zu planenden Betriebes, planungsrelevante Inneneinrichtungen. Bei der Modellierung wurden sowohl konventionelle als auch wissensbasierte Technologien eingesetzt.

Energieverbrauchsrechnung: Beratungsprogramm, welches Auskunft über den Energieverbrauch von Gewächshausanlagen liefert (Belichtung, Heizung). Grundlage ist ein konventionelles Simulationsprogramm, gekoppelt mit datenorientierten Modellansätzen.

Bestimmung der Basisdaten für die Heizungsanlagenplanung: Berechnungsprogramm, welches alle für die Planung der Heizenergieversorgung von Gewächshausanlagen notwendigen Daten ermittelt. Im wesentlichen sind dies: notwendige Heizleistungen und deren Aufteilung, Jahresenergiebedarf und Jahresstunden anfallender Heizleistungen. Das Programm besteht aus einer Simulationsrechnung und einem konventionellen Planungsalgorithmus zur Bestimmung der Auslegungsleistungen. Die Simulationsrechnung erfolgt in gleicher Weise wie im Modul 'Energieverbrauchsrechnung', jedoch mit reduzierten Eingabeparametern (Standardwerte für Klimaregelung, Heizungssystem und Energiebereitstellung). Da beide Module auf dem gleichen 'Simulationskern' aufbauen, werden sie im folgenden immer zusammen betrachtet.

Planung der Energiebereitstellung: Expertensystemprogramm, welches die Auswahl möglicher Heizenergieversorgungskonzepte für einen Gartenbaubetrieb durchführt. Durch die Kopplung mit einem ökonomischen Modell und einem Hypertextsystem werden gefundene Lösungen ökonomisch bewertet und Zusatzinformationen gegeben.

Planung der Energieausbringung: Expertensystemprogramm, welches die Aus-

wahl, Kombination und Berechnung der Heizflächen in den Gewächshäusern durchführt. Der Schwerpunkt dieses Moduls liegt auf der technischen Auslegung und Konfiguration der Heizungsanlage, wobei mehrere Lösungsmöglichkeiten entwickelt werden.

Datenbanksystem: Konventionelles Datenverwaltungssystem, welches alle für die Programme wichtigen Daten und Standardwerte, die der Benutzer verändern, erweitern oder löschen kann, enthält.

### 3.2.2 Konzepte der Wissensrepräsentation

#### 3.2.2.1 Ansatz orientiert an Produktionsregeln

Um die in dem Expertensystem HORTEX notwendigen Ableitungsbäume zu repräsentieren, wurde ein formales System entwickelt, welches die Darstellung sowohl vorwärts- als auch rückwärtsverketteter Und-Oder-Graphen ermöglicht. Als Darstellungswerkzeug wurde die Programmiersprache Prolog ausgewählt, da sie Such- und Schlußfolgerungsverfahren in einem ökonomisch sinnvollen Rahmen zur Verfügung stellt.

Die Darstellung eines Knotens eines Ableitungsbaumes erfolgt mit dem Prolog-Prädikat

$$knoten (E, EK, EZ_L) \quad (3.2-1)$$

in der Wissensbasis. Darin bedeuten

E = Name des Knotens

EK = Bezeichnung der Kante

EZ<sub>L</sub> = Liste der Zielknoten der von E ausgehenden Kanten.

Durch Angabe mehrerer Knoten mit dem gleichen Namen können Oder-Bäume erzeugt werden. Durch die Angabe mehrerer Zielknoten in der Liste EZ<sub>L</sub> können Und-Verknüpfungen dargestellt werden. EK kann folgende Werte annehmen:

$$\begin{aligned} ist\_ende &=> \text{Kante mit Endknoten (höchste Priorität)} \\ hat\_moeglichkeit &=> \text{gewöhnliche Kante (mittlere Priorität)} \\ hat\_variante &=> \text{alternative Kanten (niedrigste Priorität)} \end{aligned} \quad (3.2-2)$$

Durch den Vorsatz *ist\_ziel* vor der Kantenbezeichnung wird ein Knoten spezifi-

ziert, der eine Erklärungsfunktion für die Lösung hat. Zum Beispiel könnte für EK = *ist\_ziel\_hat\_möglichkeit* das Prädikat 3.2-1 umgangssprachlich beschrieben werden mit: Aus E folgt EZ und E muß als Erklärung oder Zwischenlösung registriert werden.

Die Erweiterung des Prädikates 3.2-1 in eine Prologregel ermöglicht es, Bedingungen zu spezifizieren, unter denen der Knoten wahr ist. Somit ergibt sich als erweiterter Knoten

$$\begin{aligned} \text{knoten (E, EK, EZ}_L) \text{ if} & & (3.2-3) \\ & \text{bedingung 1 and} \\ & \text{bedingung 2 and} \\ & \dots \end{aligned}$$

Die Bedingungen können konventionelle Prologprädikate sein, die rückwärtsverkettet ausgewertet werden oder spezielle Prädikate, bei denen der Benutzer nach Angaben gefragt wird. Der allgemeine Aufbau solcher Prädikate wird beschrieben mit

$$\text{wenn}_{--} (\text{Be, X, Ant}) \quad (3.2-4)$$

oder

$$\text{wenn}_{--}\text{zahl} (\text{Be, X, Ant}) \quad (3.2-5)$$

Be gibt die Bedingung an. X ist eine Variable in der Bedingung. Ant ist bei Prädikat 3.2-4 die notwendige Antwort des Benutzers, damit die Bedingung erfüllt wird. In Prädikat 3.2-5 ist Ant der vom Benutzer angegebene numerische Wert.

In einer speziellen Datenbasis sind alle Fragen, die im Zusammenhang mit den Wenn-Prädikaten gestellt werden, in Form des acht-stelligen Prädikates

$$f (\text{Fr}_{\text{Ch}}, \text{Be}, \text{X}_L, \text{Ant}_L, \text{Fr}_{\text{Te}}, \text{Fr}_{\text{Li}}, \text{Fr}_{\text{min}}, \text{Fr}_{\text{max}}) \quad (3.2-6)$$

gespeichert. Es bedeuten

- Fr<sub>Ch</sub> = Kennzeichnung der Fragecharakteristika (numerisch, symbolisch inklusives Oder, symbolisch exklusives Oder)
- Be = Bedingung
- X<sub>L</sub> = Liste möglicher Bedingungsvariablensymbole
- Ant<sub>L</sub> = Liste möglicher Benutzerantworten bei nicht-numerischer Eingabe
- Fr<sub>Te</sub> = Fragetext
- Fr<sub>Li</sub> = Link zum Hypertextsystem bei Benutzerfragen
- Fr<sub>min/max</sub> = Eingabegrenzen bei numerischer Eingabe

Die Bearbeitung des mit den Prädikaten 3.2-1 bis 3.2-5 definierten Ableitungsbaumes geschieht durch zwei Mechanismen:

Mechanismus 1: Ein Regelinterpreter verknüpft das erste Element  $E_1$  einer Pfadliste  $Pf_L$  mit dem Knoten  $E_2$  aus der Wissensbasis, so daß gilt:  $E_1 = E_2$ . Werden alle Bedingungen des Knotens  $E_2$  erfüllt, wird die Liste der Zielknotenbezeichnungen  $EZ_{L_2}$  der Pfadliste vorne angehängt (pre-order-Verfahren). Beginnt darüberhinaus die Knotenbezeichnung von  $E_2$  mit *ist\_ziel*, wird  $E_2$  der Lösungsliste  $Lö_L$  hinten angehängt. Dieses Verfahren wird solange wiederholt, bis die Pfadliste  $Pf_L$  leer ist. Die Liste  $Lö_L$  stellt die gesuchte Lösung dar und wird abgespeichert.

Mechanismus 2: Scheitert bei der Abarbeitung des Ableitungsbaumes eine Klausel, so werden mit Hilfe des Backtracking-Mechanismus stufenweise die letzten Aktionen rückgängig gemacht, die Lösungsliste entsprechend reduziert und alternative Lösungswege gesucht. Ist eine Lösung gefunden, wird die Lösungsliste  $Lö_L$  abgespeichert und durch ein erzwungenes Scheitern der Backtracking-Mechanismus aufgerufen. Dieses Verfahren wird solange fortgesetzt, bis der gesamte Baum durchsucht ist und das Prädikat endgültig scheitert. Die abgespeicherten Lösungslisten werden danach durch Matching mit entsprechenden Satz-elementen oder Sätzen verknüpft und zu einem Lösungstext zusammengesetzt.

Das Suchverfahren entspricht der vollständigen Suche durch Tiefensuche und Backtracking (vgl. Kap. 2.2.2.1.2). Eine Ausnahme stellt die Bearbeitung von Variantenknoten dar. Hier werden nach Aufruf des Knotens alle anderen Varianten des Ausgangsknotens vom Baum abgekoppelt und erst dann wieder angekoppelt, wenn durch Backtracking auf den Ausgangsknoten erneut von einem übergeordneten Knoten zugegriffen wird.

Die Auswahl der Reihenfolge der anzuwendenden Regeln geschieht durch die angegebenen Kantenbezeichnungen EK oder bei Gleichheit durch die Reihenfolge in der Wissensbasis.

Abbildung 3.2-1 zeigt den realisierten Regelinterpreter. Abbildung 3.2-2 stellt den implementierten Algorithmus zur Bearbeitung der Wenn-Prädikate graphisch dar. Man erkennt, daß nach jeder Benutzereingabe (User-Interface) der gesamte Regelbaum wieder von vorne analysiert wird. Hierdurch können leicht 'Was-wäre-wenn-Anfragen' realisiert werden.

```
los([],K,K).

los([Kopf|Rest],Eingang,Ausgang):-
    weiter(Kopf,Liste,Eingang,Ausgabe),
    anhaengen_Liste_an_Liste(Liste,Rest,NeueListe),
    los(NeueListe,Ausgabe,Ausgang).

los2([],K,K,Abschneidevariante):-
    assert(bekannt(Abschneidevariante),knotenbank).

los2([Kopf|Rest],Eingang,Ausgang,Abschneidepunkt):-
    weiter(Kopf,Liste,Eingang,Ausgabe),
    anhaengen_Liste_an_Liste(Liste,Rest,NeueListe),
    los2(NeueListe,Ausgabe,Ausgang,Abschneidepunkt).

weiter(Faktum,[],Ein,Aus):-
    knoten(Faktum,ist_ziel,_,Ein),
    anhaengen_Element_an_eine_Liste(Faktum,Ein,Aus).

weiter(Faktum,[],Ein,Ein):-
    knoten(Faktum,ist_ende,_,Ein).

weiter(Faktum,Moeglichkeit,Ein,Aus):-
    knoten(Faktum,ist_ziel_hat_moeglichkeit,Moeglichkeit,Ein),
    anhaengen_Element_an_eine_Liste(Faktum,Ein,Aus).

weiter(Faktum,Moeglichkeit,Ein,Ein):-
    knoten(Faktum,hat_moeglichkeit,Moeglichkeit,Ein).

weiter(Faktum,[],Ein,Aus):-
    retractall(bekannt(Faktum),knotenbank),
    knoten(Faktum,ist_ziel_hat_variante,Variante,Ein),
    not(bekannt(Faktum)),
    anhaengen_Element_an_eine_Liste(Faktum,Ein,Ein_neu),
    los2(Variante,Ein_neu,Aus,Faktum).

weiter(Faktum,[],Ein,Aus):-
    retractall(bekannt(Faktum),knotenbank),
    knoten(Faktum,hat_variante,Variante,Ein),
    not(bekannt(Faktum)),
    los2(Variante,Ein,Aus,Faktum).
```

Abbildung 3.2-1: Der entwickelte Regelinterpretier des Expertensystems HORTEX



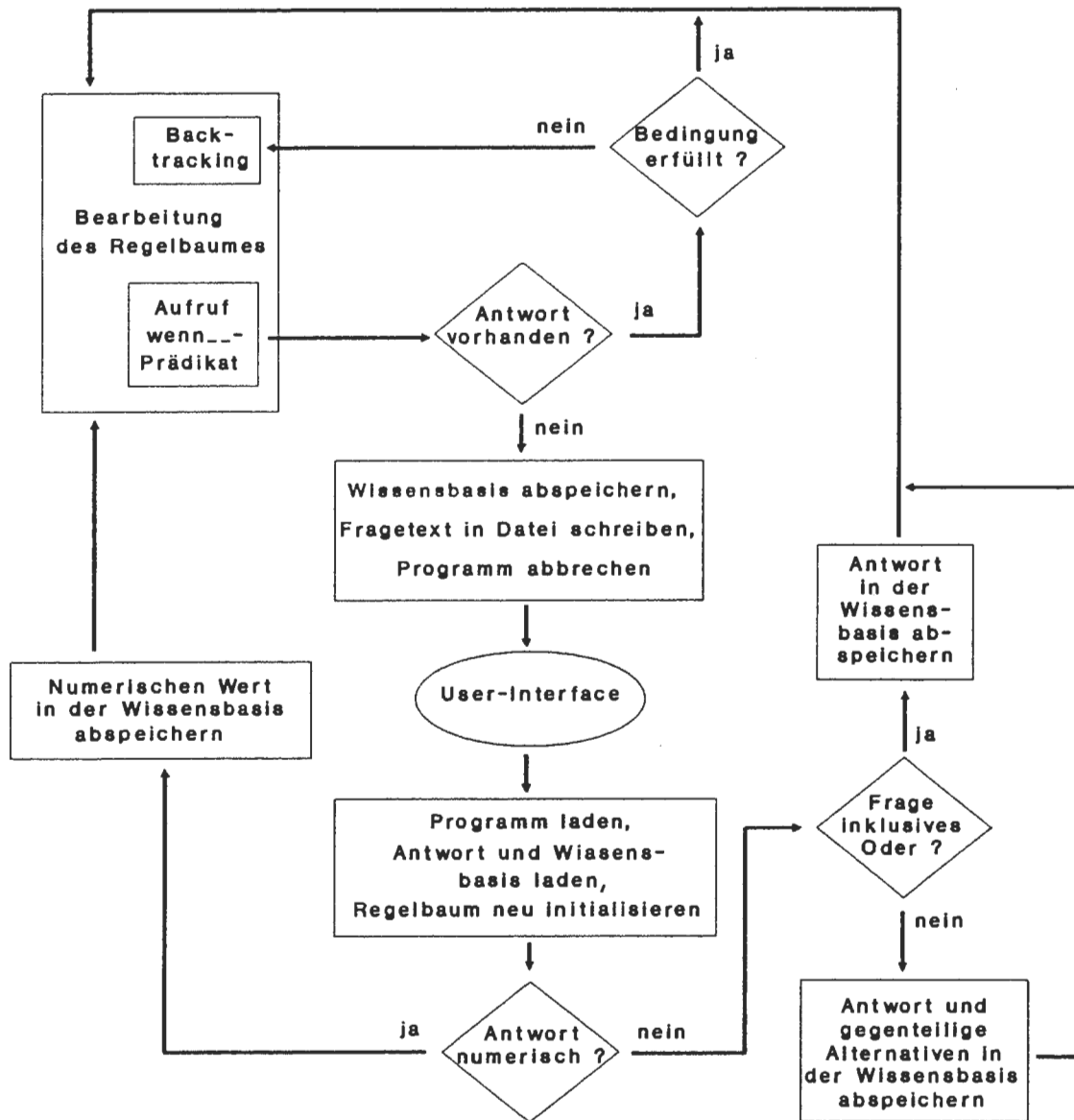


Abbildung 3.2-2: Algorithmus zur Behandlung benutzerspezifischer Angaben

### 3.2.2.2 Ansatz orientiert am Framekonzept

Für die Darstellung von framebasierten Wissensrepräsentationsansätzen wurde ein formales System erstellt, welches auf den bei SCHNUPP und NGUYEN HUU (1987, S. 117ff) und SAVORY (1988, S. 143ff) beschriebenen Ansätzen aufbaut. Der Frameraamen wurde jedoch auf ein fünfstelliges Prädikat erweitert, Objekt-Facetten hinzugenommen, die prozeduralen Bearbeitungsmöglichkeiten auf die bei technischen Planungen hauptsächlich benötigten *if-needed*-Prozeduren reduziert und ein veränderter und auf Typisierung abgestimmter Ob-

jektinterpretierer realisiert. Im einzelnen wird ein Frame im Expertensystem HOR-TEX durch das Prädikat

$$\text{frame} (E_O, Sl, Fa, In_L, Bes) \quad (3.2-7)$$

beschrieben. Es bedeuten

- $E_O$  = Bezeichnung des Objektes im Rahmen
- $Sl$  = Bezeichnung des Slots im Rahmen
- $Fa$  = Beschreibung der Art des Slots (Facette)
- $In_L$  = Liste der Slotinhalte
- $Bes$  = Beschreibung, ob das Slot dem Benutzer als Ergebnis angezeigt wird oder nicht

Speziell bezeichnete Slots kennzeichnen Hierarchieebenen innerhalb der abgebildeten Frames. Für  $Sl = \text{ist\_ein}$  gilt, daß  $In_L$  'Väter' darstellt, von denen Werte übernommen (geerbt) werden können. Für  $Sl = \text{sohn}$  gilt, daß an  $In_L$  Werte übergeben (vererbt) werden können.

Durch die Beschreibung  $Fa$  eines Slots wird definiert, auf welche Art und Weise der Slot mit Informationen gefüllt werden kann. Für  $Fa = \text{wert}$  und  $Fa = \text{standard}$  gilt, daß der Slotinhalt durch  $In_L$  bestimmt wird. Ist  $Fa = \text{wenn\_benoetigt}$  wird das Prädikat

$$\begin{aligned} \text{prozedur} (E_O, Sl, In_L) : - \\ \text{Berechnung 1,} \\ \text{Berechnung 2, ...} \end{aligned} \quad (3.2-8)$$

aufgerufen. Die Auswahl der entsprechenden Prozeduren erfolgt durch Matching von  $E_O$ ,  $Sl$  und  $In_L$  des Prädikates 3.2-7 mit 3.2-8 ( $E_O$ ,  $Sl$ ,  $In_L$  werden unifiziert), so daß Prozeduren definiert werden können, die für einzelne Objekte, einzelne Slots, aber auch für Objekt- oder Slotklassen gelten können. Ist im Prädikat 3.2-7  $Fa = \text{objekt}$ , so gilt, daß der Slotinhalt aus einem weiteren Objekt besteht, welches analysiert werden muß.

Die Wissensbasis des Expertensystems besteht somit aus Prädikaten, die alle die Form der Prädikate 3.2-7 oder 3.2-8 haben (evtl. erweitert um Berechnungsprädikate). Um aus dem so beschriebenen Objektraum Informationen erhalten zu können, ist ein Objektinterpretierer notwendig, der die notwendigen Schlußfolgerungen und Bearbeitungsprozesse bereitstellt. Im Expertensystem

HORTEX wurde ein Objektinterpretierer realisiert, der zwei Mechanismen beinhaltet.

Suche nach dem Inhalt eines speziellen Slots:

Der Objektinterpretierer versucht, durch Matching (der Anfrage mit der Wissensbasis) den Inhalt eines gewünschten Slots eines Frames zu ermitteln. Die Auswahl der möglichen Slots erfolgt nach folgender Priorität (die höchste zuerst): Objektslots, Wertslots, Wenn-benötigt-Slots, Standard-Slots. Die Suche wird durch Backtracking gesteuert. Konnte kein Slotinhalt ermittelt werden, wird versucht, von einem übergeordneten Objekt den Slotinhalt zu erben. Hierzu wird ähnlich wie bei der Produktionsregelauswertung der 'Sohn' mit dem 'Vater' verknüpft und der 'Vater' ist das neue Suchobjekt. Da sämtliche Slotinhalte Listen darstellen, ist es auch möglich, daß mehrere 'Väter' existieren (multiple Vererbung). Genauso ist es auch möglich, daß der Inhalt eines Objektslots wieder mehrere Objekte oder Werte sind. Die Suche innerhalb der Frames wird abgebrochen, sobald der gesuchte Slot mit einem Wert gefüllt ist (somit sind die 'Väter', die in der 'Vaterliste' zuerst verzeichnet sind, 'dominant' gegenüber den weiter hinten stehenden). Abbildung 3.2-3 zeigt den implementierten Auswertemechanismus.

Suche nach maximaler Informationsmenge zu einem Objekt:

Bei diesem Bearbeitungsverfahren werden alle Frames nach Informationen (Slotinhalten) zu einem Objekt durchsucht. Zuerst wird versucht, sämtliche direkte Slots eines Frames durch Matching oder Berechnung zu füllen. Anschließend werden alle 'Väterobjekte' durchsucht. Da die Unterklassen alle Eigenschaften der Oberklassen 'erben', werden Slots, die entweder noch nicht gefüllt sind oder die noch gar nicht als Slot vorhanden sind, in die Lösungsmenge mit aufgenommen. Entscheidend ist, daß Objekte nur Eigenschaften von übergeordneten Klassen 'erben' können, niemals aber von ihren 'Söhnen' ! Nach der Auswertung der Vererbungshierarchie werden die gefundenen *objekt*-Slots ausgewertet. Hierbei können neue 'Väter' auftreten, die analysiert werden müssen. Die Bearbeitung der Wissensbasis erfolgt so lange, bis keine weiteren Herleitungsmöglichkeiten mehr bestehen. Am Ende der Bearbeitung werden alle gefundenen Slots mit ihren dazugehörigen Inhalten abgespeichert. Sie stellen die gesamte Information dar, die über das Ausgangsobjekt in der Wissensbasis vorhanden ist. Eine vereinfachte Beschreibung des zweiten Mechanismus liefert die Abbildung 3.2-4. Die Programmierung erfolgte durch verschachtelte, rekursive Prologprädikate. Matching erfolgt ausschließlich durch Unifikation. Genaue Informationen gibt der Source-Code im Anhang B.

```
frage(Objekt,Slot,Wert):-
    bound(Objekt),bound(Slot),free(Wert),
    suche(Objekt,Slot,Werteliste),
    mitglied_der_liste(Wert,Werteliste).

frage(Objekt,Slot,Wert):-
    bound(Objekt),bound(Slot),bound(Wert),
    suche(Objekt,Slot,Werteliste),
    mitglied_der_liste("beliebig",Werteliste),!.

frage(Objekt,Slot,Wert):-
    bound(Objekt),bound(Slot),bound(Wert),
    suche(Objekt,Slot,Werteliste),
    mitglied_der_liste(Wert,Werteliste),!.

frage_wert(Objekt,Slot,Wert):-!,
    frage(Objekt,Slot,S),
    str_real(S,Wert).

suche(Objekt,Slot,Wert):-
    frame(Objekt,Slot,objekt,Wert,_),!.

suche(Objekt,Slot,Wert):-
    frame(Objekt,Slot,wert,Wert,_),!.

suche(Objekt,Slot,Ergebnis):-
    frame(Objekt,Slot,wenn_benoetigt,_,_),
    prozedur(Objekt,Slot,Ergebnis),!.

suche(Objekt,Slot,Wert):-
    frame(Objekt,Slot,standard,Wert,_),!.

suche(Objekt,Slot,Wert):-
    frame(Objekt,ist_ein,wert,Vaeter,_),
    suche_vaeter(Vaeter,Slot,Wert),!.

suche(Objekt,Slot,Wert):-
    findall(ObjektSlots,frame(Objekt,ObjektSlots,objekt,_,_),Slotobs),
    suche_objekte(Objekt,Slotobs,Slot,Wert),!.

suche_vaeter([],_):-fail.

suche_vaeter([Vater|_],Slot,Wert):-
    suche(Vater,Slot,Wert),!.

suche_vaeter([_|Rest],Slots,Wert):-!,
    suche_vaeter(Rest,Slots,Wert).

suche_objekte(_,[],_):-
    ....
    ....
```

Abbildung 3.2-3: Implementierter Mechanismus zur Auswertung von Slots

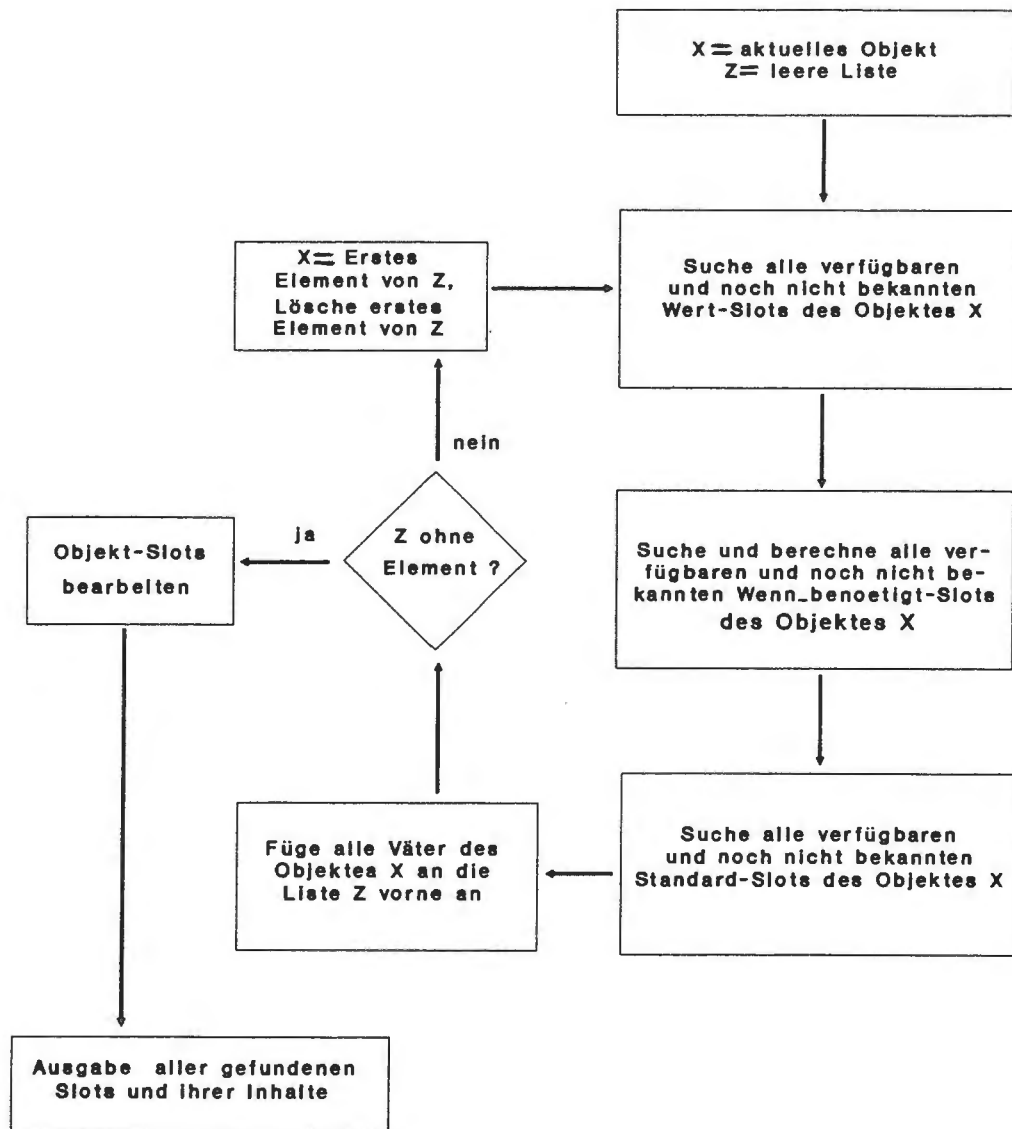


Abbildung 3.2-4: Vereinfachte Darstellung des Algorithmus zur Suche aller Informationen zu einem Objekt

### 3.2.2.3 Datenbankorientierter Ansatz

Die gesamte Datenverwaltung des Systems erfolgt durch ein Datenbanksystem, welches in Pascal realisiert wurde, auf Recordstrukturen aufbaut und einen hierarchischen Charakter hat. Auf der untersten Ebene befinden sich die anwendungsspezifischen Datensätze in Form von Pascalrecords. Mehrere solcher Datensätze werden zu einer für den Benutzer sichtbaren und editierbaren Tabelle zusammengefaßt. Mehrere Tabellen bilden eine Datenbankabteilung. Es erfolgt eine strikte Trennung zwischen Datenbasis und Abarbeitungsmechanismus. Als alleiniger Abarbeitungsmechanismus steht die Suche nach Spalten- und Zei-

lenschlüsseln zur Verfügung, sowie die Umwandlung von Datenbankzeilen in Prolog-Prädikate. Die Implementierung relationaler Abarbeitungsmechanismen erfolgt nicht. Es werden folgende Kontrollmechanismen eingesetzt (vgl. Anhang C.5 (Struktur)):

- auf Datenebene: Formatangaben, Eingabegrenzen bei numerischer Eingabe,
- auf Spaltenebene: Sperren oder Freigabe der Editierbarkeit,
- auf Zeilenebene: Standardvorgaben bei Zeilenneueingaben,
- auf Tabellenebene: Logische Links zum Hypertextsystem,
- auf Datenbankabteilungsebene: Sperren oder Freigabe des Ladens und Speicherns eigener Datenbankabteilungen.

Es bleibt festzuhalten, daß es nicht das Ziel bei der Implementierung des oben beschriebenen Konzeptes war, eine vollständige und auf Massendaten ausgerichtete Datenbank zu erstellen. Vielmehr wurde versucht, eine für den Programm benutzer veränderbare, einsichtige und einfache Datenhaltung zu schaffen. Abbildung 3.2-5 gibt ein Beispiel. Desweiteren wird das gesamte Hypertext- und Hilfesystem über zwei Datenbankabteilungen gesteuert.

**DATENBLATT: Rohrheizungssysteme**

BEZEICHNUNG	N-WAB	Mat	DU	PREIS
Stahlrohr 51 mm	134	St	47	12
PE-Rohr 3/4 Zoll	51	PE	20	4
Stahlrohr 70 mm	178	St	66	23
Stahlrohr 114 mm	278	St	107	32
Stahlrohr 27 mm	76	St	25	8
Tropfenrohr 22	134	St	30*	14
Aluflügelrohr 70	134	Alu	19	18
PE-Rohr 1 Zoll	63	PE	25	5
PP-Rohr 3/4 Zoll	51	PP	20	4
PP-Rohr 1 Zoll	63	PP	25	5

Abbildung 3.2-5: Beispiel einer Datenbanktabelle (Programmgraphik) mit Standardwerten im Bereich Heizungssysteme (N-WAB = normierte Wärmeabgabe (W/m), Du = hydraulisch wirksamer Innendurchmesser (mm), Mat = Material, Preis = Investitionsvolumen (DM/m). Quellen: TANTAU 1983; KTBL 1990; Expertengespräche GABLOFFSKI, DOMKE (vgl. Kapitel 3.1.3); TANTAU 1990; DIN 2993 (1977); \* = berechnet nach dem von VAN LEEUWEN BUIZEN (1990) angegeb. Rohrvolumen)

### 3.2.3 Modellierung und Darstellung des Wissens

#### 3.2.3.1 ... im Bereich Betriebsdateneingabe

Wichtiger Bestandteil bei der Planung und Analyse der Energieversorgung von Gewächshausanlagen ist die Erfassung des vorhandenen oder zu erstellenden Betriebes. Weil letztlich jedes Programmmodul in irgendeiner Form die räumlichen Abmessungen der Gewächshäuser benötigt, ist es sinnvoll, eine dreidimensionale, digitalisierte Abbildung des Betriebes im Computer vorzuhalten. Da aber schon die Beschreibung eines Betriebes mit mehreren Gewächshauskappen die Eingabe von sehr vielen Raumkoordinaten erfordert, muß ein Front-End-System entwickelt werden, welches eine einfache und sichere Eingabe des Betriebes ermöglicht. Unter Ausnutzung der logischen Zusammenhänge einzelner Raumkoordinaten wurde ein solches System im Expertensystem HORTEX durch eine enge Kopplung zwischen wissensbasierter und prozeduraler Repräsentation realisiert. Die Entwicklung des Betriebsmodells erfolgt auf der Basis einer zweidimensionalen Skizze des Betriebes (Draufsicht) plus räumlichen Zusatzangaben (Stehwandhöhe, Rastermaß, Dachneigungswinkel). Um das Wesentliche der Skizze später herausarbeiten zu können, wird ein grobes Raster vorgegeben, in dem 250 Rasterpunkte die möglichen Eckpunkte für die Gewächshaussegmente darstellen (vgl. Abbildung 3.2-6). Nach Beendigung der Grob-skizzierung wird das vorliegende Pixelbild in Vektorform überführt, wobei jeder Vektor beschrieben wird durch: Startpunkt, Zielpunkt und die Angabe, welche Flächen er realisiert (Stehwand oder Giebelwand oder Teile davon).

Um die vorliegende Vektorgraphik in ein adäquates Modell des Betriebes zu überführen, müssen die vom Vektor repräsentierten, tatsächlichen Gewächshauslängen bekannt sein. Die Abfrage aller realen Längen vom Benutzer ist nicht sinnvoll. Versuche ergaben, daß, wie bereits erwähnt, zu viele Angaben zu machen sind und daß bei verschachtelten Betrieben der Benutzer nicht mehr die logischen Abhängigkeiten einzelner Strecken voneinander überblickt. Im Expertensystem HORTEX wird daher vor der Frage nach der Streckenlänge versucht, durch logische Verknüpfung aller bisher bekannten Strecken die benötigte Länge des unbekanntes Vektors herzuleiten. Es muß ein paralleler und in der Skizze gleichlanger Vektor gefunden werden, dessen reale Länge schon bekannt ist. Hierbei kann der in der realen Länge unbekanntes Vektor mit in der realen Länge bekannten Vektoren verlängert oder unterteilt werden. Die Verlängerung und die Unterteilung kann sowohl vom Beginn als auch vom Endpunkt des unbekanntes Vektors erfolgen.

Das Herleitungsverfahren wurde im Programm mittels mehrerer Prologregeln abgebildet, die im Anhang unter Abschnitt C.1 zu finden sind.

Der so beschriebene Lösungsraum wird mit Hilfe des prologinternen Inferenzmechanismus durchsucht (Tiefensuche mit Backtracking). Da bei umfassenden Zeichnungen und bei ungünstiger Vektorkonstellation der mögliche Suchraum

sehr große Dimensionen annehmen kann, mußte das Verfahren heuristisch beschränkt werden, so daß jeweils nur in eine Richtung die Erweiterung/Aufteilung des unbekanntes Vektors geschehen darf.

In bestimmten Situationen führt dieses dazu, daß der Benutzer nach einer Strecke gefragt wird, die der Rechner durch kombinierte Vorwärts-/Rückwärtsverlängerung des unbekanntes Vektors theoretisch hätte herleiten können. Damit muß der Benutzer aber genau die Länge der Strecke angeben, die hergeleitet werden könnte, da sonst die räumliche Logik nicht mehr gegeben ist. Um dieses zu garantieren, wird am Ende der Längeneingabe noch einmal überprüft, ob alle Längeneingaben mit der Skizze realisiert werden können.

Die weitere Bearbeitung vorliegender Daten läßt sich in drei Bereiche gliedern. Zum einen wird die angegebene Skizze maßstabsgetreu umgewandelt und durch Verknüpfung mit den Gewächshausdaten in mehrere Schnittebenen unterteilt (Boden, Traufe, First). Anschließend können die Daten mit einem 3-D-Algorithmus (vgl. BIELIG-SCHULZ und SCHULZ 1987, S. 59ff) dargestellt werden. Auf eine Nichtdarstellung verdeckter Linien wurde verzichtet, da dieses zum einen relativ viel Rechenzeit beansprucht und zum anderen dem Glashauscharakter der Gewächshäuser nicht entspricht.

Ein weiterer Prozeß berechnet alle sich durch die bisherigen Angaben ergebenden Gewächshaushüllflächen. Abbildung 3.2-7 gibt die eingesetzten Berechnungsverfahren wieder. Für jeden Berechnungsabschnitt  $l_{BA}$ , der durch einen Vektor im Grundriß des Betriebes gegeben ist, werden die im 3D-Raum entstehenden Innenflächen ( $A_i$ ) und Außenflächen ( $A_a$ ) unter Verwendung der Segmentbreite ( $b_{Seg}$ ), Kappenbreite ( $b_{Kap}$ ), der Dachneigungswinkel ( $DW1$ ,  $DW2$ ) und der Stehwandhöhe ( $h_{Ste}$ ) bestimmt. Dabei werden sowohl die Abmessungen des aktuell zu berechnenden Segmentes ( $Akt$ ) als auch die Abmessungen des angrenzenden Nachbarsegmentes ( $Na$ ) benötigt. Es werden alle entstehenden Flächen auf Dreieck-, Rechteck- oder Trapezform aufgelöst ( $Da =$  Dach,  $Gi =$  Giebel,  $Tr(h) =$  Trapez mit der Höhe  $h$ ). Stoßen zwei versetzt stehende Giebelfronten aufeinander, wird mit einem vereinfachten Berechnungsverfahren die Aufteilung in Innen- und Außenflächen vorgenommen (vgl. Gleichung 3.2-15). Der hierdurch mögliche maximale Zuordnungsfehler ist gleich der halben Fläche der Giebel-dreiecke. Er tritt auf, wenn zwei Haussegmente mit gleichem Dachneigungswinkel und gleicher Stehwandhöhe um die halbe Kappenbreite versetzt zueinander stehen.

In einem dritten Schritt werden den einzelnen Gewächshausabteilungen die planungsrelevanten Inneneinrichtungen zugeordnet. Im einzelnen sind dieses die installierte Belichtungsanlage, charakterisiert durch die installierte elektrische Anschlußleistung und die Lampenart sowie installierte Energieschirme, charakterisiert durch das Schirmmaterial und die Abdichtungsgüte.



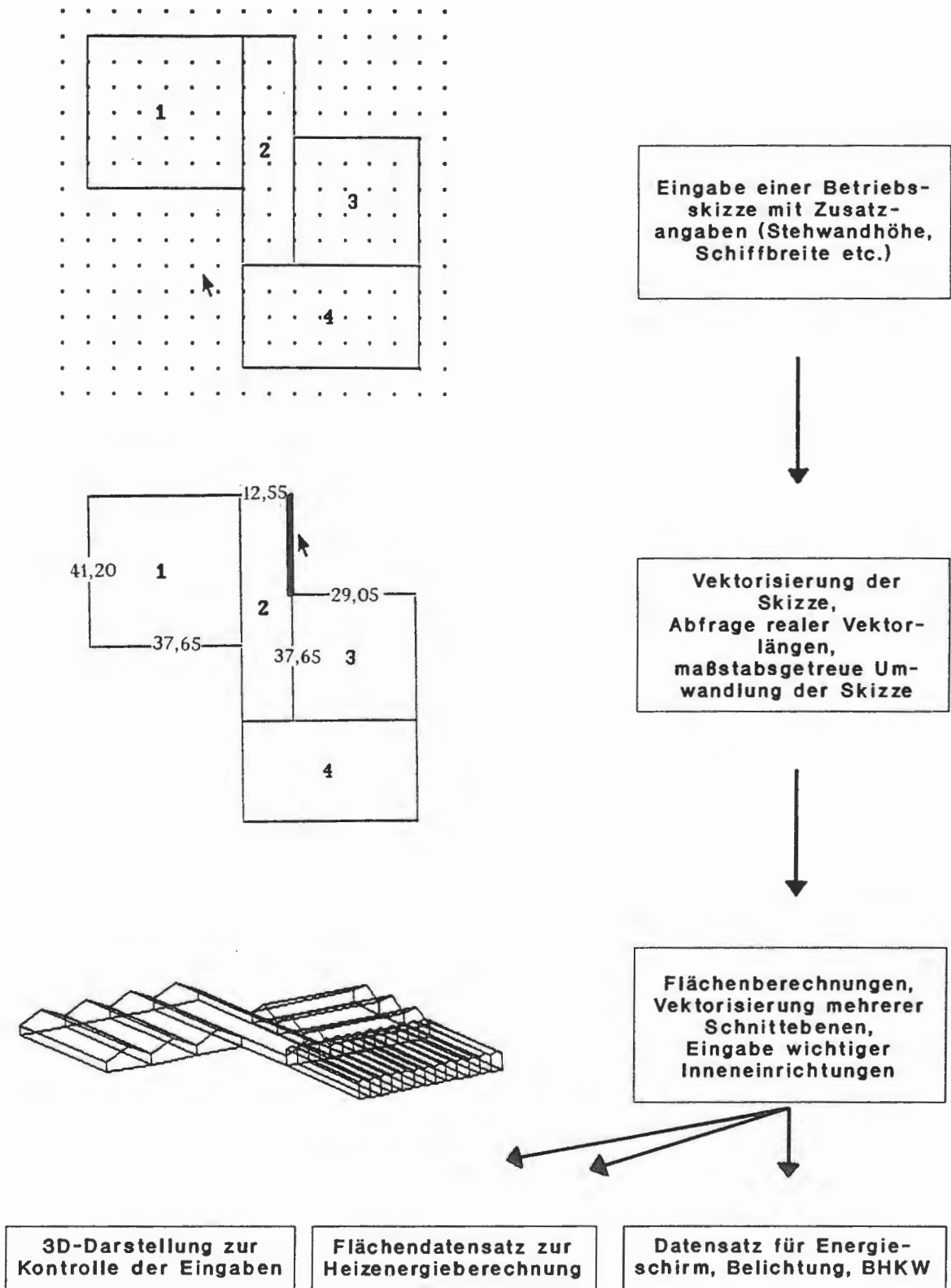


Abbildung 3.2-6: Schematische Darstellung der Eingabe der Betriebsdaten

$$A_{Da} = b_{Seg} \cdot l_{BA} \cdot \frac{\sin DW_1 + \sin DW_2}{\sin (180 - DW_1 - DW_2)} \quad (3.2-9)$$

$$A_{Gi} = h_{Ste} \cdot l_{BA} + \frac{\sin DW_1 \cdot \sin DW_2 \cdot b_{Kap} \cdot l_{BA}}{2 \cdot \sin (180 - DW_1 - DW_2)} \quad (3.2-10)$$

$$A_{Tr}(h) = \frac{2 \cdot b_{Kap} - \frac{h}{\tan DW_1} - \frac{h}{\tan DW_2}}{2} \cdot h \quad (3.2-11)$$


---

Dachfläche —  $A_i = 0$        $A_a = A_{Da}$  (3.2-12)

---

Giebel außen —  $A_i = 0$        $A_a = A_{GiAkt}$  (3.2-13)

---

Stehwand außen —  $A_i = 0$        $A_a = h_{Ste,Akt} \cdot l_{BA}$  (3.2-14)

---

Giebel gegen Giebel  $\left\{ \begin{array}{l} A_i = A_{GiNa} \quad A_a = A_{GiAkt} - A_{GiNa} \quad \text{falls } A_a > 0 \\ A_i = A_{GiAkt} \quad A_a = 0 \quad \text{sonst} \end{array} \right.$  (3.2-15)

---

Giebel gegen Stehwand  $\left\{ \begin{array}{l} A_i = l_{BA} \cdot h_{Ste,Na} \quad A_a = A_{GiAkt} - A_i \quad \text{falls } h_{Ste,Na} \leq h_{SteAkt} \\ A_i = A_{GiAkt} \quad A_a = 0 \quad \text{falls } h_{GiAkt} \leq h_{SteNa} \\ A_i = A_{Tr}(h_{Ste,Na} - h_{Ste,Akt})_{Akt} \cdot \frac{l_{BA}}{b_{KapAkt}} + l_{BA} \cdot h_{SteAkt} \\ A_a = A_{GiAkt} - A_i \quad \text{sonst} \end{array} \right.$  (3.2-16)

---

Stehwand gegen Giebel  $\left\{ \begin{array}{l} A_i = A_{GiNa} \quad A_a = l_{BA} \cdot h_{SteAkt} - A_i \quad \text{falls } h_{SteAkt} \geq h_{GiNa} \\ A_i = l_{BA} \cdot h_{SteAkt} \quad A_a = 0 \quad \text{falls } h_{SteAkt} \leq h_{SteNa} \\ A_i = A_{Tr}(h_{SteAkt} - h_{SteNa})_{Na} \cdot \frac{l_{BA}}{b_{KapNa}} + l_{BA} \cdot h_{SteNa} \\ A_a = l_{BA} \cdot h_{SteAkt} - A_i \quad \text{sonst} \end{array} \right.$  (3.2-17)

---

Stehwand gegen Stehwand  $\left\{ \begin{array}{l} A_i = h_{SteAkt} \cdot l_{BA} \quad A_a = 0 \quad \text{falls } h_{SteAkt} \leq h_{SteNa} \\ A_i = h_{Ste,Na} \cdot l_{BA} \quad A_a = (h_{SteAkt} - h_{SteNa}) \cdot l_{BA} \quad \text{sonst} \end{array} \right.$  (3.2-18)

Abbildung 3.2-7: Formeln und Verfahren zur Berechnung der Gewächshausflächen

### 3.2.3.2 ... im Bereich Wärmebedarfsrechnung

#### 3.2.3.2.1 Klima

Die Modellierung der für das System notwendigen Klimadaten erfolgt mit einem datenbankorientierten Ansatz. Zur Abbildung des jährlichen Klimaverlaufes wurde die Datensammlung TRY (Testreferenzjahr) von BLÜMEL et al. (1986) ausgewählt. Sie stellt ein charakteristisches Klimajahr auf Stundenbasis für verschiedene Regionen der Bundesrepublik (alte Bundesländer und Berlin) zur Verfügung und ist speziell für gebäudetechnische Simulationen entwickelt worden. Die Daten des Testreferenzjahres wurden auf die für weitere Berechnungen notwendigen Werte Außentemperatur, Windgeschwindigkeit und Globalstrahlung (diffus + direkt) reduziert. Um auch besonders milde und strenge Winter abbilden zu können, werden bei der Modellierung eines milden Winters die Temperaturwerte des kalendarischen Winters um 4 K heraufgesetzt, bei der Darstellung eines strengen Winters werden die TRY-Temperaturen um 4 K gesenkt (vgl. Abbildung 3.2-8). Andere Klimagrößen wie Globalstrahlung und Windgeschwindigkeit bleiben unbeeinflusst. Zur Berechnung von Energieverbräuchen, die auf tatsächlichen Klimadaten (z.B. Klimacomputer) beruhen, können auch eigene Klimadaten in das Programm geladen werden.

Da für die ostdeutschen Bundesländer (außer Berlin) noch keine Testreferenzjahre vorliegen, wurden westdeutsche Klimaregionen auf Ostdeutschland übertragen. Grundlage war dabei die regionale Einteilung der DIN 4701 (1959), in der die damalige DDR noch berücksichtigt war.

Neben dem Witterungsverlauf im Laufe des Jahres ist bei der Planung von Heizungsanlagen besonders die Auslegungsaußentemperatur von Bedeutung. In dem Programm wird hierzu die Normaußentemperatur jeweiliger Standorte der DIN 4701 (1986) verwendet. Die entsprechenden Regionen Ostdeutschlands wurden auch hier nach der alten DIN (1959) aufgeteilt. Die dargestellten Werte wurden jedoch um 2 K erhöht, damit sie dem heutigen Stand der DIN 4701 entsprechen.

Um dem Benutzer eine einheitliche Klimakarte zur genauen Definition seines Klimastandortes anzubieten, wurden aus den Regionen des Testreferenzjahres und den Normregionen der DIN 4701 Schnittmengen gebildet, die dem Programmbenutzer innerhalb des Front-End-Systems zur Auswahl angeboten werden. Entstanden zu kleinflächige Teilgebiete, wurden sie unter Beachtung der Normaußentemperatur benachbarten TRY-Regionen zugeordnet. Abbildung 3.2-9 gibt eine der drei resultierenden Klimakarten (Programmgraphik) wieder.

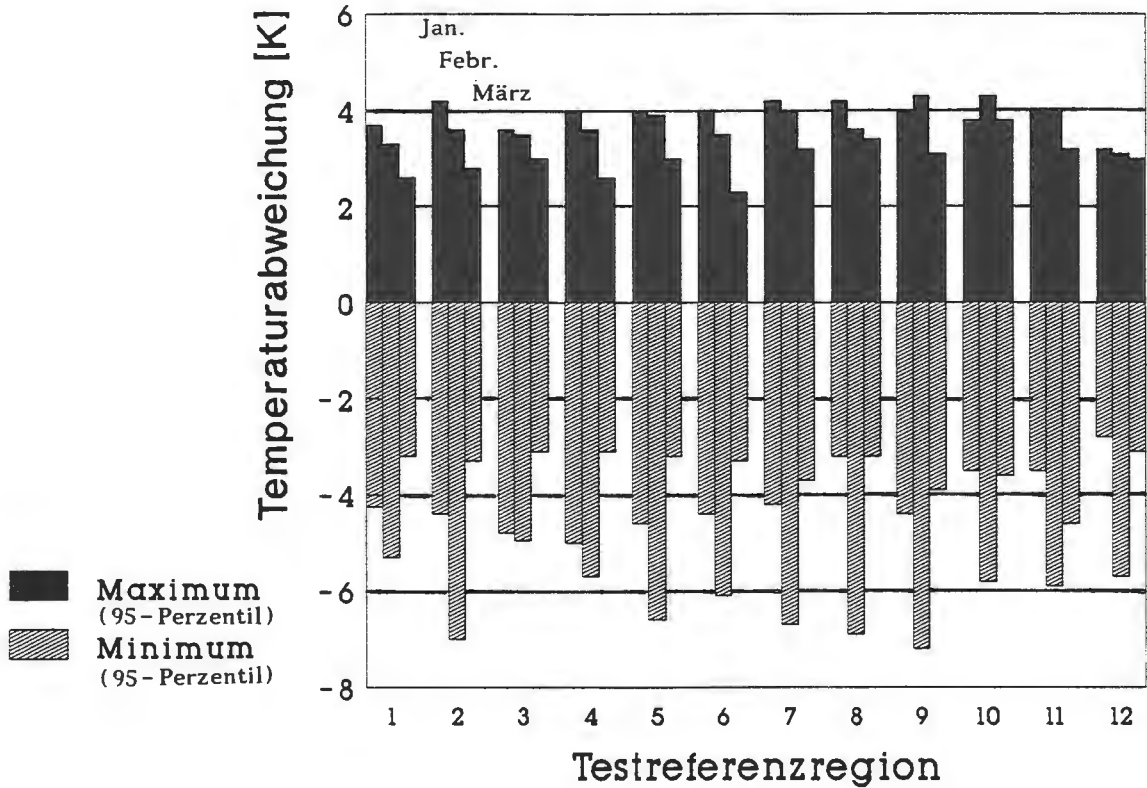


Abbildung 3.2-8: Temperaturabweichungen vom Monatsmittelwert über 30 Jahre in der BRD (nach Daten des Deutschen Wetterdienstes, (DWD 1991)).

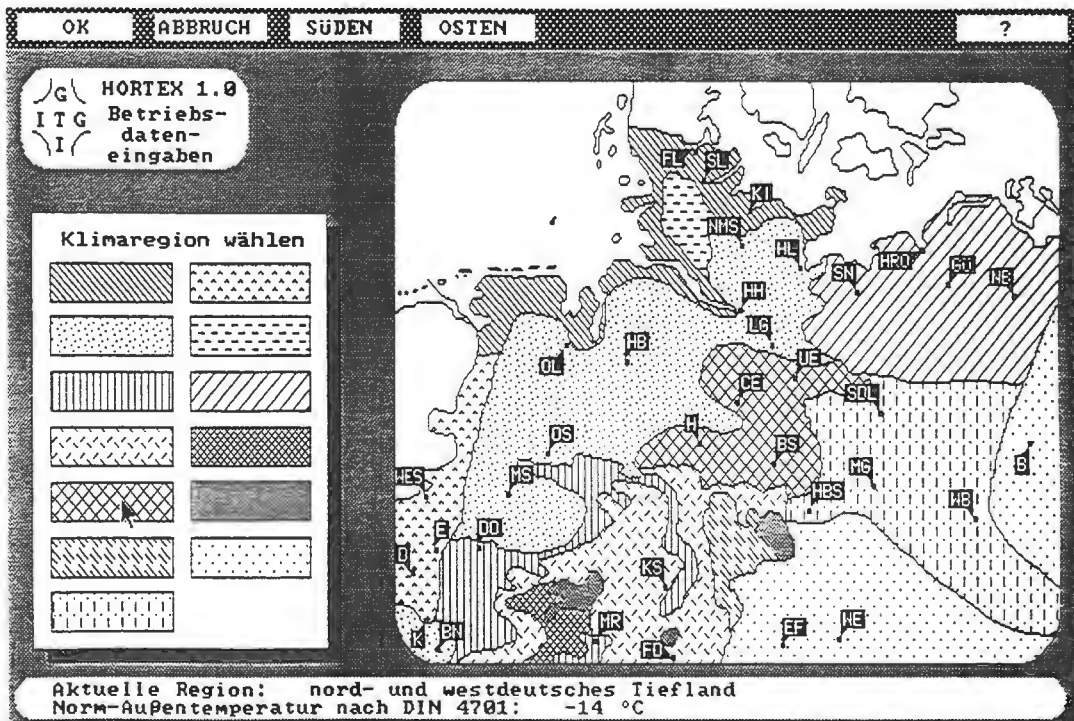


Abbildung 3.2-9: Durch Verknüpfung von TRY-Regionen und Normregionen der DIN 4701 entwickelte Klimadatenkarte (BRD-Nord)

### 3.2.3.2.2 Wärmeverbrauch

Die Abschätzung des für Planungs- und Beratungsaufgaben wichtigen Wärmeverbrauchs geschieht durch den Einsatz von arithmetischen Gleichungen. Es werden Berechnungen durchgeführt, die auf einem flächenbezogenen  $k'$ -Modell aufbauen, wobei sich der Energieverbrauch aus der notwendigen Energie zur Anhebung der Innentemperatur des unbeheizten Gewächshauses auf den gewünschten Sollwert ergibt. Abbildung 3.2-10 stellt das implementierte Gesamtmodell dar. Die genauen Berechnungsverfahren werden mit den Gleichungen 3.2-19 bis 3.2-39 beschrieben. Dabei kommen Simulationsberechnungen auf Stundenbasis mit den oben aufgeführten Klimadaten zum Einsatz.

Ausgehend von einem auf 4 m/s Windgeschwindigkeit bezogenen  $k'$ -Wert für einfach bedachte Gewächshäuser von 7,56 (W/(m<sup>2</sup>K)) und der bei VON ZABELTITZ (1986, S.182) beschriebenen Regressionsgleichung ( $k' = 0,35 \cdot v_w + 6,16$ ) kann eine allgemeingültige Korrekturgleichung zur Berücksichtigung der Windgeschwindigkeit auf den Wärmeverbrauch von Gewächshäusern entwickelt werden. Unter der Annahme, daß sich der Einfluß der Windgeschwindigkeit proportional zur Isolierungswirkung des Bedachungsmaterials verhält und bei direkter Einbeziehung des auf 4 m/s Windgeschwindigkeit normierten  $k'$ -Wertes, gilt die Gleichung 3.2-20 mit den Koeffizienten  $x_1$ ,  $x_2$  und  $x_3$  ( $x_3 = 6,16 - 7,56$ ).

Die Berücksichtigung der Globalstrahlung erfolgt mit Hilfe des von DAMRATH (1980, S. 6f) vorgeschlagenen  $\eta$ -Wertes. Andere Wärmequellen werden in voller Höhe zur Temperaturerhöhung verrechnet.

Werden in den Gewächshäusern Energieschirme eingesetzt, werden die bei MÜLLER (1987, S. 66) angegebenen Einsparwerte als Standard vorgehalten. Nach den von MÜLLER (1987, S. 102, 108ff) angegebenen Meßwerten und bei angenommener Linearität zwischen  $k'$ -Wert des Gewächshauses und prozentualer Energieeinsparwirkung eines Energieschirmes wurde das in Abbildung 3.2-11 (Mitte) und mit Gleichungen 3.2-22 bis 3.2-24 dargestellte Modell entwickelt. Auf einen ähnlichen Ansatz im Bereich der permanenten Energiesparmaßnahmen (z.B. Noppenfolie) wurde aufgrund fehlender Daten verzichtet. Die Eingabewerte für Energiesparmaßnahmen in Gleichung 3.2-19 müssen sich daher auf die im Betrieb tatsächlich herrschenden Bedingungen beziehen. Als Standard werden die von VON ZABELTITZ (1986, S. 246) angegebenen Energiesparwirkungen vorgehalten.

Bei der Berücksichtigung des Kunstlichteinsatzes wird mit den von MEYER J. (1989) für verschiedene Lampenarten angegebenen Strahlungsausbeuten gerechnet. Die Korrektur des Wärmeverbrauches durch unterschiedliche Heizungssysteme erfolgt mit den von TANTAU (1983, S. 184) angegebenen Tabellenwerten.

Bei der Berücksichtigung der Tag-Nacht-Energiespeicherung wurde davon ausgegangen, daß erstens die Speicherwirkung von den im Gewächshaus herrschenden Temperaturunterschieden zwischen Tag und Nacht abhängt, zweitens jedes Gewächshaus seine spezifische Speicherkapazität besitzt und drittens bei hoher Heizleistung ( $\Delta\vartheta_{(i-a)} \geq 25$ ) die Tag-Nacht-Speicherwirkung vollends im k'-Wert enthalten ist. Es wurde ein Meta-Modell entwickelt, welches die aufgeführten Faktoren in einem linearen Wirkungsspektrum verwendet und das den jeweiligen Betriebsbedingungen angepaßt werden kann. Die Speicherwirkung führt dabei zu einer Anhebung der Innentemperatur des Gewächshauses in der Nacht. Die Gewächshausspeicherkapazität wird durch die höchstmögliche Temperaturanhebung (im Durchschnitt einer Nacht) im Laufe des Jahres in einem ungeheizten Gewächshaus beschrieben. Sie stellt den Justierwert des Speichermodells dar. Aufgrund von Expertenangaben wird im Programm 7 K als Standard für übliche Produktionsgewächshäuser vorgegeben. Je nach Tag-Nacht-Innentemperaturverlauf und Heizlast steht diese maximal mögliche Speicherkapazität ( $\vartheta_{Sp,max}$ ) mehr oder weniger energetisch zur Verfügung. Abbildung 3.2-11 (unten) stellt die Wirkungsweise anhand zweier Beispiele graphisch dar. Aufgrund des aktuellen Speicherpotentials ( $\vartheta_{Sp,pot}$ ), welches mit Hilfe der Gewächshausspeicherkapazität und dem Jahresverlauf der Innentemperaturen des unbeheizten Gewächshauses ( $\vartheta_{i,oH}$ ) mit den Gleichungen 3.2-37 u. 3.2-38 für jede Nacht

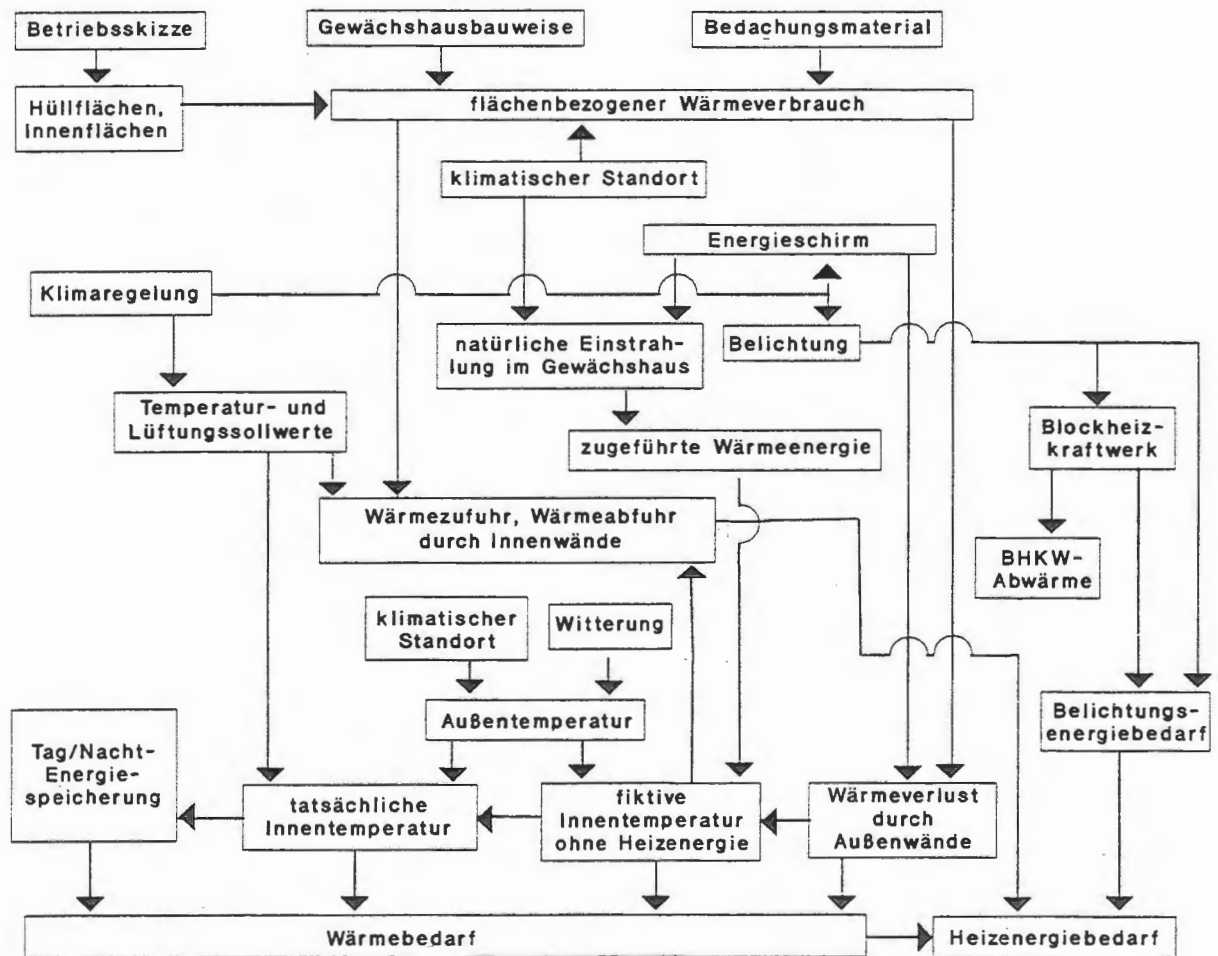


Abbildung 3.2-10: Das Energiebedarfsmodell des Expertensystems HORTEx.

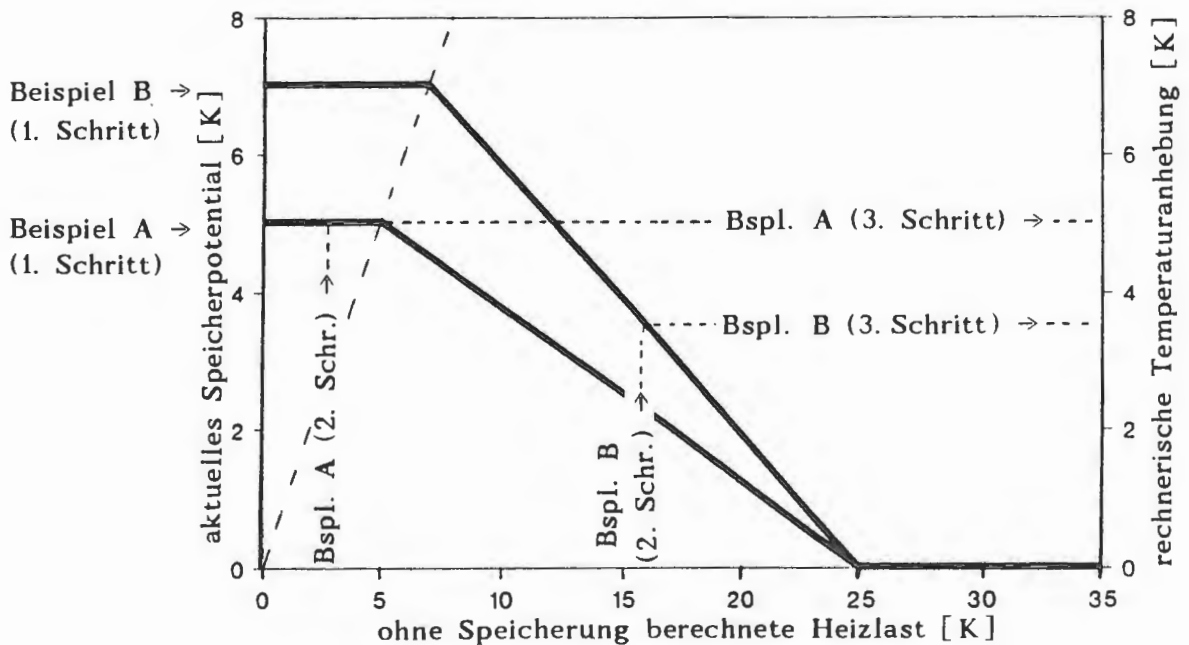
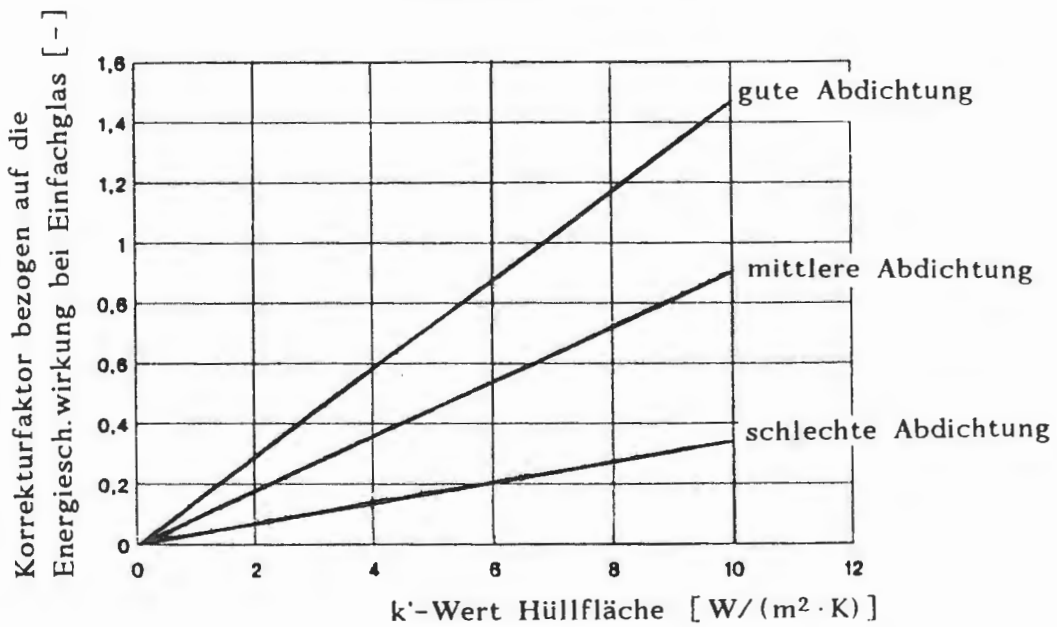
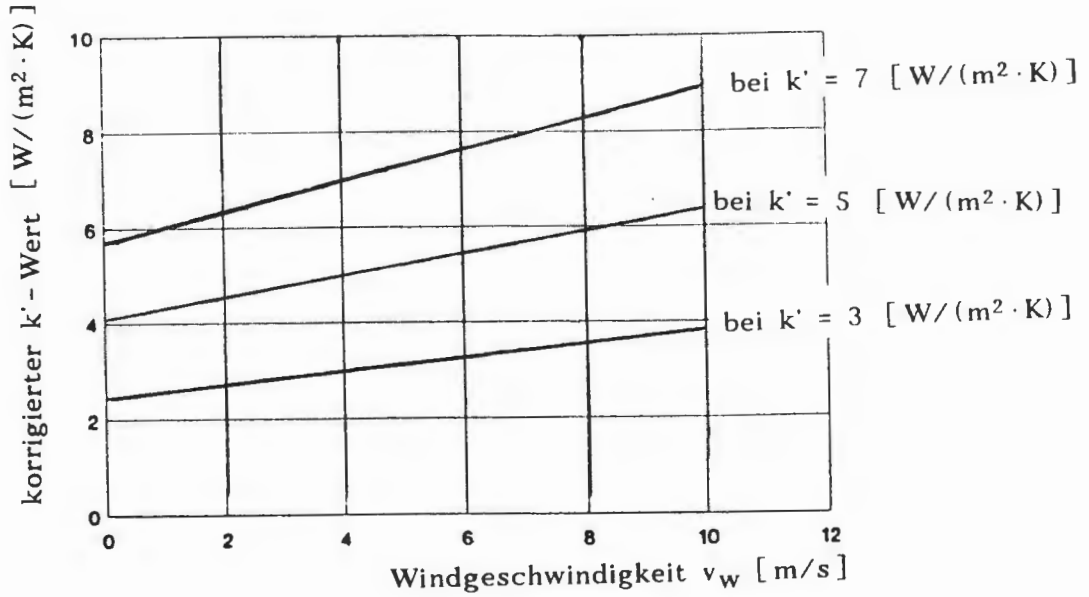


Abbildung 3.2-11: Einsatz arithmetischer Gleichungen zur Anpassung des  $k'$ -Modells an betriebspezifische Situationen.

berechnet wird, werden für jede Nachtstunde des Jahres die in der Abbildung eingezeichneten Interpolationsgeraden gebildet. Sie besagen, daß, wenn die ohne Speicherung berechnete Heizlast ( $\vartheta_i - \vartheta_{oH}$ ) gleich oder kleiner als das aktuelle Speicherpotential ist, die mit dem Simulationsmodell berechneten Innentemperaturen für das unbeheizte Gewächshaus um das aktuelle Speicherpotential angehoben werden (d.h. das Gewächshaus muß nicht beheizt werden) (Beispiel A). Beträgt die Heizlast 25 K oder darüber, erfolgt keine Korrektur der Innentemperatur. Im Zwischenbereich wird linear interpoliert (Beispiel B). Das Speichermodell wird im Programm durch zwei hintereinandergeschaltete Simulationsläufe realisiert. Die genauen Berechnungsformeln werden mit den Gleichungen 3.2-36 bis 3.2-38 beschrieben.

Im folgenden werden die Berechnungsgrundlagen des Energieverbrauchmodells zusammenfassend dargestellt.

Berechnung des k'-Wertes für die Außenflächen eines Gewächshaussegmentes

$$k'_{F,St} = k'_{TF_1,St} \cdot (1 - EE_{TF_1}) \cdot A_n + k'_{TF_2,St} \cdot (1 - EE_{TF_2}) \cdot (1 - A_n) \quad (3.2-19)$$

Es ist:

$$k'_{F,St} = \text{k'-Wert einer Fläche bei 4 m/s Windgeschwindigkeit} \left[ \frac{W}{m^2 \cdot K} \right]$$

$$k'_{TF_{1,2},St} = \text{k'-Wert einer Teilfläche bei 4 m/s Windgeschwindigkeit} \left[ \frac{W}{m^2 \cdot K} \right]$$

$$EE_{TF_{1,2}} = \text{Energieeinsparung durch Wärmedämmmaßnahmen an den Teilflächen } TF_1, TF_2 \text{ [-]}$$

$$A_n = \text{Anteil der Teilfläche } TF_1 \text{ an der Gesamtfläche } F \text{ [-]}$$

Unter Berücksichtigung der Windgeschwindigkeit ergibt sich

$$k'_F = k'_{F,St} + \frac{k'_{F,St}}{x_1} \cdot (x_2 \cdot v_w + x_3) \quad (3.2-20)$$

Es ist:

$$k'_F = \text{k'-Wert der Fläche (korrigiert um Windgeschwindigkeit)} \left[ \frac{W}{m^2 \cdot K} \right]$$

$$v_w = \text{Windgeschwindigkeit (Jahresdurchschnitt)} \left[ \frac{m}{s} \right]. \text{ Bei Innenflächen } v_w = 0.$$

$$x_1 = 7,56 \text{ [-]} \quad x_2 = 0,35 \left[ \frac{s}{m} \right] \quad x_3 = -1,4 \text{ [-]}$$

Die Berechnung des k'-Wertes der Hüllfläche (außen) geschieht mit

$$k'_a = \frac{\sum_{n \in HF_a} (k'_{F_n} \cdot A_n)}{\sum_{n \in HF_a} A_n} \quad (3.2-21)$$



Es ist:

$k'_a$  = k'-Wert der Außenflächen eines Gewächshaussegmentes  $\left[ \frac{W}{m^2 \cdot K} \right]$

$A_n$  = Fläche n  $[m^2]$

$HF_a$  = Menge aller Hüllflächen (außen) eines Gewächshaussegmentes

Berücksichtigung eines Energieschirmes

$$EE_{ES} = \frac{EE_{ES,St}}{KF_{ES}} \cdot k'_a \quad \text{gültig für } k'_a \leq 10 \text{ und } EE_{ES} \leq 0.6 \quad (3.2-22)$$

Es ist:

$EE_{ES}$  = Energieeinsparung durch Energieschirmeinsatz  $[-]$

$EE_{ES,St}$  = Einsparpotential des eingesetzten Energieschirmmaterials bei  $k'_a = 6,8$  und dichter Anbringung  $[-]$

$KF_{ES}$  = Korrekturfaktor Energieschirm  $\left[ \frac{W}{m^2 \cdot K} \right]$

Es gilt:

$$KF_{ES} = \begin{cases} 6,8 & \text{falls Energiesch. geschlossen u. gute Abdicht.güte} \\ 11,05 & \text{falls Energiesch. geschl. u. mittlere Abdicht.güte} \\ 29,43 & \text{falls Energiesch. geschl. u. schlechte Abdicht.güte} \\ \text{nicht definiert sonst} & (EE_{ES} = 0) \end{cases} \quad (3.2-23)$$

Als Schirmschaltzeitpunkte werden bei einer Regelstrategie nach der Uhrzeit die vom Benutzer anzugebenden Öffnungs- und Schließzeiten verwendet. Bei Regelung nach der Einstrahlung wird der Energieschirm geöffnet, wenn

$$SP_{ES} < \dot{q}_{GS} \cdot GIPAR \cdot D_{GH} \quad (3.2-24)$$

ist. Es ist:

$\dot{q}_{GS}$  = Globalstrahlung  $\left[ \frac{W}{m^2} \right]$

$GIPAR$  = Umwandlungsfaktor Globalstrahl. in PAR  $\left[ \frac{W(PAR)}{W} \right]$  (Standard = 0,5)

$D_{GH}$  = Lichtdurchlässigkeit Gewächshaus  $[-]$  (Standard = 0,6)

$SP_{ES}$  = Benutzereingabe Energieschirmschaltpunkt  $\left[ \frac{W(PAR)}{m^2} \right]$

Berücksichtigung einer Belichtungsanlage

$$\dot{Q}_{KL} = \begin{cases} P_{KL} \cdot KF_{KL} & \text{falls Kunstlicht angeschaltet} \\ 0 & \text{sonst} \end{cases} \quad (3.2-25)$$

Es ist:

$\dot{Q}_{KL}$  = Wärmeeintrag durch Kunstlichtanwendung im Segment [W]

$P_{KL}$  = Installierte elektrische Leistung (Kunstlichtsystem) im Segment [W]

$KF_{KL}$  = Korrekturfaktor Kunstlichtenergienutzung [-] (Standard = 0,9)

Genauso wie bei der Energieschirmsteuerung wird bei der Kunstlichtsteuerung nach der Uhrzeit die vom Benutzer angegebene Anschlag- bzw. Ausschaltzeit verwendet. Bei der Regelung nach der Lichtsumme gilt das Licht als angeschaltet, wenn

$$Li_{Sum} > \sum_{n=1}^{m-1} \left( (\dot{q}_{GS,i_n} + Li_{KL_n}) \cdot t_{Si} \right) \quad \text{und} \quad \dot{q}_{GS,i_m} = 0 \quad (3.2-26)$$

mit

$$\dot{q}_{GS,i_n} = \begin{cases} GIPAR \cdot D_{GH} \cdot D_{ES} \cdot \dot{q}_{GS_n} & \text{falls zur Stunde n ein geschlossener Energieschirm vorhanden} \\ GIPAR \cdot D_{GH} \cdot \dot{q}_{GS_n} & \text{sonst} \end{cases} \quad (3.2-27)$$

$$Li_{KL_n} = \begin{cases} \frac{P_{KL} \cdot SA_{KL}}{A_{Seg}} & \text{falls Kunstlicht zur Stunde n angeschaltet} \\ 0 & \text{sonst} \end{cases} \quad (3.2-28)$$

Es gilt:

$Li_{Sum}$  = Gewünschte Lichtsumme pro Tag  $\left[ \frac{Wh(PAR)}{m^2} \right]$

$\dot{q}_{GS,i_n}$  = Im Pflanzenbestand zur Stunde n einfallende Globalstrahlung  $\left[ \frac{W(PAR)}{m^2} \right]$

$Li_{KL_n}$  = Zur Stunde n zugeführte Kunstlichtenergie  $\left[ \frac{W(PAR)}{m^2} \right]$

$SA_{KL}$  = Strahlungsausbeute jeweiliger Lampen  $\left[ \frac{W(PAR)}{W} \right]$

$D_{ES}$  = Lichtdurchlässigkeit des Energieschirmmaterials [-]

$A_{Seg}$  = Grundfläche des Gewächshaussegmentes [m<sup>2</sup>]

$t_{Si}$  = Zeitinkrement Simulationsschritt = 1 [h]

$m$  = aktuelle Simulationsstunde (fortlaufend,  $m = 1$  bei Sonnenaufgang)

Berechnung theoretischer und tatsächlicher Innentemperaturen

$$\vartheta_{i,th} = \frac{\dot{q}_{GS} \cdot D_{GH} \cdot \eta \cdot A_{Seg}}{k'_a \cdot (1 - EE_{ES}) \cdot \sum_{n \in HF_a} A_n} + \vartheta_a \quad (3.2-29)$$

Es ist:

- $\vartheta_{i,th}$  = theoretische Innentemperatur ohne Heizung und Lüftung [ $^{\circ}\text{C}$ ]
- $\eta$  = Wirkungsgrad Energieumsetzung (sensibel/latent) im Gew.haus [-]  
(Standard = 0.7)
- $\vartheta_a$  = Außentemperatur [ $^{\circ}\text{C}$ ]

Unter Berücksichtigung des Lüftungssollwertes wird berechnet

$$\vartheta_{i,oH} = \begin{cases} \vartheta_L & \text{falls } \vartheta_{i,th} \geq \vartheta_L \text{ und } \vartheta_L \geq \vartheta_a \\ \vartheta_{i,th} & \text{falls } \vartheta_{i,th} < \vartheta_L \text{ und } \vartheta_{i,th} > \vartheta_a \\ \vartheta_a & \text{sonst} \end{cases} \quad (3.2-30)$$

Es ist:

- $\vartheta_{i,oH}$  = fiktive Innentemperatur des Gewächshaussegmentes ohne Heizung [ $^{\circ}\text{C}$ ]
- $\vartheta_L$  = Lüftungssollwert für das Gewächshaussegment [ $^{\circ}\text{C}$ ]

Die tatsächliche Innentemperatur  $\vartheta_i$  ergibt sich zu

$$\vartheta_i = \begin{cases} \vartheta_{i,S} & \text{falls } \vartheta_{i,oH} \leq \vartheta_{i,S} \\ \vartheta_{i,oH} & \text{sonst} \end{cases} \quad (3.2-31)$$

beim Einsatz einer Tag-Nacht-Regelstrategie zu

$$\vartheta_i = \begin{cases} \vartheta_{i,S_{Tag}} & \text{falls } \dot{q}_{GS} > 0 \text{ und } \vartheta_{i,oH} \leq \vartheta_{i,S_{Tag}} \\ \vartheta_{i,S_{Nacht}} & \text{falls } \dot{q}_{GS} = 0 \text{ und } \vartheta_{i,oH} \leq \vartheta_{i,S_{Nacht}} \\ \vartheta_{i,oH} & \text{sonst} \end{cases} \quad (3.2-32)$$

und bei der Temperatursummenregelung zu

$$\vartheta_i = \begin{cases} \vartheta_{i,oH} & \text{falls } \vartheta_{i,r} \leq \vartheta_{i,oH} \\ \vartheta_{i,Min} & \text{falls } \vartheta_{i,r} > \vartheta_{i,oH} \text{ und } \vartheta_{i,r} < \vartheta_{i,Min} \\ \vartheta_{i,r} & \text{sonst} \end{cases} \quad (3.2-33)$$

mit

$$\vartheta_{i,r} = \frac{\vartheta_{i,Sum} - \sum_{n=1}^{m-1} (\vartheta_{i,n} \cdot t_{Si})}{t_{Rest}} \quad (3.2-34)$$

Es ist:

- $\vartheta_i$  = tatsächliche Innentemperatur des Gewächshaussegmentes [°C]
- $\vartheta_{i,n}$  = tats. Innentemp. des Gewächshausseg. zur Stunde n [°C]
- $\vartheta_{i,S}$  = Sollwert Innentemperatur [°C]
- $\vartheta_{i,S_{Tag/Nacht}}$  = Sollwert Innentemp. Tag ( $\dot{q}_{GS} > 0$ ) bzw. Nacht ( $\dot{q}_{GS} = 0$ ) [°C]
- $\vartheta_{i,Min}$  = Minimumwert bei der Temperatursummenregelung [°C]
- $\vartheta_{i,Sum}$  = Sollwert Temperatursumme pro Tag [°C·h]
- $\vartheta_{i,r}$  = rechnerischer Sollwert [°C]
- m = aktuelle Simulationsstunde (fortlaufend, m = 1 bei Sonnenaufgang)
- $t_{Rest}$  = Restzeit bis Tagesanfang [h] . Ergibt sich aus  $(25 - m) \geq 1$ .

Berechnung der durch Innenflächen zu/abgeführten Wärmeenergie

$$\dot{Q}_{IW} = \begin{cases} \sum_{n \in HF_i} (k'_{Fn} \cdot A_n \cdot (\vartheta_{i,an} - \vartheta_i)) & \text{falls } HF_i \neq \emptyset \\ 0 & \text{sonst} \end{cases} \quad (3.2-35)$$

Es ist:

- $\dot{Q}_{IW}$  = Durch Innenwände zu/abgeführte Wärmeenergie [W]
- $HF_i$  = Menge aller Hüllflächen (innen) eines Gewächshaussegmentes
- $\vartheta_{i,an}$  = Innentemperatur des angrenzenden Gewächshaussegmentes

Berücksichtigung der Tag-Nacht-Energiespeicherung

$$\Delta\vartheta_{Sp} = \begin{cases} \Delta\vartheta_{Sp,pot} & \text{falls } \Delta\vartheta_{Sp,pot} \geq 25 \text{ und } \dot{q}_{GS} > 0 \\ \Delta\vartheta_{Sp,pot} & \text{falls } \vartheta_i - \vartheta_{i,oH} \leq \Delta\vartheta_{Sp,pot} < 25 \text{ und } \dot{q}_{GS} > 0 \\ \frac{\Delta\vartheta_{Sp,pot} \cdot (\vartheta_i - \vartheta_{i,oH} - 25)}{\Delta\vartheta_{Sp,pot} - 25} & \text{falls } 0 < \Delta\vartheta_{Sp,pot} < \vartheta_i - \vartheta_{i,oH} < 25 \text{ und } \dot{q}_{GS} > 0 \\ 0 & \text{sonst} \end{cases} \quad (3.2-36)$$

mit

$$\Delta\vartheta_{Sp,pot} = \frac{Z_d}{\max(Z_{2...365})} \cdot \Delta\vartheta_{Sp,max} \quad (3.2-37)$$

$$Z_d = \overline{\vartheta_{i,oH,Tag_{d-1}}} - \overline{\vartheta_{i,oH,Nacht_d}} \quad (3.2-38)$$

Es ist:

$\Delta\vartheta_{Sp}$  = Innentemperaturanhebung bedingt durch Tag-Nachtspeicherung in einem beheizten Gewächshaus [°C]

$\Delta\vartheta_{Sp,pot}$  = aktuelles Speicherpotential des unbeheizten Gewächshauses [°C]

$\Delta\vartheta_{Sp,max}$  = Maximal mögliche durchschnittliche Temperaturanhebung nachts in einem unbeheizten Gewächshaus [°C] (Standard = 7)

$Z_d$  = Differenz der Innentemperatur (Mittelwert) in einem unbeheizten Gewächshaus zwischen Tag ( $\dot{q}_{GS} > 0$ ) und Nacht ( $\dot{q}_{GS} = 0$ ) am aktuellen Jahrestag d [°C]

Berechnung des Jahresheizenergieverbrauches

$$Q_{Seg} = \sum_{n=1}^{8760} \left( (\vartheta_{i,n} - \vartheta_{i,oH,n} - \Delta\vartheta_{Sp,n}) \cdot k_a \cdot \sum_{r \in HFa} A_r \cdot (1 - EE_{ES,n}) - \dot{Q}_{IW,n} - \dot{Q}_{KL,n} \right) \cdot t_{Si} \quad (3.2-39)$$

Es ist:

$Q_{Seg}$  = Jahresheizenergieverbrauch des Gewächshaussegmentes [Wh]

n = Stunde des Jahres

### 3.2.3.2.3 Auslegungsleistung

Es wird folgende Berechnung durchgeführt:

$$\dot{Q}_{\text{Seg,Ausl}} = k'_a \cdot \sum_{n \in \text{HF}_a} A_n \cdot (\vartheta_{i,\text{Ausl}} - \vartheta_{a,\text{Ausl}}) \cdot (1 - EE_{\text{ES}} \cdot BF_{\text{ES}}) - \dot{Q}_{\text{IW,Ausl}} - P_{\text{KL}} \cdot KF_{\text{KL}} \cdot BF_{\text{KL}} \quad (3.2-40)$$

Es ist:

$\dot{Q}_{\text{Seg,Ausl}}$  = Heizbedarf Gewächshaussegment im Auslegungsfall [W]

$\vartheta_i$  = Innentemperatur im Auslegungsfall [°C]

$\vartheta_a$  = Außentemperatur im Auslegungsfall [°C]

$\dot{Q}_{\text{IW,Ausl}}$  = Wärmegewinn, Verlust durch Innenflächen im Auslegungsfall [W]

$BF_{\text{ES}}$  = Berücksichtigungsfaktor für Energieschirmeinsatz (0, 0,5 oder 1) [-]

$BF_{\text{KL}}$  = Berücksichtigungsfaktor für Kunstlichteinsatz (0, 0,5 oder 1) [-]

Neben dem allgemeinen Planungsfall mit  $\vartheta_i$  = Auslegungstemperatur und  $\vartheta_a$  = Normaußentemperatur nach DIN 4701 wird zur eingehenderen Analyse des Planungsfall es auch eine Auslegung mit  $\vartheta_i$  = Sicherheitstemperatur und  $\vartheta_a$  = Normaußentemperatur nach DIN 4701 berechnet. Die Sicherheitstemperatur ist dabei die Temperatur, die im Gewächshaus auf keinen Fall unterschritten werden sollte. Ferner wird noch ein 'Nullgradfall' ( $\vartheta_a = 0$  [°C]) und ein 'Extremfall' ( $\vartheta_a$  = Extremtemperatur TRY) berechnet.

Für alle Steh- und Giebelwände (außen) werden getrennte Wärmebedarfsrechnungen durchgeführt, wobei beim Wärmebedarf für die Stehwandabschirmung ein bestimmter Anteil der Dachfläche (Standard 1 m) hinzugenommen wird. Der Gesamtenergiebedarf eines Gewächshausbetriebes errechnet sich aus der Addition des Wärmebedarfes der einzelnen Segmente unter Berücksichtigung von Zu- leitungsverlusten.

### **3.2.3.3 ... im Bereich Planung der Heizenergiebereitstellung**

#### **3.2.3.3.1 Modularisierung**

Die Planung der Heizenergiebereitstellung wird von ökonomischen und technischen Einflußfaktoren gleichermaßen beeinflusst. Die wichtigsten Größen sind die Energiepreissituation, die aktuellen Investitionskosten für die Anlagenerstellung, Energieträgerlieferbedingungen und technische Realisierungsmöglichkeiten einzelner Anlagenkonzeptionen.

Das Expertensystem HORTEX verknüpft diese Einflußfaktoren miteinander und liefert Vorschläge für die Energiebereitstellung in einer konkreten Planungssituation. Hierzu wurde der Themenbereich in fünf verschiedene Aufgabenschwerpunkte unterteilt:

- Energieträgerauswahl
- Anlagenkonzeption
- Auswahl möglicher Planungsvarianten
- ökonomische Bewertung gefundener Lösungen
- sonstige Bewertung gefundener Lösungen

Die Modellierung und Darstellung des Fachwissens erfolgte bei den ersten drei Schwerpunkten mit Hilfe von Produktionsregeln, beim vierten Aufgabenschwerpunkt mit einem Datenbankansatz gekoppelt mit prozeduralen Berechnungen. Der fünfte Schwerpunkt wurde mit Hilfe eines Hypertextkonzeptes abgebildet.

#### **3.2.3.3.2 Beschreibung**

##### Energieträgerauswahl

Ziel dieses Programmteiles ist es, mögliche und in einer Planungssituation adäquate Energieträger aus einem Energieträgerpool auszuwählen und Einsatzrestriktionen und Kapazitätsgrenzen herauszufinden.

Anhand spezifischer Kriterien wird der Einsatz der im Programm berücksichtigten Energieträger überprüft. Abbildung 3.2-12 gibt die Struktur des implementierten Produktionsregelbaumes wieder. Die Abbildung 3.2-13 zeigt ein Beispiel für eine Regel aus der Wissensbasis.

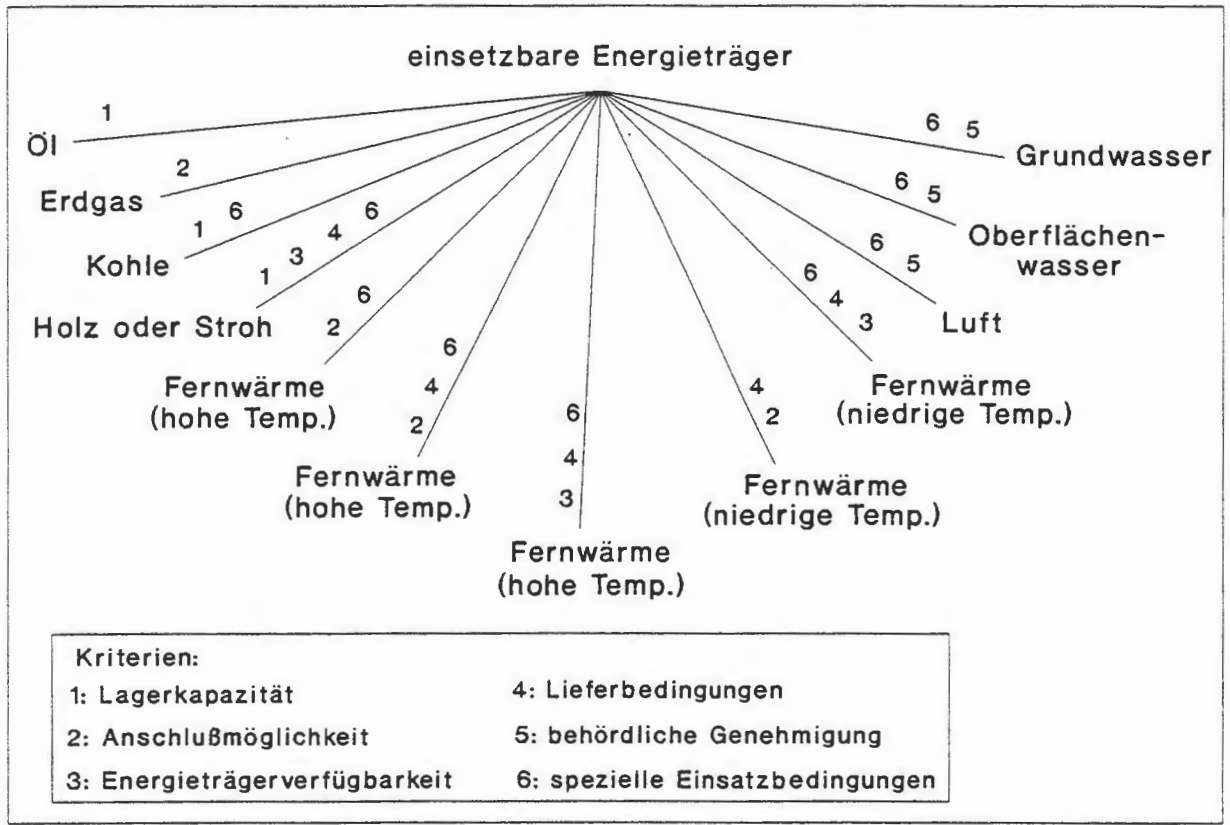


Abbildung 3.2-12: Graphische Darstellung eines Teils der Wissensbasis 'Energieträgerauswahl'

```
knoten(verfuegbar,hat_moeglichkeit,[oel],_):-  
  wenn__(oellagerung_moeglich,qq,ja).  
  
knoten(verfuegbar,hat_moeglichkeit,[gas],_):-  
  wenn__(erdgasanschluss_moeglich,qq,ja).  
  
knoten(verfuegbar,hat_moeglichkeit,[ht_fern],_):-  
  anlagendimension(Dim),  
  Abstand = trunc(((Dim/1E6) * 500)/100) * 100,  
  format(Abst,"%1.0",Abstand),  
  wenn__(fernwaermeanschluss_moeglich_meter,Abst,ja),  
  wenn__(fernwaermeanschluss_stelle,qq,vorlaufleitung).
```

Abbildung 3.2-13: Beispiel mehrerer Regeln aus der Wissensbasis 'Energieträgerauswahl'



Anschließend wird untersucht, ob in Frage kommende Energieträger aufgrund irgendwelcher Restriktionen nur eingeschränkt zur Verfügung stehen. Es werden sowohl Einsatzgrenzen für den monovalenten, bivalent-parallelen und bivalent-alternativen Einsatz der Energieträger in Grundlast und Spitzenlast festgelegt. Die Umwandlung von mengenorientierten Kapazitätsgrenzen in leistungsbezogene Auslegungsbereiche erfolgt für jeden Fall pauschal mit dem aus der Energieverbrauchsrechnung resultierenden Jahresenergiebedarf und dem entsprechenden Jahresheizenergieanteil für parallele und alternative Betriebsweisen. Abbildung 3.2-14 zeigt die Struktur des implementierten Und-Oder-Graphen. Er wird für jeden zur Verfügung stehenden Energieträger durchlaufen. Abbildung 3.2-15 zeigt ein typisches Beispiel einer Regel aus der Wissensbasis. Anstelle der Angabe mehrerer Oder-Knoten werden hierbei unterschiedliche Oder-Kanten, die alle zum gleichen Zielknoten führen, eingesetzt. Dieses Verfahren führt zu einer Reduzierung des Programmcodes (da nur ein Zielknoten definiert werden muß). Auf der anderen Seite erschwert es die Lesbarkeit der Wissensbasis. Das Ergebnis ist aber in beiden Fällen das gleiche.

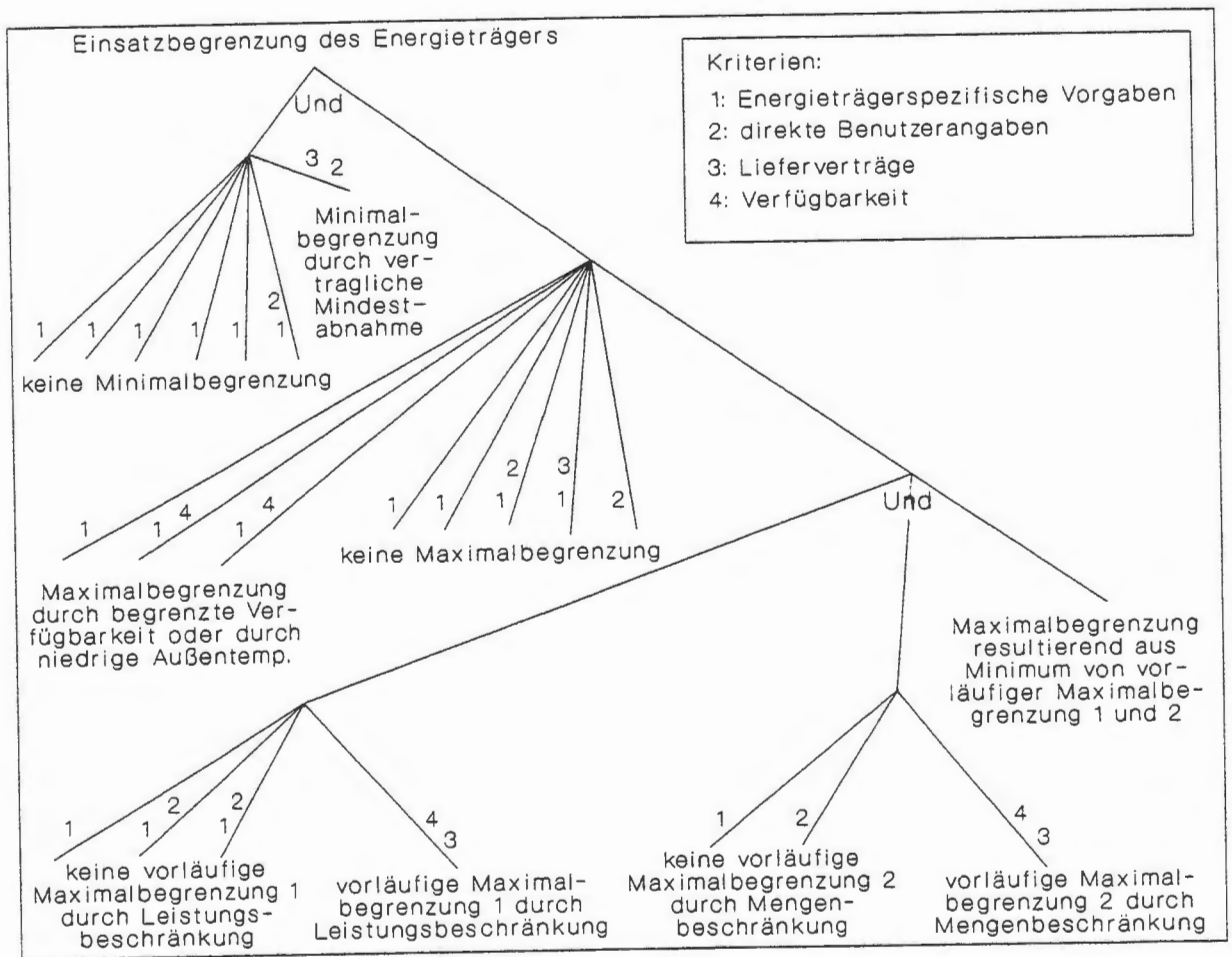


Abbildung 3.2-14: Graphische Darstellung eines Teils der Wissensbasis 'Energieträgerauswahl'

```
knoten(maximum, hat_moeglichkeit, [kein_maximum], Loesung) :-  
  vorhanden(Loesung, x, x, kohle), ! OR  
  vorhanden(Loesung, x, x, oel), ! OR  
  vorhanden(Loesung, x, x, gas),  
  wenn__(maximalbegrenzung_vorhanden, gas, nein), ! OR  
  vorhanden(Loesung, x, x, gas),  
  wenn__(maximalbegrenzung_vorhanden, gas, ja),  
  wenn__nicht(gasliefervertrag_ist, qq, teilabschaltbar), ! OR  
  eingesetzter_traeger(Loesung, x, x, Traeger),  
  not(Traeger = gas),  
  konvert(Tr, Traeger),  
  wenn__(maximalbegrenzung_vorhanden, Tr, nein), !.
```

Abbildung 3.2-15: Beispiel einer Regel aus der Wissensbasis 'Energieträgerauswahl'

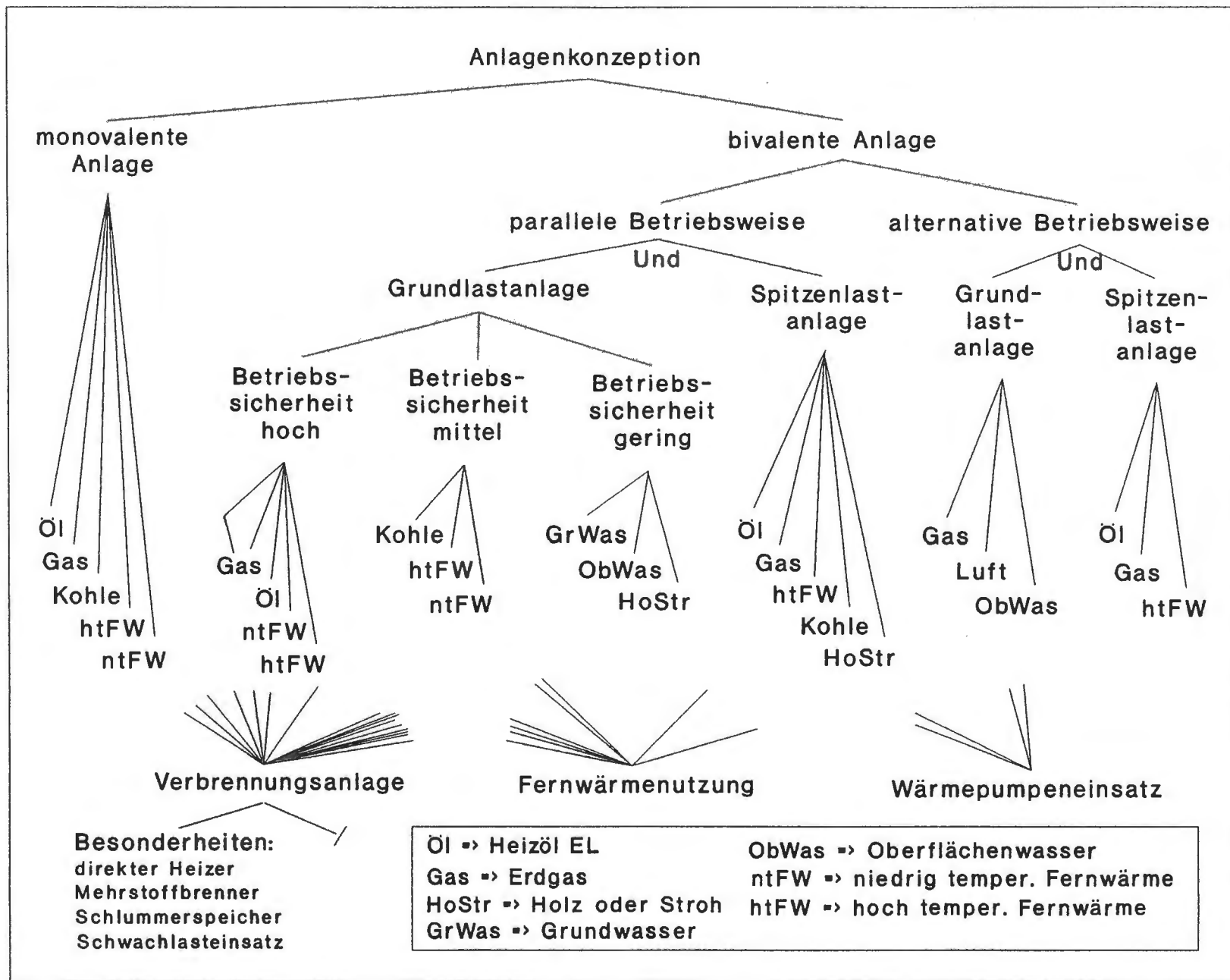
In einem letzten Schritt werden alle Energieträger, die weniger als 5 % der Auslegungsleistung abdecken können, aus dem weiteren Planungsprozeß herausgenommen.

### Anlagenkonzeption

Innerhalb dieses Programmteils werden mögliche Heizenergieversorgungskonzepte zusammengestellt. Es kommen monovalente, bivalent-parallele und bivalent-alternative Energieversorgungskonzepte zum Einsatz. Die im Bereich bivalent-paralleler Anlagen eingesetzten Grundlastenergieerzeugeranlagen werden in drei Kategorien unterteilt: hohe Betriebssicherheit, mittlere Betriebssicherheit und geringe Betriebssicherheit.

Beim Einsatz eines Grundlastenergieerzeugers niedriger Betriebssicherheit sollte die Spitzenlast die gesamte Auslegungsleistung abdecken. Bei der Verwendung eines Energieerzeugers mit mittlerer Betriebssicherheit muß die Spitzenlast zumindest die Sicherheitsauslegungsleistung abdecken. In allen anderen Fällen wird keine Einschränkung hinsichtlich der Spitzenlastauslegung vorgenommen. Abbildung 3.2-16 gibt den implementierten Und-Oder Graphen wieder. Auf die Einzeichnung verschiedener Kriterien wurde der Übersichtlichkeit halber verzichtet. Hier sei auf den Anhang C.2 verwiesen. Abbildung 3.2-17 zeigt einen charakteristischen Ausschnitt aus der Wissensbasis.

Abbildung 3.2-16: Graphische Darstellung eines Teils der Wissensbasis 'Anlagenkonzeption'



```
knoten(spitzenlast,ist_ziel_hat_moeglichkeit,[kohle],Loesung):-  
  vorhanden(Loesung,x,x,parallel),  
  wenn_traeger_vorhanden(kohle),  
  not(vorhanden(Loesung,grundlast,x,kohle)),  
  wenn__(schon_anlagen_vorhanden,qq,ja),  
  wenn_zahl(anzahl_anlagen,qq,Zahl),  
  X = trunc(Zahl),  
  unter(X,Nr),  
  wenn__(anlage_nr,Nr,verbrennungsanlage),  
  wenn__(eingesetzter_brennstoff_anlage_nr,Nr,kohle).
```

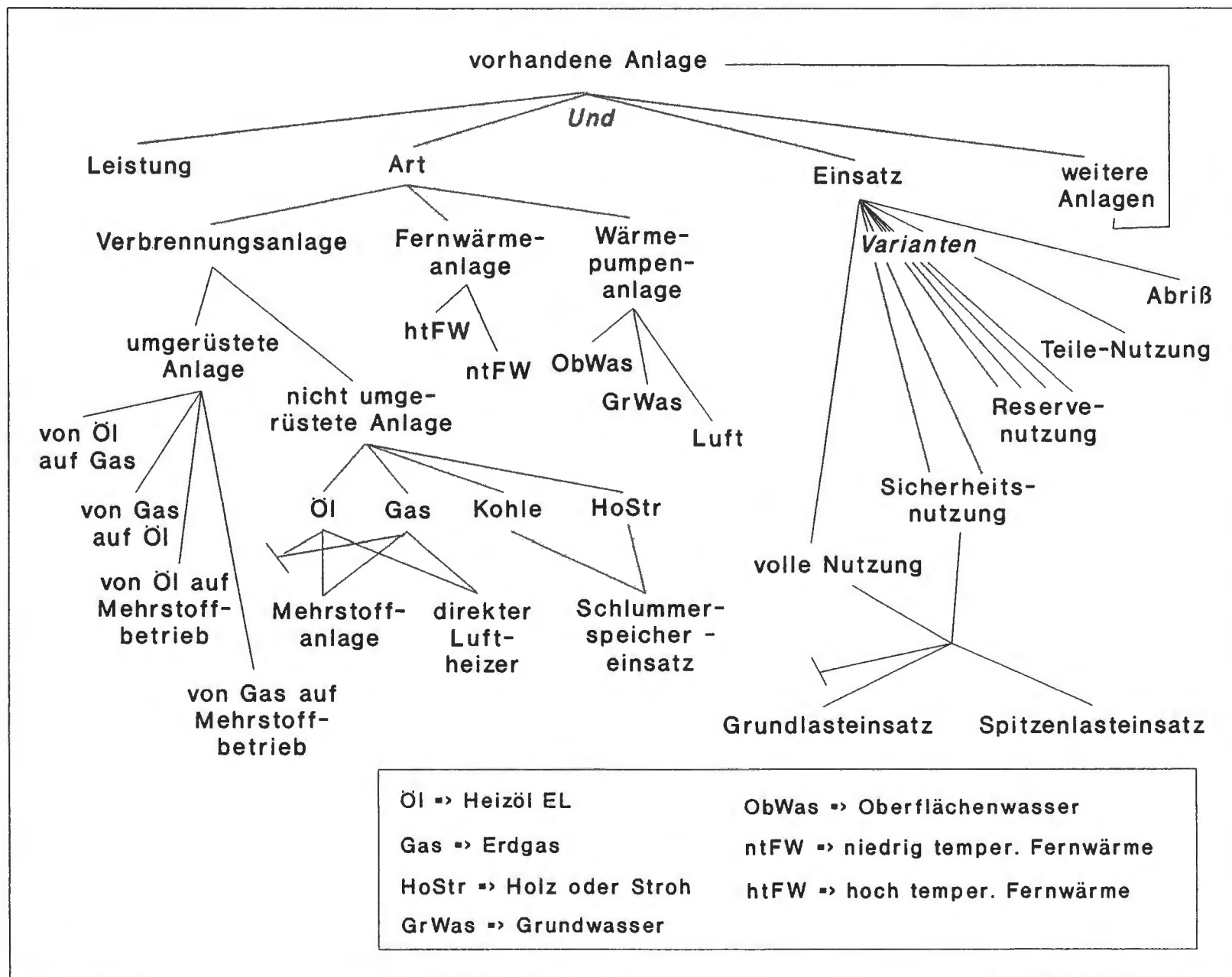
Abbildung 3.2-17: Beispiel einer Regel aus der Wissensbasis 'Anlagenkonzeption'

In einem zweiten Schritt wird versucht, vorhandene Anlagen in die definierten Energieversorgungskonzepte zu integrieren. Hierbei können vorhandene Anlagen direkt genutzt werden oder es erfolgt eine Umrüstung der Verbrennungsanlage auf einen anderen Energieträger. Als Einsatzmöglichkeiten kommen in Frage:

1. Volle Nutzung: Die vorhandene Anlage kann voll in die Planung einbezogen werden.
2. Sicherheitsnutzung: Die vorhandene Anlage sollte nur oberhalb der Sicherheitsleistung in das System integriert werden.
3. Reserve Nutzung: Die vorhandene Anlage wird nicht bei der Auslegung berücksichtigt. Sie wird aber aufgrund von Sicherheitsaspekten im Betrieb gelassen.
4. Teile Nutzung: Einzelne Teile der alten Anlage sollten für die Einrichtung der neuen Anlage genutzt werden.
5. Abriß: Die vorhandene Anlage sollte nicht weiter verwendet werden und abgerissen werden.

Die Ausweitung des Konzeptes auf mehrere vorhandene Anlagen geschieht durch die rekursive Definition eines Ableitungszweiges. Zur Zeit ist aufgrund des zur Verfügung stehenden Arbeitsspeichers die Angabe von maximal zwei vorhandenen Anlagen möglich. Abbildung 3.2-18 gibt den implementierten Und-Oder-Graphen wieder. Er wird für jede bisher in Frage kommende Energieträgerkombination vollständig durchlaufen (d.h. für jede Kombination können mehrere Planungs-fälle entstehen). Abbildung 3.2-19 zeigt eine Regel aus der Wissensbasis, in der überprüft wird, ob eine Umrüstung der Altanlage in Frage kommt.

Abbildung 3.2-18: Graphische Darstellung eines Teils der Wissensbasis 'Anlagenkonzeption'



Öl → Heizöl EL	ObWas → Oberflächenwasser
Gas → Erdgas	ntFW → niedrig temper. Fernwärme
HoStr → Holz oder Stroh	htFW → hoch temper. Fernwärme
GrWas → Grundwasser	

```
knoten(verbrennungsanlage_v,ist_ziel_hat_moeglichkeit,[umruestung],Loesung):-  
  versuch_umruesten,  
  letzte_anlage(Loesung,Nr),  str_int(Nnr,Nr),  
  not(bearbeitete_anlage(Nr)),  
  wenn_nicht(eingesetzter_brennstoff_anlage_nr,Nnr,kohle),  
  wenn_nicht(eingesetzter_brennstoff_anlage_nr,Nnr,holz_stroh),  
  wenn__(einsatz_direkte_luftheizer_anlage_nr,Nnr,nein),  
  wenn__(einsatz_mehrstoffbrenner_anlage_nr,Nnr,nein),  
  wenn_nicht(vorhandene_anlage_nr,Nnr,nur_bedingt_einsetzbar).
```

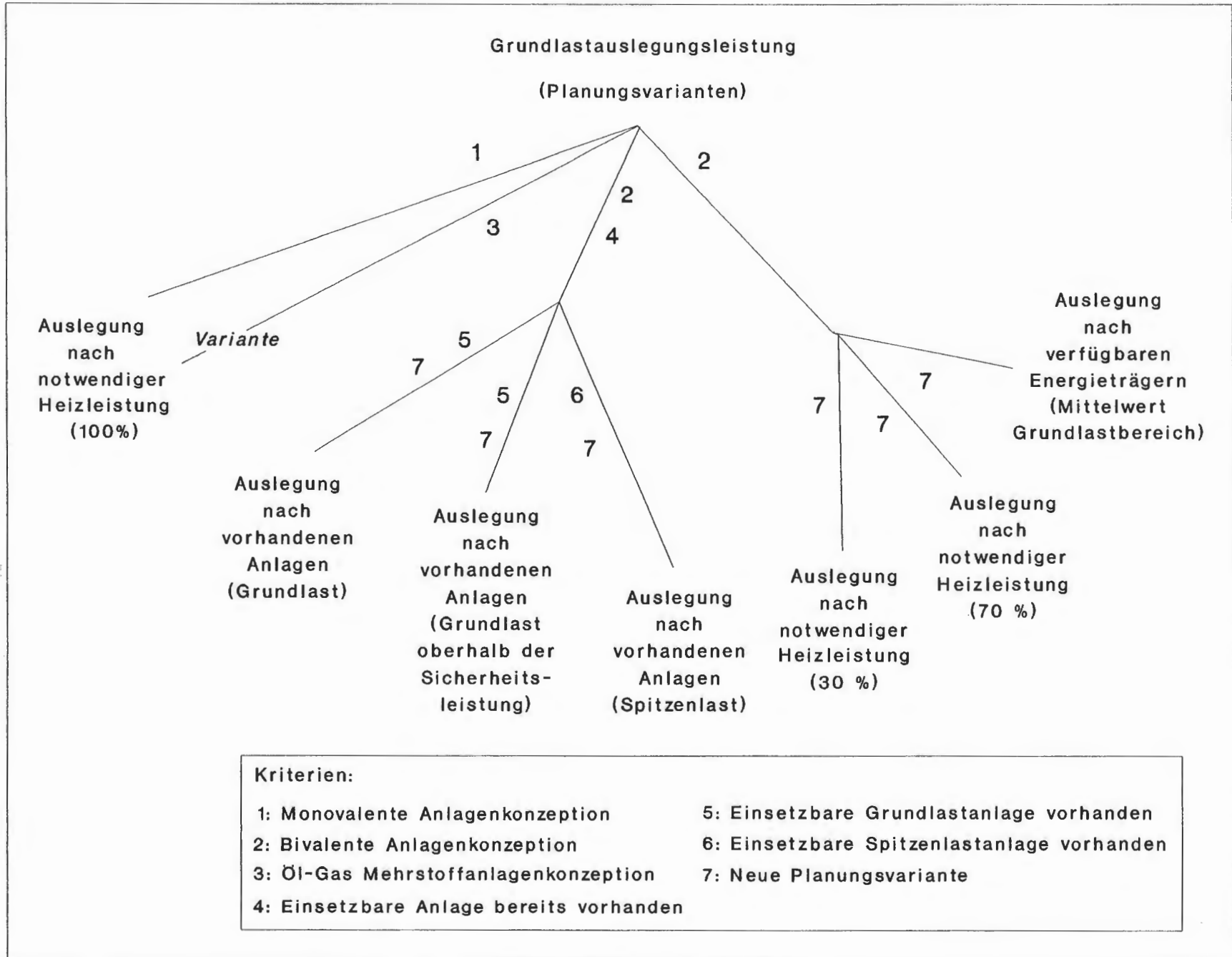
Abbildung 3.2-19: Beispiel einer Regel aus der Wissensbasis 'Anlagenkonzeption'

#### Auswahl möglicher Planungsvarianten

In einem letzten Planungsschritt werden verschiedene Auslegungs- bzw. Planungsvarianten ausgewählt und als Lösungsvorschläge dem Benutzer angeboten. Es muß für den jeweiligen Fall der notwendige Grundlastanteil, der Spitzenlastanteil und evtl. die Aufteilung der Kesselanlage bestimmt werden. Die Auswahl sinnvoller Planungsvarianten erfolgt nach dem in Abbildung 3.2-20 abgebildeten Graphen. Da auch hier eine vollständige Suche stattfindet, können mehrere Lösungen und Planungsvarianten entstehen. Unterscheidet sich eine neu entdeckte Planungsvariante um weniger als 10 % der Auslegungsleistung von einer anderen Variante, so wird sie als Lösung verworfen. Somit ist die Reihenfolge der Regeln in der Wissensbasis wichtig, da die in der Regelbasis weiter 'hinten' platzierten Auswahlregeln höchstens noch die Auslegungslücken abdecken können. Abbildung 3.2-21 verdeutlicht diesen Sachverhalt anhand eines Ausschnittes mit vier Regeln aus der Wissensbasis.

Bei bivalent-parallelen Anlagen ergibt sich die Spitzenlast aus der Differenz zwischen Grundlastanteil und Gesamtauslegungsleistung, wobei als Randbedingungen für die Grundlast notwendige Sicherheits- und Schwachlastleistungen sowie die Verfügbarkeitsgrenzen der Energieträger Beachtung finden müssen. Bei bivalent-alternativen Anlagen ist die Spitzenlast gleich der Gesamtauslegungsleistung.

Abbildung 3.2-20: Graphische Darstellung der Wissensbasis 'Planungsvarianten'



```
knoten(auslegungsfall,hat_moeglichkeit,[nach_planung],Loesung):-  
    vorhanden(Loesung,x,x,monovalent),  
    assert(auslegungsfall(100)).  
  
knoten(auslegungsfall,hat_moeglichkeit,[nach_mehrstoff],Loesung):-  
    vorhanden(Loesung,x,x,mehrstoffbrenner),  
    assert(auslegungsfall(100)),!.  
  
knoten(auslegungsfall,hat_moeglichkeit,[nach_anlage],Loesung):-  
    vorhanden(Loesung,x,x,bivalent),  
    vorhandene_anlage(_,_).  
  
knoten(auslegungsfall,hat_moeglichkeit,[nach_energie],Loesung):-  
    vorhanden(Loesung,x,x,bivalent).
```

Abbildung 3.2-21: Ausschnitt aus der Wissensbasis 'Planungsvarianten'

### Ökonomische Bewertung gefundener Lösungen

Die ökonomische Bewertung der gefundenen Planungsvarianten erfolgt durch die Angabe des erforderlichen Neu-Investitionsvolumens und durch die Angabe der jährlichen Kosten entstehend durch die Neuinvestition (berechnet aus dem Wiedergewinnungsfaktor und dem Investitionsvolumen (STORCK 1977, S. 24)), den Wartungskosten (in der Datenbank in Prozent der Investitionssumme abgespeichert) und den jährlichen Energiekosten.

Da die Investitionsvolumina für die Energiebereitstellung in entscheidendem Maße von der Leistung der erforderlichen Anlage abhängen (vgl. BYLDA 1990), wurde als charakteristische Kenngröße das Investitionsvolumen einer 0,2 MW-Anlage und einer 2,4 MW-Anlage gewählt (jeweils in DM pro kW installierte Leistung). Das Gesamtinvestitionsvolumen  $IV_{Ges}$  [DM] einer Planungsvariante wird berechnet mit

$$IV_{Ges} = \sum_{n=1}^k (IV_{EB_n} + IV_{EA_n}) + \sum_{m=1}^l IV_{Um_m} + IV_{WA,ht} + IV_{WA,nt} + \sum_{r=1}^s IV_{ZS_r} \quad (3.2-41)$$



Die Indizes bedeuten:

- GES = Gesamtanlage
- EB = Energiebereitstellung
- EA = Energieträgeranschluß
- Um = Umrüstung
- WA = Wärmeausbringung (ht = hoch temperiert, nt = niedrig temperiert)
- ZS = Zusatzeinrichtungen (Schlammerspeicher, Mehrstoffbrenner etc.)
- k = Anzahl eingesetzter Energieträger (Planung)
- l = Anzahl verwendeter Altanlagen
- s = Anzahl besonderer Zusatzeinrichtungen

Die jeweiligen Investitionsvolumina werden durch die Multiplikation der tatsächlichen Anlagengröße mit einem entsprechenden Investitionswert aus der Datenbank berechnet. Bei Anlagen zwischen 200 und 2400 kW erfolgt eine Interpolation zwischen den Datenbankwerten für die 0,2 und 2,4 MW-Anlage. Bei Anlagengrößen außerhalb dieses Bereiches wird der Investitionswert von einer 0,2 bzw. 2,4 MW-Anlage verwendet.

Die Definition der Inter- bzw. Extrapolationsgeraden lautet:

$$IV = P \cdot \left( \frac{IV_{DB0,2} - IV_{DB2,4}}{200 - 2400} \cdot P' + IV_{DB0,2} - \frac{IV_{DB0,2} - IV_{DB2,4}}{200 - 2400} \cdot 200 \right) \quad (3.2-42)$$

mit

$$P' = \begin{cases} 200 & \text{falls } P < 200 \\ 2400 & \text{falls } P > 2400 \\ P & \text{sonst} \end{cases} \quad (3.2-43)$$

Es ist:

- P = erforderliche Anlagengröße [kW]
- $IV_{DB0,2}$  = Datenbankwert Investitionsvolumen 0,2 MW-Anlage [DM/kW]
- $IV_{DB2,4}$  = Datenbankwert Investitionsvolumen 2,4 MW-Anlage [DM/kW]

Die erforderlichen Anlagengrößen ergeben sich in allen Fällen (außer bei Umrüstung) aus der für die Planung notwendigen Anlagenleistung abzüglich berücksichtbarer Altanlagen.

Für die Berechnung der im Laufe eines Jahres anfallenden Energiekosten wurde ebenfalls ein datenbankorientiertes Modell konzipiert, in dem die wesentlichen

Kennzahlen möglicher Energieversorgungsverträge und die anlagenspezifischen Charakteristika (Heizwert, Jahresbetriebswirkungsgrad, Zusatzenergiebedarf) enthalten sind. Der Energiepreis wird aufgeteilt in einen Preis für elektrische Energie, einen Preis für die gelieferte Heizenergiemenge, einen leistungsbezogenen Heizenergiebereitstellungspreis und einen absoluten Heizenergiebereitstellungspreis. Die Berechnung der Energiekosten für die einzelnen Energieträger der Heizungsanlage erfolgt mit

$$EKO = \frac{Q}{\eta_{JB} \cdot Hw} \cdot EP_{LE} + \dot{Q} \cdot EP_{LB} + EP_{AB} + \frac{Q}{\eta_{JB}} \cdot \frac{ZEB}{100} \cdot EP_{EI} \quad (3.2-44)$$


Es ist:

- $EKO$  = Energiekosten  $\left[ \frac{DM}{Jahr} \right]$   
 $Q$  = mit dem Energieträger abgedeckter Heizenergieanteil  $\left[ \frac{kWh}{Jahr} \right]$   
 $\dot{Q}$  = Anlagenleistung des Energieträgers  $[kW]$   
 $\eta_{JB}$  = Jahresbetriebswirkungsgrad Heizanlage  $[-]$   
 $Hw$  = Heizwert Energieträger  $\left[ \frac{kWh}{Liefereinheit} \right]$   
 $ZEB$  = Zusatzenergiebedarf des Energieträgers  $[\%]$   $\left( \left[ \frac{kWh (elektr.)}{kWh (Heizenergie)} \right] \right)$   
 $EP_{LE}$  = Energiepreis pro Liefereinheit  $\left[ \frac{DM}{Liefereinheit} \right]$   
 $EP_{LB}$  = Leistungsbezogener Bereitstellungspreis  $\left[ \frac{DM}{kW (install.) \cdot Jahr} \right]$   
 $EP_{AB}$  = absoluter Bereitstellungspreis  $\left[ \frac{DM}{Jahr} \right]$   
 $EP_{EI}$  = Energiepreis für im Betrieb eingesetzte elektrische Energie  $\left[ \frac{DM}{kWh (elektr.)} \right]$

### Sonstige Bewertung gefundener Lösungen

Weitere Bewertungskriterien, die nicht direkt ökonomisch erfaßbar sind, werden mit Hilfe eines Hypertextsystems dargestellt. Um eine hohe Informationsdichte zu erzielen, werden die wichtigsten Kriterien auf Bewertungspunkte reduziert, die dem Benutzer in Form einer Graphik dargestellt werden. Abbildung 3.2-22 gibt das Bewertungsschema mit den im Bereich 'Umweltschutzauflagen' angebotenen und vernetzten Hypertexten (überwiegend Graphiken) wieder.

Es bleibt anzumerken, daß das implementierte Hypertextsystem statischen Charakter hat. Das bedeutet, daß die angegebenen Texte und Graphiken nicht von der aktuellen Lösungssuche beeinflußt werden. Lediglich das im oberen Teil der Abbildung 3.2-22 dargestellte Bewertungsschema wird um die nicht in den Lösungsvarianten auftretenden Energieträger reduziert.


**HORTEX 1.0**  
 Planung  
 Energie-  
 träger

**Betrieb: BEISPIEL, Bedachung: Standard**  
**Grundfläche: 3490 [m<sup>2</sup>], Hüllfl.: 4768 [m<sup>2</sup>]**  
**Standort: Nord- und Ostseeküste, nördl. Schleswig-Holst.**  
**Auslegungsleistung: 920.4 [kW],**

Bewertungsskala: Von 2 = sehr günstig bis -2 = problematisch  (nähere Informationen durch Anklicken der Punktefelder)	St - EL	Erdgas	Kohle	Holz / Stroh	hochtemp. Abwärme	niedrigtemp. Abwärme.	Wärme- pumpe
Automatisation	1	1	-2	-2	1	1	1
Betriebssicherheit	1	1	0	-2	1	1	-2
Preisentwicklung, -stabilität	-2	-2	0	1	0	1	0
Probleme Umweltauflagen	0	0	-1	-1	1	1	0
Umweltbelastung	0	1	-1	0	2	2	1
öffentliche Fördermaßnahmen	0	0	0	0	0	1	1
kulturtechnische Vorteile	0	1	0	0	0	1	0
Probleme Energieträgerwechsel	1	1	0	0	1	-2	0

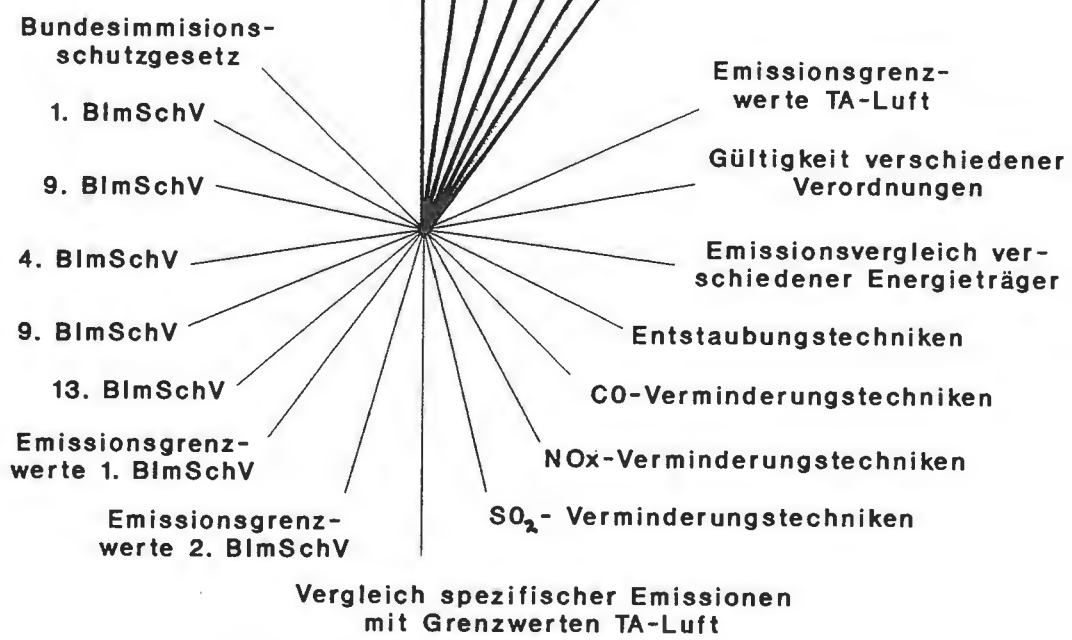


Abbildung 3.2-22: Hypertextsystem zur Bewertung der Energieträger

### 3.2.3.3 Übersicht

Abbildung 3.2-23 gibt das beschriebene Programmmodul noch einmal zusammenhängend wieder. Besonders die Modulteile Energieträgerauswahl, Anlagenkonzeption und Planungsvariantenauswahl verkörpern die 'klassische' Expertensystemtechnologie in Form von Produktionsregeln. Die anschließenden Auslegungs- und Berechnungsvorgänge stellen eine Mischform dar zwischen konventioneller und expertensystemorientierter Wissensrepräsentation. Zum Teil sind sie sehr stark prozedural ausgerichtet (z.B. ökonomische Berechnung), andererseits nutzen sie aber auch intensiv die prologinternen Auswertungs- und Schlußfolgerungsmechanismen wie Matching und Backtracking.

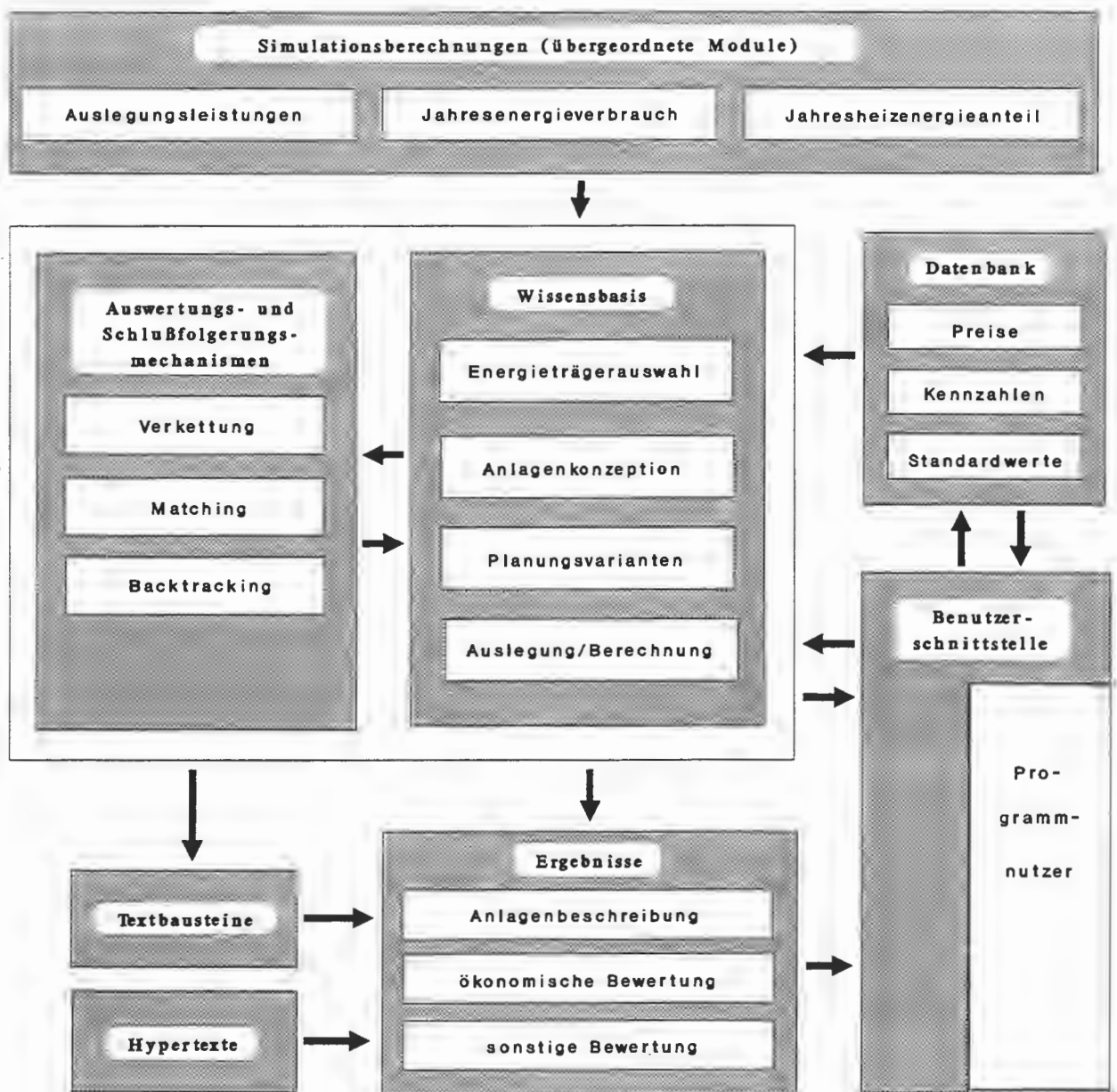


Abbildung 3.2-23: Übersicht der Darstellung und Repräsentation des Fachwissens im Bereich 'Planung der Heizenergiebereitstellung'

### **3.2.3.4 ... im Bereich Planung der Heizenergieausbringung**

#### **3.2.3.4.1 Modularisierung**

Die Planung und Auslegung von Heizungssystemen ist ein Vorgang, bei dem technisch konkretes Formelwissen sehr stark mit heuristisch unscharfem Erfahrungswissen gekoppelt wird. Unterschiedliche Heizungssysteme müssen in einem Gewächshaus so installiert werden, daß die Ansprüche der Kulturpflanzen, die ökonomischen Ansprüche und die technischen Realisierungsmöglichkeiten in Einklang zueinander stehen. Ökologische Aspekte, wie sie z.T. stark in den Bereich der Energiebereitstellung hineinwirken, sind im Bereich der Heizenergieausbringung nur von untergeordneter Bedeutung. Das entwickelte System berücksichtigt die oben beschriebenen Faktoren, verknüpft sie miteinander und gibt in einer Planungssituation konkrete Lösungsvorschläge aus. Die Bearbeitung wurde in vier Teilschritte aufgeteilt:

- Bestimmung planungsspezifischer Eckdaten
- Realisierung heuristischer Lösungsstrategien
- Aufteilung der Heizungssysteme im Gewächshaus
- Auslegung einzelner Heizungssysteme

Die Modellierung der dargestellten Teilschritte erfolgte mit unterschiedlichen Ansätzen, die zum Teil sehr speziell auf das jeweilige Problem abgestimmt wurden. Kernpunkt der eingesetzten Verfahren ist eine heuristische Lösungssuche aufgrund von vorgegebenen Planungsstrategien und Datenbankvorgaben und ein im Bereich der Auslegung eingesetztes Framekonzept.

#### **3.2.3.4.2 Beschreibung**

##### Bestimmung planungsspezifischer Eckdaten

Das Gewächshaus wird zur Planung der Heizungssysteme in vier spezifische Bereiche unterteilt: Außenwandbereich, Dachbereich, Kulturbereich und einen Bereich, der eine besondere Stellung einnimmt: den Bodenbereich. Jedem der drei zuerst genannten Bereiche wird eine Auslegungsleistung zugeordnet, die während der nachfolgenden Planungen als angestrebter Eckwert für die erforderliche Heizleistung gilt. Die für die Stehwand- und Giebelabschirmung notwendigen Heizleistungen werden bereits mit den Berechnungen in Kapitel 3.2.3.2.3 erfaßt und sind somit vorgegeben. Die Aufteilung der restlichen Heizleistungen auf den Dachbereich und den Kulturraum erfolgt mit Hilfe eines

Punktesystems. Die im Kulturbereich zu installierende Heizleistung  $\dot{Q}_K$  [ W ] wird mit

$$\dot{Q}_K = \frac{\dot{Q}_{Seg} - \dot{Q}_{SG}}{T + 1} \quad (3.2-45)$$

berechnet. Darin ist  $\dot{Q}_{Seg}$  die im Gewächshaussegment erforderliche Gesamtleistung [ W ],  $\dot{Q}_{SG}$  die notwendige Steh- und Giebelwandabschirmung [ W ] und T das angestrebte Verhältnis von im Dachraum zu im Kulturraum installierten Heizleistungen [ - ].

T wird berechnet mit

$$T = \begin{cases} \frac{T_{min} + T_{max}}{2} & \text{falls } T_{min} > T_{max} \\ T_{min} & \text{falls } \frac{\sum WF1}{\sum WF2} < T_{min} \\ T_{max} & \text{falls } \frac{\sum WF1}{\sum WF2} > T_{max} \\ \frac{\sum WF1}{\sum WF2} & \text{sonst} \end{cases} \quad (3.2-46)$$

WF1, WF2,  $T_{min}$  und  $T_{max}$  sind spezifische Punktwerte, die aus dem jeweiligen Programmverlauf resultieren. Abbildung 3.2-24 gibt das zur Zeit implementierte Punktesystem wieder. Man erkennt, daß  $T_{min}$  und  $T_{max}$  die Begrenzung des T-Wertes darstellen, während die Wichtungsfaktoren WF1 und WF2 Tendenzen manifestieren. Durch das Hinzufügen, Verändern oder Löschen einzelner Kriterien bzw. Faktoren kann das System neuen Situationen angepaßt werden.

Ursache	$T_{min}$	$T_{max}$	WF1	WF2
Sand- oder Mattenbewässerung ohne Nadelfolie (Tischkultur)	2	-	0	0
Düsenrohrbewässerung	-	-	1	1
sparsamer Einsatz der Heizenergie	-	-	0	1
höhere Substrattemperaturen bei Tischkulturen erwünscht	-	-	0	1
Energieschirm	-	1	0	1
niedrige Luftfeuchte im Bestand erwünscht	-	-	0	1
geschlossene Tischflächen bei Tischkulturen	0,5	-	0	0
empfindliche Pflanzen	0,5	-	1	1

$T_{min}, T_{max}, WF1, WF2$  = Grenz- und Wichtungsfaktoren  
(im Text erklärt)

Abbildung: 3.2-24: Implementiertes Punktesystem zur Aufteilung der Heizflächen

Die im Dachraum zu installierende Heizleistung ergibt sich entsprechend zu

$$\dot{Q}_{Da} = \dot{Q}_{Seg} - \dot{Q}_K - \dot{Q}_{SG} \quad (3.2-47)$$

Dem Bodenbereich wird keine direkte Heizleistung zugeordnet, sondern es wird festgestellt, ob eine Bodenheizung sinnvoll ist bzw. vom Benutzer gefordert wird. Nur in diesem Fall wird der Boden als ein Heizsystembereich in die Planung mit aufgenommen.

In einem zweiten Schritt werden die für die Beheizung des Gewächshauses zur Verfügung stehenden Energieträger hinsichtlich ihrer möglichen Vorlauftemperatur und Wärmeleistung analysiert. Es erfolgt eine Einteilung in drei Gruppen: Niedertemperaturwärme (< 40 °C), Mitteltemperaturwärme (40 °C - 60 °C) und 'Hochtemperaturwärme' (> 60 °C). Die von einem Brennwertgerät zur Verfügung gestellte Wärme wird mit 45 °C angesetzt.

In einem dritten Planungsschritt hat der Programmbenutzer die Möglichkeit, die Planungsfreiheit einzuschränken. Er kann die Anzahl von Heizungssystemen in bestimmten Planungsbereichen begrenzen, oder es können Planungsbereiche oder Teilbereiche davon (z.B. Kulturtische) aus dem Planungsprozeß herausgenommen werden.

Alle drei Planungsschritte wurden ähnlich wie die Wissensrepräsentation in Kapitel 3.2.3.3 in Form von Produktionsregeln implementiert.

#### Realisierung heuristischer Lösungsstrategien

Das vollständige Durchsuchen des Lösungsraumes ist bei der Heizungsauslegung nicht möglich, da mehrere Heizungssysteme im Gewächshaus kombiniert werden können und unterschiedliche Heizungssysteme von der Industrie zur Verfügung gestellt werden. Im Expertensystem HORTEX wurde daher ein heuristisches Suchverfahren in Kombination mit einem offenen Datenbankmodell eingesetzt. Das Suchverfahren kann in die Gruppe der Hill-Climbing-Verfahren eingeordnet werden. Durch das mehrmalige Durchsuchen des Problemlösungsraumes mit unterschiedlichen Strategien wird das Lösungsspektrum zusätzlich ausgeweitet.

Es wurde folgendes Verfahren entwickelt und implementiert:

Die in einem Planungsprozeß notwendigen Heizungssysteme werden aus einer Datenbank entnommen, wobei jeweils die Anordnung innerhalb der Datenbank die Entnahmepriorität festlegt. Scheitert die Auslegung oder die Anbindung des gewählten Heizungssystems, erfolgt mittels Backtracking die Auswahl des nächstmöglichen Systems. Durch Umsortierung der Datensätze innerhalb der Datenbank vor dem Datenzugriff können verschiedene Lösungsstrategien vorgegeben werden:

- Standardauslegung (1): Entnahme der Heizungssysteme aus der Datenbank nach Original-Reihenfolge.
- Ökonomische Strategie (2): Umsortierung der Heizungssysteme vor jedem Zugriff, so daß die für den speziellen Datenbankaufruf kostengünstigsten Systeme die höchste Priorität genießen.
- Benutzerdefinierte Strategie A (3): Umsortierung der Heizungssysteme, so daß die vom Benutzer angegebene Heizungssystemgruppe höchste Planungspriorität bekommt. Das weitere Prioritätskriterium ist die Strategie 2.
- Benutzerdefinierte Strategie B (4): Das vom Benutzer angegebene spezielle Heizungssystem erhält die höchste Priorität. Alle anderen Systeme werden nach Strategie 2 geordnet.

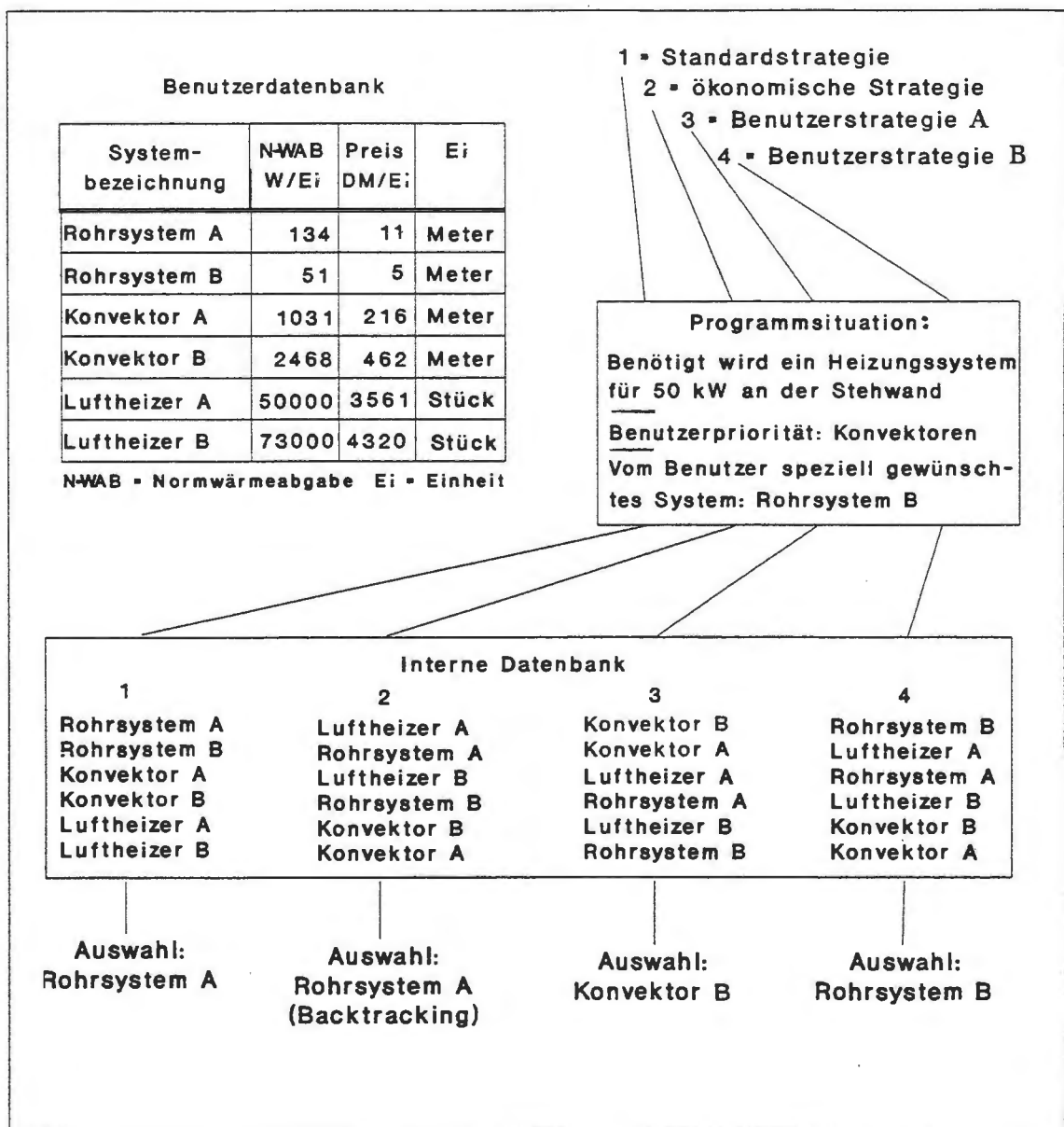


Abbildung 3.2-25: Auswahl von Heizungssystemen nach verschiedenen Strategien



Die Einteilung der Heizungssysteme innerhalb der Datenbank erfolgt in vier Gruppen: Rohrsysteme, Konvektorsysteme, Warmwasserluftheizer und direkt befeuerte Luftheizer. Notwendige Kenngrößen sind die Wärmeabgabe pro Auslegungseinheit (Meter oder Stück) unter Normbedingungen, das verwendete Material, Druckverlustangaben und der Preis pro Auslegungseinheit. Abbildung 3.2-25 gibt die Realisierung der verschiedenen Strategien an einem Datenbankbeispiel wieder.

#### Aufteilung der Heizungssysteme im Gewächshaus

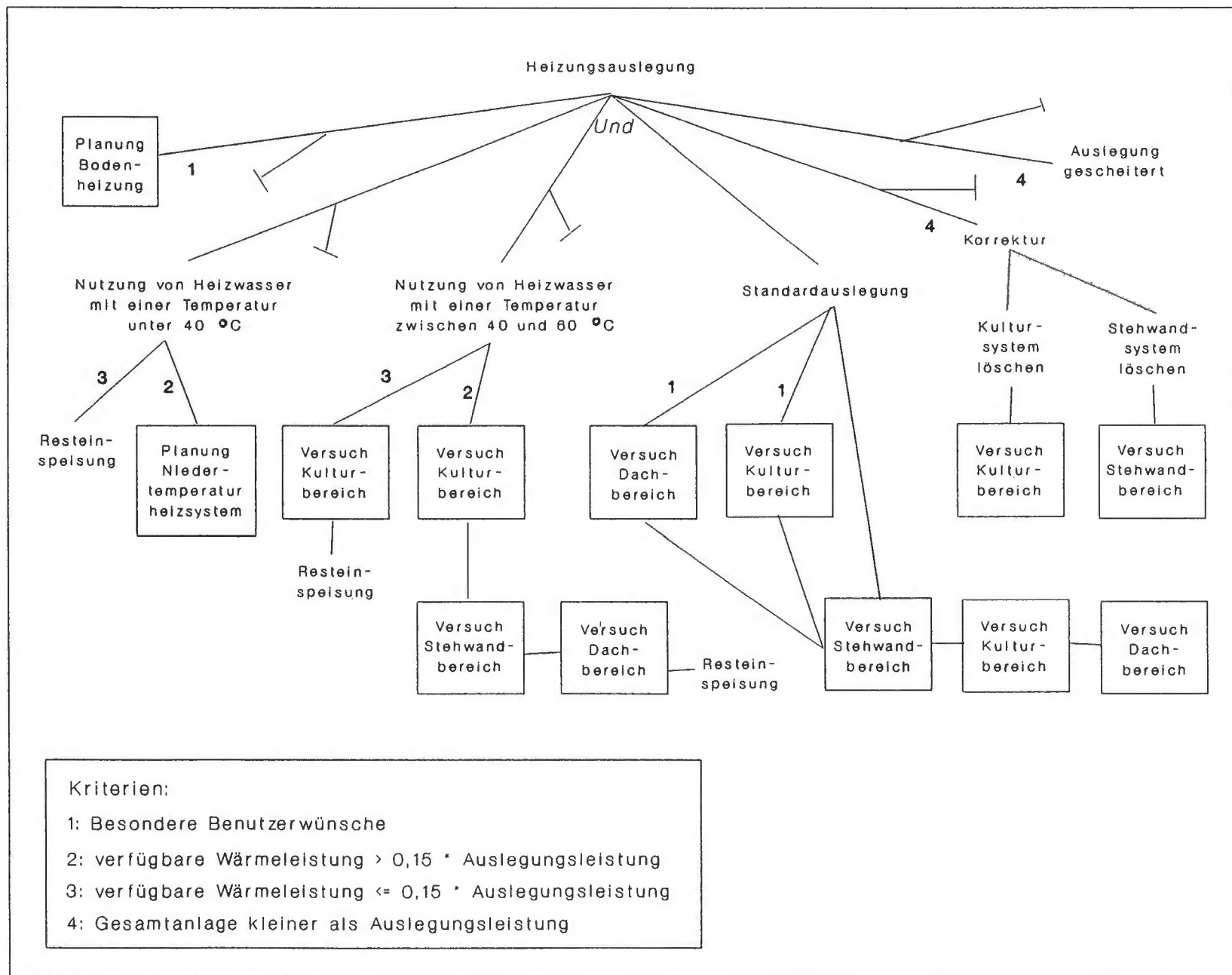
Die Aufteilung der Heizungssysteme und die Verteilung notwendiger Heizflächen im Gewächshaus erfolgt anhand des in Abbildung 3.2-26 dargestellten Und-Oder-Graphen. Der dargestellte Graph wurde innerhalb des Programms nicht mit dem in Kapitel 3.2.2.1 dargestellten Produktionsregelansatz verwirklicht, da umfangreiche Variablenübergaben notwendig waren. Es erfolgte eine direkte Implementierung mit Prologprädikaten.

Für jede Planungsstrategie wird der abgebildete Graph einmal durchlaufen und eine Lösungsmenge generiert. Wird bei der Abarbeitung die technische Beschreibung eines Heizungssystems verlangt (eingerahmte Kästchen), wird versucht, in den angegebenen Bereichen (Kultur-, Dach-, Stehwandbereich) ein Heizungssystem auszulegen oder spezielle Planungen (Bodenheizung, Niedertemperaturheizungen) durchzuführen. Steht Heizwasser mit niedrigen Vorlauftemperaturen zur Verfügung, wird zuerst versucht, dieses Heizwasser mit besonderen Heizungssystemen zu nutzen. Ist aus irgendwelchen Gründen eine Nutzung nicht möglich, erfolgt eine Einspeisung in ein 'normales' Heizungssystem (Resteinspeisung).

Am Ende der Auslegung wird überprüft, ob die Auslegungsleistung mit dem geplanten Heizungssystem abgedeckt wird. Ist dieses nicht der Fall, werden Heizungssysteme aus der Lösungsmenge gelöscht, die noch notwendige Auslegungsleistung entsprechend korrigiert und es erfolgt eine erneute Auslegung in dem entsprechenden Bereich.

Die Auslegung (bzw. der Versuch einer Auslegung) erfolgt in dem System dadurch, daß aus der Datenbank das Heizungssystem mit der aktuell höchsten Priorität an ein Auslegesystem (Framesystem) angekoppelt wird und daß Auslegungsbedingungen überprüft werden (z.B. daß ein Heizungsrohr im Dachraum keinen Rohrdurchmesser über 60 mm haben darf oder daß ein Heizungsrohr nicht in eine vom Programm benutzer gesperrte Gewächshausregion plaziert werden darf etc.). Werden alle Bedingungen erfüllt, wird das Framesystem mit der Beschreibung des entsprechenden Objektes beauftragt (ansonsten Backtracking zum nächsten System). Nach der Auslegung müssen etwaige Heizleistungsüberschüsse bzw. -defizite auf andere Planungsbereiche oder Teilbereiche übertragen werden.

Abbildung 3.2-26: Implementiertes Verfahren zur Aufteilung und Auslegung der Heizungssysteme im Gewächshaus



### Auslegung der Heizungssysteme

Die Berechnung und Auslegung der Heizungssysteme erfolgt mit Hilfe des in Kapitel 3.2.2.2 beschriebenen Frame-Ansatzes. Es wurde die in Abbildung 3.2-27 dargestellte Objekthierarchie verwirklicht. Bei Aufruf des Framesystems wird ein Planungsframe erzeugt, welches an die unterste Ebene des Objektbaumes angekoppelt wird. Bei Scheitern der Auslegung werden zuerst weitere Heizungssysteme ausprobiert (s.o.), ehe das Planungsframe an eine andere Stelle des Objektbaumes angekoppelt wird. Die Reihenfolge der Anschlußstellen gibt ein spezielles Prioritätenframe vor. Dadurch, daß das Planungsframe am Ende des Objektbaumes steht, können sämtliche Informationen der übergeordneten Objekte bei Bedarf 'ererb't werden. Der Zugriff und die gesamte Auswertung erfolgt mit dem in Kapitel 3.2.2.2 beschriebenen Objektinterpretier.

Die Übergabe von gewünschten Vorgaben (z.B. Vorlauftemperatur) in das Framekonzept geschieht dadurch, daß die zu übergebenden Werte als Slots des Planungsframes definiert werden (Vergleiche auch Anhang C.3)

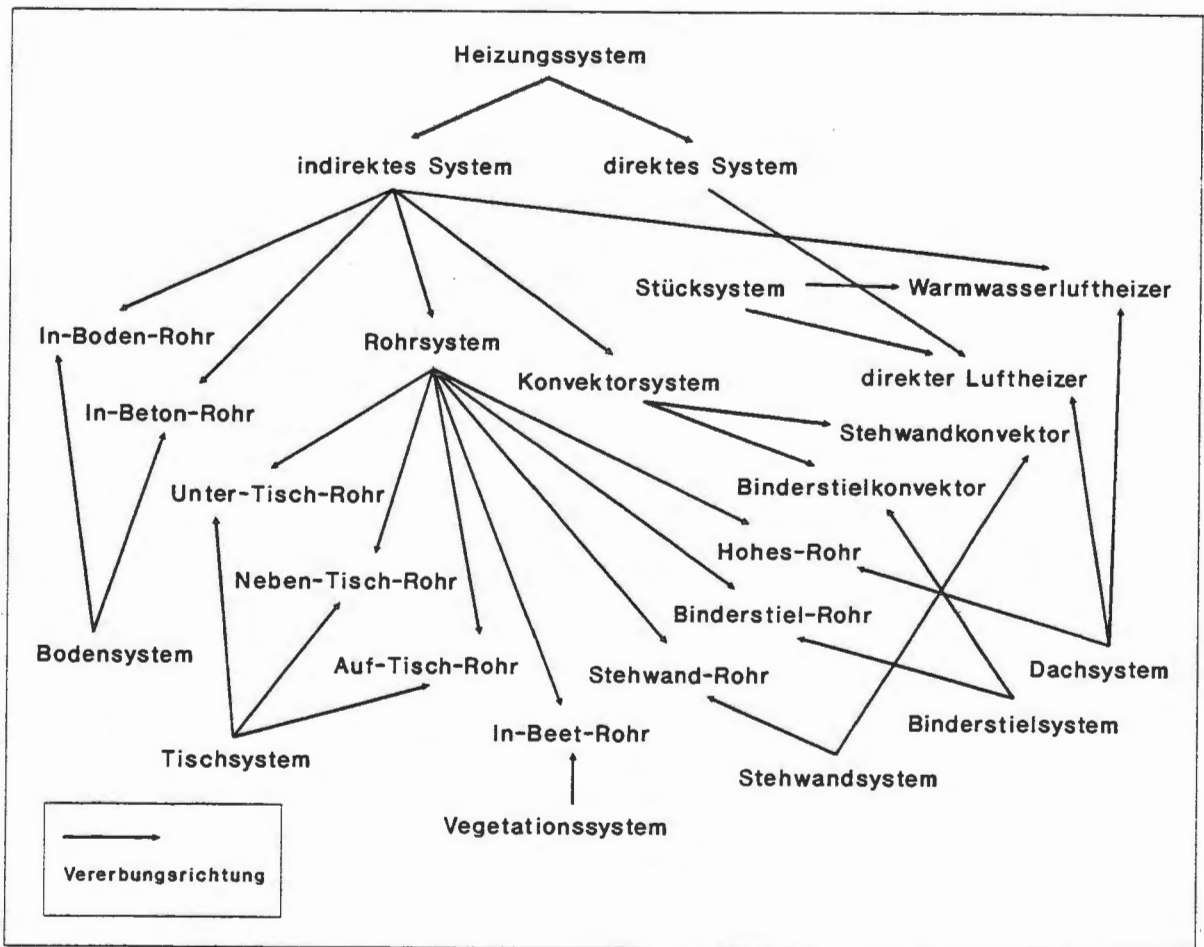


Abbildung 3.2-27: Im Bereich der Heizungssysteme implementierte Objekthierarchie

Sämtliche Berechnungsverfahren wurden als *wenn-benötigt*-Slots implementiert. Die beiden zentralen Berechnungen sind:

1. Berechnung der Wärmeabgabe und die Auslegung von Heizungssystemen, die allein zur Erhöhung der Raumtemperatur eingesetzt werden:

Die Berechnung erfolgt mit Hilfe des mittleren logarithmischen Temperaturunterschiedes zwischen Heizmittel und Raumluft und der in der Datenbank gespeicherten Heizungssystemwärmeabgabe (vgl. HÖNMANN und SPRENGER 1985, S. 597). Die von HÖNMANN und SPRENGER (1985, S. 598) angegebenen Korrekturwerte für große Temperaturspreizungen wurden in ein Polynom überführt. Die Bestimmung der erforderlichen Heizkörperexponenten erfolgte entweder nach den bei HÖNMANN und SPRENGER (1985, S. 597) angegebenen Werten oder aufgrund von Expertenangaben aus dem ITG.

2. Berechnung der Wärmeabgabe und Auslegung von Bodenheizungen:

Die Berechnung erfolgt nach dem von KAST et al. (1986) angegebenen Verfahren (Fußbodenheizung) oder nach den bei VON ZABELTITZ (1986, S. 223) angegebenen Tabellenwerten (Bodenheizung). Die angegebenen Werte bzw. Graphiken wurden durch Regression in entsprechende Polynome umgewandelt. Die Berechnungen wurden mit den von T HART et al. (1984) bzw. ROBERTS und MEARS (1980) angegebenen Parametern (Rohrabstand, Oberflächentemperatur) auf Gewächshausbedingungen übertragen und validiert. Die Auslegung erfolgt entweder nach den zur Verfügung stehenden Vorlauftemperaturen oder aufgrund vorgegebener Grenzbedingungen (Fußbodenoberflächentemperatur, Bodenbearbeitungstiefe).

Weitere Berechnungsverfahren können ebenso wie die eingesetzten Faust- und Erfahrungswerte dem Anhang C.3 entnommen werden.

Abbildung 3.2-28 gibt die für den Planungsbereich Dachraum relevanten Heizungssystem-Frames der implementierten Wissensbasis in geordneter Form wieder. Durch das Hinzufügen oder Löschen einzelner Frames kann das Modell leicht neuen Anforderungen angepaßt werden. Dabei ist es noch nicht einmal notwendig, alte und evtl. zu verbessernde Frames zu löschen. Neue Modelleigenschaften können einfach als neue Slots oder neue Objekte, die über Vererbungsslots an bestehende Objekte angekoppelt werden, hinzugefügt werden.

```
frame("hohes_rohr",ist_ein,wert,["dachsystem","rohrsystem"],intern)
frame("hohes_rohr",maximale_systemanzahl,standard,["100"],intern)
frame("luftheizer",ist_ein,wert,["dachsystem","stuecksystem","indirektes_system"],intern)
frame("luftheizer",heizkoerperexponent,wert,["1"],intern)
frame("direkter_heizer",ist_ein,wert,["dachsystem","stuecksystem","direktes_system"],intern)
frame("dachsystem",waermeabgabeort,wert,["dachraum"],intern)
frame("dachsystem",planungslaenge,wenn_benoetigt,[],intern)
frame("dachsystem",anbringung,standard,["im_dachraum"],intern)
frame("rohrsystem",gesamtheizrohrlaenge_in_meter,wenn_benoetigt,[],zeige)
frame("rohrsystem",angaben_zur_hydraulischen_dimensionierung,wenn_benoetigt,[],zeige)
frame("rohrsystem",heizungsart,wert,["rohr"],intern)
frame("rohrsystem",ist_ein,wert,["indirektes_system"],intern)
frame("rohrsystem",heizkoerperexponent,standard,["1.25"],intern)
frame("rohrsystem",gesamtanzahl_heizrohre,wenn_benoetigt,[],zeige)
frame("rohrsystem",heizrohrlaenge_in_meter,wenn_benoetigt,[],zeige)
frame("rohrsystem",anzahl_rohrregister,standard,["1"],intern)
frame("rohrsystem",anzahl_rohrregister,wenn_benoetigt,[],intern)
frame("rohrsystem",rohranzahl_multiplikator,wenn_benoetigt,[],intern)
frame("stuecksystem",gesamtanzahl_heizer,wenn_benoetigt,[],zeige)
frame("stuecksystem",waermeverteilung,wenn_benoetigt,[],zeige)
frame("stuecksystem",gesamtanzahl_einheiten,wenn_benoetigt,[],intern)
frame("stuecksystem",maximale_systemanzahl,standard,["300"],intern)
frame("stuecksystem",minimale_systemanzahl,standard,["0"],intern)
frame("stuecksystem",gesamteinheiten_ohne_anschluss,wenn_benoetigt,[],intern)
frame("indirektes_system",ist_ein,wert,["heizungssystem"],intern)
frame("indirektes_system",auslegungsvorlauftemperatur_in_grad_c,wenn_benoetigt,[],zeige)
frame("indirektes_system",waermeausbringung,wert,["indirekt mit Wasser"],zeige)
frame("indirektes_system",wassermenge_in_m3_pro_stunde,wenn_benoetigt,[],zeige)
frame("direktes_system",gesamtwaermeabgabe_in_k_xw,wenn_benoetigt,[],zeige)
frame("direktes_system",waermeabgabe_pro_einheit,wenn_benoetigt,[],intern)
frame("direktes_system",waermeausbringung,wert,["direkt mit der Waermeerzeugung"],zeige)
frame("direktes_system",ist_ein,wert,["heizungssystem"],intern)
frame("direktes_system",hinweis,wert,["Werden die Verbrennungsabgase direkt zur CO2-Duengung
eingesetzt, ist die DIN 4793 zu beachten: Es sind Einrichtungen vorzusehen, die
eine ueberkonzentration von Schadgasen im Gewaechshaus vermeiden."],zeige)
frame("heizungssystem",gesamtwaermeabgabe_in_k_xw,wenn_benoetigt,[],zeige)
frame("heizungssystem",beruecksichtigbare_waermeabgabe,wenn_benoetigt,[],intern)
frame("heizungssystem",waermeabgabe_pro_einheit,wenn_benoetigt,[],intern)
frame("heizungssystem",gesamtanzahl_einheiten,wenn_benoetigt,[],intern)
frame("heizungssystem",gesamtinvestition_in_d_xm,wenn_benoetigt,[],zeige)
frame("heizungssystem",beruecksichtigung,standard,["1"],intern)
frame("heizungssystem",normuebertemperatur,standard,["60"],intern)
frame("heizungssystem",delta_t,standard,["20"],intern)
frame("heizungssystem",maximale_systemanzahl,standard,["300"],intern)
frame("heizungssystem",minimale_systemanzahl,standard,["0"],intern)
frame("heizungssystem",systemanzahl,wenn_benoetigt,[],intern)
frame("heizungssystem",gesamteinheiten_ohne_anschluss,wenn_benoetigt,[],intern)
frame("heizungssystem",auslegungsleistung,wenn_benoetigt,[],intern)
```

Abbildung 3.2-28: Beispiel für im System implementierte Frames

Abbildung 3.2-29 zeigt *wenn\_benoetigt*-Prozeduren zur Berechnung der Wärmeabgabe des Heizungssystems. Die erste Prozedur bestimmt die Gesamtwärmeabgabe des Systems, indem sie das Planungsframe nach Informationen über die Wärmeabgabe einer Einheit und die Gesamtanzahl der Einheiten befragt. Die zweite Prozedur rechnet die Wärmeabgabe einer Einheit aus, indem der Objektbaum nach Planungssekdaten durchsucht wird und am Ende das Prädikat *berechne\_waermeabgabe* aufgerufen wird. Entscheidend ist, daß bei den Prozeduraufrufen keine Variablenübergabe von Eingangsparametern stattfindet, sondern die Prozedur alle notwendigen Informationen aus der Wissensbasis selbst abfragt, bzw. den Objektinterpretier mit der Befragung beauftragt. Somit haben alle Prozeduren die gleiche Schnittstelle und können innerhalb der Framestruktur gespeichert werden.

```
prozedur(heizungssystem,gesamtwaeirmeabgabe_in_k_xw,[Ergebnis]):-
  frage_wert(planung,waermeabgabe_pro_einheit,WPE),
  frage_wert(planung,gesamtanzahl_einheiten,GAE),
  frage_wert(planung,beruecksichtigte_waermeabgabe_anschlussleitungen...
  WAB = (WPE * GAE)/1000,
  WA = WAB + BWAL,
  wandle(Ergebnis, WA),!.

prozedur(heizungssystem,waermeabgabe_pro_einheit,[Ergebnis]):-
  frage_wert(planung,auslegungsvorlaufemperatur_in_grad_c,VL),
  frage_wert(planung,normwaermeabgabe,NW),
  frage_wert(planung,heizkoerperexponent,HKE),
  frage_wert(planung,normuebertemperatur,NUE),
  frage_wert(planung,delta_t,DT),
  inntemperatur(Ti),
  berechne_waermeabgabe(VL,NW,HKE,NUE,DT,Ti,WAB),
  str_real(Ergebnis,WAB),!.

berechne_waermeabgabe(VL,NW,HKE,NUE,DT,Ti,WAB):-
  C = (VL-Dt-Ti)/(VL-Ti),
  F1 = ((VL-Dt/2-Ti)/NUE),
  hoch(F1,HKE,F),
  F2 = 0.584875+1.120863*C-0.71859*C*C,
  minimum(1,F2,F3),
  WAB = F*F3*NW,!.

```

Abbildung 3.2-29: Darstellung von Berechnungsverfahren innerhalb der Framestruktur am Beispiel der Wärmeabgabe von Heizungssystemen

Die Einbindung der Berechnungsverfahren in *wenn\_benoetigt*-Slots ist nicht unbedingt erforderlich. Der Objektbaum kann auch mit konventionellen Prolog-Prädikaten analysiert werden. Abbildung 3.2-30 gibt hierzu ein Beispiel: Das konventionelle Prolog-Prädikat *fussbodenheizung\_auslegen* befragt den Objektbaum nach notwendigen Parametern für die Berechnung und ruft den Auslegealgorithmus *auslegen* auf.

```
fussbodenheizung_auslegen:-!,
frage_wert(planung,rohrabstand_in_mm,Teil),
frage_wert(planung,betonueberdeckung_in_mm,Ue),
frage_wert(planung,bodentemperatur_in_grad_c,BodMax),
frage_wert(planung,innendurchmesser,D),
frage_wert(planung,delta_t,SP),
innentemperatur(Ti),
anfrage(,_,Vlth,_),
auslegen(VlTh,Teil,Ue,Bodmax,Ti,D,SP),!.

auslegen(VlTh,_,_,Bod,_,_,):-
zaehler(POW,VL),
Vl <= VlTh,!, ..., ....

auslegen(VlTh,Teil,Ue,Bodmax,Ti,D,SP):-!,
fussbodenheizungsleistung(POW,HeizUeb,Teil,Ue,Bodmax,Ti,D),
Vl = Ti+HeizUeb+Sp/2,
BodmaxNeu = Bodmax -1,
retractall(zaehler(,_)), assert(zaehler(POW,VL)),
auslegen(VlTh,Teil,Ue,BodmaxNeu,Ti,D,SP).

fussbodenheizungsleistung(POW,HeizUeb,Teil,Ue,Bodmax,Ti,D):-!,
maximum(Teil,D,T),          maximum(75,T,T1),
MT=1-(T1/75),              hoch(1.24,MT,ATMT),
AUe=1.075-0.00012*T,       MUE=4.5-0.1*Ue,
hoch(AUe,MUE,AUEMue),     ARd=1+0.000351*T-5.7E-7*T*T,
Md=0.25*D-5,              hoch(ARd,Md,ARdMd),
X=5.942*ATMt*AUEMUE*ARdMd, Bg_berechnen(T,Ue,Bg1),
maximum(0,Bg1,Bg),         F1=(BodMax-Ti)/9,
hoch(F1,1.1,F),           Y=F*Bg,
N=1.043, G1 =0.09+0.00008*T+2.7E-6*T*T-0.002*Ue,
maximum(0,G1,G),          Pot=1/(N-G),
H1=Y/X,                   hoch(H1,Pot,HeizUeb),
hoch(HeizUeb,N,P1),       POW=P1*X,!.

bg_berechnen(T,Ue,Bg):- T >= 75,!, Bg=94-0.25714*T+0.4*Ue.
bg_berechnen(T,_,Bg):- T < 75,!, Bg=100-0.09211*T.
```

Abbildung 3.2-30: Beispiel für die Ankopplung konventioneller Berechnungsverfahren an das Framekonzept (Bereich Fußbodenheizung)

*auslegen* ist ein rekursives Prologprädikat, welches das von KAST et al. (1986) beschriebene und für die Verarbeitung im Programm in Polynome transformierte Berechnungsverfahren zur Bestimmung der Heizleistung von Fußbodenheizungen mit dem Prädikat *fussbodenheizungsleistung* aufruft. Nach jedem Schritt wird die maximal zulässige Bodentemperatur um 1 K herabgesetzt, bis die resultierende Vorlauftemperatur unter oder gleich der zur Verfügung stehenden Vorlauftemperatur ist. Dann werden die Ergebnisse (Wärmeabgabe, Bodenoberflächentemperatur, Vorlauftemperatur) als bekannte Slots des Planungsframes in der Wissensbasis abgespeichert.

Da bei der Abarbeitung der Frames der gesamte Objektbaum mit seinen dazugehörigen Berechnungsroutinen im Arbeitsspeicher des Rechners vorliegen muß, und die rekursive Abarbeitung einschließlich des Backtrackings ebenfalls einen hohen Anteil des zur Verfügung stehenden Hauptspeichers benötigt, wurde das vorgestellte Frame-System entsprechend den Heizungsauslegungsbereichen in vier eigenständige Module aufgeteilt. Die hierdurch bewirkte Verschlechterung der Wartbarkeit mußte in Kauf genommen werden. Durch die verbesserte Übersichtlichkeit wurde dieser Nachteil jedoch teilweise wieder ausgeglichen.

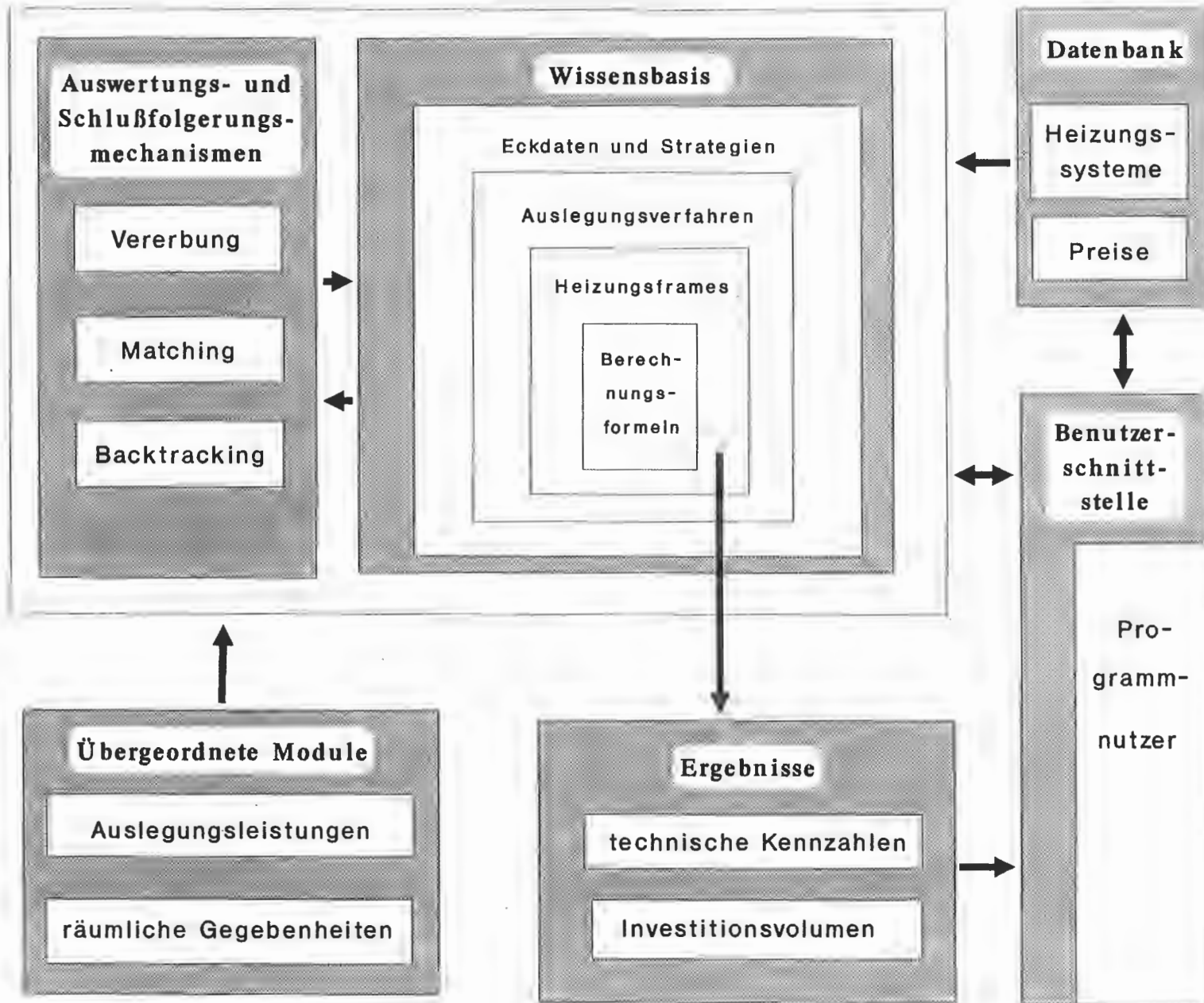
#### **3.2.3.4.3 Übersicht**

Abbildung 3.2-31 gibt eine Übersicht über die realisierten Programmteile. Es wird deutlich, daß die Heizungsframes die zentrale Planungsbasis darstellen, die jedoch von den umliegenden Modulen beeinflusst wird. In der obersten Planungsebene beeinflusst der Benutzer durch seine Anforderungen und Randbedingungen den weiteren Programmablauf in entscheidendem Maße. Im Bereich der Auslegung und der Frames bekommen zunehmend die in der Datenbank gespeicherten Heizungssysteme Bedeutung. Im Gegensatz zur Planung der Energiebereitstellung ist die ökonomische Bewertung einzelner Lösungsvarianten nicht als eigener Programmteil implementiert. Sie wird als ein Slot innerhalb der Heizungsframes dargestellt und kann somit als ein Berechnungsschritt von vielen interpretiert werden.

Die bisherigen Ausführungen und auch die dargestellte Übersicht verdeutlichen, daß die Funktionalität des Framesystems entscheidend davon abhängt, in welcher Form es in das übrige Programm integriert wird. Inwiefern die dargestellten übergeordneten Systembausteine (Strategie, Auslegung etc.) ebenfalls in das Framekonzept eingebunden werden könnten und somit eine noch einheitlichere Struktur der Wissensmodellierung erreicht werden kann, ist noch ungeklärt. Bei der eingesetzten Hard- und Software ist diese Einbindung jedoch ausgeschlossen.



Abbildung 3.2-31: Übersicht der Darstellung und Repräsentation des Fachwissens im Bereich 'Planung der Heizenergieausbringung'



### 3.3 Technische Realisierung

Das Simulationsmodell, die Datenbankkonzeption und der überwiegende Teil des Datentransfers wurde in Turbo-Pascal 5.0 (Borland) realisiert. Die Implementierung der wissensbasierten und logik-orientierten Programmelemente erfolgte mit der Programmiersprache Turbo-Prolog 2.0 (Borland).

Um das Prologsystem auch innerhalb der Graphikoberfläche anwenden zu können, mußte eine Stelle aus dem Compiler entfernt werden, die dafür sorgte, daß bei Programmbeginn jeweils in den Textmodus umgeschaltet wurde. Bei der Programmentwicklung wurden daher die Wissensbasen als C-Object-Files abgelegt und später mit den C-Object-Files des veränderten Compilers zu einem unter Graphikanwendungen lauffähigen Programm zusammengelinkt.

Die Realisierung des Programmrahmens und aller Ein- und Ausgabefunktionen erfolgte mit der Programmiersprache Turbo-PASCAL 5.0 (Borland). Zur Gestaltung der Benutzerschnittstelle wurde Metashell 3.01 (Metagraphics) eingesetzt. Es ersetzt den Borland Graphik-Treiber und erlaubt den schnellen Zugriff auf verschiedene Graphikkarten. Der Treiber wird vor Aufruf des Programms resident in den Arbeitsspeicher des Rechners geladen und benötigt ca. 90 kByte Speicherplatz. Vorgefertigte Basisroutinen (Mausinterrupts, Bildschirmspeicher-manipulationen) stehen dem Programmentwickler zur Verfügung und ermöglichen einen hervorragenden Einsatz der Software.

Die Gestaltung der Benutzerschnittstelle erfolgte auf drei verschiedene Arten:

- Direkte Darstellung von Graphiken und Bildern, die mit Paintbrush (Microsoft) oder Harvard-Graphics (Software Publishing Corporation) erstellt wurden.
- Konventionelle Menügestaltung mit Objekten wie Buttons, Scrollbalken, Pull-up-Menüs etc..
- Kombination von Pixel-Graphiken mit variablen Programmobjekten und Bildschirmmasken.

Besonders von der letztgenannten Möglichkeit wurde bei der Darstellung der Hypertexte, Zentralmenüs und Hilfefunktionen ausgiebig Gebrauch gemacht. Dabei wurde besonderer Wert darauf gelegt, eine hohe Konsistenz der Benutzermenüs zu erzielen. An jeder Stelle des Programms ist eine Hilfs- und Erklärungsfunktion vorhanden, die auf einem Hypertextsystem aufbaut.

### 3.4 Anwendungsbeispiele

#### 3.4.1 Wärmeverbrauchsrechnung

Anhand eines ausgewählten Simulationsbeispiels wird im folgenden die Anwendung des Simulationsmodells demonstriert und die Sensitivität einzelner Modellparameter dargestellt. Grundlage sind Energiebedarfsberechnungen mit dem in Abbildung 3.2-6 dargestellten Beispielbetrieb. Als Standardsimulation wird eine Simulation mit den nachfolgend aufgeführten Parametern definiert.

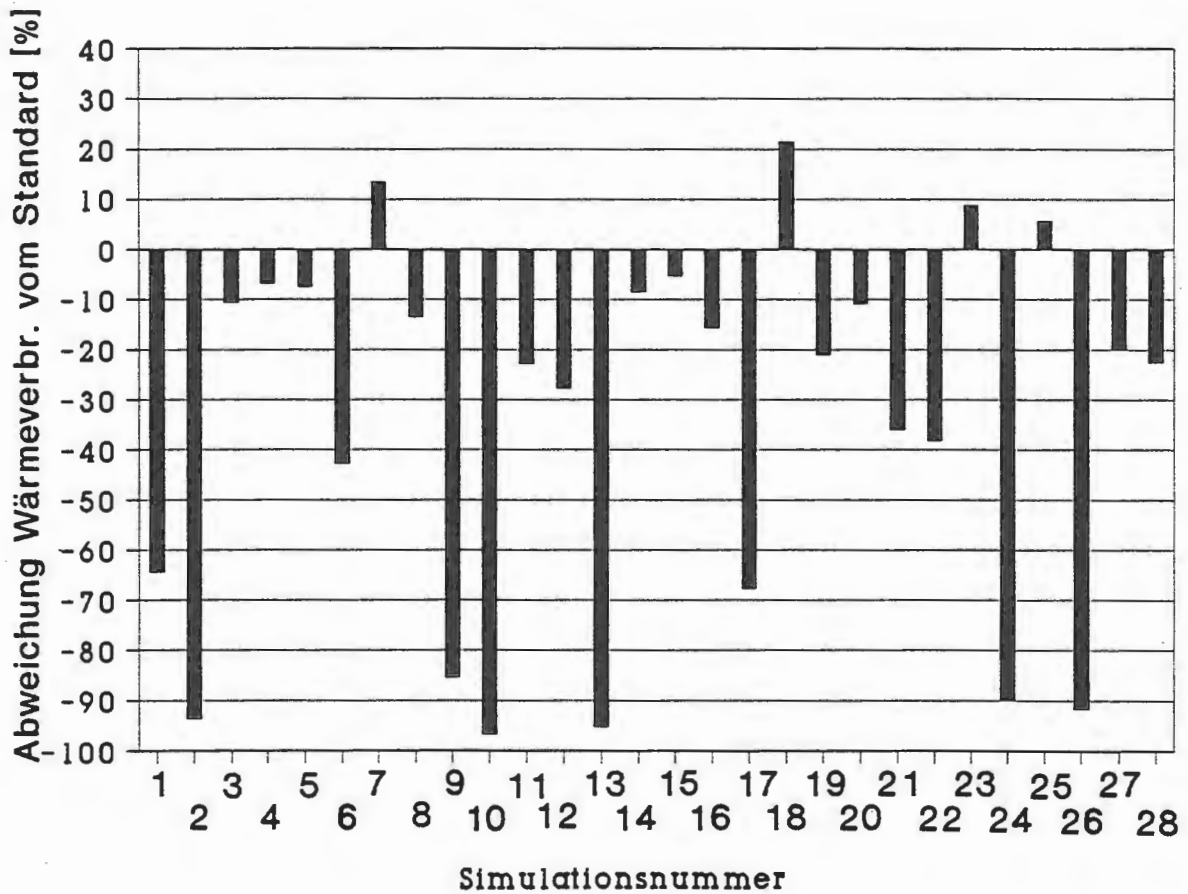
Betrieb: Beispielbetrieb aus Abb. 3.2-6, Grundfl.: 4908 m<sup>2</sup>, Hüllfl.: 7035 m<sup>2</sup>, Klimaregion: Nordwestdeutsches Tiefland, Witterung: durchschnittlich, Heizungssystem: gemischt, Bedachung: Einfachglas mit  $k' = 7,4 \text{ W}/(\text{m}^2 \cdot \text{K})$ , Sollwert Innentemperatur (konstant): 18 °C, Lüftungstemperatur (konst.): 25 °C, ferner alle im Programm zur Verfügung stehenden Standardwerte (vgl. Kapitel 3.2.3.2.2).

Die Berechnung des Heizenergieverbrauchs (Standardsimulation) führt zu einem jährlichen Energieverbrauch von 3485 MWh. Es ergibt sich die typische Jahresverlaufskurve mit geringen Energieverbräuchen zwischen der 20. und 40. Woche des Jahres und hohen Energieverbräuchen in der Zeit zwischen der 50. und 8. Woche.

Abbildung 3.4-1 zeigt die Veränderung dieses Energieverbrauches in Abhängigkeit von bestimmten Ausgangssituationen. Die Ergebnisse beziehen sich auf die von der Heizungsanlage bereitzustellende Heizenergie.

Im einzelnen wird deutlich, daß die im Gewächshaus angestrebten Innentemperaturen die dominierende Rolle des Modells spielen (Nr. 2, 3, 6). Simulation Nr. 11, 12, und 14 zeigen ferner, daß die Erfassung der vorhandenen Energieschirme wichtig ist, die Wirkung aber entscheidend von der Abdichtungsgüte des Schirms im Gewächshaus abhängt. Ebenso spielt die richtige Erfassung der  $k'$ -Werte eine wichtige Rolle (Nr. 18, 19). Dabei ist zu berücksichtigen, daß sich allein durch eine veränderte Einschätzung des  $k'$ -Wertes, auch der Energiespareffekt des Energieschirms verändert (Nr. 18, 19, 20, 21). Ein Beispiel für die Kombination der Modellparameter Bedachung, Energieschirm und Belichtung liefert die Simulation Nr. 17. Unter den gegebenen Bedingungen muß die Heizungsanlage nur 32 % der Standardenergiemenge bereitstellen.

Der Einfluß variiertes Witterung (milder, strenger Winter) liegt bei den vorgegebenen Standardsollwerten zwischen 10 und 20 % (Nr. 7, 8). Bei niedrigeren Innentemperatursollwerten erhöht sich der Einfluß (bezogen auf die erforderliche Energiemenge von Nr. 2) auf über 100 % (Nr. 9, 10).



- |   |  |
|---|--|
| Nr.1: Sollwert Innentemp.= 10 °C  | Nr.15: Belichtungsanlage 40 W/m <sup>2</sup> ,<br>Schaltp. nach Lichtsumme/Tag =<br>240 W(PAR)h/m <sup>2</sup> |
| Nr.2: Sollwert Innentemp. = 3 °C  | Nr.16: Nr. 15 und BHKW - Einsatz   |
| Nr.3: Sollwert Innentemp.=18 °C (Tag),<br>16 °C (Nacht)   | Nr.17: Nr. 16, Nr. 11, gute Abdicht.,<br>mehrlagig   |
| Nr.4: Sollwert Innentemperatur nach<br>Temp.summe/Tag = 432(Grad·h)<br>Sollwertminimum = 10 °C            | Nr.18: k'-Wert Einfachgl. = 8,8 W/(m <sup>2</sup> ·K)  |
| Nr.5: Nr. 4 mit Lüftungstemp. = 30 °C<br>und Sollwertminimum = 5 °C                                       | Nr.19: k'-Wert Einfachgl. = 6,0 W/(m <sup>2</sup> ·K)  |
| Nr.6: Stilllegung 46.-4. Woche  | Nr.20: Nr. 18 und Nr. 11   |
| Nr.7: strenger Winter   | Nr.21: Nr. 19 und Nr. 11   |
| Nr.8: milder Winter   | Nr.22: Außenflächen mit Stegdoppel-<br>platten mit k'-Wert = 4,6 W/(m <sup>2</sup> ·K)                         |
| Nr.9: Nr. 2 und Nr. 7   | Nr.23: ohne Tag-Nacht-Speicherung  |
| Nr.10: Nr. 2 und Nr. 8  | Nr.24: Nr. 2 und Nr. 23  |
| Nr.11: Energieschirm, mittl. Abdicht.,<br>einlagig, stark alum.,<br>Schaltpunkt = 2 W(PAR)/m <sup>2</sup> | Nr.25: maximaler Speicherwert = 3 K  |
| Nr.12: Nr. 11, Schaltp. = 15 W(PAR)/m <sup>2</sup>  | Nr.26: Nr.2 und Nr.25  |
| Nr.13: Nr, 2 und Nr. 11   | Nr.27: Stilllegung von Segment 3   |
| Nr.14: Nr. 11 mit schlechter Abdichtung   | Nr.28: Nr.27 ohne Wärmebilanzierung<br>der Innenflächen  |

Abb. 3.4-1: Ergebnisse mehrerer Simulationsläufe im Vergleich zur Standardsimulation

Die Nichtberücksichtigung des Tag-Nacht-Speichereffektes erhöht den Energieverbrauch des Beispielbetriebes um 9 % (Nr. 23). Bei Verringerung des maximalen Speicherwertes auf 3 K reduziert sich dieser Wert auf 6 %. Bei niedrigen Innentemperatursollwerten erhöht sich der Tag-Nacht-Speichereinfluß auf 59 bzw. 30% (Nr. 24, 26 im Vergleich zu Nr. 2).

Der Einfluß der im Modell verwirklichten Innenwandwärmebilanzierung hängt in entscheidendem Maße von dem Anteil der vorhandenen Innenflächen und den Temperaturunterschieden der einzelnen Gewächshaussegmente ab. Ein Beispiel liefern die Simulationen Nr. 27 und 28.

### **3.4.2 Planung der Heizenergiebereitstellung**

Im folgenden wird ähnlich wie in Kapitel 3.4.1 anhand des gleichen Betriebes aus Abbildung 3.2-6 der Einsatz des Systems verdeutlicht und demonstriert. Grundlage sind die im Anhang C.5 und in Kap. 3.2.3.2 angegebenen Standardwerte. Der Ölpreis wurde fiktiv mit 0.44 DM/l angesetzt und der Gaspreis (gesamt) mit 0.38 DM/m<sup>3</sup> (Jahresbetriebswirkungsgrad 0.9 in beiden Fällen). Es ergibt sich eine notwendige Auslegungsleistung für die Wärmebereitstellung von 1664 kW.

Abbildung 3.4-2 gibt die in diesem Beispiel zugrundegelegten Benutzerangaben wieder. Man sieht, daß nur Öl und Erdgas als Energieträger in Frage kommen, ein abschaltbarer Gasvertrag gewünscht wird und bereits eine funktionsfähige Öl-Anlage im Betrieb vorhanden ist. Der Rechner generiert die in Abbildung 3.4-3 dargestellten Lösungen. Der zuerst genannte Energieträger spezifiziert die Grundlastanlage, der zweite die Spitzenlastanlage. Die Zahlen in den Klammern geben den jeweiligen Auslegungsanteil (% der Gesamtleistung) wieder. Da nur Planungsvarianten mit einer vollständigen Versorgungsmöglichkeit ohne Erdgasnutzung angeboten werden, ist der abschaltbare Gasvertrag richtig 'interpretiert' worden. Abbildung 3.4-4 und 3.4-5 stellen die Programmausgaben für die ökonomische Bewertung der generierten Lösungen dar. Man erkennt die sich im Punkt Investitionsvolumen ergebende Vorteilhaftigkeit der monovalenten Öl-Anlage.

Die in Bezug auf die jährlichen Kosten beste Anlage ist der Lösungsvorschlag 3. Abbildung 3.4-6 zeigt den für diese Lösung vom Programm erzeugten Lösungs- und Erklärungstext. Für jede der gefundenen Lösungen wird ein solcher Text generiert.

Auf die Ausgabe sonstiger Bewertungskriterien mit Hilfe des Hypertextsystems wurde bereits in Abbildung 3.2-22 eingegangen.

BENUTZERANTWORTEN:

Öllagerung möglich => Ja  
Erdgasanschluss möglich => Ja  
Maximalbegrenzung bei Erdgas vorhanden => Ja  
Gasliefervertrag ist => abschaltbar  
Verzicht auf Kesselhaus => Nein  
Kesselhaus am Betriebsrand => Nein  
Holz oder Stroh steht zur Verfügung => Nein  
Fernwärmeanschluss in 800 Meter Entfernung => Nein  
Betriebslage nahe Industrie => Nein  
Wärmepumpenanlage berücksichtigen => Nein  
Schon Anlagen vorhanden => Ja  
Hohe Temperaturansprüche => Ja  
Anzahl vorhandener Anlagen => 1  
Dimension in kW vorhandene Anlage Nr 1 => 1000  
Anlage Nr 1 => Verbrennungsanlage  
Eingesetzter Brennstoff Anlage Nr 1 => Öl  
Einsatz direkte Luftheizer Anlage Nr 1 => Nein  
Einsatz Mehrstoffbrenner Anlage Nr 1 => Nein  
Vorhandene Anlage Nr 1 => voll funktionsfähig  
Alter vorhandene Anlage Nr 1 => unter 15 Jahre  
Hohes Sicherheitsbestreben vorhanden => Nein  
Einsetzbare Ht-Heizflächen vorhanden => Nein  
Strompreis PF pro kWh => 20  
Bereits Gasanschluss bezahlt => Nein  
Abschaltbare Stunden Gasvertrag im Jahr => 20

Abbildung 3.4-2: Anwendungsbeispiel HORTEX (Planung Heizenergiebereitstellung, Benutzerantworten)

Kurzbeschreibung gefundener Lösungen

Lösung 1:	Monovalent	ÖL	
Lösung 2:	Bivalent alternativ,	GAS/ÖL	(100/100)
Lösung 3:	Bivalent alternativ,	GAS/ÖL	(100/100)
Lösung 4:	Bivalent alternativ,	GAS/ÖL	(50/100)
Lösung 5:	Bivalent alternativ,	GAS/ÖL	(30/100)
Lösung 6:	Bivalent alternativ,	GAS/ÖL	(70/100)

Abbildung 3.4-3: Anwendungsbeispiel HORTEX (Planung der Heizenergiebereitstellung, Kurzbeschreibung gefundener Lösungen)

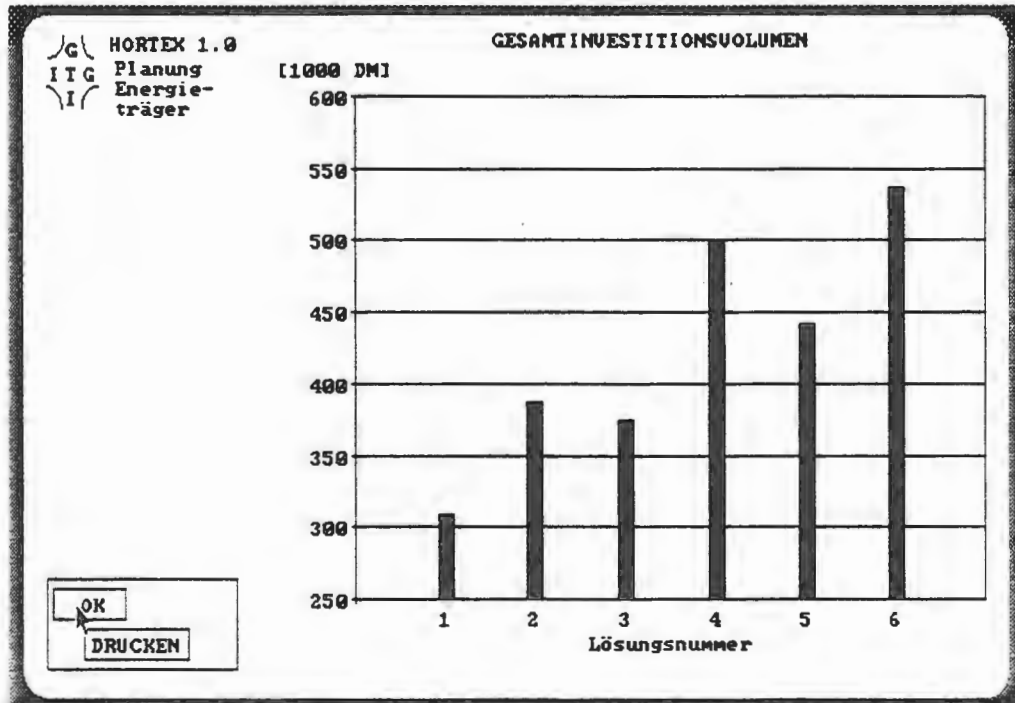


Abbildung 3.4-4: Anwendungsbeispiel HORTEX (Planung der Heizenergiebereitstellung, Investitionsvolumina gefundener Lösungen)

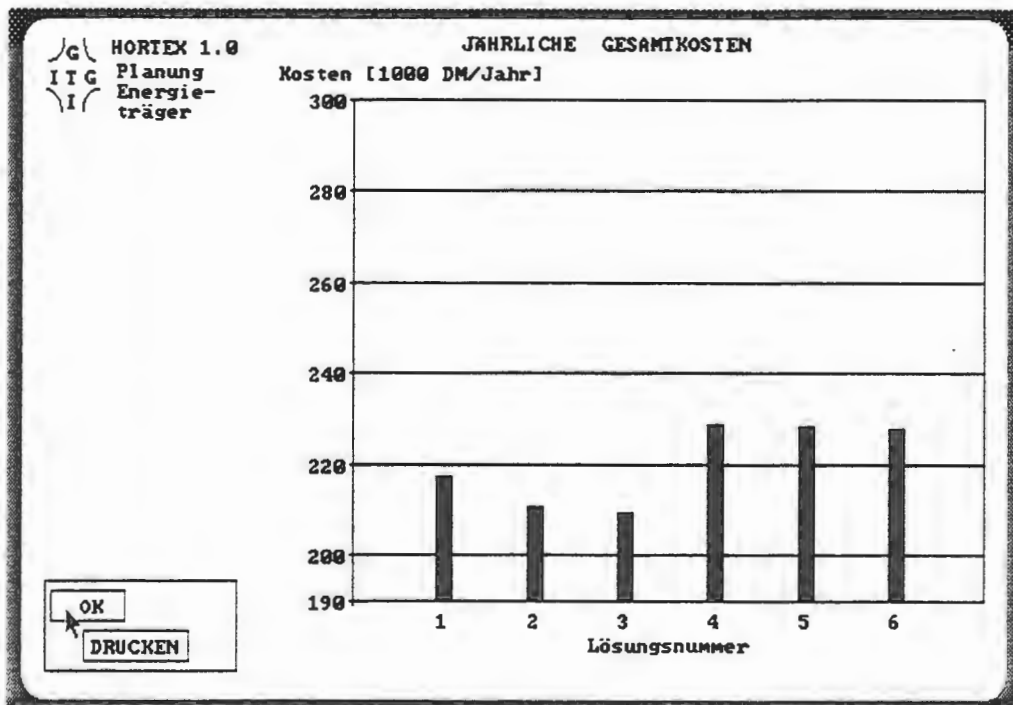


Abbildung 3.4-5: Anwendungsbeispiel HORTEX (Planung der Heizenergiebereitstellung, jährliche Kosten gefundener Lösungen)

BESCHREIBUNG LÖSUNG 3

Bivalent alternativ, Gas / Öl  
Grund-Spitzenlastverhältnis: 100/100

ANLAGENBESCHREIBUNG:

Es wird eine bivalente (d.h. mit zwei Energieträgern) Heizungsanlage geplant. Der Grundlastbetrieb wird nur vom Grundlastenergieträger abgedeckt. In Spitzenlastzeiten wird der Grundlastenergieträger abgeschaltet und die Versorgung geschieht nur mit dem zweiten Energieträger. Da der Energieträger Gas in Spitzenzeiten evtl. nicht zur Verfügung steht, wird ein Öl-Gas-Mehrstoffbrenner eingesetzt.

Einsatz der bestehenden Anlagen :

Nr. 1: Die bestehende Öl-Verbrennungsanlage wird von Ölbetrieb mit Hilfe eines Mehrstoffbrenners auf Gas-Öl-Mehrstoffbetrieb umgerüstet. Die Anlage kann sowohl in der Grundlast als auch in der Spitzenlast eingesetzt werden. Die Anlage wird voll bei der Planung der Neuanlage berücksichtigt.

TECHNISCHE ECKDATEN:

Gesamtleistungsbedarf der Anlage : 1664 kW  
Schon vorhanden und einplanbar: 1000 kW  
Neu zu installieren: 664 kW

ÖKONOMISCHE ECKDATEN:

Neu-Investitionsvolumen: 375177 DM  
Umrüstung vorhandener Anlagen: 53636 DM  
Neue Wärmebereitstellungsanlagen: 150823 DM  
Neue Anschlusskosten: 22776 DM  
Neue Investitionen Wärmeausbringung (Heizungssystem): 147942 DM

Jährliche Energiekosten (bei 3485 MWh): 165528 DM

Gesamtaufwendungen pro Jahr (Kapitalkosten, Energiekosten, Wartungskosten): 209513 DM

Abbildung 3.4-6: Anwendungsbeispiel HORTEX (Planung der Heizenergiebereitstellung, Ausgabertext einer ausgewählten Lösung)



### 3.4.3 Planung der Heizenergieausbringung

Die Abbildungen 3.4-7 bis 3.4-10 geben ein Planungsbeispiel wieder, welches sich auf das erste Segment des in Abbildung 3.2-6 dargestellten Beispielbetriebes bezieht. Die notwendige Auslegungsleistung für das Segment beträgt 499 kW. Für weitere Berechnungen werden die im Programm vorgehaltenen Standarddatenbankdaten (Heizungssysteme, Wärmeabgaben, Preise) (vgl. Abb. 3.2-5) verwendet.

Abbildung 3.4-7 zeigt die in dem Beispiel verwendeten Benutzerantworten. Die restlichen Abbildungen stellen Ausschnitte der vom Rechner gefundenen Lösungen dar (gefundene Slots mit ihren Inhalten, reduziert auf charakteristische und zur Verdeutlichung des Prinzips wichtige Slots).

Abbildung 3.4-8 zeigt die Standardlösung. Abbildung 3.4-9 zeigt den gleichen Planungsfall, nur daß hier die Lösung durch Einsatz der ökonomischen Strategie ermittelt wurde. Abbildung 3.4-10 zeigt die Veränderung der Ausgaben, wenn auf die Kulturtische im Gewächshaus verzichtet wird und die Heizenergieausbringung durch ein Fußbodensystem unterstützt werden soll (Standardauslegung).

Man erkennt schon an diesem kleinen Beispiel, wie flexibel das System auf unterschiedliche Anforderungen und Randbedingungen reagiert. Hinzu kommt noch, daß zahlreiche Daten aus vorherigen Berechnungen (Auslegungsleistung, Gewächshausgeometrie etc.) in die Berechnungen mit eingehen. In allen Fällen wird jedoch die erforderliche Auslegungsleistung im Gewächshaus korrekt ausgelegt. Dazu kommt, je nach Planungssituation, eine unterschiedliche Anzahl von Heizungssystemen zum Einsatz. Es wird deutlich, daß selbst das angestrebte Verhältnis von im Kulturbereich zu im Dachraumbereich auszulegender Heizleistung (welches in allen drei Fällen gleich ist) offenbar durch Planungsrestriktionen modifiziert werden mußte. Der Benutzer kann diese Eigenständigkeit des Systems voll nutzen. Er kann aber auch die Planungsfreiheit so stark einschränken, daß der Rechner praktisch seine gewollte Lösung berechnet (oder keine Lösung findet).

**BENUTZERANTWORTEN:**

Tisch oder Bankkulturen => Ja  
Binderabstand in Längsrichtung in Meter => 4.0  
Manuelle Planungsunterstützung => Nein  
Unnutzbarer Stehwandanteil in Prozent => 1.0  
Rohre im Dachraum notwendig => Nein  
Tischlänge in Meter => 19.0  
Tischanzahl gesamt => 30.0  
Wegeanteil in Prozent => 26.0  
Heizungssysteme unter Tisch möglich => Ja  
Heizungssysteme neben Tisch möglich => Ja  
Gitterunterzug vorhanden => Nein  
Heizungssysteme an Binderstielen möglich => Ja  
Sand/Matten-Bewässerung => Nein  
Düsenrohrbewässerung => Nein  
Rohre im Kulturraum notwendig => Nein  
Höhere Substrattemperatur erwünscht => Ja  
Empfindliche Kulturen => Nein  
Hohe Luftfeuchte im Bestand schlecht => Nein  
Tischflächen geschlossen => Ja  
Brennwerteinsatz => Nein

**Abbildung 3.4-7: Anwendungsbeispiel HORTEX (Planung der Heizenergieausbringung, Benutzerantworten)**

**Abbildung 3.4-8: Anwendungsbeispiel HORTEX (Planung der Heizenergieausbringung, Lösungsausgabe (verkürzt) Standardstrategie)**

**LÖSUNG NR : 1**

Abgedeckte Auslegungsleistung : 501.38 kW.  
Gesamtinvestitionsvolumen: 43097 DM

**Beschreibung Komponente Nr. 1**

**BEZEICHNUNG:** Stahlrohr 51 mm  
Anbringung: an Stehwand  
-> Gesamtwärmeabgabe in kW = 80.92  
-> Auslegungsvorlauftemperatur in Grad C = 90.00  
-> Heizrohrlänge in Meter = 115.33  
-> Gesamtanzahl Heizrohre = 5.00

**Beschreibung Komponente Nr. 2**

**BEZEICHNUNG:** Stahlrohr 51 mm  
Anbringung: unter Tisch  
-> Gesamtwärmeabgabe in kW = 335.17  
-> Auslegungsvorlauftemperatur in Grad C = 90.00  
-> Heizrohrlänge in Meter = 19.00  
-> Anzahl Heizrohre pro Tisch = 4.00

**Beschreibung Komponente Nr. 3**

**BEZEICHNUNG:** Stahlrohr 51 mm  
Anbringung: im Dachraum  
-> Gesamtwärmeabgabe in kW = 85.29  
-> Auslegungsvorlauftemperatur in Grad C = 90.00  
-> Gesamtlänge Verbindungsrohre in Meter = 75.30  
-> Heizrohrlänge in Meter = 41.20  
-> Gesamtanzahl Heizrohre = 13.00

Abbildung 3.4-9: Anwendungsbeispiel HORTEX (Planung der Heizenergieausbringung, Lösungsausgabe (verkürzt) ökonomische Strategie)

Abgedeckte Auslegungsleistung : 502.45 kW  
Gesamtinvestitionsvolumen: 40534 DM

Beschreibung Komponente Nr. 1

BEZEICHNUNG: Stahlrohr 51 mm  
Anbringung: an Stehwand  
-> Gesamtwärmeabgabe in kW = 80.92  
-> Heizrohrlänge in Meter = 115.33  
-> Gesamtanzahl Heizrohre = 5.00

Beschreibung Komponente Nr. 2

BEZEICHNUNG: PP-Rohr 3/4 Zoll  
Anbringung: unter Tisch  
-> Gesamtwärmeabgabe in kW = 248.71  
-> Auslegungsvorlauftemperatur in Grad C = 90.00  
-> Heizrohrlänge in Meter = 19.00  
-> Anzahl Heizrohre pro Tisch = 8.00

Beschreibung Komponente Nr. 3

BEZEICHNUNG: Stahlrohr 51 mm  
Anbringung: an Binderstielen  
-> Gesamtwärmeabgabe in kW = 53.02  
-> Auslegungsvorlauftemperatur in Grad C = 90.00  
-> Heizrohrlänge in Meter = 41.20  
-> Anzahl Heizrohre pro Binderstiel = 3.00

Beschreibung Komponente Nr. 4

BEZEICHNUNG: Stahlrohr 51 mm  
Anbringung: im Dachraum  
-> Gesamtwärmeabgabe in kW = 119.80  
-> Auslegungsvorlauftemperatur in Grad C = 90.00  
-> Gesamtlänge Verbindungsrohre in Meter = 75.30  
-> Heizrohrlänge in Meter = 41.20  
-> Gesamtanzahl Heizrohre = 19.00

Abbildung 3.4-10: Anwendungsbeispiel HORTEX (Planung der Heizenergieausbringung, Lösungsausgabe (verkürzt) bei veränderten Benutzerantworten, Standardstrategie)

Abgedeckte Auslegungsleistung : 504.88 kW  
Gesamtinvestitionsvolumen: 53773 DM

Beschreibung Komponente Nr. 1

BEZEICHNUNG: PE-Rohr 3/4 Zoll  
Anbringung: in Beton  
-> Gesamtheizrohrlänge in Meter = 4653.54  
-> Gesamtwärmeabgabe in kW = 95.84  
-> Auslegungsvorlauftemperatur in Grad C = 40

Beschreibung Komponente Nr. 2

BEZEICHNUNG: Stahlrohr 51 mm  
Anbringung: an Stehwand  
-> Gesamtwärmeabgabe in kW = 80.92  
-> Auslegungsvorlauftemperatur in Grad C = 90.00  
-> Heizrohrlänge in Meter = 115.33  
-> Gesamtanzahl Heizrohre = 5.00

Beschreibung Komponente Nr. 3

BEZEICHNUNG: Stahlrohr 51 mm  
Anbringung: an Binderstielen  
-> Gesamtwärmeabgabe in kW = 104.79  
-> Auslegungsvorlauftemperatur in Grad C = 90.00  
-> Heizrohrlänge in Meter = 41.20  
-> Gesamtanzahl Heizrohre = 18.00  
-> Anzahl Heizrohre pro Binderstiel = 6.00

Beschreibung Komponente Nr. 4

BEZEICHNUNG: Stahlrohr 51 mm  
Anbringung: im Dachraum  
-> Gesamtwärmeabgabe in kW = 223.33  
-> Auslegungsvorlauftemperatur in Grad C = 90.00  
-> Gesamtlänge Verbindungsrohre in Meter = 75.30  
-> Heizrohrlänge in Meter = 41.20  
-> Gesamtanzahl Heizrohre = 37.00

## 4 Diskussion

### 4.1 Modellierung und Darstellung von Fachwissen mit wissensbasierten Systemen

Die wichtigsten Wissensrepräsentationsformen heutiger Expertensysteme sind Produktionsregeln, strukturierte Objekte und der Einsatz logischer Kalküle (COY und BONSIPEEN 1989, S. 52). Sie wurden in dem beschriebenen Expertensystem HORTEX an unterschiedlichen Stellen und in verschiedenen Zusammenhängen verwendet, erprobt, weiterentwickelt und mit arithmetischen und datenbankorientierten Ansätzen gekoppelt.

Abbildung 4.1-1 stellt einen Vergleich zwischen wissensbasierter und arithmetischer Modellierung und Wissensdarstellung dar. Abbildung 4.1-2 vergleicht wissensbasierte Methoden mit Datenbankansätzen. Es wird deutlich, daß jeder Ansatz seine spezifischen Einsatzschwerpunkte und von der Aufgabenstellung abhängige Vor- und Nachteile besitzt. Obwohl durch die objektorientierte Programmierung bei der konventionellen Softwareentwicklung, durch die verschiedenen Frame-Ansätze in wissensbasierten Systemen und durch die Verwendung relationaler Datenbankkonzepte der Unterschied der einzelnen Methoden zunehmend geringer wird (ROTHENBERG 1989, S. 85; STEFIK und BOBROW 1986), bleibt festzuhalten, daß der Auswahl des für ein Problem am besten geeigneten Konzeptes eine entscheidende Bedeutung zukommt. Die Entwicklung des Expertensystems HORTEX hat gezeigt, daß zumindest im Bereich der Gartenbautechnik eine Darstellung von Fachwissen mit wissensbasierten Konzepten allein nicht sinnvoll ist, da verfügbares Fachwissen zu einem großen Teil aus technischen Berechnungen und Kennzahlen besteht. Es besteht keinerlei Veranlassung, gut funktionierende arithmetische Modelle durch logisch geprägte Ansätze zu ersetzen. Auf der anderen Seite betonten bisherige Modellansätze im Bereich der Gartenbautechnik übermäßig stark die quantitativen Eigenschaften der Modellparameter und griffen nur zu einem unbedeutenden Anteil auf Entscheidungs- oder Objektbäume als Modellierungskonzept zurück. Das System HORTEX ist ein Beispiel dafür, daß eine sinnvolle Erweiterung prozeduraler, algorithmischer Modellierungsansätze erfolgen kann. Im einzelnen bieten sich Ansätze zur gezielten Anwendung numerischer Berechnungen, zur Interpretation und Anwendung von Simulationsergebnissen und zur Unterstützung der Parametereingabe von arithmetischen Modellen an. Durch die sehr enge Kopplung von logischer und arithmetischer Wissensdarstellung ist es möglich, ein gut funktionierendes Hybridsystem zu entwickeln.

arithmetisches Gleichungssystem	wissensbasiertes System
<ul style="list-style-type: none"> <li>- numerische Auswertung</li> <li>- Eingabeparameter numerische Werte</li> <li>- Modell stellt direkt Abarbeitungsvorschrift dar</li> <li>- häufig Modellierung zeitlicher Prozesse</li> <li>- Beschreibung funktionaler Abhängigkeiten zwischen Modellobjekten (quantitativ)</li> <li>- Darstellung von Unschärfe durch statistische Maßzahlen</li> </ul>	<ul style="list-style-type: none"> <li>- beweistheoretische Auswertung, Inferenz, Matching</li> <li>- Eingabeparameter diskrete Wert-Menge</li> <li>- strikte Trennung zwischen Wissensbasis und Abarbeitungsmechanismus</li> <li>- Modellierung statischer Zustände</li> <li>- Beschreibung logischer Abhängigkeiten zwischen Modellobjekten (qualitativ)</li> <li>- Beschreibung von Unschärfe häufig durch unscharfe Mengen oder Sicherheitsfaktoren.</li> </ul>

Abbildung 4.1-1: Gegenüberstellung der Modellierung und Darstellung von Fachwissen mit wissensbasierten Systemen und arithmetischen Gleichungen

Datenbanksystem	wissensbasiertes System
<ul style="list-style-type: none"> <li>- algorithmische Informationsverarbeitung mit abgeschlossener Spezifikation</li> <li>- große Datenmengen auf Massenspeichern</li> <li>- Konsistenzüberprüfung für Daten leicht</li> <li>- hohe Frequenz beim Modifizieren der Daten</li> <li>- keine negierten und unscharfen Daten</li> <li>- keine Erklärungskomponente</li> </ul>	<ul style="list-style-type: none"> <li>- beweistheoretische Auswertung, Inferenz, Matching</li> <li>- kleine Daten-/Regelmengen häufig komplett im Speicher notwendig</li> <li>- Konsistenz des Regelwerkes schwer zu prüfen</li> <li>- geringe Frequenz beim Modifizieren der Daten und Regeln</li> <li>- negierte und unscharfe Daten/Regeln möglich</li> <li>- Erklärungskomp. durch Aufzeichnung des Beweisganges möglich</li> </ul>

Abbildung 4.1-2: Gegenüberstellung der Modellierung und Darstellung von Fachwissen mit wissensbasierten Systemen und Datenbankkonzeptionen

Wissensbasierte Systeme werden auch als qualitative Modelle bezeichnet (FOX 1990). Man geht davon aus, daß bei fehlender Genauigkeit vorhandener Wissensstrukturen und beim Fehlen exakter algorithmischer Lösungen in einem Wissensgebiet Expertensystemtechnologien eingesetzt werden können. Häufig werden Systeme konzipiert, bei denen die Wissensmodelle sehr grob und mehr oder weniger zufällig entwickelt werden. Systemschwächen und -fehler sollen dann durch explorative Entwicklung behoben werden. Aussagen wie die, daß Expertensysteme niemals fertig sind und immer Fehler enthalten (vgl. COY und BONSIEN 1989, S. 124), sind die logische Konsequenz. Obwohl sicher Aufgaben existieren, die eine solche Modellentwicklung notwendig machen, führt diese in der Regel zu unkontrollierbaren Systemkosten und schlecht strukturierten Problemlösungsansätzen (SOMMERVILLE 1990, S. 13). Vielen gescheiterten Expertensystemen liegt sicherlich der fehlende Modellierungsaspekt zugrunde. RADERMACHER (1988) meint daher, daß nur wissenschaftlich fundierte Modellierung der Wissensstrukturen zu wirklich effektiven Lösungsansätzen führen kann. Aufgrund der Entwicklung des Expertensystems HORTOX kann diese Ansicht nur bestätigt werden. Erste Versuche, ein Wissensmodell mit einer regelorientierten Shell mit einem rapid-prototyping ähnlichen Vorgehen zu erstellen, führten relativ schnell zu einem lauffähigen Programm. Durch die fehlenden tieferen Strukturen der Wissensbasis war jedoch so gut wie keine Anpassung des Systems an Praxisbedingungen möglich. Ferner versagte das System relativ schnell bei neuen Anforderungen, so daß ständig die Regelbasis verändert werden mußte.

Es bleibt demnach festzuhalten, daß

- der Einsatz wissensbasierter Modellansätze nicht unbedingt zu besseren und effektiveren Problemlösungen führt.
- wissensbasierte Methoden eine sehr genaue und exakte Wissensstruktur voraussetzen. Fehlende oder nur grob vorliegende Wissensselemente können nicht modelliert werden.
- logische Kalküle aufgrund ihrer eingeschränkten Wertigkeit und durch die leicht unübersichtlich werdende Verknüpfung von logischen Operatoren nur gezielt und fundiert zur Modellierung eingesetzt werden können.
- die Darstellung von Wissen mit Hilfe wissensbasierter Systeme - wenn sie erfolgreich sein soll - eine tiefe Modellierung des Wissensgebietes voraussetzt. Im Bereich der objektorientierten Programmierung wird diese Modelltiefe immer wieder hervorgehoben.

Neben dem logischen Aspekt ist bei der wissensbasierten Modellierung die Trennung von Wissensbasis und Abarbeitungsmechanismus von zentraler Bedeutung.

Hieraus ergibt sich eine leichte und schnelle Veränderbarkeit des implementierten Wissens. Im Vergleich zu rein prozeduralen oder datenorientierten Ansätzen werden nicht nur einzelne Modellparameter verändert, sondern das gesamte Wissensmodell kann erweitert, verkleinert oder auch korrigiert werden, ohne daß die prinzipielle Funktionalität des Modells darunter leidet. Der beschriebene Vorteil kann jedoch häufig nicht ausgenutzt werden, da die logischen Abhängigkeiten innerhalb der Wissensbasis bei komplexen Anwendungen nicht mehr zu durchschauen sind. So berichten COY und BONSIPEP (1989, S. 125) von zahlreichen Fällen, in denen Projekte trotz anfänglicher Einsatzfähigkeit in eine massive Wartungskrise geraten sind. Änderungen in der Wissensbasis waren nicht mehr möglich. Unter anderem auch weil Expertensysteme aufgrund ihrer praxisorientierten Funktionalität und ihres Anwendungsbezuges auf diese Änderbarkeit zwingend angewiesen sind, führte das Problem letztlich zum Scheitern der Projekte.

Bei der Entwicklung des Expertensystems HORTEX konnte dieser Effekt bisher aufgrund der durch eingeschränkte Rechnerkapazitäten erzwungenen Modularisierung weitgehend vermieden werden. Eine Ausweitung des Programms sollte daher in erster Linie auf die Erstellung neuer Module anstatt auf die Erweiterung einzelner Wissensbasen hinwirken. Eine Modulgröße von unter 100 logischen Verknüpfungen scheint angemessen zu sein. Dabei sollten die einzelnen Wissensbasen streng voneinander getrennt sein und über definierte Schnittstellen miteinander kommunizieren.

Eine Formalisierung von Wissen mit wissensbasierten Systemen setzt eine einheitlich strukturierte Wissensbasis und einen auf das jeweilige Problem abgestimmten aber dennoch allgemeingültigen Schlußfolgerungsmechanismus voraus. Die Entwicklung des Systems HORTEX hat gezeigt, daß es zum Teil sehr schwierig ist, eine Idee zur Lösung eines nicht trivialen Problems in die von einem Repräsentationsformalismus zur Verfügung gestellte Struktur hineinzupassen. Natürlich kann für jede Problemvariante oder für jedes Teilproblem der geeignete Bearbeitungsmechanismus erstellt werden. Letztendlich führt dieser Weg jedoch weg von der Expertensystemtechnologie hin zur konventionellen prozeduralen Programmierung. Auf der anderen Seite ist es natürlich auch nicht sinnvoll, ein Problem nur wegen des methodischen Aspektes in ein vorgefertigtes Konzept hineinzuzwängen.

Die im Bereich der Planung der Heizungsanlagen implementierte Wissensbasis, die sowohl aus einheitlichen Frames als auch aus übergeordneten speziellen

Suchstrategien und Auslegungsalgorithmen besteht, scheint ein praktikabler und effektiver Kompromiß zwischen den beiden Ansätzen zu sein. Durch den Frameansatz kann das strukturierte und gut formalisierbare Wissen über Heizungssysteme und deren Einsatz im Gewächshaus ideal gespeichert werden. Übergeordnete Auslegungs- und Suchstrategien wenden dieses Wissen mit vorgegebenen Verfahren mehr oder weniger prozedural an. Die dabei eingesetzten heuristischen Auslegungsverfahren (Strategien) haben sich als sinnvolle und pragmatisch orientierte Problemlösung erwiesen. Sie ermöglichen trotz der Komplexität der Aufgabe die Ausgabe konkreter und für den Programmnutzer wertvoller Informationen. Bisherige Expertensysteme im Bereich der Agrartechnik (vgl. Kap. 2.1.3) berücksichtigten so gut wie gar nicht derartige direkte und heuristische Auswahlverfahren. Bei ihnen bestimmen die Programmnutzer durch die Angabe von Modellparametern und deren Spezifikation maßgeblich die Auswahl von Planungsvarianten. Zukünftige Entwicklungen sollten daher die Implementierung verschiedener Planungsstrategien zumindest als Erweiterung bestehender Problemlösungsparadigmen einsetzen. Kapitel 3.2.3.4.2 zeigt Realisierungsmöglichkeiten auf.

Die Trennung von Wissensbasis und Abarbeitungsmechanismus ermöglicht es, eine durch die Wissensbasis generierte Erklärungskomponente zur Verfügung zu stellen. Diese Erklärungskomponente soll eine gute Programmtransparenz und dadurch bedingt eine hohe Benutzerakzeptanz hervorrufen. Zahlreiche Anstrengungen zur Realisierung dieses Zieles wurden bisher unternommen (vgl. RÖSNER 1986, S. 11ff; APPELRATH 1985, S. 78ff). Sie führten aber offensichtlich bis heute zu keinem brauchbaren Ergebnis. Verfügbare Erklärungskomponenten können zwar den Lösungsgang exakt wiedergeben, diese Angaben sind aber für den Programmnutzer relativ wertlos, da er die Zusammenhänge des Gesamtableitungsbaumes nicht kennt. So wurde beim Einsatz von HORTEx mehrfach festgestellt, daß den Benutzer nicht so sehr die Erklärungen zu gefundenen Lösungen interessieren, sondern er möchte vielmehr wissen, wieso bestimmte Lösungen nicht in Frage kommen. In diesen Fällen kann nur die Angabe des gesamten Lösungsbaumes helfen. Versuche, im Expertensystem HORTEx zumindest Teilbereiche des Ableitungsbaumes als Erklärungskomponente anzubieten, scheiterten letztlich auch hier an der geringen Aussagekraft der gegebenen Informationen.

Noch problematischer sieht es im Frame-Bereich aus: Eine benutzergerechte Erklärungskomponente ist hier noch nicht einmal in Ansätzen zu erkennen und ist auch in Zukunft aufgrund der Komplexität des Modellierungsansatzes nicht zu erwarten.

Insgesamt wird deutlich, daß Wissensrepräsentation und Erklärung des Wis-



sens inhaltlich nicht identisch sind, da sie unterschiedliche Ziele verfolgen und auf unterschiedlichen Abstraktionsebenen aufbauen (vgl. CHANDRASEKARAN et al. 1988). Eine Selbsterklärungsfähigkeit von Expertensystemen wird wahrscheinlich in absehbarer Zeit nicht zu realisieren sein. COY und BONSIEPEN (1989, S. 69) schreiben dazu: "Erklärungskomponenten als Programmierhilfen sind ein wertvolles Hilfsmittel für den Systemersteller..... Das Expertensystem als 'Lehrer' gehört dagegen zu den uneingelösten Ansprüchen der KI-Ideologie." Bei heutigen Expertensystemen verbirgt sich aus den beschriebenen Gründen hinter der Erklärungskomponente in der Regel nicht mehr als eine Trace-Funktion für den Systemersteller (VOß 1988). Die Erklärungen für den Benutzer werden als statische Hilfsfunktionen implementiert, die dann nicht mehr direkt durch die Wissensbasis generiert werden. COY und BONSIEPEN (1989, S. 74) weisen auf die dadurch entstehenden Probleme hin, daß durch die Abtrennung der Erklärungskomponente von der Wissensbasis letztendlich zwei unterschiedliche Wissensmodelle im Rechner vorhanden sind, die eventuell zu Inkonsistenzen führen können.

Im Expertensystem HORTEX konnte diese Problematik nicht festgestellt werden. Vielmehr wurde deutlich, daß durch eine übersichtliche Gestaltung der Benutzerführung (unterstützt durch Hypertext und Graphikelemente) die Wichtigkeit einer auf der Wissensbasis aufbauenden Erklärungskomponente reduziert werden kann (vgl. auch STROTHOTTE und BÖCKE 1989; RATH 1991b). Der Ersatz einer dynamischen Erklärungskomponente durch ein statisches Hilfs- und Erklärungssystem bedeutet in diesem Punkt aber eine Abkehr von der Expertensystemtechnologie und somit den Einsatz konventioneller Programmier- und Darstellungstechniken.

Neben einer bildhaften und ergonomischen Gestaltung der Erklärungskomponente scheint für den praktischen Einsatz von wissensbasierten Systemen eine adäquate Modellierung des Benutzerkreises von großer Wichtigkeit zu sein, da die unterschiedlichen Sichtweisen einzelner Benutzer maßgeblich die Effektivität der Mensch-Maschine-Schnittstelle und somit auch den Nutzen des Gesamtsystems beeinflussen (vgl. LEHNER und KRALJ 1988). Darüberhinaus sollten sich sowohl Dialogschnittstellen als auch Erklärungskomponenten dem aktuellen Wissensstand des Benutzers anpassen (vgl. LENTZ 1991, S. 93).

Da bei der Entwicklung des Expertensystems HORTEX die methodischen Aspekte der Wissensrepräsentation im Vordergrund standen und keine direkte Spezifikation der Endbenutzer erfolgte, fand dieses wichtige Kriterium im Rahmen der vorgestellten Arbeiten nur geringe Beachtung. Weitere Entwicklungen sollten daher erstens den Kreis der Endbenutzer eines Systems klar abgrenzen und zweitens das User-Interface auf diese Gruppe abstimmen (oder unter-

schiedliche Schnittstellen für unterschiedliche Benutzergruppen vorhalten). Inwieweit die Expertensystemtechnologie in diesem Bereich einen Vorteil gegenüber konventionellen Programmen hat, müssen jedoch erst noch weitere Forschungen zeigen.

#### **4.2 Darstellung von unscharfem Wissen**

Die in Kapitel 3.2.3.3 und 3.2.3.4 beschriebenen Systeme verwenden keine Ansätze zur direkten Modellierung von Unsicherheiten, Unschärfen etc.. Lediglich das im Bereich der Heizungsauslegung konzipierte Bewertungspunktesystem führt in gewisser Weise zu einer Aggregation mehrerer unterschiedlich gewichteter Wissensselemente. Die vorgestellten Produktionsregeln und Frames sind ansonsten rein zweiwertig ausgerichtet, d.h. jeder Knoten/jedes Objekt repräsentiert einen aussagenlogisch eindeutigen Zusammenhang. Bei bisherigen Programmanwendungen wurde offensichtlich, daß der Programmbenutzer diese 'Entscheidungshärte' bei der Abarbeitung der Regel/Objektbäume merkt. Es wurde vielfach die Meinung geäußert, daß gerne ein 'Sowohl-als-auch' oder aber ein 'abgeschwächtes Ja bzw. Nein' als Eingabe auf eine Rechnerfrage gemacht würde. Gleichzeitig wurde allerdings auch gefordert, daß das System konkrete Planungsvorschläge erbringen soll, die vor allen Dingen nachvollziehbar und einleuchtend sein sollen. Erste Programmversionen verfolgten daher das Ziel, Regeln mit Certainty-Factors darzustellen. Als Ergebnis wurden dem Benutzer alle Lösungsvarianten mit Bewertungspunkten angezeigt. Eine Abbildung mit statistischen Ansätzen kam aufgrund fehlender Grunddaten nicht in Frage. Der Einsatz von Schwellenwerten erfolgte nicht. Der beschriebene Ansatz wurde nicht weiterverfolgt. Die wesentlichen Gründe hierfür sind:

- Es war nicht möglich, dem Benutzer die Verarbeitung von Punktsystemen bei einer größeren Anzahl von Regeln transparent zu machen. Der Programmbenutzer konnte nicht nachvollziehen, wie Entscheidungen zustande gekommen waren.
- Die Modellierung der Entscheidungsbäume wurde in erheblichem Maße erschwert. Insbesondere die Verarbeitung der logischen Operatoren Und, Oder, Nicht und der Einsatz von Schlußfolgerungsmechanismen führte zu Systeminkonsistenzen. Die programmiertechnische und logische Validierung der Wissensbasis war so gut wie nicht möglich.
- Die zentralen Planungsregeln eines Themenbereiches gingen durch den Ein-

satz von Unschärferepräsentationsformen verloren. Es zeigte sich, daß gerade durch die beschriebene Entscheidungshärte der zweiwertigen Und-Oder-Graphen dem Benutzer Ursache-Wirkungsmechanismen besser vor Augen geführt werden können.

- Die Verdichtung von unterschiedlich bedingten Unsicherheiten (im Bereich der Energieversorgung insbesondere fehlende Informationen und unterschiedliche Expertenmeinungen) auf einen Sicherheitswert war unbefriedigend. Das System konnte nicht aufzeigen, ob ein niedriger Sicherheitswert durch fehlende Benutzereingaben, durch fehlende Information in der Wissensbasis, durch widersprüchliche Expertenangaben oder durch die Kombination von negativen mit positiven Bedingungen zustande gekommen war.

Die Einsatzmöglichkeit unscharfer Menge (fuzzy sets) wurde nicht untersucht. Es ist jedoch unwahrscheinlich, daß die oben beschriebenen Grundprobleme durch Fuzzifikation, mengenorientierte Verknüpfungsregeln und Defuzzifikation behoben werden können. So schreiben GRAHAM und JONES (1988, S. 150f) auch, daß der Einsatz der Fuzzy-Logik innerhalb von Expertensystemen nur in abgegrenzten und statistisch überprüfbaren Bereichen erfolgen sollte. Eine Ausstattung des beschriebenen Frame-Konzeptes mit Fuzzy-Ansätzen ist darüberhinaus programmier- und modellierungstechnisch sehr schwierig. Zukünftige Arbeiten sollten in diesem Bereich Klärung bringen.

Als Fazit bleibt festzuhalten, daß die Problematik des Einsatzes der zweiwertigen Logik trotz verfügbarer Vorschläge von Seiten der KI nicht zufriedenstellend gelöst werden konnte. Behauptungen, daß Expertensysteme besonders bei unscharfem Wissen eingesetzt werden können, wurden für den Bereich der Gartenbautechnik nicht bestätigt. Vielmehr besteht die schon von RADERMACHER (1988) beschriebene Gefahr, daß versucht wird, Modellierungsdefizite durch Unschärfefaktoren auszugleichen, welche dann letztlich die Defizite nur noch verschlimmern. RICH (1988, S. 214) schreibt dazu: "...wir sollten Zeichen lieber als Menge hochwertiger Eigenschaften denn als scheinbar stark zufällige Sammlungen von Tintenflecken darstellen. ... Wenn die Eingabe aus von Menschen gelieferten groben Schätzwerten besteht, ist es nicht nur eine Zeitverschwendung, sondern auch trügerisch, zu versuchen, einen hochpräzisen Schätzwert der Wahrscheinlichkeit der Schlußfolgerung zu berechnen."

Zur Überwindung beschriebener Probleme wurde im Expertensystem HORTEX eine pragmatische Lösung implementiert, bei der Bewertungsregeln für bestimmte Lösungsansätze mit einem Punktesystem direkt angezeigt werden ohne weitere Datenverdichtung (vgl. Abb. 3.2-22). Diese Art der Wissensrepräsentation

tion führt zwar nicht zur direkten Anwendung von Expertenwissen, sie stellt aber ein wertvolles Informationsangebot dar und verfälscht zumindest nicht die durch logische Ableitung gefundenen Lösungen. Zukünftige Entwicklungen werden nach COY und BONSIEPEN (1989, S. 140) verstärkt diese oder ähnliche Repräsentationsmechanismen nutzen.

#### **4.3 Möglichkeiten und Grenzen beschriebener Wissensrepräsentationsformen**

Grundlage jeder wissensbasierten Darstellung im Expertensystem HORTEX sind Aussagen in Horn-Klausel-Form, realisiert in der Programmiersprache Prolog. Der ausgewählte Dialekt Turbo-Prolog (Borland) stellt eine Mischform zwischen Pascal und Prolog dar, da er eine Variablentypisierung zwingend voraussetzt. Eine prädikatenlogische Programmierung ist somit nur eingeschränkt möglich. Trotz dieser Einschränkungen wurde die Sprache intensiv eingesetzt. Die Variablen-Typisierung wurde dabei nicht unbedingt als Nachteil empfunden, vielmehr wurde sie zur Code- und Systemüberprüfung genutzt.

Durch die in der Programmiersprache Prolog implementierten Schlußfolgerungsmechanismen (einschließlich Unifikation und Backtracking) und die optimale Möglichkeit, rekursive Listenverarbeitung durchzuführen, stellt Prolog fast alle notwendigen Eigenschaften für die Entwicklung von Expertensystemen zur Verfügung. So ist es auch nicht verwunderlich, daß zahlreiche Lisp-orientierte Shells Prolog als Wissensrepräsentationsformalismus anbieten. Besonders durch die integrierten Unifikationsalgorithmen lassen sich in kürzester Zeit effektive Abfragemechanismen für die verschiedenen Arten von Datenbanken und Wissensbasen entwickeln.

Im Expertensystem HORTEX wurden auch zahlreiche prozedural orientierte Lösungsansätze in Prolog implementiert, um notwendige Datenübergaben zu prozeduralen Programmiersprachen zu umgehen und um den integrierten Backtrackingmechanismus zu nutzen. Die vielfach beschriebene Problematik, Fehler in Prolog-Programmen nur sehr schwer entdecken zu können (vgl. ALTENKRÜGER 1987, S. 111), konnte nicht bestätigt werden. Der integrierte Trace-Mechanismus bietet eine sehr gute Möglichkeit, Programmabläufe zu verfolgen. Da man alle Wissensrepräsentationsformen heutiger Expertensysteme als einen Ausschnitt der Prädikatenlogik betrachten kann (vgl. RÖSNER 1986, S. 24), wird zum gegenwärtigen Zeitpunkt von Seiten des Autors die Wissensdarstellung in Form von Prolog als eine hervorragende Basis für die Entwicklung von Expertensystemen gesehen. Ob Prolog in einer Shell, in Lisp oder in konventionellen Programmiersprachen realisiert wird, ist dabei zweitrangig.

Der Einsatz des von der Firma Borland nicht mehr vertriebenen Prologcompilers wird als sehr problematisch bewertet. Obwohl die beschriebenen Maßnahmen zur Einbindung des Prologsystems in eine konventionelle Graphikoberfläche erfolgreich waren, stellten sie eine erhebliche Einschränkung bei der Systementwicklung dar, da ein direktes Tracing unter der Graphikoberfläche nicht möglich war. Ebenso erschwerten massive Softwarefehler besonders beim Zugriff auf externe Datenbanken den Einsatz des verwendeten Systems. Bei der gesamten Entwicklung wurde deswegen darauf geachtet, keine Borland-spezifischen Prologprädikate einzusetzen. Dem Einsatz eines anderen Prologdialektes steht daher nichts im Wege, solange er die Integration in die bestehende Softwareumgebung ermöglicht. Auf eine voll graphische Benutzeroberfläche sollte dabei allerdings aus ergonomischen Gründen auf keinen Fall verzichtet werden (vgl. HIBBELER 1989; CHARLIER 1990). Inwieweit der direkte Nachfolger von Turbo-Prolog (PDC-Prolog, Prolog Development Center) einzusetzen ist, bleibt abzuwarten. Ein durchaus positives Einsatzbeispiel des 'neuen Turbo-Prolog' zeigen VOGES und EPKE (1991).

In der Literatur wird vielfach auf Probleme hingewiesen, die durch die Abstraktion von Problemen auf die logische oder auch prädikatenlogische Ebene entstehen. So schreibt ALTENKRÜGER (1987, S. 111) im Hinblick auf die Schwächen des Prologeinsatzes: "Vielen fällt logisches Denken schwer". SHADBOLT (1989) bestätigt diese Aussage, indem er schreibt: "...all the research in this area shows that at a fundamental level the basic laws of logic and associated proof strategies are difficult for people to apprehend and follow". SAVORY (1988, S.144) meint: "... Dennoch haben Experten in anderen Gebieten Probleme, Prädikatenlogik der ersten Stufe zu verstehen bzw. Wissen zu verwenden, das in der Prädikatenlogik erster Stufe dargestellt ist, weil die Granularität der Sprache sehr fein ist und sie keine adäquaten Möglichkeiten bietet, komplexere Konstrukte zu definieren".

Zur Spezifizierung und einfacheren Handhabung von Prologregeln wurde daher in Kapitel 3.2.2.1 beschriebene Produktionsregelformalismus entwickelt. Durch die Definition von spezifischen Regelgruppen (*hat\_moeglichkeit*, *hat\_variante etc.*) konnte ein effektiv arbeitendes Metaregelsystem aufgebaut werden, welches die sonst bei Prologregeln auftretende Schwierigkeit der Regelauswahl nach Stellung im System entschärft.

Der Einsatz von Verkettungslisten ermöglicht ferner die Darstellung von vorwärts verketteten Ableitungsbäumen, welche sonst mit einem reinen (rückwärtsverkettenden) Prologsystem nicht oder nur sehr schwer möglich sind. Diese Erweiterung führte bei der Systementwicklung zur entscheidenden Verbesserung der Modellierung. Durch die Verwendung von Lösungslisten anstelle einer dynamischen Wissensbasis kann die im Ableitungsbaum erfolgte Suche leicht

dokumentiert und gespeichert werden. Ebenso ist das Zurücksetzen gefundener Lösungen beim Scheitern eines Ableitungszweiges ohne Schwierigkeiten und unter voller Ausnutzung des Backtracking-Mechanismus möglich. Es wird allerdings durch einen erhöhten Arbeitsspeicherbedarf erzielt. Durch die Entwicklung und den Einsatz der *wenn*-Prädikate ist es möglich, in Verbindung mit Backtracking Problemlösungsräume situationsspezifisch zu durchsuchen. Dabei ist der Einsatz von variablen Regeln besonders effektiv, da er eine Reduzierung des Regelumfangs ermöglicht und grundlegende Regelaspekte herauskristallisiert. Ebenso hat sich das dargestellte Fragesystem mit Erklärungseinbindungen und der Unterscheidung in Oder-Fragen, Alternativ-Fragen (ausschließendes Oder) und numerische Fragen bewährt.

Probleme stellen bei dem implementierten Produktionsregelsystem der notwendige Speicherbedarf und die Gefahr der rekursiven Regelaufrufe dar. Durch sorgfältige Modularisierung und bedachte Regelimplementierung lassen sich die beiden Nachteile jedoch umgehen.

Ferner muß klar gesagt werden, daß der beschriebene Mechanismus keinerlei Hilfen bei der Erstellung der Ableitungsbäume zur Verfügung stellt. In diesem Punkt haben induktive Shells oder Regeltools mit graphischer Unterstützung mit Sicherheit große Vorteile (als eindrucksvolles Beispiel: FAIRCHILD et al. 1988). Trotzdem lassen sich mit dem entwickelten Produktionsregelschema einfach, effektiv und ökonomisch angemessen logische Ableitungsbäume implementieren und in hybride Systeme einbinden. Je datenorientierter und an Objekten ausgerichteter die zu modellierenden Wissensstrukturen sind, desto mehr muß dabei auf konzeptfremde Mechanismen (z.B. Prologdatenbankprädikate) zurückgegriffen werden. Der durch das Regelsystem bedingte Vorteil wirkt sich immer weniger aus.

In diesen Fällen sollte dann auf das beschriebene Frame-Konzept zurückgegriffen werden, da hierbei besonders die Verknüpfung von Objekten mit ihren Eigenschaften im Vordergrund steht. Bisherige Expertensysteme im Agrarbereich berücksichtigten dieses Konzept nur vereinzelt. Die wesentlichen Vorteile framebasierter Wissenrepräsentation sehen PHAM und PHAM (1988) darin, daß tiefliegende Wissensstrukturen angemessen modelliert werden können. Im Expertensystem HORTEx konnte durch den Einsatz von Standardwerten und Vererbungsmechanismen das Modell immer weiter verfeinert werden, ohne daß das Ursprungsmodell neu konzipiert werden mußte. Da das Framekonzept besonders auf die Kopplung zwischen prozeduraler und deklarativer Wissensdarstellung aufbaut, kann es als das ideale Konzept zur Modellierung zumindest von Teilbereichen gartenbautechnischer Probleme bezeichnet werden. Es setzt

allerdings einen überlegten und fundierten Einsatz der Vererbungsstrategie und der Defaultwerte voraus. Bei größeren Systemen kann diese Tatsache leicht zu Problemen führen (vgl. COY und BONSIEPEN 1989, S. 56), die dann das gesamte Konzept in Frage stellen. Ebenso wie beim Produktionsregelansatz ist demnach auch hier auf eine strenge Modularisierung Wert zu legen.

Durch den Einsatz von *zeige/intern*-Slotfacetten wurde der direkte Anschluß an die Ausgabeschnittstelle geschaffen. Wesentliche Angaben können hervorgehoben werden, unwesentliche Informationen bleiben nur internen Rechengängen zugänglich. Die Reduzierung der prozeduralen 'Anhänger' auf *wenn-benötigt*-Slots (vgl. Kap. 3.2.2.2) scheint bei technischen Planungssystemen nur von Vorteil zu sein, da der Frameansatz einfacher und transparenter zu gestalten ist. Die von SCHNUPP und NGUYEN HUU (1987, S. 185) vertretene Ansicht, daß in wissenschaftlich-technischen Expertensystemen *if-needed*-Prozeduren die dominierende Rolle spielen, konnte somit bestätigt werden. Gleichzeitig wird die von SAVORY (1988, S. 163f) beschriebene Möglichkeit und Wichtigkeit des Einsatzes von *if-added*-Prozeduren, mit denen zum Beispiel Wertebereiche und Standardvorgaben von neu definierten Frames bestimmt werden können, nicht als unbedingt notwendig bewertet. Allgemein sollte bei der Realisierung des Framekonzeptes darauf geachtet werden, daß nicht zu viele Slotfacetten zur Verfügung stehen, da die gesteigerte Funktionalität durch zunehmende Komplexität des Ansatzes in Frage gestellt wird. Die in Kapitel 3.2.2.2 eingesetzte Anzahl von 4 Slotarten (Facetten) war bei der Systementwicklung ausreichend.

Die Realisierung des Frame-Konzeptes führte zu erheblichen programmiertechnischen Problemen, für die fast ausschließlich der unzureichende Prolog-Compiler der Firma Borland verantwortlich war. Eine Weiterentwicklung mit der eingesetzten Software ist daher besonders im Bereich des Frame-Ansatzes auf keinen Fall sinnvoll. Dazu muß ebenso wie bei der Diskussion des Produktionsregelansatzes gesagt werden, daß der implementierte Frame-Mechanismus keinerlei Maßnahmen zum Abfangen von Ringschlüssen oder Fehlbelegungen von Frames und Slots besitzt. Ebenso konnte nicht eine direkte Einbindung von Benutzerangaben in die Framestruktur erfolgen, weil das - ähnlich wie beim Produktionsregelansatz - durch die fehlende Grafikschnittstelle des Prologsystems den Abbruch der Bearbeitung der Frames und den Aufruf der Grafikschnittstelle notwendig machen würde. Eine erneute Analyse des Framesystems nach jeder Benutzerangabe wäre die logische Konsequenz. Dieses ist jedoch bei den ohnehin schon kritischen Rechenzeiten des Frame- und Auslegesystems nicht sinnvoll. Die im Expertensystem HORTEX praktizierte Lösung, zuerst alle Benutzereingaben einzuholen und dann das Auslegesystem zu starten, ist zwar ein umständliches aber dennoch praktikables Vorgehen. Es führt aber eventuell da-

zu, daß der Benutzer nicht notwendige Eingaben in das System machen muß. Der Einsatz einer voll ausgestatteten Shell würde in diesem Bereich sicherlich große Vorteile erbringen. Für kleinere, wohl strukturierte und bedacht programmierte Anwendungen, die in ein größeres hybrides Gesamtkonzept integriert werden sollen, reicht der beschriebene Formalismus jedoch aus. Bei einem grafikfähigen Prologsystem dürfte es ferner keine Schwierigkeit sein, den Ansatz entsprechend zu modifizieren.

Ebenso wie der Produktionsregelansatz hat auch das Frame-Konzept seine Einsatzgrenzen. Nehmen die prozeduralen Algorithmen eine so starke Dominanz ein, daß die logischen Verknüpfungen einzelner Objekte von zu vernachlässigender Rolle sind, sollte auf den Ansatz verzichtet werden. Rein prozedurale Wissensaspekte können mit konventionellen Programmieransätzen wesentlich effektiver beschrieben werden.

Wichtigster Modellierungsansatz in dem beschriebenen Expertensystem HORTEX ist die Vereinigung von wissensbasierten Ansätzen mit Simulationsrechnungen und Datenbankkonzeptionen zu hybriden Systemen. Die Entwicklung entsprechender Umwandlungsalgorithmen und ASCII-Schnittstellendateien war mit erheblichem Arbeitsaufwand verbunden, so daß versucht wurde, mit möglichst wenig Transformationen auszukommen.

Zukünftige Entwicklungen sollten daher von Anfang an das Prologsystem so auswählen, daß eine möglichst direkte Kopplung unterschiedlicher Programmierparadigmen möglich ist. Sollten große hybride Shells in Zukunft in eine für den Gartenbau ökonomisch angemessene Preisklasse kommen, stellen sie sicherlich die beste Möglichkeit dar, unterschiedliche Konzepte zu vereinigen. Bis dahin müssen aber wahrscheinlich 'selfmade-Schnittstellen' und 'Eigenkonstruktionen' diese Problematik lösen.

#### **4.4 Einsatz von Expertensystemen in der Gartenbautechnik**

Der Einsatz von Expertensystemen wird zur Zeit als ernüchternd bezeichnet. Die in Kapitel 2.1.3 beschriebenen Expertensysteme zeigen, daß nur ein sehr geringer Anteil erfolgreich ist. Es wird deutlich, daß die Expertensystemtechnologie deutlich hinter den vielfach aufgezeigten Entwicklungsmöglichkeiten (vgl. HOFER 1987; NEIDHARDT 1989; HAUG 1990; BERLIN 1991) hinterherhinkt. Auf dem Softwaremarkt dominieren nicht etwa lauffähige Expertensysteme, sondern Expertensystem-Shells (RADERMACHER 1988). Viele Anwendungen werden von vornherein als "Einstieg in zukünftig bedeutsame Technik" betrachtet (WURR 1989).



Das Thema Expertensystem wird in der Fach- und vor allem auch in der populärwissenschaftlichen Diskussion überbewertet (MERTENS 1987). Im wesentlichen werden folgende Ursachen für die bisher ernüchternde Bilanz bestehender Systeme gegeben (RADERMACHER 1988; MERTENS 1987; HAUG 1990; CHARLIER 1989) :

- Fehlende Wissensmodellierung, unzureichender Modellierungsrahmen
- Wahl des falschen Projektes
- Fehlende Einbettung in die bestehende Softwareumgebung
- Fehlende Kombination von unterschiedlichen Modellierungsansätzen
- Fehlende Akzeptanz beim Benutzer

Obwohl HARMON und KING (1989, S.11) noch von einer zweiten Expertensystemwelle sprechen (nachdem die erste nicht eingetroffen ist), hat der Expertensystemmarkt mit höchstens 1 % des Softwaremarktes in der Bundesrepublik zur Zeit nur eine geringe Bedeutung (COY und BONSIEPEN 1989, S. 82). Wahrscheinlich wird auch in Zukunft die wissensbasierte Softwaretechnik nur eine von vielen Modellierungstechniken sein. Schwerpunkte werden weiterhin konventionelle Systeme bleiben, die um einen wissensbasierten, logikorientierten Programmier- und Darstellungsaspekt erweitert werden.

Inwieweit die stark aufkommende objektorientierte Programmierung den wissensbasierten Aspekt übernehmen wird, bleibt abzuwarten. Tatsache ist, daß bei vielen Systemen schon heute keine eindeutige Zuordnung zur Expertensystemtechnologie oder objektorientierten Programmierung zu machen ist. Sollte es von Seiten der relationalen Datenbanken gelingen, effektive Inferenz- und Ableitungsmechanismen in die Datenbankverwaltung zu integrieren, könnten auch Datenbanken einen wesentlichen Anteil der wissensbasierten Darstellungsformen übernehmen.

Im Bereich der Gartenbautechnik ist ebenso mit einem eher geringen Einsatz von reinen Expertensystemen zu rechnen, obwohl auch hier das prinzipielle Potential (vgl. HOLT 1989) bei weitem noch nicht ausgeschöpft ist. Aufgrund der existierenden ökonomischen Randbedingungen des Gartenbaues ist es zur Zeit nahezu unmöglich, daß arbeits- und wartungsintensive Expertensysteme ohne öffentliche Unterstützung erstellt und im Bereich der Gartenbautechnik eingesetzt werden können. Es ergibt sich das in Abbildung 4.4-1 schematisch dargestellte Einsatzspektrum. Die von Liebig (1991, zitiert in LINK 1991, S. 64) gemachte Aussage, daß der Einsatz von Expertensystemen in bestimmten gartenbaulichen Bereichen in 5 bis 10 Jahren bei über 50 % liegen wird, scheint entweder auf einem ganz anderen Expertensystemverständnis zu beruhen oder aber ist reine Spekulation. Vielmehr ist auch im Bereich des Gartenbaues und da

besonders im technischen Bereich, zu erwarten, daß konventionelle, aus wissenschaftlichen Versuchen resultierende Modelle mit logischen und gut formalisierten Lösungsansätzen so ergänzt werden, daß ein praktischer Einsatz in einer EDV-Anlage erfolgen kann. Die vorliegende Arbeit kann als Grundlage für derartige Entwicklungen dienen.

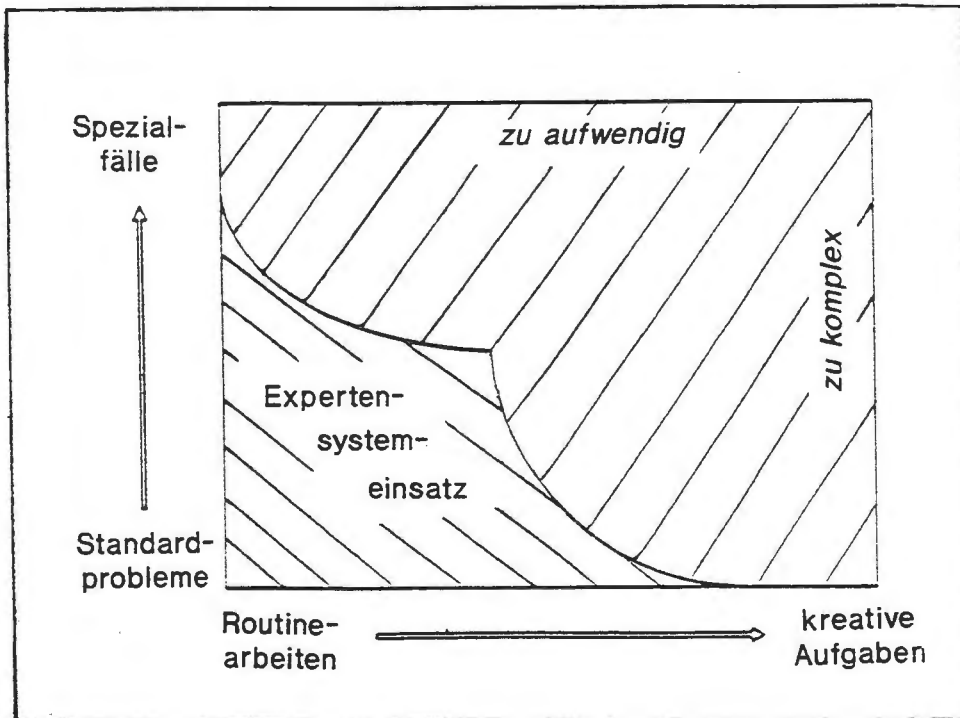


Abbildung 4.4-1: Schematische Darstellung des möglichen Einsatzgebietes von Expertensystemen im Bereich der Gartenbautechnik

#### 4.5 Bewertung des beschriebenen Systems

##### 4.5.1 Validierung und Einsatzfähigkeit

Wissensbasierte Systeme gelten im allgemeinen aufgrund ihres Aufgaben- und Einsatzbereiches als inhaltlich schwer validierbar. Darüberhinaus ist bei komplexen Systemen aufgrund der kombinatorischen Möglichkeiten auch die logische und formale Validierung aufwendig, wenn nicht sogar unmöglich (vgl. ENAND und KAHN 1988). Vielfach können nur punktuelle Tests mit konstruierten Fällen zur ansatzweisen Validierung eingesetzt werden. Neben der Gültig-

keit und Korrektheit des implementierten Wissensmodells spielt dabei besonders die Einsatzfähigkeit und Nutzbarkeit des Systems eine wichtige Rolle. Auf die Probleme der explorativen System- und Modellentwicklung wurde bereits in Kapitel 4.1 eingegangen.

Das beschriebene System wurde anhand zahlreicher fiktiver und realer Beispiele überprüft und verbessert. Schwerpunkt aller Betrachtungen war und ist dabei die grundsätzliche Einsatzfähigkeit und die logische und inhaltliche Korrektheit des Programms. Interessante Verbesserungsvorschläge wurden im Rahmen von Demonstrationen und Vorführungen auf der Fachmesse 'plantec 1991' (vgl. RATH und TANTAU 1991) aufgegriffen und anschließend in die Wissensbasis implementiert. Ebenso wurden Verbesserungen und Praxiskorrekturen, die von Dipl.-Ing. W. Gabloffsky (LWK Hannover) im Rahmen einer Programmsichtung vorgeschlagen wurden, anschließend übernommen. Wertvolle Anregungen ergab die Systemüberprüfung mit den Gartenbautechnikberatern der LWK Rheinland. Weitere Verbesserungen hinsichtlich der Benutzerführung erfolgten aufgrund zahlreicher Programmtests mit unterschiedlichen Personen. Die Ergebnisse bzw. Teilergebnisse von Energiebedarfsberechnungen wurden bereits im Rahmen einer Studie veröffentlicht (VON ELSNER 1991).

Bisherige Anwendungen haben gezeigt, daß mit dem beschriebenen Front-End-System Betriebe modelliert werden können. Der Nutzen des Systems ist dabei um so größer, je einheitlicher die Gewächshausanlage ist. Die Betriebsgröße spielt keine Rolle. Vielmehr sind Rechtwinkligkeit, Verschachtelung einzelner Gewächshausabteilungen und Einheitlichkeit des Bedachungsmaterials wichtige Kriterien.

Der implementierte und relativ starre Abfragemechanismus bei der Betriebsdateneingabe hat sich bei ungeübten Benutzern und Gewächshausanlagen mit geringer Komplexität bewährt. Er führt relativ sicher und schnell zur Eingabe der notwendigen Betriebsdaten. So kann ein Betrieb, bestehend aus mehreren gleichen Gewächshauschiffen in Blockbauweise, in kürzester Zeit im Rechner abgebildet werden. Im Minimalfall sind nur 13 Benutzereingaben erforderlich. Der Vorteil des Systems bei einfach strukturierten Betrieben hat sich bei komplexen Betrieben eher als Nachteil herausgestellt. Der starre Abfrage- und Eingabemechanismus führt zu einer unflexiblen Betriebsdateneingabe und erfordert zum Teil eine gewisse Übung. So waren zum Beispiel zur Modellierung des realen Betriebes Behre (s.u.) (9 Abteilungen, unterschiedliche Bedachungsmaterialien, Energieschirm, Kunstlicht) 158 Benutzerangaben notwendig. Für einen geübten Programmbenutzer muß mit einer Eingabezeit von ca. 1 Stunde gerechnet werden.

Bisherige Programmeinsätze haben gezeigt, daß bei noch komplexeren Betrieben das vorgegebene Punktraster zur Erstellung der Betriebsskizze zum Teil nicht ausreicht. Ebenso stellt die maximale Eingabe und Berechnung von 15 Gewächshaussegmenten besonders in 'gewachsenen' Betrieben oft eine Einschränkung dar. In beiden Fällen muß der Betrieb in zwei Betriebshälften aufgeteilt werden, die getrennt berechnet werden. So mußte beispielsweise bei der Eingabe des für die Validierung der Energiebedarfsberechnungen ausgewählten Betriebes Platz (s.u.) (13 Abteilungen, unterschiedliche Isolationsmaßnahmen, Belichtung, Energieschirme) eine Aufteilung in zwei Abschnitte erfolgen. Dadurch waren insgesamt 344 Benutzerangaben notwendig, bis der Betrieb vollständig vektorisiert und analysiert im Rechner vorlag. Die Eingabezeit betrug ca. 2 Stunden.

Zukünftige Entwicklungen sollten deswegen zusätzlich zu dem bisher implementierten Ansatz dem Programm benutzer die Möglichkeit geben, den abzubildenden Betrieb relativ frei zu entwickeln und zu gestalten.

Die Überprüfung der Einsatzfähigkeit und Gültigkeit des Energieverbrauchmodells erfolgte anhand von vier hinsichtlich ihrer Klimaführung sehr unterschiedlichen niedersächsischen Gartenbaubetrieben (Rieche, Koldingen; Platz, Sarstedt; Behre, Gehrden-Benthe; Wörmke, Bexhövede). Bei allen Betrieben erfolgt die Klimaregelung mit einem Klimaregelcomputer. Mit den von den Betriebsleitern verfügbaren Angaben über Gewächshäuser, Klimaregelung, Stilllegung von Betriebsteilen im Winter, Bedachungsmaterialien, Isolationsmaßnahmen, Energieschirm- und Kunstlichteinsatz wurden die angegebenen Gartenbaubetriebe im Rechner abgebildet. Die Anpassung weiterer Parameter (z.B. spezifische  $k'$ -Werte, Speicherwerte etc.) erfolgte nicht. Hier wurden die Standardwerte des Programms (siehe Kapitel 3.2.3.2) verwendet. Es wurden für alle verfügbaren Zeitabschnitte Energiebedarfsrechnungen mit realen Klimadaten (Daten des ITG-Klimacomputers) und programminternen Daten des Testreferenzjahres (TRY) durchgeführt. Abbildung 4.5-1 gibt den Vergleich zwischen tatsächlichem (aus verfügbaren Angaben der Betriebsleiter oder Gasabrechnungen resultierendem) Energieverbrauch und simuliertem Energieverbrauch wieder (Wö = Wörmke, Be = Behre, Pl = Platz, Ri = Rieche). Der Jahresbetriebswirkungsgrad der Heizungsanlage wurde in allen Fällen mit 0,9 angesetzt (vgl. EGGER 1990).

Die Berechnungen des Jahres 1990 beziehen sich auf den Zeitraum von Jahresanfang bis Jahresmitte. Die Berechnung des Betriebes Platz bezieht sich auf den Zeitraum vom 1.6. 1988 bis 31.5. 1989.

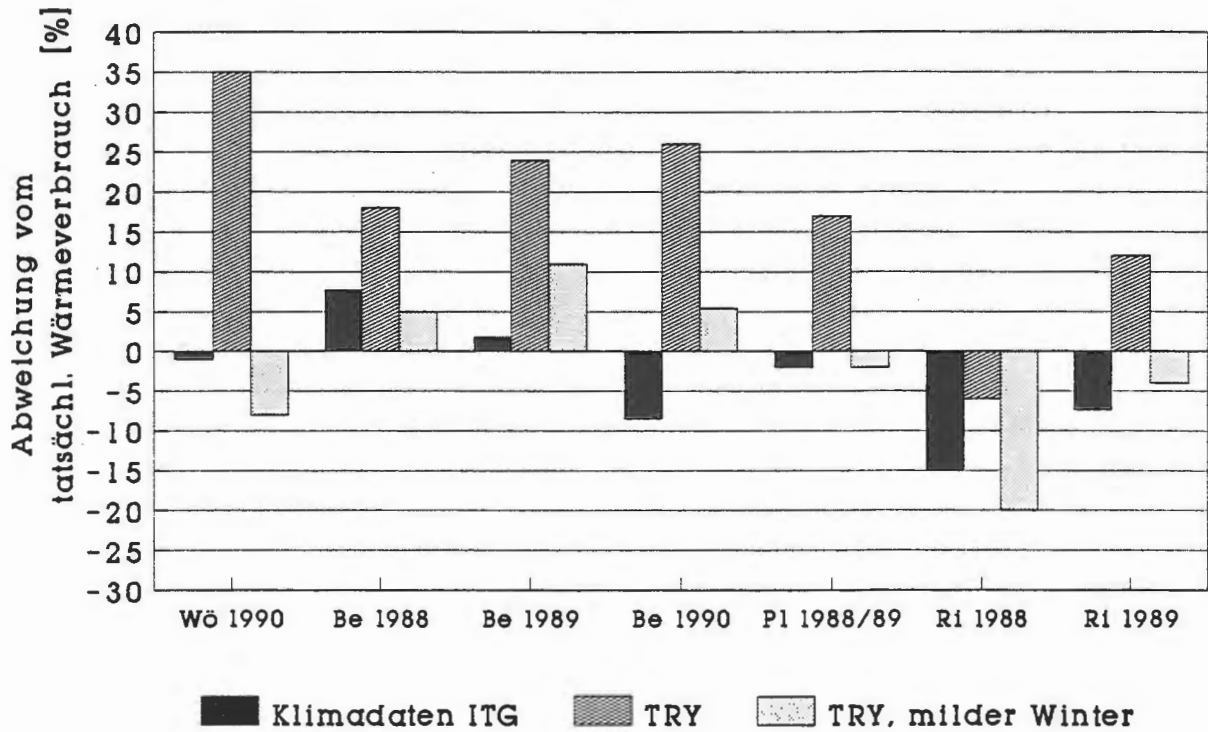


Abbildung 4.5-1: Vergleich gemessener und simulierter Energieverbräuche verschiedener niedersächsischer Gartenbaubetriebe.

Man erkennt, daß mit den verwendeten Berechnungsverfahren eine gute Übereinstimmung zwischen Modell und Realität erzielt werden konnte. Bis auf die Werte in einem Fall liegen die Abweichungen zwischen simuliertem und tatsächlichem Wert unter 10 %. Die höhere Abweichung bei der Berechnung des Betriebes Rieche 1988 ist evtl. auf eine unsichere Angabe des Betriebsleiters über die Stillungszeiten zweier Abteilungen zurückzuführen. Möglicherweise wurden aber auch die als sehr dicht eingestuftten Energieschirme überbewertet. Betrachtet man die Simulationsergebnisse unter Verwendung des Testreferenzjahres, so erkennt man, daß deutliche Abweichungen zwischen Modell und Realität vorliegen. Insbesondere bei Betrieben mit niedrigen Innentemperatursollwerten (Wörmke) macht sich der Unterschied sehr stark bemerkbar. Will man in diesen Fällen für ein bestimmtes Jahr genauere Energieverbrauchsangaben erhalten, müssen die Originalklimadaten oder angepaßte Standardklimadaten eingesetzt werden. In Abbildung 4.5-1 verdeutlicht dieses der Einsatz des TRY mit mildem Winter.

Die Überprüfung der Gültigkeit der Planungsprogramme läßt sich nur sehr schwer anhand der gegebenen Beispielbetriebe durchführen, da die mit dem Programm erzielbaren Ergebnisse in entscheidendem Maße von der subjektiven Beantwortung der vom Rechner an den Programmnutzer gestellten Fragen abhängen. So kann ein bestehender Betrieb bei entsprechenden Eingaben mit dem Planungsprogramm real nachgeplant werden. Da es das optimale Planungsergebnis aber nicht gibt und es keinerlei definierte Meßkriterien für eine gute Planung gibt (eindeutige technische Fehlplanungen sind hiermit nicht gemeint), kann das System in diesem Bereich nur mit laufenden Tests anwenderorientiert überprüft werden. Diese Tatsache hat weniger mit den in diesen Bereichen eingesetzten wissensbasierten Methoden zu tun, als vielmehr mit dem inhaltlichen Aufgabengebiet (das allerdings charakteristisch für den Expertensystemeinsatz ist).

Bisherige Einsätze im Programmbereich 'Energiebereitstellung' haben gezeigt, daß die Verknüpfung von technischen Planungsvorgaben, Investitionskosten und Energiepreisen wertvolle Hinweise für die Anlagenplanung liefert. Insbesondere durch die leichte Veränderung der Energieträgerpreise und -tarifbedingungen können aufschlußreiche 'Was-wäre-wenn-Berechnungen' bezogen auf die konkrete Planungssituation durchgeführt werden. Die große Breite der implementierten Energieträger wurde bei bisherigen Anwendungen allerdings nicht genutzt. Es wurde vielmehr deutlich, daß der Programmnutzer in der Regel eine feste Vorstellung von den zu realisierenden Energieversorgungskonzepten hat und das System als Rechenprogramm und weniger als Beratungssystem versteht. Das implementierte Energieträgerbewertungssystem in Hypertextform fand deswegen bisher so gut wie keine Beachtung (man erwartet offensichtlich von einem technischen Planungssystem in erster Linie, daß Ergebnisse errechnet werden). Ferner zeigten bisherige Anwendungen, daß eine genaue Spezifizierung des Anwenderkreises unerläßlich für die Einsatzfähigkeit des Systems, aber auch für die Gültigkeit des implementierten Wissens ist. So wurden die vom Rechner an den Benutzer gestellten Fragen von verschiedenen Anwendergruppen zum Teil unterschiedlich interpretiert. Zahlreich vorhandene Erklärungs- und Hilfstexte halfen hier auch nicht weiter, da sie erst gar nicht aufgerufen wurden. Diese Tatsache birgt die große Gefahr in sich, daß durch 'falsche' Benutzereingaben falsche Lösungen generiert werden (logisch jedoch völlig korrekt).

Ebenso wie bei der Energiebereitstellung ist auch die Validierung des Programmbereichs 'Heizenergieausbringung' nur schwer möglich, da es keine fest definierten Kriterien für eine optimale Planung gibt (selbst eine überdimensionierte Heizungsanlage sagt noch nichts über die zugrundegelegte Planungsqualität aus). Der Benutzer greift ferner durch die Eingabe beliebiger Heizungs-

systeme, deren Preise und deren Anordnung in der Datenbank massiv in den Auslegungsablauf ein. Das Programm ist somit an dieser Stelle als ein sehr offenes System zu betrachten, welches letztendlich nur durch laufende Anwendung validiert und verbessert werden kann. Die in der Standarddatenbank enthaltenen Heizungssystemdaten und die technischen Auslegungsberechnungen entsprechen dem Stand der Technik. Auf die Überprüfung der Richtigkeit wurde daher verzichtet.

Die Funktionalität der implementierten Auslegungsalgorithmen wurde anhand zahlreicher Testfälle überprüft. Besonders effektiv erwies sich hierbei der Einsatz der ökonomisch orientierten Planungsstrategie, da dem Programmbenutzer die ökonomische und technische Weite möglicher Planungen vor Augen geführt wird, er gleichzeitig aber auch konkrete Lösungsvorschläge bekommt. Die implementierten Heuristiken führten in fast allen Fällen dazu, daß die ökonomisch orientierte Auslegungsvariante am preiswertesten war, wobei die Auslegungen mit den vorgegebenen Standardwerten aus der Heizungssystemdatenbank durchgeführt wurden. Sie lassen sich somit nur sehr schwer verallgemeinern, weil die Effektivität der implementierten heuristischen Suche maßgeblich von den Unterschieden der Preise und der Wärmeabgaben der einzelnen Heizungssysteme abhängt.

#### **4.5.2 Anwendung**

Bisherige Berechnungsansätze und -verfahren zur Heizenergieberechnung von Gewächshausanlagen in der Bundesrepublik Deutschland bauen in der Regel auf den von VICKERMANN (1974) und DAMRATH (1980) veröffentlichten Tabellenwerten auf. Von einer Umsetzung dieser Berechnungsverfahren oder -werte in allgemein einsetzbare Software wird zwar berichtet (z.B. DOMKE 1984; RIEDEL 1982), sie scheint bisher allerdings relativ erfolglos gewesen zu sein oder aber hat sich zumindest in der Gartenbautechnikberatung nicht durchgesetzt. Ebenso kommen die aus der konventionellen Heiztechnik zur Verfügung gestellten Berechnungsprogramme nicht zum Einsatz (vgl. RATH 1990).

Das Energieverbrauchsmodell des Expertensystems HORTEX kann diese Einsatzlücke eventuell schließen, da durch die Energiebedarfsrechnung auf Stundenbasis mit dem TRY des deutschen Wetterdienstes, den implementierten Klimaregelstrategien, der Berücksichtigung von wichtigen Einrichtungen wie Energieschirm und Belichtung und dem durch Speicherfunktion und flächenbezogene Berechnung erweiterten  $k'$ -Modellansatz eine gute Praxisnähe bei angemessener Modellkomplexität erzielt werden konnte.

Die wichtigste Voraussetzung zum Einsatz des Modells in der Praxis ist allerdings eine adäquate und auf den Benutzer abgestimmte Eingabe- und Ausgabeschnittstelle. Durch das beschriebene Front-End-System und durch die graphische Benutzerschnittstelle ist auch ein relativ unbedarfter Programmbenutzer in der Lage, das Programm und seine vielfältigen Möglichkeiten zu nutzen und den gewünschten Betrieb zu modellieren. Zur Verbesserung sollte jedoch noch die in Kapitel 4.5.1 angesprochene Erweiterung in das System aufgenommen werden. Einem erfolgreichen Einsatz steht dann aufgrund des als relativ gering einzuschätzenden Wartungsaufwandes und durch die vielseitigen Einsatzmöglichkeiten nichts mehr im Wege. Probleme wie sie GOLLWITZER (1988) beim praktischen Einsatz von computergestützten Planungshilfsmitteln für den Bereich der Anbauplanung beschreibt, können jedoch auch hier auftreten. Sie dürften jedoch aufgrund der inhaltlich technischen Ausrichtung des Systems eine geringere Bedeutung haben.

Das Einsatzspektrum des Wärmeberechnungssystems beschränkt sich nicht nur auf die gartenbautechnische Beratung, sondern auch Industrie, Ausbildung und die Praxis selbst zeigen großes Interesse an der Nutzung des Programms. Darüberhinaus ist mit der Schaffung einer standardisierten DSIF-Schnittstelle (DICOTU 1989) ein erster Schritt verwirklicht worden, der eventuell einen direkten Einsatz im Bereich der Kostenrechnung und Kulturplanung ermöglicht. Zukünftige Arbeiten sollten hier Klärung bringen.

Im Bereich der Heizungsanlagenplanung und -auslegung ist bisher keine gartenbauspezifische Software vorhanden. Die vom Kuratorium für Technik und Bauwesen in der Landwirtschaft e.V. vorgehaltene Datenbank für Investitionskosten im Gartenbau steht nur auf einem Großrechner zur Verfügung und dient in erster Linie der Erstellung des Tabellenwerkes "Kalkulationsunterlagen im Gartenbau" (KTBL 1990). Eine betriebsspezifische Anwendung ist aufgrund der rein datenbankorientierten Wissensdarstellung nicht möglich. Spezielle Planungssoftware, wie sie in Abbildung 3.1-2 dargestellt ist, wird von der Gewächshausindustrie und den Heizungsanlagenbauern zum Teil eingesetzt. Sie unterstützt jedoch jeweils nur die letzten Auslegungsschritte im Planungsablauf. Eine konkrete Abstimmung der technischen Planung auf gartenbauliche Belange erfolgt hiermit nicht.

Das Expertensystem HORTEX kann mit seinen Planungsmodulen diese Planungslücke verringern, indem es Entscheidungshilfen gibt. Die Kopplung von betriebsspezifischen Berechnungen mit technischen und ökonomischen Kennzahlen, Standardwerten und Faustregeln bildet hierfür den geeigneten Planungsrahmen. Durch die im Bereich der Investitionskosten und der herstellereinspezifischen Heizungsanlagen implementierte Systemoffenheit ist das Programm an spezielle Situationen anpaßbar. Inwieweit diese Anpassungsmöglichkeit im realen Programmeinsatz erfolgen kann und erfolgt, bleibt abzuwarten. Auf jeden



Fall setzt sie eine gewisse fach- und systemspezifische Kenntnis voraus. Bisherige Programmanwendungen im Bereich der Beratung haben gezeigt, daß der für die Modellierung der Anlagenpreise gewählte Modellrahmen den Programm-  
benutzern zum Teil zu grob erschien. Es entstand die Schwierigkeit, die Eckdaten der Preisdatenbank mit Hilfe von konkret vorliegenden Angeboten für Anlagenteile zu verifizieren bzw. anzupassen.

Der Einsatz von Computerprogrammen eignet sich besonders im Bereich kurzfristiger Planungs- und Entscheidungsprobleme aufgrund der evtl. zu automatisierenden Datenbeschaffung, der Softwareamortisation und der Programmkenntnis des Anwenders zur Unterstützung von Entscheidungsprozessen (vgl. BERG 1985). Da alle heizungstechnischen Planungsaufgaben von Natur aus relativ selten in einem Gartenbaubetrieb anfallen, ist ein direkter Einsatz der Planungssoftware in der Praxis nicht zu erwarten. Das Anwendungspotential der unter 3.2.3.3 und 3.2.3.4 beschriebenen Systeme wird wahrscheinlich auf die Technikberatung und die mit der Heizungsanlagenplanung vertrauten Fachleute beschränkt bleiben. Der Einsatz der Systeme wird dabei maßgeblich von der (ökonomisch) realisierbaren Daten- und Programmpflege abhängen. Obwohl das implementierte Wissen auf strukturierten Modellansätzen aufbaut, ist eine Wartung unerlässlich. Eine Programmpflege auf Benutzerbasis oder durch selbstlernende Algorithmen, wie sie früher KI-Firmen propagierten, ist ausgeschlossen. Die starke Diskrepanz zwischen der Zahl der Anwender, deren finanziellen Potential und dem durch das Aufgabengebiet anfallenden Wartungsaufwand wird den Praxiseinsatz der Planungsprogramme erschweren. Diese Tatsache resultiert allein aus den beschriebenen pragmatischen und ökonomischen Sachzwängen. Die wissensbasierte Softwareentwicklung unterscheidet sich in diesem Punkt nicht von der konventionellen Programmierung.

#### 4.6 Übertragbarkeit der Konzeption

Das beschriebene Expertensystem HORTEX kann strukturell und inhaltlich in vier verschiedene Konzepte unterteilt werden.

1. Ein Front-End-System zur Eingabe betriebsspezifischer Planungsparameter und zur Interpretation und Erklärung von Lösungen.
2. Ein arithmetisches Modell zur Berechnung planungsspezifischer Basisdaten.
3. Ein Produktionsregelmodell zur Auswahl von Problemlösungsvarianten.
4. Ein Frame-Modell zur Auslegung und Spezifikation von Planungsvarianten.

Abbildung 4.6-1 zeigt die Übertragung der entsprechenden Konzeptionen auf den Bereich der Bewässerungsanlagenplanung. Ähnliche Ansätze lassen sich auch noch für weitere gartenbautechnische Planungsbereiche wie Kunstlichtein-satz oder Gewächshausplanung erstellen. Das hybride System HORTEX und die darin eingesetzten Methoden bekommen somit einen allgemeingültigen Charakter, obwohl jede Aufgabe natürlich ihre spezifischen Lösungsansätze benötigt. Ob und wie stark die beschriebenen Wissensdarstellungsmethoden zur Geltung kommen, hängt entscheidend von der inhaltlichen Struktur der Aufgabenstel-lung und den eingesetzten Implementierungsinstrumentarien ab.

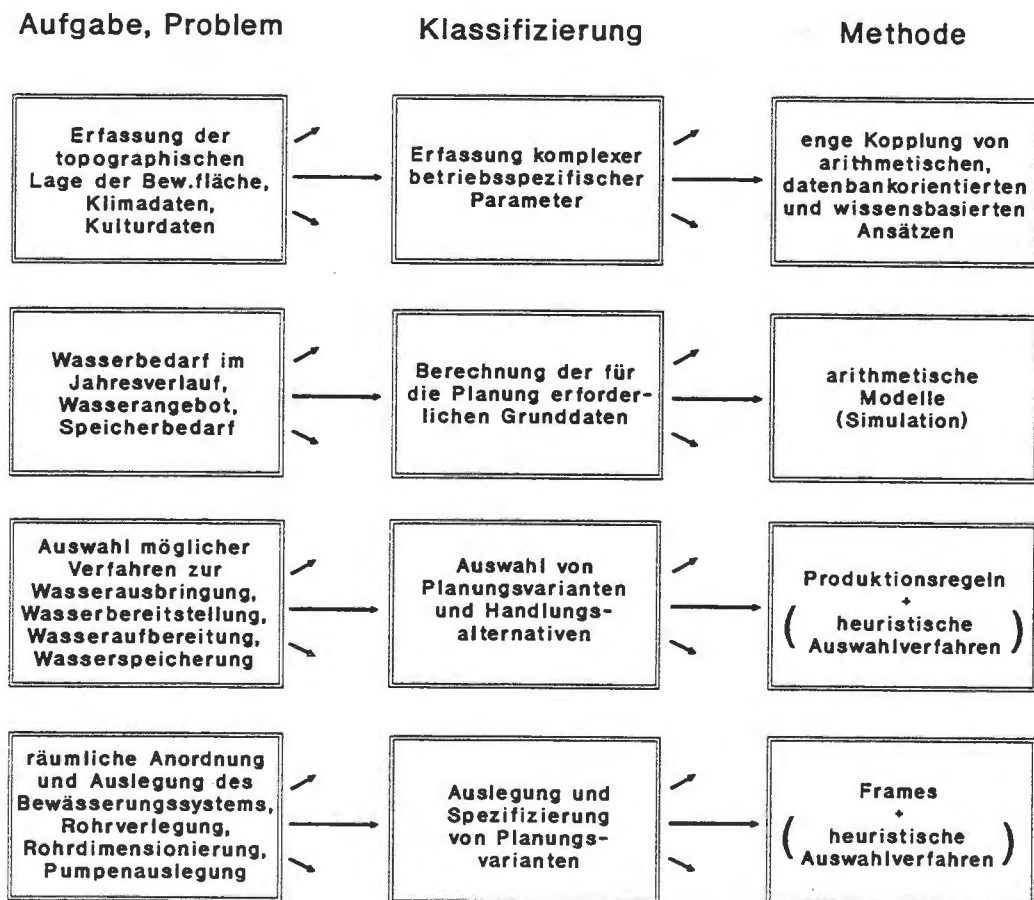


Abbildung 4.6-1: Übertragung der methodischen und inhaltlichen Konzeption des Expertensystems HORTEX auf den Bereich der Bewässerungsanlagenplanung

Auf jeden Fall sollten Ableitungs- oder Objektbäume fester Bestandteil praxisorientierter und wissenschaftlich fundierter EDV-Modelle werden, da nur so nicht nur die Diskrepanz zwischen Modell und Wirklichkeit, sondern auch die Diskrepanz zwischen Modell und Modellanwender verringert werden kann. Das setzt allerdings voraus, daß auch nichtnumerische Darstellungen als vollwertige Modellierungsformalismen akzeptiert und anerkannt werden und nicht nur als 'Lückenbüßer' für fehlendes Wissen dienen.

#### **4.7 Zukünftige Arbeiten**

Zukünftige Arbeiten sollten die vorgestellten Modellierungsansätze auch auf andere Arbeitsbereiche übertragen. Dabei sollte dem Frame-Ansatz noch stärkere Bedeutung beigemessen werden, da er zumindest von seinen theoretischen Ansprüchen her die optimale Wissensrepräsentationsform für wissensbasierte Systeme in der Gartenbautechnik zu sein scheint. Ebenso sollte der Einsatz des relationalen Datenbankkonzeptes in die weiteren Betrachtungen mit aufgenommen werden. Durch die Verknüpfung einer großen Anzahl von Massendaten mit Hilfe effektiver und kostengünstiger Software könnte eine wesentliche Verbesserung des Wartungsproblems erzielt werden.

Inwieweit die Ausweitung zweiwertiger Und-Oder-Graphen auf Mehrwertigkeit eine Verbesserung der Modellierungsgüte erbringt, bleibt abzuwarten. Der gezielte Einsatz der Fuzzy-Logik in inhaltlich validierbaren und abgegrenzten Systemen sollte ebenso getestet werden, wie die Einbeziehung statistischer Lösungsansätze in konventionellen Simulationsberechnungen.

Weitere Arbeiten sollten den Einsatz multimedialer Systemkomponenten vorsehen. Er wird allerdings maßgeblich von den von der EDV-Branche zur Verfügung gestellten Software- und Hardwaremöglichkeiten abhängen.

Die inhaltliche Ausweitung, Verbesserung und Vervollständigung des Systems ist anzustreben. Zukünftige Maßnahmen in dieser Richtung werden jedoch in erster Linie vom Einsatz des Systems abhängen. Eine Erweiterung des Systems um Blockheizkraftwerk- und Kunstlichtplanung müßte realisierbar sein.

Die Thematik der Bewässerungsanlagenplanung, die weltweit sehr starke Bedeutung findet, sollte auch im deutschsprachigen Raum weiter verfolgt werden, da im Bereich der Bewässerung viel Fachwissen zur Verfügung steht und auf der anderen Seite entscheidungsunterstützende Systeme aus aktuellen ökologischen und ökonomischen Zwängen benötigt werden.

Denn nur durch konkrete Entwicklungen und Anwendungen, orientiert an praktischen Problemen, kann dem Ziel näher gekommen werden, 'intelligente' und nützliche Software zu erstellen, die weniger mit der erschreckenden Vorstellung des ersetzbaren Menschen zu tun hat, als vielmehr eine sinnvolle Hilfestellung bei gartenbautechnischen Aufgaben darstellt.

## 5. Zusammenfassung

Die vorliegende Arbeit beschreibt den Einsatz von wissensbasierten Systemen (Expertensystemen) zur Modellierung und Darstellung von Fachwissen im Bereich der Gartenbautechnik. Im einzelnen wurde untersucht, inwieweit die Methoden zur Entwicklung von Expertensystemen im Bereich der Gartenbautechnik eingesetzt werden können, inwieweit sie angepaßt werden müssen oder inwieweit sie eine Erweiterung konventioneller Modellierung ermöglichen. Wissensbasierte Systeme werden dabei als Computerprogramme verstanden, bei denen eine strikte Trennung zwischen einer Wissensbasis und einem Ableitungsmechanismus existiert.

Expertensysteme bilden Wissen in Form von Und-Oder-Graphen ab, bei denen die Knoten Objekte, Oberklassen etc. eines Objektbaumes oder Regeln und Fakten eines Ableitungsbaumes darstellen. Die Kanten (die Verbindung der Knoten untereinander) repräsentieren die hierarchischen Verbindungen zwischen Objekten oder die Bedingungen eines Regelsystems. Im einzelnen werden verschiedene Formen der Repräsentation von Und-Oder-Graphen eingesetzt. Die wichtigsten sind: Produktionsregeln der Art 'Wenn....dann...', Frames, bei denen Wissen durch eine Ansammlung einheitlicher Daten- und Berechnungsstrukturen (Objekte) repräsentiert wird, sowie verschiedene Logikkalküle. Besonders bedeutend und für viele Anwendungen grundlegend sind die Aussagenlogik und die Prädikatenlogik, die eingeschränkt mit der Programmiersprache Prolog realisiert werden können. Die Auswertung der so definierten Graphen erfolgt mit prozeduralen Algorithmen, wobei verschiedene Such- und Schlußfolgerungsverfahren eingesetzt werden. Wichtige Auswertemechanismen sind verschiedene Verkettungsverfahren, Modus Ponens, Backtracking und Matching. Eine Erweiterung des Und-Oder-Graphen-Konzeptes auf Mehrwertigkeit kann durch den Einsatz von statistischen Kennzahlen, Sicherheits-Faktoren oder durch die Definition von unscharfen Mengen (fuzzy sets) erfolgen.

Neben der wissensbasierten Modellierung von Fachwissen können konventionelle EDV-Konzepte zur Darstellung von Fachwissen eingesetzt werden. Die beiden wichtigsten sind arithmetische (Un)Gleichungen und verschiedene Datenbankkonzeptionen, heute meistens in Form von relationalen Datenbanken. Neuere Systeme verknüpfen verschiedene Ansätze zu einem effektiven Gesamtkonzept.

Das hybride System HORTEX stellt den Versuch dar, wissensbasierte Methoden im Bereich der Gartenbautechnik anzuwenden, an die situationsspezifischen Besonderheiten anzupassen und mit konventionellen Wissensrepräsentationsformen zu koppeln. Als thematischer inhaltlicher Schwerpunkt wurde die Heizenergieversorgung von Gewächshausanlagen ausgewählt. Es erfolgte die Aufteilung in fünf Module.

Modul 1: Ein Front-End-System zur Eingabe von betriebsspezifischen Daten. Wesentliches Ziel dieses Moduls ist es, auf möglichst effektive Art und Weise ein räumliches Modell des Gartenbaubetriebes im Rechner abzubilden.

Modul 2: Ein konventionelles Simulations- und Berechnungsprogramm, welches den Energieverbrauch und -bedarf von Gewächshausbetrieben nach einem modifizierten k'-Modell berechnet.

Modul 3: Ein wissensbasiertes System, basierend auf Produktionsregeln zur Planung und zur Bewertung von Energiebereitstellungskonzeptionen. Im einzelnen erfolgt die Auswahl möglicher Energieträger, deren Kombination und die Zusammenstellung möglicher Heizenergieversorgungsanlagen. Im Anschluß daran werden die gefundenen Planungsvarianten ökonomisch bewertet und Zusatzinformationen gegeben.

Modul 4: Ein wissensbasiertes System zur Planung der Heizenergieausbringung in Gewächshäusern. Aufbauend auf einem Frame-Konzept werden Heizungssysteme im Gewächshaus situationsspezifisch ausgelegt. Dabei werden mehrere Planungsvarianten erstellt und dem Benutzer als Problemlösung vorgeschlagen.

Modul 5: Eine Datenbankkonzeption, die die für den Benutzer wichtigen Daten editierbar bereitstellt.

Die Realisierung konventioneller Programmmodule erfolgte in Pascal. Die wissensbasierten Methoden wurden mit Hilfe der Programmiersprache Prolog implementiert.

Zur besseren Handhabung und zur Beschreibung einer einheitlichen Wissensstruktur wurde ein Produktionsregelformalismus und ein Framekonzept entwickelt.

Der Produktionsregelmechanismus besteht aus einem vierstelligen Prologprädikat, welches jeweils zwei Knoten und die verbindende Kante darstellt. Zur besseren Verarbeitung werden sechs unterschiedliche Kantentypen vorgegeben. Bei Abarbeitung der Wissensbasis werden alle gefundenen Lösungen in einer Liste gespeichert, welche am Ende mit Textelementen zu einem Lösungstext verknüpft wird. Es erfolgt ein vollständiges Durchsuchen der Wissensbasis nach Lösungen mittels Backtracking.

Der Frame-Formalismus wurde mit einem fünfstelligen Prologprädikat realisiert. Die Variablen kennzeichnen das Objekt, den Slot, die Art des Slots, die Liste aller Slotinhalte und eine Kennzeichnung, ob der Slot nur internen Rechenvorgängen dient oder nicht. Die prozeduralen Berechnungen werden mit einem dreistelligen Prologprädikat eingebunden. Die Verbindung zum Objekt erfolgt ausschließlich über *wenn\_benoetigt*-Slots.

Vererbung geschieht durch die Definition bestimmter Vater/Sohn-Slots innerhalb der Frames. Die Auswertung des so beschriebenen Objektbaumes erfolgt durch einen Mechanismus, der den Objektbaum vollständig auswertet oder den Inhalt eines bestimmten Objekt-Slots sucht.

Zur Nutzung des Framesystems im Bereich der Heizenergieausbringung (Heizungsauslegung) wurde ein Auslegungsverfahren implementiert, welches zur Berechnung der Heizungssysteme jeweils auf das Framekonzept zugreift. Um mehrere Planungsstrategien verfolgen zu können, wurden vier heuristische Suchstrategien in das Auslegungsverfahren integriert.

Die Entwicklung des Systems HORTEX hat gezeigt, daß im Bereich der Gartenbautechnik eine alleinige Modellierung mit wissensbasierten Ansätzen nicht sinnvoll ist. Expertensystemtechnologien sollten konventionelle, aus wissenschaftlichen Versuchen resultierende arithmetische Modelle unterstützen. Ein großer Vorteil kann dabei in der Trennung von Wissensbasis und Inferenzmechanismus gesehen werden, weil sie eine leichte Änderbarkeit des Wissensmodells ermöglicht. Zwingend notwendig hierbei ist allerdings eine Modularisierung der Wissensbasis. Die Darstellung von unzureichend modelliertem oder fehlendem Wissen mit Expertensystemtechnologien ist nicht möglich. Vielmehr kann die Entwicklung zweiwertiger Und-Oder-Graphen als eine tiefe Modellierung eines Gegenstandsbereiches betrachtet werden, besonders dann, wenn Objekte und ihre hierarchischen Beziehungen zueinander definiert werden.

Aus den im Bereich des Expertensystems HORTEX durchgeführten Entwicklungen wurde deutlich, daß eine aus der Wissensbasis generierte Erklärungskomponente wegen des fehlenden Informationsgehaltes nur bedingt zu realisieren ist. Ebenso ist der Einsatz von Unschärferepräsentationsformen, wie Certainty-Factors, kritisch zu betrachten, da er eventuell dazu führt, daß Modellierungsdefizite noch verstärkt werden. Im Expertensystem HORTEX wurde daher ein statisches Hilfs- und Erklärungssystem und eine direkte Darstellung von unscharfen Bewertungsfaktoren eingesetzt. Beides baut auf einem Hypertextansatz auf.

Die Wissensrepräsentation in Form von Hornklauseln (Prolog) wird als sehr positiv bewertet, da Backtracking, Unifikation und Listeneinsatz grundlegende Mechanismen für die Bearbeitung und Auswertung von Und-Oder-Graphen darstellen. Das beschriebene Produktionsregelsystem kann in den Fällen effektiv eingesetzt werden, in denen nicht numerische oder funktionale Beziehungen, sondern logische Abhängigkeiten einzelner Modellobjekte bei der Modellbildung im Vordergrund stehen. Handelt es sich bei den Wissensstrukturen schwerpunktmäßig um Objekte mit bestimmten Eigenschaften, sollte das Framekonzept Anwendung finden. Bei sehr stark auf arithmetischen Modellen aufbauendem Wissen sollte auf einen Expertensystemansatz ganz verzichtet werden. Konventionelle Ansätze haben in diesen Bereichen erhebliche Vorteile.

Der Einsatz von Expertensystemen im Bereich der Gartenbautechnik ist besonders aufgrund der entstehenden Wartungs- und Entwicklungskosten schwierig. Am ehesten können Systeme in den Bereichen ökonomisch eingesetzt werden, in denen eine häufige und breit gefächerte Anwendung erfolgt. Das beschriebene System wird deshalb in dem Bereich der Heizanlagenplanung Schwierigkeiten haben, über den Prototypstatus hinauszukommen. Im Bereich der Energiebedarfsrechnung scheint aufgrund der existierenden Anwendungsbreite eine Einsatzmöglichkeit gegeben zu sein. Außerdem können die in der Arbeit entwickelten und untersuchten methodischen Ansätze als Grundlage für weitere Entwicklungen dienen.



## 6 Literatur

- Altenkrüger D.E. (1987): Wissensdarstellung für Expertensysteme. Mannheim: BI-Wissenschaftsverlag.
- Anderson D.; Graves L.; Reinert W.; Kreider J.F.; Dow J.; Wubbena H. (1989): A Quasi-Real-Time Expert System for Commercial Building HVAC Diagnostics. ASHRAE Transactions, 95 (2), S. 954-960.
- Appelrath H.J. (1985): Von Datenbanken zu Expertensystemen. Informatik-Fachberichte 102. Berlin: Springer.
- Baer K.; Diemer R. (1990): Wissensbasiertes System zur Vorplanung von Heizungsanlagen. HLH, 41 (8), S. 659-663.
- Barr A.; Feigenbaum E.A. (1981): The handbook of artificial intelligence. Vol. 1. Los Altos, Ca: William Kaufmann.
- Bazlen E. (1985): Modelluntersuchungen zu altersabhängigen Klimareaktionen als Basis für die Simulation des Wachstums am Beispiel von Radies (*Raphanus sativus var. sativus*). Dissertation Universität Hannover, Fachbereich Gartenbau.
- Beck H.W.; Jones J.W. (1989): Simulation and Artificial Intelligence. In: Barrett J.R.; Jones D.D. (Hrsg.): Knowledge Engineering in Agriculture. St. Joseph, Mi: American Society of Agricultural Engineers.
- Belcher H.W.; Merva G.E. (1988): Expert System to evaluate subirrigation site. ASAE Paper No. 88-7535. St. Joseph, Mi: American Society of Agricultural Engineers.
- Berg E. (1985): Mikrocomputereinsatz im landwirtschaftlichen Betrieb. Ber. Ldw., 63, S. 376-389.
- Bergmann B.; Noll H. (1977): Mathematische Logik mit Informatik-Anwendungen. Berlin: Springer.
- Berlin F.U. (1991): Expertensysteme ersetzen noch keine Experten. VDI nachrichten, Nr. 14, 5.April 1991, S. 16.
- Bielig-Schulz G.; Schulz C. (1987): 3D-Graphik in PASCAL. Stuttgart: Teubner.

- Billerbeck J.D.; Bönsch R.; Zimmermann H.J. (1991): Unschärfe Logik stößt auf großes Interesse. VDI nachrichten, Nr. 20, 17. Mai 1991, S. 14.
- Blümel K.; Hollan E.; Kähler M.; Peter R. (1986): Entwicklung von Testreferenzjahren (TRY) für Klimaregionen der Bundesrepublik Deutschland. Forschungsbericht T 86-051. Technologische Forschung und Entwicklung - Nichtnukleare Energietechnik. Bundesminist. für Forschung und Technologie.
- Böck J. (1991): Procedural, Functional, Object-Oriented and Logic-Oriented Programming - Overview and Evaluation - . In: Hälker M.; Jaeschke A. (Hrsg.): Informatik für den Umweltschutz. 6. Symposium München, Dezember 1991. Informatik-Fachberichte 296. Berlin: Springer.
- Boose J.H. (1988): A survey of knowledge acquisition techniques and tools. 3rd AAAI-Sponsored Knowledge Acquisition for Knowledge-Based Systems Workshop. Banff, Canada, November 1988, S. 3.1-3.22.
- Bosch R. (1988): Umfrage: Laufende KI-Projekte in der Industrie. KI, 2 (4), S. 33-45.
- Bradley T.; Sowell R.S.; Sneed R.E. (1988): An expert system for irrigation planning and design. ASAE Paper No. 88-5021. St. Joseph, Mi: American Society of Agricultural Engineers.
- Brewka G. (1986): Wissensrepräsentation und Inferenztechniken. In: Krallmann H. (Hrsg.): Expertensysteme im Unternehmen. Berlin: Ernst Schmidt.
- Bronstein I.N.; Semendjajew K.A. (1987): Taschenbuch der Mathematik. Frankfurt: Harri Deutsch.
- Buchanan B.G.; Duda R.O. (1983): Principles of rule-based expert systems. In: Yovits M.C. (Hrsg.): Advances in Computers, Vol. 22, S. 164-210. New York: Academic Press.
- Bundy A. (1986): Catalogue of Artificial Intelligence Tools. Berlin: Springer.
- Bylda C. (1990): Die Wahl des Grund- und Spitzenlastverhältnisses bei der Auslegung bivalenter Gewächshausheizungsanlagen unter Berücksichtigung verschiedener Energieträger. Diplomarbeit Universität Hannover, Institut für Technik in Gartenbau und Landwirtschaft, unveröffentlicht.
- Camejo P.J.; Hittle D.C. (1989): An Expert System for the Design of Heating, Ventilating, and Air-Conditioning Systems. ASHRAE Transactions, 95 (1), S. 379-386.

- Case M.P.; McConkey I.; McGraw K.; Lu S.C-Y. (1990): Multiple Cooperating Knowledge Sources for the Design of Building Energy Systems. ASHRAE Transactions, 96 (2), S. 490-500.
- Ceri S.; Gottlob G.; Wiederhold G. (1987): Interfacing Relational Databases and Prolog Efficiently. In: Kerschberg L. (Hrsg.): Expert Database Systems, S. 207-223. Menlo Park, Ca: Benjamin/Cummings.
- Chandrasekaran B.; Tanner M.C.; Josephson J.R. (1988): Explanation: The Role of Control Strategies and Deep Models. In: Hendler J.A. (Hrsg.): Expert Systems: The User Interface. Norwood, NJ: Ablex.
- Charlier M. (1989): Informationsmanagement: Versicherungen denken vor. ComputerMagazin, 18 (6-7), S. 18-22.
- Charlier M. (1990): Graphische Oberflächen - Ergonomie und mehr. ComputerMagazin, 19 (8), S. 36-40.
- Clarke N.D.; Miles G.E.; Barrett J.R.; Christmas E.P.; Doster D.H. (1990): Coupled Expert System and Simulation to Assess Crambe Production. Applied Engineering in Agriculture, 6 (5), S. 661-666.
- Claus V.; Schwill A. (1989): Programmiersprachenkonzepte orientiert an PASCAL (Teil 2). Studienmaterial 1611-7-04-S1. Hagen: Fernuniversität - Gesamthochschule - Hagen, Fachbereich Mathematik und Informatik.
- Clocksinn W.F.; Mellish C.S. (1981): Programming in Prolog. Berlin: Springer.
- Coy W.; Bonsiepen L. (1989): Erfahrung und Berechnung: Kritik der Expertensystemtechnik. Informatik-Fachberichte 229. Berlin: Springer.
- Damrath J. (1980): Tabellen zur Heizenergieermittlung von Gewächshäusern. Gartenbautechnische Informationen, Heft 8. Hannover: Institut für Technik in Gartenbau und Landwirtschaft, Universität Hannover.
- Deutsche Landwirtschafts-Gesellschaft (1988): Internationaler DLG-Computerkongreß. Wissensbasierte Systeme in der Landwirtschaft - Auf dem Weg zum Anwender. Frankfurt: Deutsche Landwirtschafts-Gesellschaft.
- DICOTU (1989): Dicotu Standaard Interface Formaat (Versie 1.0). DICOTU, Wassenaarseweg 80, 2596 CZ Den Haag.
- Diederich J. (1987): Wissensakquisition. Arbeitspapiere der GMD 245. Sankt Augustin: Gesellschaft für Mathematik und Datenverarbeitung.

- Diemer W.R. (1989): Relationale Datenbanken kurz und bündig: Sicherheit, Integrität und Unabhängigkeit in der Datenverwaltung. Würzburg: Vogel.
- Dilger W. (1991): Expertensysteme - eine Einführung. ComputerMagazin, 19 (1-2), S. 30-36.
- DIN 2993 (1977): Stahlfittings mit Gewinde. Rohrverschraubungen. Berlin: Beuth.
- DIN 4701 (1959): Regeln für die Berechnung des Wärmebedarfs von Gebäuden. Berlin: Beuth.
- DIN 4701 (1986): Regeln für die Berechnung des Wärmebedarfs von Gebäuden. Berlin: Beuth.
- Domke O. (1984): Umstellungsberechnungen macht der Computer leicht. Gb+Gw, 84 (19), S. 454-457.
- Duda R.O.; Hart P.E.; Nilsson N.J. (1976): Subjective Bayesian methods for rule-based inference systems. In: Proceedings of the 45. AFIPS National Computer Conference, S. 1075-1082.
- DWD (1991): Schriftliche Mitteilung Deutscher Wetterdienst, Zentralamt Offenbach. 6050 Offenbach am Main.
- Dzaebel F. (1990): Neuronale Netze. In: BremTec '90 - Seminar. Der Einsatz von Expertensystemen in der Industrie, 28. Juni 1990. Bremen: Universität Bremen, PC-Labor.
- Egger R. (1990): Reduzierung von Schadstoffemissionen durch moderne Kesselkonstruktionen. HLH, 41 (12), S. 1013-1032.
- von Elsner B. (1982): Das Kleinklima und der Wärmeverbrauch von geschlossenen Gewächshäusern. Dissertation Universität Hannover, Fakultät für Maschinenwesen.
- von Elsner B. (1991): Energiespargewächshaus 'System Fischer'. Abschlußbericht der wissenschaftlichen Betreuung zum Forschungs- und Entwicklungsvorhaben für den Umweltschutz im Agrarbereich (FuE-Vorhaben 83 UM 22). Hannover: Institut für Technik in Gartenbau und Landwirtschaft, Universität Hannover.
- Enand R.; Kahn G.S. (1988): A Methodology for Validating Large Knowledge Bases. 3rd AAAI-Sponsored Knowledge Acquisition for Knowledge-Based Systems Workshop. Banff, Canada, November 1988, S. 5.1-5.11.

- Engesser H. (Hrsg.) (1988): Duden "Informatik". Mannheim: Dudenverlag.
- Fairchild K.M.; Poltrock S.E.; Furnas G.W. (1988): SemNet: Three-Dimensional Graphic Representations of Large Knowledge Bases. In: Guindon R. (Hrsg.): Cognitive science and its applications for human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum.
- Fischer D.; Menges R. (1989): Wissensbasierte Materialauswahl. In: Expertensysteme in Produktion und Engineering. IAO-Forum 20. April 1989. Stuttgart: Fraunhofer-Institut für Arbeitswirtschaft und Organisation.
- Flitman A.M.; Hurrion R.D. (1987): Linking discrete-event simulation models with expert systems. Journal of the Operational Research Society, 38 (8), S. 723-733.
- Forgy C.L.; Shepard S.J. (1987): RETE: A Fast Match Algorithm. AI Expert, January 1987, S. 34-40.
- Fox M.S. (1990): AI and Expert System. Myths, Legends, and Facts. IEEE Expert, February 1990, S. 8-20.
- Freeland R.S.; Howard K.D. (1990): TIP - An Expert System for Ground Sprayer Nozzle Tip Selection. Applied Engineering in Agriculture, 6 (6), S. 697-700.
- Freksa C. (1988): Cognitive Science - eine Standortbestimmung. In: Heyer G.; Krems J.; Görz G. (Hrsg.): Wissensarten und ihre Darstellung. Berlin: Springer.
- Furbach U. (1988): Wissensrepräsentation und Programmiersprachen. In: Heyer G.; Krems J.; Görz G. (Hrsg.): Wissensarten und ihre Darstellung. Berlin: Springer.
- Futo I.; Gergely T. (1990): Artificial intelligence in simulation. New York: Ellis Horwood.
- Gaultney L.; Harlow S.; Ooms W. (1987): An expert system for diagnosing problems in hydraulic systems. ASAE Paper No. 87-1025. St. Joseph, Mi: American Society of Agricultural Engineers.
- Gerster H.D. (1972): Aussagenlogik, Mengen, Relationen. Freiburg: Herder.
- Geske U. (1988): Programmieren mit PROLOG. München: Oldenbourg.

- Giacomelli G.A.; van Weel P.; van der Schilden M. (1987): Expert systems development for greenhouse potted plant production. ASAE Paper No. 87-5012. St. Joseph, Mi: American Society of Agricultural Engineers.
- Gibbert R. (1991): UMEX ein wissensbasiertes System zur umweltbereichsübergreifenden Plausibilitätsprüfung. In: Hälker M., Jaeschke A. (Hrsg.): Informatik für den Umweltschutz. 6. Symposium München, Dezember 1991. Informatik-Fachberichte 296. Berlin: Springer.
- Glück B. (1987): Zustands- und Stoffwerte. Berlin: Marhold.
- Göhler A. (1990): - Expertensystem Gurke - Düngeplanung und Diagnose von Ernährungsstörungen. Dissertation Universität Hannover, Fachbereich Gartenbau.
- Göhler A.; Rath T.; Schmidt-Paulsen T. (1989): Chancen und Risiken von Expertensystemen. Taspo Magazin, 16 (11), S. 15-16.
- Gollwitzer S. (1988): Probleme beim praktischen Einsatz von computergestützten Planungshilfsmitteln in Betrieben und Beratung. In: AID (Hrsg.): Bericht über das 27. Betriebswirtschaftliche Seminar für Gartenbauberater vom 26. bis 30. September 1988 in Grünberg. Bonn: Auswertungs- und Informationsdienst für Ernährung, Landwirtschaft und Forsten (AID).
- Goltz H.J.; Herre H. (1990): Grundlagen der logischen Programmierung. Berlin: Akademie-Verlag.
- Graham I.; Jones P.L. (1988): Expert systems: knowledge, uncertainty and decision. London: Chapman and Hall.
- Granier J. (1991): Artificial Intelligence Techniques for Irrigation System Design. 26 th EAAE Seminar "Economics and Artificial Intelligence in Agriculture". Grignon, 10.-12. September 1991.
- Greenwell M. (1988): Knowledge engineering for expert systems. Chichester: Ellis Horwood.
- Habel C.; Rollinger C.R. (1985): Lernen und Wissensakquisition. In: Habel C. (Hrsg.): Repräsentation von Wissen und natürlichsprachliche Systeme. Informatik-Fachberichte 93. Berlin: Springer.
- Haberl J.S.; Claridge D.E. (1987): An Expert System for Building Energy Consumption Analysis: Prototype Results. ASHRAE Transactions, 93 (1), S. 979-998.

- Hälker M.; Bubeck F.; Stahl V. (1986): Möglichkeiten der Bearbeitung von Umweltschutz-Vorschriften mit Hilfe der Expertensystem-Technik. In: Hommel G.; Schindler S. (Hrsg.): GI - 16. Jahrestagung II. Informatik-Anwendungen - Trends und Perspektiven. Berlin, Oktober 1986. Berlin: Springer.
- Hanus M. (1989): Logisches Programmieren (Teil 1). Studienmaterial 1816-2-01-S1. Hagen: Fernuniversität - Gesamthochschule - Hagen, Fachbereich Mathematik und Informatik.
- Harary F. (1974): Graphentheorie. München: Oldenbourg.
- Harmon P.; King D. (1989): Expertensysteme in der Praxis: Perspektiven, Werkzeuge, Erfahrungen. München: Oldenbourg.
- 't Hart C.; Heyna B.J.; van Tongeren P.; van Weel P. (1984): Heated floors with a flooded watering system. Acta Horticulturae, 148, S. 65-72.
- Hart E.; Yitayew M.; Ekholt B.A.; Kim T.G. (1989): Irrigation system selection. ASAE Paper No. 89-7042. St. Joseph, Mi: American Society of Agricultural Engineers.
- Haug H. (1990): Expertensysteme lassen noch manchen Wunsch offen. VDI nachrichten, Nr. 38, 21. Sept. 1990, S. 37.
- Heatwole C.D.; Zhang T.L. (1988): Representing Uncertainty: Theoretical and Practical Considerations. ASAE Paper No. 88-7538. St. Joseph, Mi: American Society of Agricultural Engineers.
- Herberholz P. (1991): Harter Markt für weiche Ware - Software als Helfer für das Energiemanagement. Energie Spektrum, Mai 1991, S. 13-15.
- Hertzberg J. (1986): Planen und die Repräsentation der realen Welt. Informatik Berichte Nr. 50. Bonn: Institut für Informatik, Universität Bonn.
- Hibbeler M. (1989): Benutzerfreundlichkeit: Graphische Benutzungsoberflächen. ComputerMagazin, 18 (9), S. 48-49.
- Hönmann W.; Sprenger E. (Hrsg.) (1985): Taschenbuch für Heizung und Klimatechnik. München: Oldenbourg.
- Hofer R. (1987): Künstliche Intelligenz: Vom Mauerblümchen zur Stardisziplin. MEGA Sonderheft Nr. 250, Expertensysteme, S. 18-21. München: Franzis.

- Holt D.A. (1989): The Growing Potential of Expert Systems in Agriculture. In: Barrett J.R.; Jones D.D. (Hrsg.): Knowledge Engineering in Agriculture. St. Joseph, Mi: American Society of Agricultural Engineers.
- Hoshi T.; Kozai T. (1984): Knowledge-based and hierarchically distributed online control system for greenhouse management. *Acta horticulturae*, 148, S. 301-308.
- Jacobson B.K.; Jones P.H.; Jones J.W.; Paramore J.A. (1989): Real-time greenhouse monitoring and control with an expert system. *Computers and Electronics in Agriculture*, 3, S. 273-285.
- Janson A. (1989): *Expertensysteme und Turbo-Prolog: Der sichere Weg zum eigenen Programm*. München: Franzis.
- Jones J.W.; Jones P.; Everett P.A. (1987): Combining Expert Systems and Agricultural Models: A Case Study. *Transactions of the ASAE*, 30 (5), S. 1308-1314.
- Kaemena K. (1990): Automatisierte Wissensakquisition. In: *BremTec '90 - Seminar 28. Juni 1990. Der Einsatz von Expertensystemen in der Industrie*. Bremen: Universität Bremen, PC-Labor.
- Kafka L.; Petzoldt T.; Petersohn U.; Recknagel F. (1991): Unscharfe und fallbasierte Inferenzmethoden zur Vorhersage von Algenmassenentwicklungen. In: Hälker M.; Jaeschke A. (Hrsg.): *Informatik für den Umweltschutz*. 6. Symposium München, Dezember 1991. *Informatik-Fachberichte 296*. Berlin: Springer.
- Kaindl H. (1989): *Problemlösen durch heuristische Suche in der Artificial Intelligence*. Wien: Springer.
- Kaler Jr. G.M. (1990): Embedded Expert System Development for Monitoring Packaged HVAC Equipment. *ASHRAE Transactions*, 96 (2), S. 733-742.
- Kandel A. (1986): *Fuzzy mathematical techniques with applications*. Reading, Ma: Addison-Wesley.
- Karagiannis D. (1987): *Repräsentation und Verarbeitung von Wissensstrukturen: Das hybride System KANON*. Dissertation Technische Universität Berlin, Fachbereich Informatik.



- Karras D.; Kredel L.; Pape U. (1987): Entwicklungsumgebungen für Expertensysteme: Vergleichende Darstellung ausgewählter Systeme. Berlin: de Gruyter.
- Kast W.; Klan H.; Bohle J. (1986): Wärmeleistung von Fußbodenheizungen. HLH, 37 (4), S. 175-182.
- Kaufmann A. (1975): Theory of Fuzzy Subsets. New York: Academic Press.
- Kaufmann A.; Gupta M.M. (1988): Fuzzy mathematical models in engineering and management science. Amsterdam: Elsevier Science Publishers.
- Kavka M.; Vesely P.; Kovar K. (1988): Möglichkeiten der Anwendung von Expertensystemen in der Landwirtschaft und eine Applikation von Agroexpert in der Motoröldiagnostik. In: Geidel H.; Mangstl A. (Hrsg.): Referate der 9. GIL-Jahrestagung in Münster, September 1988. Stuttgart: Ulmer.
- O'Keefe M.R. (1989): The Role of Artificial Intelligence in Discrete-Event Simulation. In: Widmann L.E.; Loparo K.A.; Nielsen N.R. (Hrsg.): Artificial intelligence, simulation, and modeling. New York: John Wiley & Sons.
- Kline D.E.; Bender D.A.; McCarl B.A.; van Donge C.E. (1988): Machinery selection using expert systems and linear programming. Computers and Electronics in Agriculture, 3, S. 45-61.
- Kozai T. (1985): Ideas of greenhouse climate control based on knowledge engineering techniques. Acta horticulturae, 174, S. 365-373.
- KTBL (1990): KTBL-Taschenbuch Gartenbau. Münster-Hiltrup: Landwirtschaftsverlag.
- van Leeuwen Buizen (1990): Produktinformation der Firma van Leeuwen Buizen, Lindsedijk 20, Postbus 1, 3330 AA Zwijndrecht.
- Lehner P.E.; Kralj M.M. (1988): Cognitive impacts of the user interface. In: Hendler J.A. (Hrsg.): Expert systems: The user interface. Norwood, NJ: Ablex.
- Lentz W. (1987): Bestimmung von Temperaturstrategien für Unterglasgemüsekulturen mittels bio-ökonomischer Modelle und numerischer Suchverfahren. Dissertation Universität Hannover, Fachbereich Gartenbau.
- Lentz W. (1991): Mögliche Entwicklungslinien für computergestützte Entscheidungshilfsmittel. Ber. Ldw., 69, S. 69-99.

- Li B.; Heatwole C.D; Perumpral J.V. (1990): Knowledge-Based Assistant for Component Selection and Preliminary Design of Hydrostatic Drivers. Applied Engineering in Agriculture, 6 (3), S. 359-366.
- Link P. (1991): Expertensysteme in der Landwirtschaft. Diplomarbeit Justus - Liebig - Universität Giessen, Institut für landwirtschaftliche Betriebslehre.
- Lipschutz S. (1987): Datenstrukturen. Hamburg: McGraw-Hill.
- Lloyd J.W. (1987): Foundations of Logic Programming. Berlin: Springer.
- Loewer O.J.; Kocher M.F.; Solaimanian J. (1990): An Expert System for Determining Bottlenecks in On-Farm Grain Processing Systems. Applied Engineering in Agriculture, 6 (1), S. 69-72.
- Mainka C.; Mangstl A. (Hrsg.) (1991): Beiträge zum Workshop Faktendatenbanken im Agrarbereich, April 1991 in Bonn. Agrarinformatik Band 23. Stuttgart: Ulmer.
- Marcellus D.H. (1989): Expert systems programming in Turbo Prolog. Englewood Cliffs, NJ: Prentice-Hall.
- Marcus S. (Hrsg.) (1988): Automating knowledge acquisition for expert systems. Boston: Kluwer Academic Publishers.
- McLeod J. (1982): Computer Modeling and Simulation. Simulation Series, 10 (2), La Jolla, Ca: Simulation Councils.
- Merritt D. (1989): Building expert systems in Prolog. New York: Springer.
- Mertens P. (1987): Expertensysteme in den betrieblichen Funktionsbereichen - Chancen, Erfolge, Mißerfolge. In: Wildemann H. (Hrsg.): Expertensysteme in der Produktion. München: Gesellschaft für Management und Technologie.
- Meyer J. (1989): Technische Grundlagen der Pflanzenbelichtung. KTBL-Arbeitsblatt Lfd. Nr. 0658. Braunschweig: Thalacker.
- Meyer M. (1989): SO2XPS, an Expert System for the Damage Assessment of SO2 on Plants. In: Jaeschke A.; Geiger W.; Page B. (Hrsg.): Informatik im Umweltschutz. 4. Symposium Karlsruhe, November 1989. Informatik-Fachberichte 228. Berlin: Springer.

- Meyer U.; Schmidt T. (1991): Konzeption und Implementierung eines Expertensystems im Bereich des städtebaulichen Lärmschutzes. In: Hälker M.; Jaeschke A. (Hrsg.): Informatik für den Umweltschutz. 6. Symposium München, Dezember 1991. Informatik-Fachberichte 296. Berlin: Springer.
- Minsky M. (1975): A framework for representing knowledge. In: Winston P.H. (Hrsg.): The psychology of computer vision. New York: McGraw-Hill.
- Moser J. (1991): Objektorientiertes Programmieren. Computer Magazin, 20 (3), S. 48-52.
- Müller G. (1989): Objektorientierter Datenbank-Entwurf am Beispiel des sozio-ökonomischen Panels. In: Schiefer G. (Hrsg.): Expertensysteme in der Agrarwirtschaft: Entwicklung, Erfahrung, Perspektiven. Vorträge der ITMA-Fachtagung "Expertensysteme 1988". Kiel: Wissenschaftsverlag Vauk.
- Müller G.J. (1987): Energieschirme unter Praxisbedingungen - Bewertung und Optimierung im Hinblick auf Energieverbrauch und Klimaführung. Gartenbautechnische Informationen, Heft 28. Hannover: Institut für Technik in Gartenbau und Landwirtschaft, Universität Hannover.
- Nastansky L.; Seidensticker F.J. (1990): Anwendungen und Konzepte für hypermedia-basiertes Informationsmanagement am netzintegrierten Managerarbeitsplatz. Wirtschaftsinformatik, 32 (6), S. 519-537.
- Negahban B.; Fluck R.C.; Shoup W.D. (1988): Selection of lawn and garden tractors using an expert system. Applied Engineering in Agriculture, 4 (3), S. 211-216.
- Negoita C.V. (1985): Expert Systems and Fuzzy Systems. Menlo Park, Ca: Benjamin/Cummings.
- Neidhardt J. (1989): Expertensysteme haben sich noch nicht durchgesetzt. Die Welt, Nr. 254, 31. Oktober 1989, S. 31.
- Neumann B. (1986): Was sind Expertensysteme ? In: Krallmann H. (Hrsg.): Expertensysteme im Unternehmen. Berlin: Ernst Schmidt.
- Niemeyer G. (1977): Kybernetische System- und Modelltheorie. München: Franz Vahlen.
- Peart R.M. (1989): Applications of expert systems in agriculture. In: Barrett J.R.; Jones D.D (Hrsg.): Knowledge Engineering in Agriculture. St. Joseph, Mi: American Society of Agricultural Engineers.

- Pham D.T.; Pham P.T.N. (1988): Expert systems: A review. In: Pham D.T. (Hrsg.): Expert systems in engineering. Berlin: Springer.
- Pigorsch J. (1991): VULKAN - Schadstoffbelastung an Arbeitsplätzen in der Gummiindustrie und ihre Überwachung. Hannover: Universität Hannover, Institut für anorganische Chemie.
- Pinto I.C. (1989): Wissensbasierte Unterstützung bei der Lösung von Optimierungsaufgaben. Dissertation Universität Dortmund, Fachbereich Informatik.
- Prade H. (1985): A Computational Approach to Approximate and Plausible Reasoning with Applications to Expert Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 7 (3), S. 260-283.
- Preiß N. (1986): Aspekte der Datenbankunterstützung für PROLOG. Forschungsbericht 174. Karlsruhe: Universität Karlsruhe, Institut für angewandte Informatik und formale Beschreibungsverfahren.
- Preiß N.; Stucky W. (1988): Probleme des Datenmanagements für entscheidungsunterstützende Systeme. In: Wolff M.R. (Hrsg.): Entscheidungsunterstützende Systeme im Unternehmen. Wirtschafts-Informatik-Symposium der IBM Deutschland GmbH, 26.-28. November 1986, Bad Neuenahr. München: Oldenbourg.
- Puppe F. (1987): Diagnostisches Problemlösen mit Expertensystemen. Informatik-Fachberichte 148. Berlin: Springer.
- Radermacher F.J. (1988): Entwicklungsperspektiven rechnergestützter Entscheidungsfindung. In: Wolff M.R. (Hrsg.): Entscheidungsunterstützende Systeme im Unternehmen. Wirtschafts-Informatik-Symposium der IBM Deutschland GmbH, 26.-28. Nov. 1986, Bad Neuenahr. München: Oldenbourg.
- Radig B. (1987): Künstliche Intelligenz - Überblick und Orientierung. München: Digital Equipment.
- Rath T. (1990): Hilfsmittel für die Technik-Beratung. Deutscher Gartenbau, 44 (4), S. 200-201.
- Rath T. (1991a): Expertensysteme für den Gartenbau - Fachwissen für den Rechner aufbereitet. GbGw, 91 (43), S. 2134-2139.
- Rath T. (1991b): Neue Informationsmöglichkeiten durch gartenbautechnische Expertensysteme? Gartenbaureport, 17 (12), S. 12-13.

- Rath T.; von Elsner B. (1989): Rechnergestützte Planung von Bewässerungsanlagen im Gartenbau. *Gartenbauwissenschaft*, 54 (4), S. 171-176.
- Rath T.; Tantau H.J. (1991): Lehrschau: Expertensystem für die Technik im Gartenbau. *plantec 1991*, internationale Fachmesse für Gartenbau, Frankfurt.
- Reddy Y.V.R.; Fox M.S.; Husain N.; McRoberts M. (1986): The Knowledge-Based Simulation System. *IEEE Software*, 3 (2), S. 26-37.
- Reuter A. (1987): Kopplung von Datenbank- und Expertensystemen. *Informationstechnik*, 29 (3), S. 164-175.
- Rich E. (1988): KI - Einführung und Anwendungen. Hamburg: McGraw-Hill.
- Riedel W. (1982): Computerberatung Energieeinsparung. *Deutscher Gartenbau*, 36 (37), S. 1522-1524.
- Rimek G. (1991): Anspruch und Wirklichkeit. *ComputerMagazin*, 20 (6), S. 49-50.
- Roberts W.J.; Mears D.R. (1980): Floor heating of greenhouses. ASAE Paper No. 80-4027. St. Joseph, Mi: American Society of Agricultural Engineers.
- Robinson (1965): A Machine-Oriented Logic Based on the Resolution Principle. *Journal of the Association for Computing Machinery*, 12 (1), S. 23-41.
- Röhrich E.; Taeger J. (1991): Gewässerschutz - ein integriertes Informationssystem zum Umweltrecht. In: Hälker M.; Jaeschke A. (Hrsg.): *Informatik für den Umweltschutz*, 6. Symposium München, Dezember 1991. *Informatik-Fachberichte* 296. Berlin: Springer.
- Rösner H. (1986): Generierung von Erklärungen aus formalen Wissensrepräsentationen. *Verbundvorhaben WISBER*, Bericht Nr. 3, Dokument Nr. 5-12-74, Nixdorf Computer AG.
- Rothenberg J. (1989): The Nature of Modeling. In: Widman L.E.; Loparo K.A.; Nielsen N.R. (Hrsg.): *Artificial Intelligence, Simulation, and Modeling*. New York: John Wiley & Sons.
- Sager W. (1991): Objektorientierte Programmierung - Übersicht und Begriffe. *ComputerMagazin*, 20 (7-8), S. 34-40.

- Salski A.; Kandzia P. (1991): Einsatz der Fuzzy-Set-Theorie in der Ökosystemforschung. In: Hälker M.; Jaeschke A. (Hrsg.): Informatik für den Umweltschutz. 6. Symposium München, Dezember 1991. Informatik-Fachberichte 296. Berlin: Springer.
- Savory S.E. (1988): Grundlagen von Expertensystemen. München: Oldenbourg.
- Schlageter G. (1983): Datenbanksysteme 1. Studienmaterial 1665-9-03-S1. Hagen: Fernuniversität - Gesamthochschule - Hagen, Fachbereich Mathematik und Informatik.
- Schmidt B. (1985): Systemanalyse und Modellaufbau: Grundlagen der Simulationstechnik. Berlin: Springer.
- Schneider K. (1987): Quick Bibliography Series. Expert systems and computer aids to decision-making 1970-1986. Beltsville Maryland: United States Department of Agriculture, National Agricultural Library.
- Schnupp P.; Nguyen Huu C.T. (1987): Expertensystem-Praktikum. Berlin: Springer.
- Shadbolt N. (1989): Knowledge representation in man and machine. In: Forsyth R. (Hrsg.): Expert systems: Principles and case studies. London: Chapman and Hall.
- Shortliffe E.H. (1976): Computer-based medical consultations: MYCIN. New York: Elsevier.
- Shortliffe E.H.; Buchanan B.G. (1975): A Model of Inexact Reasoning in Medicine. *Mathematical Biosciences*, 23, S. 351-379.
- Sommerville I. (1990): Software Engineering. Wokingham: Addison-Wesley.
- Sprague R.H.; Carlson E.D. (1982): Building effective decision support systems. London: Prentice-Hall.
- Sprague R.H.; Watson H.J. (1989): Decision support systems. Putting theory into practice. London: Prentice-Hall.
- Stefik M.; Bobrow D.G. (1986): Object-Oriented Programming: Themes and Variations. *The AI Magazine*, 6 (4), S. 40-62.
- Stender J. (1988): Eine andere Sicht von Knowledge Engineering. *KI*, 2 (1), S. 41-42.

- Storck H. (1977): Investitionsentscheidungen im Gartenbau: Prinzipien und Kalkulationsverfahren. Berlin: Parey.
- Stoyan H. (1988): Wissen wissensbasierte Programme etwas ? In: Heyer G.; Krems J.; Görz G. (Hrsg.): Wissensarten und ihre Darstellung. Berlin: Springer.
- Strothotte T.; Böcke D. (1989): Informationsvermittlung in interaktiven wissensbasierten Systemen durch bildhafte Darstellung. In: Maaß S.; Oberquelle H. (Hrsg.): Software-Ergonomie '89. Aufgabenorientierte Systemgestaltung und Funktionalität. Stuttgart: Teubner.
- Strunz H. (1977): Entscheidungstabellentechnik. München: Carl Hanser.
- Sviokla J. (1989): Business implications of knowledge-based systems. In: Sprague R.H.; Watson H.J. (Hrsg.): Decision support systems. Putting theory into practice. London: Prentice-Hall.
- Tanimoto S.L. (1990): KI: Die Grundlagen. München: Oldenbourg.
- Tantau H.J. (1983): Heizungsanlagen im Gartenbau. Stuttgart: Ulmer.
- Tantau H.J. (1990): Heizungssysteme im Gewächshaus. In: Ruhrgas AG: Erdgas - wirtschaftliche und umweltfreundliche Energie für den Gartenbau. Essen: Ruhrgas-Aktiengesellschaft.
- Tello E.R. (1989): Object-Oriented Programming for Artificial Intelligence. Reading, Ma: Addison-Wesley.
- Thieme R.H.; Whittaker A.D. (1986): Representation and reasoning issues in expert systems. ASAE Paper No. 86-4512. St. Joseph, Mi: American Society of Agricultural Engineers.
- Thomson S.J.; Peart R.M. (1986): An expert system for soil moisture-based scheduling of center pivot irrigation. ASAE Paper No. 86-4518. St. Joseph, Mi: American Society of Agricultural Engineers.
- Tischendorf M. (1989): Möglichkeiten der Kontrolle und Analyse von Umweltdaten durch Kopplung von Datenbank- und Expertensystemen. In: Jaeschke A.; Geiger W.; Page B. (Hrsg.): Informatik im Umweltschutz. 4. Symposium Karlsruhe, November 1989. Informatik-Fachberichte 228. Berlin: Springer.

- Tuluca A.N.; Turner G.E.; Krarti M.; Antipa V. (1989): Expert System for Selection of Standardized Research Greenhouses. ASHRAE Transactions, 95 (2), S. 938-946.
- Turban E.; Watkins P.R. (1986): Integrating Expert Systems and Decision Support Systems. MIS Quarterly, June 1986, S. 121-136.
- Ultsch A.; Siemon H.P.; Halmans G. (1991): Neuronale Netze zur Unterstützung der Umweltforschung. In: Hälker M.; Jaeschke A. (Hrsg.): Informatik für den Umweltschutz. 6. Symposium München, Dezember 1991. Informatik-Fachberichte 296. Berlin: Springer.
- VDI (1980): VDI-Richtlinie 3781. Ausbreitung luftfremder Stoffe in der Atmosphäre. Blatt 4: Bestimmung der Schornsteinhöhe für kleinere Feuerungsanlagen. Berlin: Beuth.
- VDI (1983): VDI-Richtlinie 2067. Berechnung der Kosten von Wärmeversorgungsanlagen. Blatt 1: Betriebstechnische und wirtschaftliche Grundlagen. Berlin: Beuth.
- VDI (1989): VDI-Richtlinie 2067. Berechnung der Kosten von Wärmeversorgungsanlagen. Blatt 6: Wärmepumpen.
- Vickermann E. (1974): Kalkulation der Heizkosten im Gartenbaubetrieb. Berlin: Paul Parey.
- Voges U.; Epke K. (1991): PRO-PLANT. Ein Beratungssystem für umweltgerechten Pflanzenschutz. Shell-Entwicklung und Wissensrepräsentation. In: Hälker M.; Jaeschke A. (Hrsg.): Informatik für den Umweltschutz. 6. Symposium München, Dezember 1991. Informatik-Fachberichte 296. Berlin: Springer.
- Vorwerk R. (1991): Objektorientierte Sprachen als Software-Entwicklungsumgebung. ComputerMagazin, 20 (7-8), S. 28-33.
- Voß H. (1988): Werkzeuge für Expertensysteme - Hardware- und Softwarebasis für Entwicklung und Betrieb. In: Expertensysteme in der betrieblichen Praxis - Ergebnisse des AWF/VDI Arbeitskreises. Fachtagung 7.-8. März 1988. Eschborn: Ausschuß für Wirtschaftliche Fertigung (AWF).
- Walker A. (Hrsg.) (1987): Knowledge systems and prolog. Reading, Ma: Addison-Wesley.



- Watson D.G.; Brook R.C. (1990): Aeration System Design for Flat Grain Storages with an Expert System. *Applied Engineering in Agriculture*, 6 (2), S. 183-188.
- Widman L.E.; Loparo K.A. (1989): Artificial Intelligence, Simulation, and Modeling: A Critical Survey. In: Widman L.E.; Loparo K.A.; Nielsen N.R. (Hrsg.): *Artificial Intelligence, Simulation, and Modeling*. New York: Jon Wiley & Sons.
- Winston P.H. (1987): *Künstliche Intelligenz*. Bonn: Addison-Wesley.
- Winter R. (1990): Integration von Abstraktionshierarchien mit der Bedarfsauflösung - Eine Voraussetzung für "intelligente" Datenbank Anwendungen in der mehrstufigen Produktionsplanung. *Wirtschaftsinformatik*, 32 (6), S. 550-565.
- Wirth N. (1975): *Algorithmen und Datenstrukturen*. Stuttgart: Teubner.
- Wittur K.H.; di Primio F. (1989): Verfahren der Künstlichen Intelligenz für die industrielle Produktion. *Chip*, Februar 1989 (Chip Plus), S. 18-22.
- Wurr P.R. (1989): Expertensysteme. *ComputerMagazin*, 18 (6-7), S. 43-45.
- von Zabeltitz C. (1986): *Gewächshäuser: Planung und Bau*. Stuttgart: Ulmer.
- Zadeh L.A. (1965): Fuzzy sets. *Information and control*, 8, S. 338-353.
- Zadeh L.A. (1989): Knowledge Representation in Fuzzy Logic. *IEEE Transactions on Knowledge and Data Engineering*, 1 (1), S. 89-100.
- Zahn M. (1991): Multimedia: Woher und wohin? *mini Mikro magazin*, 7 (3), S. 68-69.
- Ziegler J.; Koller F. (1991): Fraunhofer-Gesellschaft. Wir informieren: MULTEX, ein multi-mediales Expertensystem. Stuttgart: Fraunhofer-Institut für Arbeitswirtschaft und Organisation.
- Zimmermann H.J. (1988): Modellierung von Unsicherheit in Expertensystemen. In: Wolff M.R. (Hrsg.): *Entscheidungsunterstützende Systeme im Unternehmen*. *Wirtschafts-Informatik-Symposium der IBM Deutschland GmbH*, 26.-28. November 1986, Bad Neuenahr. München: Oldenbourg.

Verwendete Firmenunterlagen:

ALBO-DATA-SYSTEM Vetriebs-GmbH  
Anton-Aulke-Str. 1  
4425 Billerbeck

ETU-Datentechnik  
Amsterdamer Straße 91  
5000 Köln 60

Gerald Bax  
Schliemannstr. 20  
4800 Bielefeld 1

MW Software  
Markert Welfens & Partner GmbH  
Ubierring 11  
5000 Köln 1

PDS  
Programm + Datenservice GmbH  
Mühlenstr. 22  
2720 Rotenburg

Resch Verlag  
Postfach 1260  
8032 Gräfelfing

Dr. Saß  
Bleiweißstr. 40  
6000 Frankfurt 70

Solar Computer  
Ingenieurgesellschaft Rössel  
Postfach 1480  
6455 Erlensee

Stegemann Kaminbau GmbH  
Oststraße 5  
4405 Nottuln

Sykaplan Computer GmbH  
St. Rochus-Straße 45  
8700 Würzburg

## **7 Eingesetzte Software**

### Turbo Prolog 2.0

Heimsoeth & Borland

### Turbo Pascal 5.0

Heimsoeth & Borland

### Lotus 123 3.0

Lotus Development Corporation

### MetaWindow 3.4B

Metagraphics Software Corporation

### graphics-MENU 2.0

Island Systems

### FontWindow 4.0

Metagraphics Software Corporation

### Harvard Graphics 2.4

Software Publishing Corporation

### Paintbrush 2.0

Microsoft, ZSoft

**Anhang**

**A Regelinterpretier**

```
/* KNOTEN.PRO T.R. */  
/* Regelinterpretierer */
```

```
database-knotenbank  
bekannt(element)
```

```
predicates
```

```
los(elementeliste,elementeliste,elementeliste)  
los2(elementeliste,elementeliste,elementeliste,element)  
weiter(element,elementeliste,elementeliste,elementeliste)  
anhaengen_Liste_an_Liste(elementeliste,elementeliste,elementeliste)  
anhaengen_Element_an_eine_Liste(element,elementeliste,elementeliste)  
vorhanden(elementeliste,element,element,element)  
listensuchen(element,elementeliste,elementeliste)  
dran(element,elementeliste,elementeliste,elementeliste)  
mitglied_einer_elementeliste(element,elementeliste)
```

```
clauses
```

```
los([],K,K).
```

```
los([Kopf|Rest],Eingang,Ausgang):-  
weiter(Kopf,Liste,Eingang,Ausgabe),  
anhaengen_Liste_an_Liste(Liste,Rest,NeueListe),  
los(NeueListe,Ausgabe,Ausgang).
```

```
los2([],K,K,Abschneidevariante):-  
assert(bekannt(Abschneidevariante),knotenbank).
```

```
los2([Kopf|Rest],Eingang,Ausgang,Abschneidepunkt):-  
weiter(Kopf,Liste,Eingang,Ausgabe),  
anhaengen_Liste_an_Liste(Liste,Rest,NeueListe),  
los2(NeueListe,Ausgabe,Ausgang,Abschneidepunkt).
```

```
weiter(Faktum,[],Ein,Aus):-  
knoten(Faktum,ist_ziel,_,Ein),  
anhaengen_Element_an_eine_Liste(Faktum,Ein,Aus).
```

```
weiter(Faktum,[],Ein,Ein):-  
knoten(Faktum,ist_ende,_,Ein).
```

```
weiter(Faktum,Moeglichkeit,Ein,Aus):-  
knoten(Faktum,ist_ziel_hat_moeglichkeit,Moeglichkeit,Ein),  
anhaengen_Element_an_eine_Liste(Faktum,Ein,Aus).
```

```
weiter(Faktum,Moeglichkeit,Ein,Ein):-  
knoten(Faktum,hat_moeglichkeit,Moeglichkeit,Ein).
```

```
weiter(Faktum,[],Ein,Aus):-  
retractall(bekannt(Faktum),knotenbank),  
knoten(Faktum,ist_ziel_hat_variante,Variante,Ein),  
not(bekannt(Faktum)),  
anhaengen_Element_an_eine_Liste(Faktum,Ein,Ein_neu),  
los2(Variante,Ein_neu,Aus,Faktum).
```

```
weiter(Faktum,[],Ein,Aus):-  
retractall(bekannt(Faktum),knotenbank),
```

```
knoten(Faktum,hat_variante,Variante,Ein),  
not(bekannt(Faktum)),  
los2(Variante,Ein,Aus,Faktum).
```

```
vorhanden(Liste,x,E2,Ziel):-  
  not(E2 = x),!,  
  dran(E2,Liste,[],Raus),  
  mitglied_einer_elementeliste(Ziel,Raus).
```

```
vorhanden(Liste,E1,x,Ziel):-  
  not(E1 = x),!,  
  listensuchen(E1,Liste,KK),  
  mitglied_einer_elementeliste(Ziel,KK).
```

```
vorhanden(Liste,x,x,Ziel):-!,  
  mitglied_einer_elementeliste(Ziel,Liste).
```

```
vorhanden(Liste,E1,E2,Ziel):-!,  
  listensuchen(E1,Liste,KK),  
  dran(E2,KK,[],Raus),  
  mitglied_einer_elementeliste(Ziel,Raus).
```

```
listensuchen(_,[],[]):-!.
```

```
listensuchen(Start,[Kopf|Rest],A):-  
  Kopf = Start,  
  A = Rest,!.
```

```
listensuchen(Start,[_|Rest],Ergebnis):-!,  
  listensuchen(Start,Rest,Ergebnis).
```

```
dran(_,[],A,A):-!.
```

```
dran(Start,[Kopf|_] ,A,A):-  
  Kopf = Start,!.
```

```
dran(Start,[Kopf|Rest],Alt,Ergebnis):-!,  
  anhaengen_Element_an_eine_Liste(Kopf,Alt,Neu),  
  dran(Start,Rest,Neu,Ergebnis).
```

```
anhaengen_Element_an_eine_Liste(X,Y,Z):-  
  anhaengen_Liste_an_Liste(Y,[X],Z).
```

```
anhaengen_Liste_an_Liste([],Z,Z):-!.  
anhaengen_Liste_an_Liste([X|Y],Z,[X|U]):-  
  anhaengen_Liste_an_Liste(Y,Z,U).
```

```
mitglied_einer_elementeliste(Kopf,[Kopf|_] ).  
mitglied_einer_elementeliste(Kopf,[_|Rest]):-  
  mitglied_einer_elementeliste(Kopf,Rest).
```

```
/* FRAGEN.PRO T.R. */
/* Handling der Benutzeranfragen bei Produktionsregeln*/

domains
  symbolliste = symbol*

database-internebank
  bekannt_ja(symbol,symbol)
  bekannt_nein(symbol,symbol)
  bekannt_zahl(symbol,real)
  bekannt_auswahlliste(symbol,string)
  antwort(integer,symbol,symbol,symbol)
  f(integer,symbol,symbolliste,symbolliste,string,symbol,real,real)

global database-fragedatenbank
  ende
  frage(string)
  moege(symbolliste)

predicates
  fragedatei_laden(string)
  pascal_antworten_laden(string)
  fertiges_prolog_ziel_rausgeben
  wenn_zahl(symbol,symbol,real)
  bekannt_erstellen
  vergleiche(symbol,symbol,symbol)
  wenn_eingabe(symbol,symbol,symbol)
  fragen(symbol,symbol)
  wenn__(symbol,symbol,symbol)
  wenn_nicht(symbol,symbol,symbol)
  wenn_auswahlliste(symbol,symbol,string)
  alle_abspeichern(integer,symbol,symbol,symbolliste)

clauses

fragedatei_laden(Fragedatei):-
  retractall(_,loesungsbank),
  retractall(_,internebank),
  retractall(_,fragedatenbank),
  consult(Fragedatei,internebank),!.

pascal_antworten_laden(Datei):-
  consult(Datei,internebank),
  bekannt_erstellen,!.

fertiges_prolog_ziel_rausgeben:-
  retractall(_,fragedatenbank),
  asserta(ende),
  save("FRAGE.VIR",fragedatenbank),!,
  save("LOESUNG.VIR",loesungsbank).

wenn_eingabe(T1,T2,Aus):-
  concat(T1,T2,T3),
  bekannt_ja(T3,Aus2),!,
  Aus2 = Aus.

wenn_eingabe(T1,T2,_):-
  fragen(T1,T2).
```

```
wenn_zahl(T1,T2,Aus):-  
    concat(T1,T2,T3),  
    bekannt_zahl(T3,Neu),!,  
    Neu = Aus.
```

```
wenn_zahl(T1,T2,CC):-  
    bound(CC),  
    fragen(T1,T2).
```

```
wenn_zahl(T1,T2,0):-  
    fragen(T1,T2).
```

```
wenn__(T1,T2,Sp) :-  
    concat(T1,T2,T3),  
    bekannt_ja(T3,Sp),!.
```

```
wenn__(T1,T2,Sp) :-  
    concat(T1,T2,T3),  
    bekannt_nein(T3,Sp),!,  
    fail.
```

```
wenn__(T1,T2,VV):-  
    bound(VV),  
    fragen(T1,T2).
```

```
wenn__(T1,T2,ja):-  
    fragen(T1,T2).
```

```
wenn__nicht(T1,T2,Sp) :-  
    concat(T1,T2,T3),  
    bekannt_nein(T3,Sp),!.
```

```
wenn__nicht(T1,T2,Sp) :-  
    concat(T1,T2,T3),  
    bekannt_ja(T3,Sp),!,  
    fail.
```

```
wenn__nicht(Frage,T2,_):-  
    fragen(Frage,T2).
```

```
wenn__auswahlliste(T1,T2,Sp) :-  
    concat(T1,T2,T3),  
    bekannt_auswahlliste(T3,Sp),!.
```

```
wenn__auswahlliste(Frage,T2,"Unbekannt"):-  
    fragen(Frage,T2).
```

```
fragen(Frage,Variable) :-!,  
    f(Art,Frage,_,Moeglich,Text,Hilfe,Mini,Maxi),  
    format(Aus,"%*%*%*%*%*%*%*%*%",Art,Frage,Variable,Text,Hilfe,Mini,Maxi),  
    assertz(frage(Aus),fragedatenbank),  
    assertz(moege(Moeglich),fragedatenbank),  
    save("frage.vir",fragedatenbank),  
    exit.
```

```
bekannt_erstellen:-  
    antwort(Art,Frage,Variable,Antwort),
```



```
retractall(antwort(Art,Frage,Variable,Antwort),internebank),
f(Art,Frage,_,Moeglich,_,_,_,_),!,
concat(Frage,Variable,Komplett),
alle_abspeichern(Art,Komplett,Antwort,Moeglich),
    bekannt_erstellen.

bekannt_erstellen.

alle_abspeichern(1,_,_,[]):-!.

alle_abspeichern(1,T3,Antwort,[Kopf|Rest]):-!,
    vergleich(T3,Antwort,Kopf),
    alle_abspeichern(1,T3,Antwort,Rest).

alle_abspeichern(2,Frage,Antwort,_):-!,
    assertz(bekannt_ja(Frage,Antwort),internebank).

alle_abspeichern(3,Frage,Antwort,_):-!,
    str_real(Antwort,Zahl),
    asserta(bekannt_zahl(Frage,Zahl),internebank).

alle_abspeichern(4,Frage,Antwort,_):-!,
    asserta(bekannt_auswahlliste(Frage,Antwort),internebank).

vergleich(Tatsache,Antwort,Kopf):-
    Antwort = Kopf,!,
    assertz(bekannt_ja(Tatsache,Kopf),internebank).

vergleich(Tatsache,_,Kopf):-
    assertz(bekannt_nein(Tatsache,Kopf),internebank).
```

**B Objektinterpreter**

```
/* FRAME.PRO T.R. */  
/* Objektinterpreter */
```

predicates

```
frage(string,slot,string)  
frage_wert(string,slot,real)  
suche(string,slot,stringeliste)  
suche_vaeter(stringeliste,slot,stringeliste)  
suche_objekte(string,slotliste,slot,stringeliste)  
suche_einzelobjekte(stringeliste,slot,stringeliste)  
prozedur(string,slot,stringeliste)  
mitglied_der_liste(string,stringeliste)  
mitglied_der_slotliste(slot,slotliste)  
zeige_objekt(string,stringeliste,slotliste)  
zeige_werte(string,facet,slotliste)  
zeige_wenn_slots(string,slotliste)  
zeige_vater_slots(stringeliste)  
zeige_abgeleitetes_objekt(string)  
zeige_alle_vaeter_slots(string)  
zeige_objekt_slots(stringeliste)  
zeige_alle_objekt_slots(string,slotliste)  
gebe_slots_raus(slot,stringeliste)  
gebe_end_soehne_raus(string,string)  
pruefe_vater(string,string)  
ergaenze_planung(slot,string)
```

clauses

```
pruefe_vater(Objekt,Vater):-  
    frame(Objekt,ist_ein,wert,Werteliste,_),  
    mitglied_der_liste(Vater,Werteliste),!.  
  
pruefe_vater(Objekt,Vater):-  
    frame(Objekt,ist_ein,wert,Werteliste,_),  
    mitglied_der_liste(Vater2,Werteliste),  
    pruefe_vater(Vater2,Vater).  
  
frage(Objekt,Slot,Wert):-  
    bound(Objekt),bound(Slot),free(Wert),  
    suche(Objekt,Slot,Werteliste),  
    mitglied_der_liste(Wert,Werteliste).  
  
frage(Objekt,Slot,Wert):-  
    bound(Objekt),bound(Slot),bound(Wert),  
    suche(Objekt,Slot,Werteliste),  
    mitglied_der_liste("beliebig",Werteliste),!.  
  
frage(Objekt,Slot,Wert):-  
    bound(Objekt),bound(Slot),bound(Wert),  
    suche(Objekt,Slot,Werteliste),  
    mitglied_der_liste(Wert,Werteliste),!.  
  
frage_wert(Objekt,Slot,Wert):-!,  
    frage(Objekt,Slot,S),  
    str_real(S,Wert).  
  
suche(Objekt,Slot,Wert):-
```

```
frame(Objekt,Slot,objekt,Wert,_),!.

suche(Objekt,Slot,Wert):-
  frame(Objekt,Slot,wert,Wert,_),!.

suche(Objekt,Slot,Ergebnis):-
  frame(Objekt,Slot,wenn_benoetigt,_,_),
  prozedur(Objekt,Slot,Ergebnis),!.

suche(Objekt,Slot,Wert):-
  frame(Objekt,Slot,standard,Wert,_),!.

suche(Objekt,Slot,Wert):-
  frame(Objekt,ist_ein,wert,Vaeter,_),
  suche_vaeter(Vaeter,Slot,Wert),!.

suche(Objekt,Slot,Wert):-
  findall(ObjektSlots,frame(Objekt,ObjektSlots,objekt,_,_),Slotobs),
  suche_objekte(Objekt,Slotobs,Slot,Wert),!.

suche_vaeter([],_,_):-fail.

suche_vaeter([Vater|_],Slot,Wert):-
  suche(Vater,Slot,Wert),!.

suche_vaeter([_|Rest],Slots,Wert):-!,
  suche_vaeter(Rest,Slots,Wert).

suche_objekte(_,[],_,_):-fail.

suche_objekte(Objekt,[ObjektSlot|_],Slot,Wert):-
  frame(Objekt,ObjektSlot,objekt,SlotObjekt,_),
  suche_einzelobjekte(SlotObjekt,Slot,Wert),!.

suche_objekte(Objekt,[_|Rest],Slot,Wert):-!,
  suche_objekte(Objekt,Rest,Slot,Wert).

suche_einzelobjekte([],_,_):-fail.

suche_einzelobjekte([Objekt|_],Slot,Wert):-
  suche(Objekt,Slot,Wert),!.

suche_einzelobjekte([_|Rest],Slot,Wert):-!,
  suche_einzelobjekte(Rest,Slot,Wert).

zeige_objekt(Objekt,XX,YY):-
  retractall(objekt_beschreibung(_)),assert(objekt_beschreibung([])),
  retractall(gezeigte_slots(_)),assert(gezeigte_slots([])),
  zeige_abgeleitetes_objekt(Objekt),!,
  objekt_beschreibung(XX),!,
  gezeigte_slots(YY),!,
  retractall(objekt_beschreibung(_)),
  retractall(gezeigte_slots(_)),!.

zeige_abgeleitetes_objekt(Objekt):-!,
  findall(Slot,frame(Objekt,Slot,wert,_,zeige),WertSlots),
  findall(Slot,frame(Objekt,Slot,objekt,_,zeige),ObjektSlots),
  findall(Slot,frame(Objekt,Slot,standard,_,zeige),StandardSlots),
```

```
findall(Slot,frame(Objekt,Slot,wenn_benoetigt,_,zeige),WennSlots),
zeige_werte(Objekt,wert,Wertslots),!,
zeige_werte(Objekt,wert,Objektslots),!,
zeige_wenn_slots(Objekt,WennSlots),!,
zeige_werte(Objekt,standard,Standardslots),!,
zeige_alle_vaeter_slots(Objekt),!.

zeige_alle_objekt_slots(Objekt,ObjektSlots),!.

zeige_werte(,_,[]):-!.

zeige_werte(Objekt,Facette,[Anfang|Rest]):-
    gezeigte_slots(Slots),
    not(mitglied_der_slotliste(Anfang,Slots)),
    frame(Objekt,Anfang,Facette,Wert,zeige),
    not(Anfang = ist_ein),
    not(Anfang= sohn),
    gebe_slots_raus(Anfang,Wert),!,
    zeige_werte(Objekt,Facette,Rest).

zeige_werte(Objekt,Facette,[_|Rest]):-!,
    zeige_werte(Objekt,Facette,Rest).

zeige_wenn_slots(,[]):-!.

zeige_wenn_slots(Objekt,[Anfang|Rest]):-
    gezeigte_slots(Slots),
    not(mitglied_der_slotliste(Anfang,Slots)),
    frame(ergebnis,Anfang,wert,Ergebnis,_),
    gebe_slots_raus(Anfang,Ergebnis),!,
    zeige_wenn_slots(Objekt,Rest).

zeige_wenn_slots(Objekt,[Anfang|Rest]):-
    gezeigte_slots(Slots),
    not(mitglied_der_slotliste(Anfang,Slots)),
    prozedur(Objekt,Anfang,Ergebnis),
    bound(Ergebnis),
    gebe_slots_raus(Anfang,Ergebnis),!,
    zeige_wenn_slots(Objekt,Rest).

zeige_wenn_slots(Objekt,[_|Rest]):-!,
    zeige_wenn_slots(Objekt,Rest).

zeige_alle_vaeter_slots(Objekt):-!,
    frame(Objekt,ist_ein,_,Vaeter,_),!,
    zeige_vater_slots(Vaeter),!.

zeige_vater_slots([]):-!.

zeige_vater_slots([Anfang|Rest]):-
    zeige_abgeleitetes_objekt(Anfang),!,
    zeige_vater_slots(Rest).

zeige_vater_slots([_|Rest]):-!,
    zeige_vater_slots(Rest).

zeige_alle_objekt_slots(,[]):-!.
```

```
zeige_alle_objekt_slots(Objekt,[Slot|Rest]):-!,
  frame(Objekt,Slot,objekt,Slotobjekte,_),!,
  zeige_objekt_slots(Slotobjekte),!,
  zeige_alle_objekt_slots(Objekt,Rest).

zeige_alle_objekt_slots(Objekt,[_|Rest]):-!,
  zeige_alle_objekt_slots(Objekt,Rest).

zeige_objekt_slots([]):-!.

zeige_objekt_slots([Anfang|Rest]):-
  zeige_abgeleitetes_objekt(Anfang),!,
  zeige_objekt_slots(Rest).

zeige_objekt_slots([_|Rest]):-!,
  zeige_objekt_slots(Rest).

gebe_slots_raus(_,[]).

gebe_slots_raus(Slot,[Ergebnis|Rest]):-!,
  retract(objekt_beschreibung(S)),!,
  assert(objekt_beschreibung([Ergebnis|S])),
  retract(gezeigte_slots(GS)),!,
  assert(gezeigte_slots([Slot|GS])),
  gebe_slots_raus(Slot,Rest).

gebe_end_soehne_raus(Objekt,Vater):-
  frame(Sohn,ist_ein,wert,Vaterliste,_),
  mitglied_der_liste(Vater,Vaterliste),
  retractall(frame(Objekt,sohn,wert,[Vater],_)),
  retractall(frame(Objekt,sohn,wert,[Sohn],_)),
  assert(frame(Objekt,sohn,wert,[Sohn],intern)),
  gebe_end_soehne_raus(Objekt,Sohn),
  fail.

gebe_end_soehne_raus(_,_).

ergaenze_planung(Slot,Wert):-
  retract(frame(ergebnis,Slot,wert,_,_)),!,
  assert(frame(ergebnis,Slot,wert,[Wert],intern)),!.

ergaenze_planung(Slot,Wert):-
  assert(frame(ergebnis,Slot,wert,[Wert],intern)),!.

mitglied_der_liste(Kopf,[Kopf|_]).
mitglied_der_liste(Kopf,[_|Rest]):-
  mitglied_der_liste(Kopf,Rest).

mitglied_der_slotliste(hinweis,_):-!,fail.
mitglied_der_slotliste(Kopf,[Kopf|_]):-!.
mitglied_der_slotliste(Kopf,[_|Rest]):-!,
  mitglied_der_slotliste(Kopf,Rest).
```

## **C Implementierungsbeispiele**

### **C.1 Wissensbasiertes Modell (Prolog) im Bereich Betriebsdateneingabe**

```
/* LAENGE.PRO T.R */
/* leitet aus bekannten, parallelen Vektoren die Länge unbekannter */
/* Vektoren durch Verlängerung her */

domains

database
v(integer, integer, char, char, integer, integer, integer, integer, integer)
ve(integer, integer, integer, char, char, integer, integer, integer, integer)
vl(integer, integer, char, char, integer, integer, integer, integer, real)
vz(integer, integer, integer, char, char, integer, integer, integer, integer, real)
fehler(integer, integer, char, char, integer, integer, integer, integer, real)
frage(integer, integer, char, char, integer, integer, integer, integer)
antwort(integer, integer, char, char, integer, integer, integer, integer, real)
gwh(integer, char, integer, integer, integer, integer)
benutzt(integerliste)

predicates

fehl
los
los1
antworten_laden(string)
entsprechend_loeschen(string)
vektoren_ausrichten(integer)
ausrichtung_vektorbegin(char, integer, integer, integer, integer)
laenge_erfragen(real, integer, integer, char, char, integer, integer, integer, integer)
gesamtlaeenge_ermitteln(realliste, real, real)
dachvektoren_entwickeln(string)
mini_suchen(integer, integer, integer, integer, integer, integer, integer, integer, integer)
ueberpruefung(string)
laenge_bestimmen(char, integer, integer, integer, integer, real, real)
laenge_ermitteln(integer, integer, char, char, integer, integer, integer, integer, real)
mitglied_liste(integer, integerliste)
kopieren

clauses

los:-!,
  retractall(_),
  comline(XXX),
  consult("gwh.pit"),
  antworten_laden(XXX),
  not(vektoren_ausrichten(1)),
  los1,
  kopieren,
  entsprechend_loeschen(XXX),
  consult("gwh.int"),
  not(dachvektoren_entwickeln(XXX)),
  retractall(gwh(_,_,_,_,_,_)),!,
  not(ueberpruefung(XXX)),
  retractall(benutzt(_)),
  save("laenge.pix").
```



kopieren:-

```
retract(vz(_,A,AX,B,BX,C,D,E,F,Laenge)),
assert(vl(A,AX,B,BX,C,D,E,F,Laenge)),
fail.
```

kopieren.

ueberpruefung("Antwort"):-

```
vl(A,AX,B,BX,C,D,E,F,Laenge),
Laenge <= 0,!,
retractall(_),
assertz(fehler(A,AX,B,BX,C,D,E,F,Laenge)),
fail.
```

ueberpruefung("Frage"):-!,fail.

dachvektoren\_entwickeln("Antwort"):-!,

```
gwh(Nr,Seite,C,D,E,F),!,
retractall(gwh(Nr,Seite,C,D,E,F)),
findall(Laenge,vl(Nr,_,Seite,_,_,_,_,Laenge),Liste),
gesamtlaenge_ermitteln(Liste,0,Gesamtlaenge),
assertz(vl(Nr,10,Seite,'D',C,D,E,F,Gesamtlaenge)),
dachvektoren_entwickeln("Antwort").
```

dachvektoren\_entwickeln("Frage"):-fail,!.

gesamtlaenge\_ermitteln([],Ein,Gesamtlaenge):-!,
Gesamtlaenge = Ein.

gesamtlaenge\_ermitteln([Kopf|Rest],Ein,Gesamtlaenge):-!,
Aus = Ein + Kopf,
gesamtlaenge\_ermitteln(Rest,Aus,Gesamtlaenge).

entsprechend\_loeschen("Antwort"):-!,

```
retractall(frage(_,_,_,_,_,_,_,_)),
retractall(antwort(_,_,_,_,_,_,_,_)).
```

entsprechend\_loeschen("Frage"):-!,

```
retractall(vl(_,_,_,_,_,_,_,_)).
```

antworten\_laden("Antwort"):-!,

```
consult("Laengen.Ant").
```

antworten\_laden("Frage"):-!.

vektoren\_ausrichten(Nr):-!,

```
v(A,AX,B,BX,C,D,E,F,_)!,
retractall(v(A,AX,B,BX,C,D,E,F,_)),
mini_suchen(C,D,E,F,CX,DX,EX,FX),
assertz(ve(Nr,A,AX,B,BX,CX,DX,EX,FX)),
NNr = Nr + 1,
vektoren_ausrichten(NNr).
```

mini\_suchen(C,D,E,F,CX,DX,EX,FX):-

```
C = E, D < F,!,
CX = C, DX = D, EX = E, FX = F.
```

mini\_suchen(C,D,E,F,CX,DX,EX,FX):-

```
C = E, D > F,!,
CX = C, DX = F, EX =E, FX = D.

mini_suchen(C,D,E,F,CX,DX,EX,FX):-
  C < E, D = F,!,
  CX = C, DX = D, EX =E, FX = F.

mini_suchen(C,D,E,F,CX,DX,EX,FX):-
  C > E, D = F,!,
  CX = E, DX = D, EX =C, FX = F.

los1:-
  retract(ve(Nr,A,AX,B,BX,C,D,E,F)),!,
  laenge_ermitteln(A,AX,B,BX,C,D,E,F,Laenge),!,
  assertz(vz(Nr,A,AX,B,BX,C,D,E,F,Laenge)),!,
  los1.

los1:-!.

laenge_ermitteln(_,_,_,_C,D,E,F,Laenge):-
  retractall(benutzt(_)), assert(benutzt([])),
  laenge_bestimmen('R',C,D,E,F,0,Laenge),!.

laenge_ermitteln(_,_,_,_C,D,E,F,Laenge):-
  retractall(benutzt(_)), assert(benutzt([])),
  laenge_bestimmen('L',C,D,E,F,0,Laenge),!.

laenge_ermitteln(_,_,_,_C,D,E,F,Laenge):-
  retractall(benutzt(_)), assert(benutzt([])),
  laenge_bestimmen('U',C,D,E,F,0,Laenge),!.

laenge_ermitteln(_,_,_,_C,D,E,F,Laenge):-
  retractall(benutzt(_)), assert(benutzt([])),
  laenge_bestimmen('O',C,D,E,F,0,Laenge),!.

laenge_ermitteln(A,AX,B,BX,C,D,E,F,Laenge):-
  retractall(benutzt(_)), assert(benutzt([])),
  laenge_erfragen(Laenge,A,AX,B,BX,C,D,E,F),!.

laenge_bestimmen(_Anfang,X,Ende,X,Laengeein,Laengeaus):-
  vz(Nr,_,_,_,_Anfang,Y,Ende,Y,Laenge),
  benutzt(Liste),
  not(mitglied_liste(Nr,Liste)),
  Laengeaus = Laengeein + Laenge,!.

/* verlaengern nach rechts */

laenge_bestimmen('R',Anfang,X,Ende,X,Laengeein,Laengeaus):-
  vz(Nr,_,_,_,_Ende,Y,NeuesEnde,Y,Laenge),
  benutzt(Liste),
  not(mitglied_liste(Nr,Liste)),
  retractall(benutzt(_)),assert(benutzt([Nr|Liste])),
  Laengeeinneu = Laengeein - Laenge,
  laenge_bestimmen('R',Anfang,X,NeuesEnde,X,Laengeeinneu,Laengeaus).

/* verlaengern nach links */

laenge_bestimmen('L',Anfang,X,Ende,X,Laengeein,Laengeaus):-
```

```
    vz(Nr,_,_,_,_,NeuerAnfang,Y,Anfang,Y,Laenge),
    benutzt(Liste),
    not(mitglied_liste(Nr,Liste)),
    retractall(benutzt(_)),assert(benutzt([Nr|Liste])),
    Laengeeinneu = Laengeein - Laenge,
    laenge_bestimmen('L',NeuerAnfang,X,Ende,X,Laengeeinneu,Laengeaus).

/* zusammensetzen von vorne */

laenge_bestimmen('R',Anfang,X,Ende,X,Laengeein,Laengeaus):-
    vz(Nr,_,_,_,_,Anfang,Y,NeuerAnfang,Y,Laenge),
    NeuerAnfang < Ende,
    benutzt(Liste),
    not(mitglied_liste(Nr,Liste)),
    retractall(benutzt(_)),assert(benutzt([Nr|Liste])),
    Laengeeinneu = Laengeein + Laenge,
    laenge_bestimmen('R',NeuerAnfang,X,Ende,X,Laengeeinneu,Laengeaus).

/* zusammensetzen von hinten */

laenge_bestimmen('L',Anfang,X,Ende,X,Laengeein,Laengeaus):-
    vz(Nr,_,_,_,_,NeuesEnde,Y,Ende,Y,Laenge),
    NeuesEnde > Anfang,
    benutzt(Liste),
    not(mitglied_liste(Nr,Liste)),
    retractall(benutzt(_)),assert(benutzt([Nr|Liste])),
    Laengeeinneu = Laengeein + Laenge,
    laenge_bestimmen('L',Anfang,X,NeuesEnde,X,Laengeeinneu,Laengeaus).

laenge_bestimmen(_,X,Anfang,X,Ende,Laengeein,Laengeaus):-
    vz(Nr,_,_,_,_,Y,Anfang,Y,Ende,Laenge),
    benutzt(Liste),
    not(mitglied_liste(Nr,Liste)),
    Laengeaus = Laengeein + Laenge,! .

/* verlaengern nach unten */

laenge_bestimmen('U',X,Anfang,X,Ende,Laengeein,Laengeaus):-
    vz(Nr,_,_,_,_,Y,Ende,Y,NeuesEnde,Laenge),
    benutzt(Liste),
    not(mitglied_liste(Nr,Liste)),
    retractall(benutzt(_)),assert(benutzt([Nr|Liste])),
    Laengeeinneu = Laengeein - Laenge,
    laenge_bestimmen('U',X,Anfang,X,NeuesEnde,Laengeeinneu,Laengeaus).

/* verlaengern nach oben */

laenge_bestimmen('O',X,Anfang,X,Ende,Laengeein,Laengeaus):-
    vz(Nr,_,_,_,_,Y,NeuerAnfang,Y,Anfang,Laenge),
    benutzt(Liste),
    not(mitglied_liste(Nr,Liste)),
    retractall(benutzt(_)),assert(benutzt([Nr|Liste])),
    Laengeeinneu = Laengeein - Laenge,
    laenge_bestimmen('O',X,NeuerAnfang,X,Ende,Laengeeinneu,Laengeaus).

/* zusammensetzen von vorne */

laenge_bestimmen('U',X,Anfang,X,Ende,Laengeein,Laengeaus):-
```

```
    vz(Nr,_,_,_,Y,Anfang,Y,NeuerAnfang,Laenge),
    NeuerAnfang < Ende,
    benutzt(Liste),
    not(mitglied_liste(Nr,Liste)),
    retractall(benutzt(_)),assert(benutzt([Nr|Liste])),
    Laengeeinneu = Laengeein + Laenge,
    laenge_bestimmen('U',X,NeuerAnfang,X,Ende,Laengeeinneu,Laengeaus).

/* zusammensetzen von hinten */

laenge_bestimmen('O',X,Anfang,X,Ende,Laengeein,Laengeaus):-
    vz(Nr,_,_,_,Y,NeuesEnde,Y,Ende,Laenge),
    NeuesEnde > Anfang,
    benutzt(Liste),
    not(mitglied_liste(Nr,Liste)),
    retractall(benutzt(_)),assert(benutzt([Nr|Liste])),
    Laengeeinneu = Laengeein + Laenge,
    laenge_bestimmen('O',X,Anfang,X,NeuesEnde,Laengeeinneu,Laengeaus).

ausrichtung_vektorbegin(XX,C,_,E,):-
    C = E,!,
    XX = 'S'.

ausrichtung_vektorbegin(XX,_,D,_,F):-
    D = F,!,
    XX = 'W'.

laenge_erfragen(Laenge,A,Ax,B,Bx,C,D,E,F):-
    comline(XXX),
    XXX = "Frage",!,
    assertz(frage(A,AX,B,BX,C,D,E,F)),
    Laenge = 1.

laenge_erfragen(Laenge,A,Ax,B,Bx,C,D,E,F):-
    comline(XXX),
    XXX = "Antwort",!,
    antwort(A,AX,B,Bx,C,D,E,F,Laenge),!.

mitglied_liste(Kopf,[Kopf|_]):-!.
mitglied_liste(Kopf,[_|Rest]):-!,
    mitglied_liste(Kopf,Rest).

fehl :-
    exit(99).

goal
    trap(los,_,fehl).
```

**C.2 Wissensbasiertes Modell (Produktionsregelansatz) im Bereich Heizenergiebereitstellung**

```
/* KONZEPT.PRO T.R. */
/* Erstellt ein Konzept von der zu planenden Heizungsanlage unter Berück-
/* sichtigung vorhandener Anlagen */

domains

element = x;

oel;gas;kohle;holz_stroh;ht_fern;nt_fern;grundwasser;
oberflaechenwasser;luft;
oel_v;gas_v;kohle_v;holz_stroh_v;ht_fern_v;nt_fern_v;grundwasser_v;
oberflaechenwasser_v;luft_v;

erweiterung;neubau;
monovalent;bivalent;parallel;alternativ;
anlagenbeschreibung;
grundlast;spitzenlast;
verbrennungsanlage;fernwaermenutzung;waermepumpeneinsatz;
low_risk;middle_risk;high_risk;
besonderheit;
schlammerspeicher;direkter_luftheizer;mehrstoffbrenner;
teillasteinsatz;
bestehende_anlage;
eins;zwei;drei;vier;fuenf;
fernwaermenutzung_v;waermepumpeneinsatz_v;verbrennungsanlage_v;
grundlastnutzung;spitzenlastnutzung;
schlammerspeicher_v;direkter_luftheizer_v;mehrstoffbrenner_v;
umruestung;keine_umruestung;
umruestung_gas;umruestung_oel;umruestung_gas_mehr;
umruestung_oel_mehr;
volle_nutzung;teile_nutzung;sicherheits_nutzung;
reserve_nutzung;abbriss;

nichts;naechste;normal;
dimension;einsatz;nutzungsart;anzahl;nummer;art;
schwachlasteinsatz;energiebereitstellungssystem;beschreibung;
anlagenkonzeption;
strategie

elementeliste = element*

knotenvariation =ist_ende;ist_ziel;ist_ziel_hat_moeglichkeit;
                 ist_ziel_besteht_aus;ist_ziel_hat_variante;
                 hat_moeglichkeit;hat_variante;besteht_aus

integerliste = integer*

database - loesungbank
  anlage(elementeliste,integer,integer,integer)
  anzahl_anlagen(integer)
  vorhandene_anlage(integer,real)
  vorhandener_traeger(element,real,real,real,real,real,real,real,real)

database
  versuch_umruesten
  bearbeitete_anlage(integer)

predicates
```

```
knoten(element,knotenvariation,elementeliste,elementeliste)
ziel
unter(integer,string)

include "knoten.pro"
include "fragen.pro"

predicates
  konv(integer,element)
  wenn_traeger_vorhanden(element)
  eingesetzter_traeger(elementeliste,element,element,element)
  bestehender_traeger(elementeliste,element,element,element)
  letzte_anlage(elementeliste,integer)
  beginnen
  fehler(integer)

include "konvert.pro"

clauses

beginnen:-
  retractall(_),
  fragedatei_laden("KONZEPT.TXT"),
  pascal_antworten_laden("ANTWORT.VIR"),
  consult("LOESUNG.VIR",loesungbank),
  not(ziel),
  fertiges_prolog_ziel_rausgeben,!,
  save("x2.vir",loesungbank).

ziel:-
  los([energiebereitstellungssystem],[],Loesung),
  not(anlage(Loesung,0,0,0)),
  assert(anlage(Loesung,0,0,0),loesungbank),fail.

knoten(energiebereitstellungssystem,hat_moeglichkeit,[neubau],_-) :-
  wenn__(schon_anlagen_vorhanden,qq,nein).

knoten(energiebereitstellungssystem,hat_moeglichkeit,[erweiterung],_-) :-
  wenn__(schon_anlagen_vorhanden,qq,ja).

knoten(erweiterung,ist_ziel_hat_moeglichkeit,[anlagenkonzeption,bestehende_anlage],_-).
knoten(neubau,ist_ziel_hat_moeglichkeit,[anlagenkonzeption],_-).

knoten(oel,ist_ziel_hat_moeglichkeit,[verbrennungsanlage],_-).
knoten(gas,ist_ziel_hat_moeglichkeit,[verbrennungsanlage],_-).
knoten(holz_stroh,ist_ziel_hat_moeglichkeit,[verbrennungsanlage,teillasteinsatz],_-).
knoten(kohle,ist_ziel_hat_moeglichkeit,[verbrennungsanlage,teillasteinsatz],_-).
knoten(nt_fern,ist_ziel_hat_moeglichkeit,[fernwaermenutzung],_-).
knoten(ht_fern,ist_ziel_hat_moeglichkeit,[fernwaermenutzung],_-).
knoten(grundwasser,ist_ziel_hat_moeglichkeit,[waermepumpeneinsatz],_-).
knoten(oberflaechenwasser,ist_ziel_hat_moeglichkeit,[waermepumpeneinsatz],_-).
knoten(luft,ist_ziel_hat_moeglichkeit,[waermepumpeneinsatz],_-).
knoten(fernwaermenutzung,ist_ziel,[],_-).
knoten(verbrennungsanlage,ist_ziel,[],_-).
knoten(waermepumpeneinsatz,ist_ziel,[],_-).
```

```
knoten(schwachlasteinsatz,ist_ziel,[],_).
knoten(schlummerspeicher,ist_ziel,[],_).
knoten(mehrstoffbrenner,ist_ziel,[],_).
knoten(direkter_luftheizer,ist_ziel,[],_).

knoten(anlagenkonzeption,hat_moeglichkeit,[anlagenbeschreibung,besonderheit],_).

knoten(anlagenbeschreibung,ist_ziel_hat_moeglichkeit,[monovalent],_).
knoten(anlagenbeschreibung,ist_ziel_hat_moeglichkeit,[bivalent],_).
knoten(bivalent,ist_ziel_hat_moeglichkeit,[alternativ],_).
knoten(bivalent,ist_ziel_hat_moeglichkeit,[parallel],_).

knoten(monovalent,ist_ziel_hat_moeglichkeit,[oel],_-):-
    wenn_traeger_vorhanden(oel).

knoten(monovalent,ist_ziel_hat_moeglichkeit,[gas],_-):-
    wenn_traeger_vorhanden(gas),
    wenn__(maximalbegrenzung_vorhanden,gas,nein).

knoten(monovalent,ist_ziel_hat_moeglichkeit,[kohle],_-):-
    wenn_traeger_vorhanden(kohle).

knoten(monovalent,ist_ziel_hat_moeglichkeit,[ht_fern],_-):-
    wenn_traeger_vorhanden(ht_fern),
    wenn__(ht_abwaermelieferung_sicher,qq,ja).

knoten(monovalent,ist_ziel_hat_moeglichkeit,[nt_fern],_-):-
    wenn_traeger_vorhanden(nt_fern),
    wenn__(nt_abwaermelieferung_sicher,qq,ja).

knoten(alternativ,ist_ziel_hat_moeglichkeit,[grundlast,spitzenlast],_).
knoten(parallel,ist_ziel_hat_moeglichkeit,[grundlast,spitzenlast],_).

knoten(grundlast,ist_ziel_hat_moeglichkeit,[low_risk],Loesung):-
    vorhanden(Loesung,x,x,parallel).
knoten(grundlast,ist_ziel_hat_moeglichkeit,[middle_risk],Loesung):-
    vorhanden(Loesung,x,x,parallel).
knoten(grundlast,ist_ziel_hat_moeglichkeit,[high_risk],Loesung):-
    vorhanden(Loesung,x,x,parallel).

knoten(low_risk,ist_ziel_hat_moeglichkeit,[gas],_-):-
    wenn_traeger_vorhanden(gas),
    wenn__(maximalbegrenzung_vorhanden,gas,ja),
    wenn__nicht(gasliefERVERTRAG_IST,qq,abschaltbar) OR
    wenn_traeger_vorhanden(gas),
    wenn__(maximalbegrenzung_vorhanden,gas,nein).

knoten(low_risk,ist_ziel_hat_moeglichkeit,[oel],_-):-
    wenn_traeger_vorhanden(oel).

knoten(low_risk,ist_ziel_hat_moeglichkeit,[nt_fern],_-):-
    wenn_traeger_vorhanden(nt_fern),
    wenn__(nt_abwaermelieferung_sicher,qq,ja).

knoten(low_risk,ist_ziel_hat_moeglichkeit,[ht_fern],_-):-
    wenn_traeger_vorhanden(ht_fern),
    wenn__(ht_abwaermelieferung_sicher,qq,ja).
```



```
knoten(middle_risk,ist_ziel_hat_moeglichkeit,[kohle],_):-
  wenn_traeger_vorhanden(kohle).

knoten(middle_risk,ist_ziel_hat_moeglichkeit,[ht_fern],_):-
  wenn_traeger_vorhanden(ht_fern),
  wenn__(ht_abwaermelieferung_sicher,qq,nein).

knoten(middle_risk,ist_ziel_hat_moeglichkeit,[nt_fern],_):-
  wenn_traeger_vorhanden(nt_fern),
  wenn__(nt_abwaermelieferung_sicher,qq,nein).

knoten(high_risk,ist_ziel_hat_moeglichkeit,[grundwasser],_):-
  wenn_traeger_vorhanden(grundwasser).

knoten(high_risk,ist_ziel_hat_moeglichkeit,[oberflaechenwasser],_):-
  wenn_traeger_vorhanden(oberflaechenwasser),
  wenn__nicht(oberflwasser_friert_zu,qq,ja).

knoten(high_risk,ist_ziel_hat_moeglichkeit,[holz_stroh],_):-
  wenn_traeger_vorhanden(holz_stroh),
  wenn__(schon_anlagen_vorhanden,qq,ja),
  wenn__zahl(anzahl_anlagen,qq,Zahl),
  X = trunc(Zahl),
  unter(X,Nr),
  wenn__(anlage_nr,Nr,verbrennungsanlage),
  wenn__(eingesetzter_brennstoff_anlage_nr,Nr,holz_stroh).

knoten(grundlast,ist_ziel_hat_moeglichkeit,[gas],Loesung):-
  vorhanden(Loesung,x,x,alternativ),
  wenn_traeger_vorhanden(gas),
  wenn__(maximalbegrenzung_vorhanden,gas,ja),
  wenn__(gasliefervertrag_ist,qq,abschaltbar).

knoten(grundlast,ist_ziel_hat_moeglichkeit,[luft],Loesung):-
  vorhanden(Loesung,x,x,alternativ),
  wenn_traeger_vorhanden(luft).

knoten(grundlast,ist_ziel_hat_moeglichkeit,[oberflaechenwasser],Loesung):-
  vorhanden(Loesung,x,x,alternativ),
  wenn_traeger_vorhanden(oberflaechenwasser),
  wenn__(oberflwasser_friert_zu,qq,ja).

knoten(spitzenlast,ist_ziel_hat_moeglichkeit,[oel],Loesung):-
  vorhanden(Loesung,x,x,parallel),
  wenn_traeger_vorhanden(oel),
  not(vorhanden(Loesung,grundlast,x,oel)).

knoten(spitzenlast,ist_ziel_hat_moeglichkeit,[gas],Loesung):-
  vorhanden(Loesung,x,x,parallel),
  wenn_traeger_vorhanden(gas),
  not(vorhanden(Loesung,grundlast,x,gas)),
  wenn__(maximalbegrenzung_vorhanden,gas,nein).

knoten(spitzenlast,ist_ziel_hat_moeglichkeit,[ht_fern],Loesung):-
  vorhanden(Loesung,x,x,parallel),
  wenn_traeger_vorhanden(ht_fern),
  not(vorhanden(Loesung,grundlast,x,ht_fern)).
```

```
knoten(spitzenlast,ist_ziel_hat_moeglichkeit,[kohle],Loesung):-  
    vorhanden(Loesung,x,x,parallel),  
    wenn_traeger_vorhanden(kohle),  
    not(vorhanden(Loesung,grundlast,x,kohle)),  
    wenn__(schon_anlagen_vorhanden,qq,ja),  
    wenn__zahl(anzahl_anlagen,qq,Zahl),  
    X = trunc(Zahl),  
    unter(X,Nr),  
    wenn__(anlage_nr,Nr,verbrennungsanlage),  
    wenn__(eingesetzter_brennstoff_anlage_nr,Nr,kohle).
```

```
knoten(spitzenlast,ist_ziel_hat_moeglichkeit,[holz_stroh],Loesung):-  
    vorhanden(Loesung,x,x,parallel),  
    wenn_traeger_vorhanden(holz_stroh),  
    not(vorhanden(Loesung,grundlast,x,holz_stroh)),  
    wenn__(schon_anlagen_vorhanden,qq,ja),  
    wenn__zahl(anzahl_anlagen,qq,Zahl),  
    X = trunc(Zahl),  
    unter(X,Nr),  
    wenn__(anlage_nr,Nr,verbrennungsanlage),  
    wenn__(eingesetzter_brennstoff_anlage_nr,Nr,holz_stroh).
```

```
knoten(spitzenlast,ist_ziel_hat_moeglichkeit,[oel],Loesung):-  
    vorhanden(Loesung,x,x,alternativ),  
    wenn_traeger_vorhanden(oel),  
    not(vorhanden(Loesung,grundlast,x,oel)).
```

```
knoten(spitzenlast,ist_ziel_hat_moeglichkeit,[gas],Loesung):-  
    vorhanden(Loesung,x,x,alternativ),  
    wenn_traeger_vorhanden(gas),  
    not(vorhanden(Loesung,grundlast,x,gas)),  
    wenn__(maximalbegrenzung_vorhanden,gas,nein).
```

```
knoten(spitzenlast,ist_ziel_hat_moeglichkeit,[ht_fern],Loesung):-  
    vorhanden(Loesung,x,x,alternativ),  
    wenn_traeger_vorhanden(ht_fern),  
    not(vorhanden(Loesung,grundlast,x,ht_fern)).
```

```
knoten(besonderheit,ist_ziel_hat_variante,[direkter_luftheizer],Loesung):-  
    vorhanden(Loesung,x,x,monovalent),  
    vorhanden(Loesung,x,x,gas),  
    wenn__nicht(hohe_temperaturansprueche,qq,ja).
```

```
knoten(besonderheit,ist_ziel_hat_variante,[direkter_luftheizer],Loesung):-  
    vorhanden(Loesung,x,x,monovalent),  
    vorhanden(Loesung,x,x,oel),  
    wenn__nicht(hohe_temperaturansprueche,qq,ja).
```

```
knoten(besonderheit,ist_ziel_hat_variante,[mehrstoffbrenner],Loesung):-  
    wenn__(kesselhausverzicht,qq,nein),  
    vorhanden(Loesung,x,x,alternativ),  
    vorhanden(Loesung,grundlast,spitzenlast,gas),  
    vorhanden(Loesung,spitzenlast,x,oel).
```

```
knoten(besonderheit,hat_variante,[nichts],Loesung):-  
    wenn__(kesselhausverzicht,qq,ja),  
    not(vorhanden(Loesung,x,x,oel)),  
    not(vorhanden(Loesung,x,x,gas)),
```

```
not(vorhanden(Loesung,x,x,kohle)),
not(vorhanden(Loesung,x,x,holz_stroh)).

knoten(besonderheit,hat_variante,[nichts],_):-
wenn__(kesselhausverzicht,qq,nein).

knoten(teillasteinsatz,hat_moeglichkeit,[schlammerspeicher],Loesung):-
vorhanden(Loesung,x,x,monovalent).

knoten(teillasteinsatz,hat_moeglichkeit,[schwachlasteinsatz],Loesung):-
vorhanden(Loesung,x,x,bivalent).

knoten(bestehende_anlage,ist_ziel_hat_moeglichkeit,[strategie,anzahl,beschreibung],_).

knoten(strategie,ist_ende,[],_):-
retractall(versuch_umruesten).

knoten(strategie,ist_ende,[],_):-
assert(versuch_umruesten).

knoten(anzahl,ist_ende,[],_):-
retractall(bearbeitete_anlage(_)),
wenn__zahl(anzahl_anlagen,qq,Zahl),
retractall(anzahl_anlagen(_)),
X = trunc(Zahl),
assert(anzahl_anlagen(X),loesungbank),
retractall(vorhandene_anlage(_,_)!,!).

knoten(beschreibung,hat_moeglichkeit,[nummer,dimension,art,einsatz,naechste],_).

knoten(naechste,hat_moeglichkeit,[nichts],Loesung):-
anzahl_anlagen(Zahl),
letzte_anlage(Loesung,Nr),
assert(bearbeitete_anlage(Nr)),
(Nr + 1) > Zahl,!.

knoten(naechste,hat_moeglichkeit,[beschreibung],_).

knoten(dimension,ist_ende,[],Loesung):-
letzte_anlage(Loesung,Nr),
str_int(Nnr,Nr),
wenn__zahl(dimension_kw_vorhandener_anlage_nr,Nnr,Dimension),
Dimension_W = Dimension * 1E3,
retractall(vorhandene_anlage(Nr,_)),
assert(vorhandene_anlage(Nr,Dimension_W),loesungbank),!.

knoten(nummer,hat_moeglichkeit,[Zahl],Loesung):-!,
letzte_anlage(Loesung,Nr),
Nnr = Nr + 1,
konv(Nnr,Zahl),!.

knoten(eins,ist_ziel,[],_):-!.
knoten(zwei,ist_ziel,[],_):-!.
knoten(drei,ist_ziel,[],_):-!.
knoten(vier,ist_ziel,[],_):-!.
knoten(fuenf,ist_ziel,[],_):-!.
```

```
knoten(art,hat_moeglichkeit,[verbrennungsanlage_v],Loesung):-  
  letzte_anlage(Loesung,Nr),  
  str_int(Nnr,Nr),  
  wenn__(anlage_nr,Nnr,verbrennungsanlage).
```

```
knoten(art,hat_moeglichkeit,[fernwaermenutzung_v],Loesung):-  
  letzte_anlage(Loesung,Nr),  
  str_int(Nnr,Nr),  
  wenn__(anlage_nr,Nnr,fernwaermenutzung).
```

```
knoten(art,hat_moeglichkeit,[waermepumpeneinsatz_v],Loesung):-  
  letzte_anlage(Loesung,Nr),  
  str_int(Nnr,Nr),  
  wenn__(anlage_nr,Nnr,waermepumpeneinsatz).
```

```
knoten(waermepumpeneinsatz_v,ist_ziel_hat_moeglichkeit,[Traeger],Loesung):-  
  letzte_anlage(Loesung,Nr),  
  str_int(Nnr,Nr),  
  wenn__(eingesetzte_umweltwaerme_anlage_nr,Nnr,Tr),  
  konvert_v(Tr,Traeger).
```

```
knoten(fernwaermenutzung_v,ist_ziel_hat_moeglichkeit,[ht_fern_v],Loesung):-  
  letzte_anlage(Loesung,Nr),  
  str_int(Nnr,Nr),  
  wenn__zahl(temperatur_fernwaerme_anlage_nr,Nnr,Temperatur),  
  Temperatur > 60.
```

```
knoten(fernwaermenutzung_v,ist_ziel_hat_moeglichkeit,[nt_fern_v],Loesung):-  
  letzte_anlage(Loesung,Nr),  
  str_int(Nnr,Nr),  
  wenn__zahl(temperatur_fernwaerme_anlage_nr,Nnr,Temperatur),  
  Temperatur <= 60.
```

```
knoten(verbrennungsanlage_v,ist_ziel_hat_moeglichkeit,[umruestung],Loesung):-  
  versuch_umruesten,  
  letzte_anlage(Loesung,Nr), str_int(Nnr,Nr),  
  not(bearbeitete_anlage(Nr)),  
  wenn__nicht(eingesetzter_brennstoff_anlage_nr,Nnr,kohle),  
  wenn__nicht(eingesetzter_brennstoff_anlage_nr,Nnr,holz_stroh),  
  wenn__(einsatz_direkte_luftheizer_anlage_nr,Nnr,nein),  
  wenn__(einsatz_mehrstoffbrenner_anlage_nr,Nnr,nein),  
  wenn__nicht(vorhandene_anlage_nr,Nnr,nur_bedingt_einsetzbar).
```

```
knoten(verbrennungsanlage_v,ist_ziel_hat_moeglichkeit,[keine_umruestung],Loesung):-  
  letzte_anlage(Loesung,Nr),  
  not(bearbeitete_anlage(Nr)),!.
```

```
knoten(umruestung,ist_ziel_hat_moeglichkeit,[umruestung_gas],_).  
knoten(umruestung,ist_ziel_hat_moeglichkeit,[umruestung_oel],_).  
knoten(umruestung,ist_ziel_hat_moeglichkeit,[umruestung_oel_mehr],_).  
knoten(umruestung,ist_ziel_hat_moeglichkeit,[umruestung_gas_mehr],_).
```

```
knoten(keine_umruestung,hat_moeglichkeit,[Traeger],Loesung):-  
  letzte_anlage(Loesung,Nr),  
  str_int(Nnr,Nr),  
  wenn__(eingesetzter_brennstoff_anlage_nr,Nnr,Tr),
```

konvert\_v(Tr,Traeger).

```
knoten(oel_v,ist_ziel_hat_moeglichkeit,[nichts],Loesung):-  
  letzte_anlage(Loesung,Nr),konv(Nr,Nnnr),str_int(Nnr,Nr),  
  wenn__(einsatz_direkte_luftheizer_anlage_nr,Nnr,nein),  
  vorhanden(Loesung,Nnnr,x,umruestung),  
  not(vorhanden(Loesung,Nnnr,x,oel_v)),!.
```

```
knoten(oel_v,ist_ziel_hat_moeglichkeit,[direkter_luftheizer_v],Loesung):-  
  letzte_anlage(Loesung,Nr),konv(Nr,Nnnr),str_int(Nnr,Nr),  
  wenn__(einsatz_direkte_luftheizer_anlage_nr,Nnr,ja),  
  not(vorhanden(Loesung,Nnnr,x,oel_v)),!.
```

```
knoten(oel_v,ist_ziel_hat_moeglichkeit,[gas_v,mehrstoffbrenner_v],Loesung):-  
  letzte_anlage(Loesung,Nr),konv(Nr,Nnnr),str_int(Nnr,Nr),  
  wenn__(einsatz_direkte_luftheizer_anlage_nr,Nnr,nein),  
  wenn__(einsatz_mehrstoffbrenner_anlage_nr,Nnr,ja),  
  not(vorhanden(Loesung,Nnnr,x,oel_v)),!.
```

```
knoten(oel_v,ist_ziel_hat_moeglichkeit,[nichts],_):-!.
```

```
knoten(gas_v,ist_ziel_hat_moeglichkeit,[nichts],Loesung):-  
  letzte_anlage(Loesung,Nr),konv(Nr,Nnnr),str_int(Nnr,Nr),  
  wenn__(einsatz_direkte_luftheizer_anlage_nr,Nnr,nein),  
  vorhanden(Loesung,Nnnr,x,umruestung),  
  not(vorhanden(Loesung,Nnnr,x,gas_v)),!.
```

```
knoten(gas_v,ist_ziel_hat_moeglichkeit,[direkter_luftheizer_v],Loesung):-  
  letzte_anlage(Loesung,Nr),konv(Nr,Nnnr),str_int(Nnr,Nr),  
  wenn__(einsatz_direkte_luftheizer_anlage_nr,Nnr,ja),  
  not(vorhanden(Loesung,Nnnr,x,gas_v)),!.
```

```
knoten(gas_v,ist_ziel_hat_moeglichkeit,[oel_v,mehrstoffbrenner_v],Loesung):-  
  letzte_anlage(Loesung,Nr),konv(Nr,Nnnr),str_int(Nnr,Nr),  
  wenn__(einsatz_direkte_luftheizer_anlage_nr,Nnr,nein),  
  wenn__(einsatz_mehrstoffbrenner_anlage_nr,Nnr,ja),  
  not(vorhanden(Loesung,Nnnr,x,gas_v)),!.
```

```
knoten(gas_v,ist_ziel_hat_moeglichkeit,[nichts],_):-!.
```

```
knoten(kohle_v,ist_ziel_hat_variante,[schlammerspeicher_v],_).
```

```
knoten(kohle_v,ist_ziel,[],_).
```

```
knoten(holz_stroh_v,ist_ziel_hat_variante,[schlammerspeicher_v],_).
```

```
knoten(holz_stroh_v,ist_ziel,[],_).
```

```
knoten(luft_v,ist_ziel,[],_).
```

```
knoten(oberflaechenwasser_v,ist_ziel,[],_).
```

```
knoten(grundwasser_v,ist_ziel,[],_).
```

```
knoten(ht_fern_v,ist_ziel,[],_).
```

```
knoten(nt_fern_v,ist_ziel,[],_).
```

```
knoten(umruestung_oel_mehr,ist_ziel_hat_moeglichkeit,[oel_v,gas_v,mehrstoffbrenn  
er_v],Loesung):-  
  letzte_anlage(Loesung,Nr), str_int(Nnr,Nr),  
  wenn_traeger_vorhanden(gas),  
  vorhanden(Loesung,x,x,mehrstoffbrenner),  
  wenn__(eingesetzter_brennstoff_anlage_nr,Nnr,oel).
```

```
knoten(umruestung_oel,ist_ziel_hat_moeglichkeit,[gas_v],Loesung):-
```

```
letzte_anlage(Loesung,Nr), str_int(Nnr,Nr),  
wenn_traeger_vorhanden(gas),  
not(vorhanden(Loesung,x,x,mehrstoffbrenner)),  
wenn__(eingesetzter_brennstoff_anlage_nr,Nnr,oel),  
vorhanden(Loesung,x,x,gas),  
not(vorhanden(Loesung,x,x,oel)).
```

```
knoten(umruistung_gas_mehr,ist_ziel_hat_moeglichkeit,[gas_v,oel_v,mehrstoffbrenn  
er_v],Loesung):-
```

```
letzte_anlage(Loesung,Nr), str_int(Nnr,Nr),  
wenn_traeger_vorhanden(oel),  
vorhanden(Loesung,x,x,mehrstoffbrenner),  
wenn__(eingesetzter_brennstoff_anlage_nr,Nnr,gas).
```

```
knoten(umruistung_gas,ist_ziel_hat_moeglichkeit,[oel_v],Loesung):-
```

```
letzte_anlage(Loesung,Nr), str_int(Nnr,Nr),  
wenn_traeger_vorhanden(oel),  
wenn__(eingesetzter_brennstoff_anlage_nr,Nnr,gas),  
not(vorhanden(Loesung,x,x,mehrstoffbrenner)),  
vorhanden(Loesung,x,x,oel),  
not(vorhanden(Loesung,x,x,gas)).
```

```
knoten(direkter_luftheizer_v,ist_ziel,[],Loesung):-
```

```
letzte_anlage(Loesung,Nr), str_int(Nnr,Nr),konv(Nr,Nnnr),  
not(vorhanden(Loesung,Nnnr,x,mehrstoffbrenner_v)),  
vorhanden(Loesung,Nnnr,x,oel_v),  
wenn__(einsatz_direkte_luftheizer_anlage_nr,Nnr,ja),!.
```

```
knoten(direkter_luftheizer_v,ist_ziel,[],Loesung):-
```

```
letzte_anlage(Loesung,Nr), str_int(Nnr,Nr),konv(Nr,Nnnr),  
not(vorhanden(Loesung,Nnnr,x,mehrstoffbrenner_v)),  
vorhanden(Loesung,Nnnr,x,gas_v),  
wenn__(einsatz_direkte_luftheizer_anlage_nr,Nnr,ja),!.
```

```
knoten(mehrstoffbrenner_v,ist_ziel,[],Loesung):-
```

```
letzte_anlage(Loesung,Nr), str_int(Nnr,Nr),konv(Nr,Nnnr),  
wenn__(einsatz_direkte_luftheizer_anlage_nr,Nnr,nein),  
vorhanden(Loesung,Nnnr,x,oel_v),!.
```

```
knoten(mehrstoffbrenner_v,ist_ziel,[],Loesung):-
```

```
letzte_anlage(Loesung,Nr), str_int(Nnr,Nr),konv(Nr,Nnnr),  
wenn__(einsatz_direkte_luftheizer_anlage_nr,Nnr,nein),  
vorhanden(Loesung,Nnnr,x,gas_v),!.
```

```
knoten(schlummerspeicher_v,ist_ziel_hat_moeglichkeit,[nichts],Loesung):-
```

```
letzte_anlage(Loesung,Nr),  
str_int(Nnr,Nr),  
wenn__(schlummerspeicher_einsatz_anlage_nr,Nnr,ja).
```

```
knoten(einsatz,hat_variante,[volle_nutzung],Loesung):-
```

```
letzte_anlage(Loesung,Nr), str_int(Nnr,Nr),  
not(bearbeitete_anlage(Nr)),  
wenn__(vorhandene_anlage_nr,Nnr,voll_funktionsfaehig),  
wenn__(alter_vorhandene_anlage_nr,Nnr,unter_15_jahre).
```

```
knoten(einsatz,hat_variante,[sicherheits_nutzung],Loesung):-
```

```
letzte_anlage(Loesung,Nr), str_int(Nnr,Nr),
```

```
not(bearbeitete_anlage(Nr)),  
wenn__(vorhandene_anlage_nr,Nnr,alt_aber_brauchbar).
```

```
knoten(einsatz,hat_variante,[sicherheits_nutzung],Loesung):-  
  letzte_anlage(Loesung,Nr), str_int(Nnr,Nr),  
  not(bearbeitete_anlage(Nr)),  
  wenn__(vorhandene_anlage_nr,Nnr,voll_funktionsfaehig),  
  wenn__(alter_vorhandene_anlage_nr,Nnr,ueber_15_jahre).
```

```
knoten(einsatz,hat_variante,[reserve_nutzung],Loesung):-  
  letzte_anlage(Loesung,Nr),  
  not(bearbeitete_anlage(Nr)),  
  vorhanden(Loesung,x,x,middle_risk).
```

```
knoten(einsatz,hat_variante,[reserve_nutzung],Loesung):-  
  letzte_anlage(Loesung,Nr),  
  not(bearbeitete_anlage(Nr)),  
  vorhanden(Loesung,x,x,monovalent),  
  eingesetzter_traeger(Loesung,x,x,kohle).
```

```
knoten(einsatz,hat_variante,[reserve_nutzung],Loesung):-  
  letzte_anlage(Loesung,Nr),  
  not(bearbeitete_anlage(Nr)),  
  vorhanden(Loesung,x,x,monovalent),  
  eingesetzter_traeger(Loesung,x,x,ht_fern),  
  wenn__(ht_abwaermelieferung_sicher,qq,nein).
```

```
knoten(einsatz,hat_variante,[reserve_nutzung],Loesung):-  
  letzte_anlage(Loesung,Nr),  
  not(bearbeitete_anlage(Nr)),  
  wenn__(hohes_sicherheitsbestreben_vorhanden,qq,ja).
```

```
knoten(einsatz,hat_variante,[teile_nutzung],Loesung):-  
  letzte_anlage(Loesung,Nr), konv(Nr,Nnr),  
  not(bearbeitete_anlage(Nr)),  
  vorhanden(Loesung,x,x,verbrennungsanlage),  
  vorhanden(Loesung,Nnr,x,verbrennungsanlage_v).
```

```
knoten(einsatz,hat_variante,[teile_nutzung],Loesung):-  
  letzte_anlage(Loesung,Nr),konv(Nr,Nnr),  
  not(bearbeitete_anlage(Nr)),  
  vorhanden(Loesung,x,x,fernwaermenutzung),  
  vorhanden(Loesung,Nnr,x,fernwaermenutzung_v).
```

```
knoten(einsatz,hat_variante,[teile_nutzung],Loesung):-  
  letzte_anlage(Loesung,Nr), konv(Nr,Nnr),  
  not(bearbeitete_anlage(Nr)),  
  vorhanden(Loesung,x,x,waermepumpeneinsatz),  
  vorhanden(Loesung,Nnr,x,waermepumpeneinsatz_v).
```

```
knoten(einsatz,hat_variante,[abriss],Loesung):-  
  letzte_anlage(Loesung,Nr),  
  not(bearbeitete_anlage(Nr)).
```

```
knoten(reserve_nutzung,ist_ziel,[],Loesung):-  
  letzte_anlage(Loesung,Nr),  
  konv(Nr,Nnr),
```

```
not(vorhanden(Loesung,Nnr,x,umruestung)).

knoten(teile_nutzung,ist_ziel_hat_moeglichkeit,[nutzungsart],Loesung):-
  letzte_anlage(Loesung,Nr),
  konv(Nr,Nnr),
  not(vorhanden(Loesung,Nnr,x,umruestung)).

knoten(abriss,ist_ziel,[],Loesung):-
  letzte_anlage(Loesung,Nr),
  konv(Nr,Nnr),
  not(vorhanden(Loesung,Nnr,x,umruestung)).

knoten(sicherheits_nutzung,ist_ziel_hat_moeglichkeit,[nutzungsart],Loesung):-
  letzte_anlage(Loesung,Nr),
  konv(Nr,Nnr),
  not(vorhanden(Loesung,Nnr,x,umruestung)).

knoten(volle_nutzung,ist_ziel_hat_moeglichkeit,[nutzungsart],_).

knoten(nutzungsart,ist_ende,[],Loesung):-
  vorhanden(Loesung,x,x,monovalent),
  letzte_anlage(Loesung,Nr),
  konv(Nr,Nnr),
  bestehender_traeger(Loesung,Nnr,x,Traeger),
  eingesetzter_traeger(Loesung,x,x,Traeger).

knoten(nutzungsart,ist_ende,[],Loesung):-
  vorhanden(Loesung,x,x,bivalent),
  vorhanden(Loesung,x,x,mehrstoffbrenner),
  letzte_anlage(Loesung,Nr),
  konv(Nr,Nnr),
  vorhanden(Loesung,Nnr,x,mehrstoffbrenner_v).

knoten(nutzungsart,hat_moeglichkeit,[grundlastnutzung],Loesung):-
  vorhanden(Loesung,x,x,bivalent),
  not(vorhanden(Loesung,x,x,mehrstoffbrenner)).

knoten(nutzungsart,hat_moeglichkeit,[spitzenlastnutzung],Loesung):-
  vorhanden(Loesung,x,x,bivalent),
  not(vorhanden(Loesung,x,x,mehrstoffbrenner)).

knoten(grundlastnutzung,ist_ziel,[],Loesung):-
  letzte_anlage(Loesung,Nr),
  konv(Nr,Nnr),
  bestehender_traeger(Loesung,Nnr,x,Traeger),
  eingesetzter_traeger(Loesung,x,spitzenlast,Traeger).

knoten(spitzenlastnutzung,ist_ziel,[],Loesung):-
  letzte_anlage(Loesung,Nr),
  konv(Nr,Nnr),
  bestehender_traeger(Loesung,Nnr,x,Traeger),
  eingesetzter_traeger(Loesung,spitzenlast,eins,Traeger).

knoten(nichts,ist_ende,[],_).

eingesetzter_traeger(Loesung,Von,Bis,oel):-
  vorhanden(Loesung,Von,Bis,oel),!.
eingesetzter_traeger(Loesung,Von,Bis,gas):-
```



```
vorhanden(Loesung,Von,Bis,gas),!.
eingesetzter_traeger(Loesung,Von,Bis,kohle):-
  vorhanden(Loesung,Von,Bis,kohle),!.
eingesetzter_traeger(Loesung,Von,Bis,holz_stroh):-
  vorhanden(Loesung,Von,Bis,holz_stroh),!.
eingesetzter_traeger(Loesung,Von,Bis,nt_fern):-
  vorhanden(Loesung,Von,Bis,nt_fern),!.
eingesetzter_traeger(Loesung,Von,Bis,ht_fern):-
  vorhanden(Loesung,Von,Bis,ht_fern),!.
eingesetzter_traeger(Loesung,Von,Bis,grundwasser):-
  vorhanden(Loesung,Von,Bis,grundwasser),!.
eingesetzter_traeger(Loesung,Von,Bis,oberflaechenwasser):-
  vorhanden(Loesung,Von,Bis,oberflaechenwasser),!.
eingesetzter_traeger(Loesung,Von,Bis,luft):-
  vorhanden(Loesung,Von,Bis,luft),!.
```

```
bestehender_traeger(Loesung,Von,Bis,oel):-
  vorhanden(Loesung,Von,Bis,oel_v),!.
bestehender_traeger(Loesung,Von,Bis,gas):-
  vorhanden(Loesung,Von,Bis,gas_v),!.
bestehender_traeger(Loesung,Von,Bis,kohle):-
  vorhanden(Loesung,Von,Bis,kohle_v),!.
bestehender_traeger(Loesung,Von,Bis,holz_stroh):-
  vorhanden(Loesung,Von,Bis,holz_stroh_v),!.
bestehender_traeger(Loesung,Von,Bis,nt_fern):-
  vorhanden(Loesung,Von,Bis,nt_fern_v),!.
bestehender_traeger(Loesung,Von,Bis,ht_fern):-
  vorhanden(Loesung,Von,Bis,ht_fern_v),!.
bestehender_traeger(Loesung,Von,Bis,grundwasser):-
  vorhanden(Loesung,Von,Bis,grundwasser_v),!.
bestehender_traeger(Loesung,Von,Bis,oberflaechenwasser):-
  vorhanden(Loesung,Von,Bis,oberflaechenwasser_v),!.
bestehender_traeger(Loesung,Von,Bis,luft):-
  vorhanden(Loesung,Von,Bis,luft_v),!.
```

```
letzte_anlage(Loesung,Nr):-
  konv(Nr,XX),
  vorhanden(Loesung,x,x,XX),!.
```

```
letzte_anlage(_,0):-!.
```

```
konv(6,x).
konv(5,fuenf).
konv(4,vier).
konv(3,drei).
konv(2,zwei).
konv(1,eins).
```

```
unter(X,Y):-
  str_int(Y,X).
unter(2,"1").
```

```
wenn_traeger_vorhanden(X):-
  vorhandener_traeger(X,'','','','','').
```

```
fehler(0):-!.
```

```
fehler(_):-!,
  exit(99).
```

```
goal
  trap(beginnen,Aus,fehler(Aus)).
```

**C.3 Wissensbasiertes und arithmetisches Modell (Frameansatz) im Bereich Heizenergieausbringung**

```
/* KULTUR.PRO T.R */
/* Programm zum Auslegen von Heizungssystemen im Kulturbereich */

domains

facet = wert; wenn_benoetigt; standard; objekt
facet2 = zeige;intern

slot = x;nichts;sohn;ist_ein;

hinweis; beschreibung; heizungssystem; heizungsart;
gesamtwuermeabgabe_in_k_xw; beruecksichtigung;
beruecksichtigbare_wuermeabgabe; wuermeabgabe_pro_einheit;
gesamtanzahl_einheiten; systemanzahl; anbringung;
beruecksichtigte_wuermeabgabe_anschlussleitungen_in_k_xw; preis;
gesamtinvestition_in_d_xm;
grobe_abschaetzung_gesamtinvestition_in_d_xm;
normwuermeabgabe; normuebertemperatur; heizkoerperexponent; delta_t;
maximale_systemanzahl; minimale_systemanzahl; bezeichnung;
wuermeabgabeort; wuermeausbringung; maximale_vorlauf_temperatur;
standarddruckverlust; gesamteinheiten_ohne_anschluss;
kosten_pro_m2; wuermeverteilung; material; rauhigkeit; rohrsystem;
heizrohrlaenge_in_meter; gesamtanzahl_heizrohre;
gesamtlaenge_verbindungsrohre_in_meter; innendurchmesser;
anzahl_rohrregister; rohranzahl_multiplikator;
anzahl_maximal_hintereinanderschaltbare_heizrohre;
maximale_rohrlaenge_in_meter; anzahl_heizrohre_pro_tisch;
anzahl_heizrohre_pro_binderstiel; konvektorsystem;
einzel_konvektorlaenge_in_meter; gesamtanzahl_konvektoren;
konvektoranzahl_multiplikator; minimale_konvektorlaenge;
maximale_konvektorlaenge; konvektor_anbringungsspielraum;
druckverlust_pro_konvektor_in_mbar; gesamtanzahl_heizer;
planungslaenge; auslegungsvorlauf_temperatur_in_grad_c;
wassermenge_in_m3_pro_stunde;
sinnvoller_innenrohrdurchmesser_anschlussleitung_in_mm;
beruecksichtigte_gesamtlaenge_anschlussleitungen_in_meter;
materialgruppe; rohrabstand_in_mm; betonueberdeckung_in_mm;
bodentemperatur_in_grad_c; bodenverlustfaktor;
gesamtheizrohrlaenge_in_meter; ausleistungsleistung;
verlegungstiefe_in_mm;
angaben_zur_hydraulischen_dimensionierung; kultur;stehwand;
dachraum;boden;nt_spezial; minimaler_rohrabstand; rohr;
direkter_heizer; luftheizer;konvektor;spezial_system;
unter_tisch; auf_tisch; neben_tisch; im_dachraum; an_stehwand;
in_kultur; an_binderstielen;im_boden; in_beton; variabel;
st;pe;pp;ku;al;kunststoff

symbolliste = symbol*
stringliste = string*
slotliste = slot*

file = schreibe;loesungen;konvert;speicher
energieform = nt;mt;ht

database-konvertierungsbank
slotstring(String)
stringslot(String)
```

*database*

*kds(real,real,real)*  
*nt\_neu(real,real)*  
*mt\_neu(real,real)*  
*rest(real)*  
*datei(integer)*  
*uebergabe(integer)*  
*vorhandenes\_system(slot,slot,real)*  
*zaehler(real,real)*  
*frame(string,slot,facet,stringeliste,facet2)*  
*objekt\_beschreibung(stringeliste)*  
*gezeigte\_slots(slotliste)*  
*anfrage(integer,real,real,slot)*  
*laufende\_nummer(integer)*  
*heizungssystem(integer,string,slot,real,slot,real,real,real)*  
*frame\_aktiver\_typ(integer,string)*  
*frame\_aktive\_planung(slot)*  
*frame\_oertl\_begrenzung(slot,slot,real,real),*  
*frame\_oertl\_zwang(slot,slot),*  
*frame\_energie(energieform,real,real)*  
*frame\_theoretische\_vorplanung(real,real,real)*  
*gesamtleistung(real)*  
*stehwandleistung(real)*  
*gesamtenergiemenge(real)*  
*innentemperatur(real)*  
*gewaechshauslaenge(real)*  
*gewaechshausbreite(real)*  
*aussenstehwandlaenge(real)*  
*energieschirm*  
*bhkw*  
*binderabstand(real)*  
*dachtransport*  
*kulturtransport*  
*anzahl\_kappen(real)*  
*unbenutzte\_flaeche(real)*  
*wegeanteil(real)*  
*tischlaenge(real)*  
*tischanzahl(real)*  
*abschlag\_tueren(real)*  
*beetlaenge(real)*  
*anzahl\_binderstiele(real)*  
*brennwertnutzen(real)*  
*bhkw\_therm\_wirkgrad(real)*  
*zinssatz(real)*  
*bodentemperatur(real)*  
*bodenbearbeitungstiefe(real)*  
*unbenutzte\_bodenflaeche(real)*  
*bodenheizung*  
*ueberdeckung(real)*  
*teilung(real)*

*include "frame.pro"*

*predicates*

*los*  
*fehler(integer)*

```
wandle(string,real)
suche_heizung(integer,string,slot,real,slot,real,real,real)
einbindung(string,integer,string,slot,real,slot,real,real,real)
vorlaufemperatur_testen(slot,slot)
anfrage_an_framesystem
oertliche_begrenzung_ermitteln(slot,slot,real,real)
oertlicher_zwang_ueberpruefen(slot,slot)
schon_vorhanden(slot,slot,real)
einsatztest(slot,string,real,slot,slot)
lambda(real,real,real,real)
iteration(real,real,real,real,real,real)
funktion_rohr_druckverlust(real,real,real,real,real,real,real)
hoch(real,real,real)
maximum(real,real,real)
minimum(real,real,real)
konvert(symbol,slot,string)
iteration2(real,real,real,real)
berechne_waermeabgabe(real,real,real,real,real,real,real)

clauses

los:-
  retractall(_),
  comline(Strategie),
  str_int(Strategie,St),
  STNr = trunc(St/10), str_int(DAT,StNr),
  consult("ARBEIT.VIR"),
  consult("HEIZSYS.KUL"),
  consult("HEIZSYS.1"),
  concat(DAT,".nr",Datei),
  openappend(loesungen,Datei),
  anfrage_an_framesystem,
  closefile(loesungen),!.

/* Anfrage Standardstrategie */

anfrage_an_framesystem:-
  comline(Strategie),
  str_int(Strategie,Strat),
  Strat < 20,
  anfrage(LfdNr,POW,_,kultur),
  POW > 0,

/* Backtracking zum naechstmoeeglichen Frame */
  frame("kulturloesung",Gruppe,wert,[System],intern),

/* Backtracking zum naechstmoeeglichen Heizungssystem */
  suche_heizung(Nr,Bezeichnung,Gruppe,Normabgabe,Material,D,Delta_P,Preis),

/* Versuch, ob das System in die Anlage paßt */
  einbindung(System,Nr,Bezeichnung,Gruppe,Normabgabe,Material,D,Delta_P,Preis),

/* Auswertung des Objektbaumes */
  frage_wert("planung",systemanzahl,SA),
  SA > 0,!,
  zeige_objekt("planung",ObjektListe,Slotliste),
  frage_wert("planung",gesamtwaermeabgabe__in__k_xw,GWA),
```

```
frage_wert("planung",systemanzahl,SA),!,
frage_wert("planung",gesamtinvestition_in_d_xm,INV),!,
frage_wert("planung",beruecksichtigbare_waermeabgabe,BWA),!,
frage_wert("planung",auslegungsvorlauftemperatur_in_grad_c,VL),!,

/* gefundene Lösung abspeichern */
frage(planung,anbringung,AnbringungST),!,
konvert(anb,Anbringung,AnbringungST),!,
writedevic(Devi),
writedevic(loesungen),
write("heizsystem(",LfdNr,"",GWA,"",INV,"",SA,"",BWA,"",VL,"",Gruppe,
",",kultur,"",Anbringung,"",'"',Bezeichnung,'"',"",Objektliste,"",Slotliste,")\n"
),
writedevic(Devi),!.

/* Anfrage ökonomische Strategie */
anfrage_an_framesystem:-
comline(Strategie),
str_int(Strategie,Strat),
Strat >= 20,

anfrage(LfdNr,POW,_,kultur), POW > 0,

suche_heizung(Nr,Bezeichnung,Gruppe,Normabgabe,Material,D,Delta_P,Preis),
frame("kulturloesung",Gruppe,wert,[System],intern),
einbindung(System,Nr,Bezeichnung,Gruppe,Normabgabe,Material,D,Delta_P,Preis),

frage_wert("planung",systemanzahl,SA), SA > 0,!,
zeige_objekt("planung",ObjektListe,Slotliste),
frage_wert("planung",gesamtwaermeabgabe_in_k_xw,GWA),
frage_wert("planung",systemanzahl,SA),!,
frage_wert("planung",gesamtinvestition_in_d_xm,INV),!,
frage_wert("planung",beruecksichtigbare_waermeabgabe,BWA),!,
frage_wert("planung",auslegungsvorlauftemperatur_in_grad_c,VL),!,
frage(planung,anbringung,AnbringungST),!,

konvert(anb,Anbringung,AnbringungST),!,
writedevic(Devi),
writedevic(loesungen),

write("heizsystem(",LfdNr,"",GWA,"",INV,"",SA,"",BWA,"",VL,"",Gruppe,"",kultur
',
",",Anbringung,"",'"',Bezeichnung,'"',"",Objektliste,"",Slotliste,")\n"),
writedevic(Devi),!.

anfrage_an_framesystem:-
anfrage(LfdNr,_,_,_),
writedevic(Devi),
writedevic(loesungen),
write("heizsystem(",LfdNr,"",0,0,0,0,0,x,x,x,\"x\",[,[]]\n"),
writedevic(Devi),!.

einbindung(System,_,Bezeichnung,Gruppe,Normabgabe,Material,D,Delta_P,Preis):-
retractall(frame("planung",_,_,_,_)),
retractall(frame(ergebnis,_,_,_,_)),
```

```
assert(frame("planung",ist_ein,wert,[System],intern)),!,

/* Einsatztest 1 */
frage("planung",anbringung,AnSt),!,
konvert(anb,Anbring,AnST),!,
oertliche_begrenzung_ermitteln(Gruppe,Anbring,Min,Max),!,
Min <= Max, Max > 0,!,

/* Einsatztest 2 */
konvert(mat,Material,MaterialST),!,
vorlaufemperatur_testen(Gruppe,Material),!,

/* Einsatztest 3 */
oertlicher_zwang_ueberpruefen(Gruppe,Anbring),!,

/* Einsatztest 4 */
einsatztest(Gruppe,MaterialSt,D,kultur,Anbring),!,

/* weitere Anbindung */

str_real(PreisSt,Preis),
str_real(NormSt,Normabgabe),
str_real(Durchmesser,D),
str_real(Druckverlust,Delta_P),
assert(frame("planung",material,wert,[MaterialST],intern)),
assert(frame("planung",bezeichnung,wert,[Bezeichnung],intern)),
assert(frame("planung",preis,wert,[PreisSt],intern)),
assert(frame("planung",innendurchmesser,wert,[Durchmesser],intern)),
assert(frame("planung",standarddruckverlust,wert,[Druckverlust],intern)),
assert(frame("planung",normwaermeabgabe,wert,[NormSt],intern)),!.

suche_heizung(A,B,C,D,E,F,G,H):-
    heizungssystem(A,B,C,D,E,F,G,H).

schon_vorhanden(Gruppe,Anbringung,Anz):-
    vorhandenes_system(Gruppe,Anbringung,Anz),!.

schon_vorhanden(_,_,0):-!.

oertliche_begrenzung_ermitteln(_,Anbringung,_,_):-
    frame_oertl_begrenzung(x,Anbringung,_,X),
    X <= 0,!,fail.

oertliche_begrenzung_ermitteln(Gruppe,Anbringung,_,_):-
    vorhandenes_system(Gruppe2,Anbringung,_),
    not(Gruppe2 = Gruppe),!,fail.

oertliche_begrenzung_ermitteln(_,unter_tisch,_,_):-
    vorhandenes_system(_,neben_tisch,_),!,fail.

oertliche_begrenzung_ermitteln(_,unter_tisch,_,_):-
    vorhandenes_system(_,auf_tisch,_),!,fail.

oertliche_begrenzung_ermitteln(_,auf_tisch,_,_):-
    vorhandenes_system(_,unter_tisch,_),!,fail.

oertliche_begrenzung_ermitteln(_,auf_tisch,_,_):-
    vorhandenes_system(_,neben_tisch,_),!,fail.
```

```
oertliche_begrenzung_ermitteln(,neben_tisch,_,_):-
    vorhandenes_system(,unter_tisch,_,_,fail.

oertliche_begrenzung_ermitteln(,neben_tisch,_,_):-
    vorhandenes_system(,auf_tisch,_,_,fail.

oertliche_begrenzung_ermitteln(Gruppe,Anbringung,Mini,Maxi):-
    frame_oertl_begrenzung(x,Anbringung,Min,Max),!,
    schon_vorhanden(Gruppe,Anbringung,Anz),
    Mini = Min- Anz,
    Maxi = Max - ANZ,!.

oertliche_begrenzung_ermitteln(Gruppe,Anbringung,Mini,Maxi):-
    frame_oertl_begrenzung(Gruppe,Anbringung,Min,Max),!,
    schon_vorhanden(Gruppe,Anbringung,Anz),
    Mini = Min- Anz,
    Maxi = Max - ANZ,!.

oertliche_begrenzung_ermitteln(Gruppe,Anbringung,Mini,Maxi):-!,
    frage_wert(planung,maximale_systemanzahl,MaSA),
    frage_wert(planung,minimale_systemanzahl,MiSA),
    schon_vorhanden(Gruppe,Anbringung,Anz),
    Maxi = MaSA - Anz,
    Mini = MiSA - Anz, !.

oertlicher_zwang_ueberpruefen(Gruppe,Anbringung):-
    frame_oertl_zwang(Gruppe,Anbringung),!.

oertlicher_zwang_ueberpruefen(,Anbringung):-
    frame_oertl_zwang(,Anbringung),!,fail.

oertlicher_zwang_ueberpruefen(,_,_):-!.

einsatztest(rohr,st,D,_,_,in_kultur):-
    kulturtransport,
    D < 40, !,fail.

einsatztest(rohr,st,D,_,_,in_kultur):-
    D >=40, !,fail.

einsatztest(rohr,Mat,_,_,auf_tisch):-!,
    frage(Mat,materialgruppe,kunststoff),!.

einsatztest(rohr,Mat,_,_,an_binderstielen):-
    frage(Mat,materialgruppe,kunststoff),!,fail.

einsatztest(rohr,_,_,_,_):-!.

einsatztest(konvektor,_,_,_,an_binderstielen):-!.
einsatztest(konvektor,_,_,_,_):-!,fail.

vorlauftemperatur_testen(rohr,Material):-
    konvert(mat,Material,MaterialST),
    frage_wert(MaterialST,maximale_vorlauf_temperatur,VLMax),
    frage_wert("planung",auslegungsvorlauftemperatur__in_grad_c,VL),
    VLMax < VL,!,fail.
```



vorlauftemperatur\_testen(,\_).

```
/*-----*/
/* ----- prozedurale, arithmetische Berechnungsverfahren ----- */
/*-----*/
```

```
wandle(AST,B):-!,
  format(AST,"%-0.2",B),!.
```

```
iteration2(5,D_mm,V_m_pro_s,Wasser_m3_pro_h):-
  Pi = 3.1415927,
  D_mm=1000*sqrt((Wasser_m3_pro_h/900)/(V_m_pro_s*Pi)),!.
```

```
iteration2(Nr,Erg,V_m_pro_s,Wasser_m3_pro_h):-!,
  Nrn = Nr +1,
  Pi = 3.1415927,
  D_mm=1000*sqrt((Wasser_m3_pro_h/900)/(V_m_pro_s*Pi)),
  Oeko_V = 0.562755-0.00741*D_mm+0.000123*D_mm*D_mm-2.685069E-7*D_mm*D_mm*D_mm,
  iteration2(Nrn,Erg,Oeko_V,Wasser_m3_pro_h).
```

```
iteration(,_A,A,5,_,_):-!.
iteration(Re,A,AA,X,Eta,Innen):-!,
  Y = X+1,
  B = 1/(-2*log((2.51/(Re*sqrt(A)))+(Eta/Innen)/3.71))),
  BB = B*B,
  iteration(Re,BB,AA,Y,Eta,Innen).
```

```
lambda(Re,Lambda,_,_):-
  Re < 2320,
  Lambda = 64/Re,!.

```

```
lambda(Re,Lambda,Eta,Innen):-!,
  Re >= 2320,
  iteration(Re,0.03,Lambda,0,Eta,Innen).
```

```
funktion_rohr_druckverlust(P_mbar,Wassermenge_M3_pro_h,
  Innendurch_mm,Rohr_laenge_m,
  Eta_mm,Delta_T_VLRL,VL):-
  Pi = 3.1415927,
  Temp = VL - Delta_T_VLRL/2,
  Vis=1/(556406.7+19689.27*Temp+124.6096*Temp*Temp-0.3783792*Temp*Temp*Temp),
  (GLÜCK 1987)
  V=1000000*(Wassermenge_m3_pro_h/900)/(Innendurch_mm*Innendurch_mm*Pi),
  Re =(V*(Innendurch_mm/1000))/Vis,
  lambda(Re,Lambda,Eta_mm,Innendurch_mm),
  P_mbar = (Lambda*(Rohr_laenge_m/(Innendurch_mm/1000))*500*V*V)/100,
  !.
```

```
berechne_waermeabgabe(Vl,NW,HKE,NUE,DT,Ti,WAB):-
  C = (Vl-Dt-Ti)/(Vl-Ti),
  F1 = ((Vl-Dt/2-Ti)/NUE),
  hoch(F1,HKE,F),
  F2 = 0.584875+1.120863*C-0.71859*C*C,
  minimum(1,F2,F3),
  WAB = F*F3*NW,!.

```

```
prozedur(heizungssystem,auslegungsleistung,[Ergebnis]):-
```

```
anfrage(,POW,_,_),
str_real(Ergebnis,POW),!,
ergaenze_planung(auslegungsleistung,Ergebnis),!.

prozedur(heizungssystem,beruecksichtigbare_waermeabgabe,[Ergebnis]):-
frage_wert(planung,beruecksichtigung,BER),
frage_wert(planung,gesamtwaeermeabgabe__in__k_xw,GWA),
BWA = GWA * 1000* BER,
wandle(Ergebnis,BWA),
ergaenze_planung(beruecksichtigbare_waermeabgabe,Ergebnis),!.

prozedur(heizungssystem,gesamtwaeermeabgabe__in__k_xw,[Ergebnis]):-
frage_wert(planung,waermeabgabe__pro__einheit,WPE),
frage_wert(planung,gesamtanzahl_einheiten,GAE),
frage_wert(planung,beruecksichtigte_waermeabgabe__anschlussleitungen__in__k_xw,
           BWAL),
WAB = (WPE * GAE)/1000,
WA = WAB + BWAL,
wandle(Ergebnis,WA),
ergaenze_planung(gesamtwaeermeabgabe__in__k_xw,Ergebnis),!.

prozedur(heizungssystem,gesamtinvestition__in__d_xm,[Ergebnis]):-
frage_wert(planung,gesamtanzahl_einheiten,GAE),
frage_wert(planung,preis,PR),
Invest = PR * GAE,
format(Ergebnis,"%-0.0",Invest),
ergaenze_planung(gesamtinvestition__in__d_xm,Ergebnis),!.

prozedur(heizungssystem,waermeabgabe__pro__einheit,[Ergebnis]):-
frage_wert(planung,auslegungsvorlauftemperatur__in__grad_c,VL),
frage_wert(planung,normwaermeabgabe,NW),
frage_wert(planung,heizkoerperexponent,HKE),
frage_wert(planung,normuebertemperatur,NUE),
frage_wert(planung,delta_t,DT),
innentemperatur(Ti),
berechne_waermeabgabe(VL,NW,HKE,NUE,DT,Ti,WAB),
str_real(Ergebnis,WAB),
ergaenze_planung(waermeabgabe__pro__einheit,Ergebnis),!.

prozedur(heizungssystem,gesamtanzahl_einheiten,[Ergebnis]):-
pruefe_vater(planung,rohrsystem),
frage_wert(planung,heizrohrlaenge__in__meter,HRL),
frage_wert(planung,gesamtanzahl_heizrohre,GAH),
frage_wert(planung,gesamtlaeenge_verbindungsrohre__in__meter,GVBR),
GE = HRL*GAH+GVBR,
str_real(Ergebnis,GE),
ergaenze_planung(gesamtanzahl_einheiten,Ergebnis),!.

prozedur(heizungssystem,gesamtanzahl_einheiten,[Ergebnis]):-
pruefe_vater(planung,konvektorsystem),
frage_wert(planung,einzel_konvektorlaenge__in__meter,EKL),
frage_wert(planung,gesamtanzahl_konvektoren,GAK),
GE = EKL*GAK,
str_real(Ergebnis,GE),
ergaenze_planung(gesamtanzahl_einheiten,Ergebnis),
!.

prozedur(heizungssystem,gesamteinheiten__ohne__anschluss,[Ergebnis]):-
```

```
pruefe_vater(planung,konvektorsystem),
frage_wert(heizungssystem,auslegungsleistung,POW),
frage_wert(planung,waermeabgabe__pro_einheit,WPE),
frage_wert(planung,einzel_konvektorlaenge__in_meter,EKL),
frage(planung,anbringung,AnbringungSt),
konvert(anb,Anbringung,AnbringungSt),
oertliche_begrenzung_ermitteln(konvektor,Anbringung,MIN,MAX),
frage_wert(konvektorsystem,konvektoranzahl_multiplikator,KAMU),
TH_MAX = trunc(KAMU * MAX),
TH_MIN = trunc(KAMU * MIN+1-1E-10),
DUM1 = POW/WPE/EKL+1-1E-10,
Th_Anz = trunc(DUM1),
minimum(TH_MAX,Th_ANZ,W1),maximum(TH_MIN,W1,ANZ),
str_real(Ergebnis,ANZ),
ergaenze_planung(gesamteinheiten__ohne_anschluss,Ergebnis), !.
```

```
prozedur(heizungssystem,gesamteinheiten__ohne_anschluss,[Ergebnis]):-
pruefe_vater(planung,rohrsystem),
frage_wert(heizungssystem,auslegungsleistung,POW),
frage_wert(planung,waermeabgabe__pro_einheit,WPE),
frage_wert(planung,anzahl_rohrregister,AR),
frage_wert(planung,heizrohrlaenge__in_meter,RL),
frage_wert(planung,gesamtlaege_verbindungsrohre__in_meter,GV),
frage_wert(planung,rohranzahl_multiplikator,RAMU),
frage(planung,anbringung,AnbringungSt),
konvert(anb,Anbringung,AnbringungSt),
oertliche_begrenzung_ermitteln(rohr,Anbringung,MIN,MAX),
DUM1 = (RAMU*MAX)/AR,
TH_MAX = trunc(DUM1),
DUM2 = (RAMU*MIN)/AR+1-1E-10,
TH_MIN = trunc(DUM2),
DUM3 = ((POW/WPE)-GV)/RL/AR+1-1E-10,
TH_Anz = trunc(DUM3),
minimum(TH_ANZ,TH_MAX,W1),maximum(W1,TH_MIN,W2),
ANZ = W2 * AR,
str_real(Ergebnis,ANZ),
ergaenze_planung(gesamteinheiten__ohne_anschluss,Ergebnis),!.
```

```
prozedur(heizungssystem,systemanzahl,[Ergebnis]):-
pruefe_vater(planung,tischsystem),
pruefe_vater(planung,rohrsystem),
frage_wert(planung,gesamtanzahl_heizrohre,GAH),
frage_wert(planung,heizrohrlaenge__in_meter,HRL),
frage_wert(planung,planungslaenge,PL),
gewaechshauslaenge(GL),gewaechshausbreite(GB),
wegeanteil(WA),tischlaenge(TL),tischanzahl(TA),
TB = (GL * GB - ((WA/100)* GL * GB))/TA/TL,
SA = GAH/TA/TB/(PL/HRL),
str_real(Ergebnis,SA),
ergaenze_planung(systemanzahl,Ergebnis), !.
```

```
prozedur(heizungssystem,systemanzahl,[Ergebnis]):-
pruefe_vater(planung,vegetationssystem),
pruefe_vater(planung,rohrsystem),
frage_wert(planung,gesamtanzahl_heizrohre,GAH),
frage_wert(planung,heizrohrlaenge__in_meter,HRL),
frage_wert(planung,planungslaenge,PL),
gewaechshauslaenge(GL),gewaechshausbreite(GB),
```

```
unbenutzte_flaeche(UF),beetlaenge(BL),
BB = ((GL*GB)-((UF/100)*GL*GB))/BL,
SA = GAH/(PL/HRL)/BB,
str_real(Ergebnis,SA),
ergaenze_planung(systemanzahl,Ergebnis), !.

prozedur(heizungssystem,systemanzahl,[Ergebnis]):-
pruefe_vater(planung,binderstielsystem),
pruefe_vater(planung,rohrsystem),
frage_wert(planung,gesamtanzahl_heizrohre,GAH),
frage_wert(planung,heizrohrlaenge__in_meter,HRL),
frage_wert(planung,planungslaenge,PL),
anzahl_binderstiele(AB),
SA = GAH/AB/(PL/HRL),
str_real(Ergebnis,SA),
ergaenze_planung(systemanzahl,Ergebnis), !.

prozedur(heizungssystem,systemanzahl,[Ergebnis]):-
pruefe_vater(planung,binderstielsystem),
pruefe_vater(planung,konvektorsystem),
frage_wert(planung,gesamtanzahl_konvektoren,GAK),
frage_wert(planung,einzel_konvektorlaenge__in_meter,KL),
frage_wert(planung,planungslaenge,PL),
anzahl_binderstiele(AB),
SA = GAK/AB/(PL/KL),
str_real(Ergebnis,SA),
ergaenze_planung(systemanzahl,Ergebnis), !.

prozedur(indirektes_system,auslegungsvorlaufemperatur__in_grad_c,[Ergebnis]):-
anfrage(,_,VL,_),
wandle(Ergebnis,VL),!.

prozedur(indirektes_system,wassermenge__in_m3__pro_stunde,[Ergebnis]):-
frage_wert(planung,gesamtwaermeabgabe__in_k_xw,GWA),
frage_wert(planung,delta_t,Delta_t_VLRL),
Waermemenge_W = GWA * 1000,
Wassermenge_m3_pro_h = (Waermemenge_W/(1.163 * Delta_t_VLRL))/1000,
wandle(Ergebnis,Wassermenge_m3_pro_h),!.

prozedur(indirektes_system,sinnvoller_innenrohrdurchmesser_anschlussleitung__in_mm,
[Ergebnis]):-
frage_wert(planung,gesamteinheiten_ohne_anschluss,GAOH),
frage_wert(planung,waermeabgabe__pro_einheit,WPE),
Waermemenge_W = GAOH * WPE,
frage_wert(planung,delta_t,Delta_t_VLRL),
WM = (Waermemenge_W/(1.163 * Delta_t_VLRL))/1000,
iteration2(1,Innendurch_mm,1,WM),!,
IDS = round(Innendurch_mm/10)*10,
wandle(Ergebnis,IDS),!.

prozedur(indirektes_system,beruecksichtigte_gesamtlaenge_anschlussleitungen__in_meter
,
[Ergebnis]):-
gewaechshausbreite(GHB),
GAL = GHB * 2,
wandle(Ergebnis,GAL),!.

prozedur(indirektes_system,beruecksichtigte_waermeabgabe_anschlussleitungen__in_k_xw
```

```
,[Ergebnis]):-  
  frage_wert(planung,auslegungsvorlaufemperatur_in_grad_c,VL),  
  frage_wert(planung,beruecksichtigte_gesamtlaenge_anschlussleitungen_in_meter,  
    GVB),  
  frage_wert(planung,sinnvoller_innenrohrdurchmesser_anschlussleitung_in_mm,  
    SRD),  
  frage_wert(rohrsystem,heizkoerperexponent,HKE),  
  frage_wert(planung,normuebertemperatur,NUE),  
  frage_wert(planung,delta_t,DT),  
  innentemperatur(Ti),  
  NW = 18 + 2.405 * SRD,  
  berechne_waermeabgabe(Vl,NW,HKE,NUE,DT,Ti,WAB),  
  BWAB = (WAB * GVB)/1000,  
  wandle(Ergebnis,BWAB),!.
```

```
prozedur(konvektorsystem,gesamtanzahl_konvektoren,[Ergebnis]):-  
  frage_wert(heizungssystem,auslegungsleistung,POW),  
  frage_wert(planung,waermeabgabe_pro_einheit,WPE),  
  frage_wert(planung,einzel_konvektorlaenge_in_meter,EKL),  
  frage_wert(planung,beruecksichtigte_waermeabgabe_anschlussleitungen_in_k_xw,  
    BWAL),  
  frage(planung,anbringung,AnbringungSt),  
  konvert(anb,Anbringung,AnbringungSt),  
  oertliche_begrenzung_ermitteln(rohr,Anbringung,MIN,MAX),  
  frage_wert(konvektorsystem,konvektoranzahl_multiplikator,KAMU),  
  TH_MAX = trunc(KAMU * MAX),  
  TH_MIN = trunc(KAMU * MIN+1-1E-10),  
  DUM = (POW-(BWAL*1000))/WPE/EKL+1-1E-10,  
  Th_Anz = trunc(DUM),  
  minimum(TH_MAX,Th_ANZ,W1),maximum(TH_MIN,W1,ANZ),  
  wandle(Ergebnis,ANZ),  
  ergaenze_planung(gesamtanzahl_konvektoren,Ergebnis), !.
```

```
prozedur(konvektorsystem,konvektoranzahl_multiplikator,[Ergebnis]):-  
  pruefe_vater(planung,binderstielsystem),  
  frage_wert(planung,einzel_konvektorlaenge_in_meter,EKL),  
  frage_wert(planung,planungslaenge,PL),  
  anzahl_binderstiele(ABST),  
  KAMU = ABST * (PL/EKL),  
  str_real(Ergebnis,KAMU),!.
```

```
prozedur(konvektorsystem,einzel_konvektorlaenge_in_meter,[Ergebnis]):-  
  frage_wert(planung,minimale_konvektorlaenge,KMi),  
  frage_wert(planung,maximale_konvektorlaenge,KMa),  
  frage_wert(planung,konvektor_anbringungsspielraum,KAS),  
  binderabstand(BSDA),  
  BSD = BSDA - KAS,  
  minimum(BSD,KMa,Min),  
  maximum(Min,KMi,Max),  
  wandle(Ergebnis,Max),!.
```

```
prozedur(konvektorsystem,druckverlust_pro_konvektor_in_mbar,[Ergebnis]):-  
  frage_wert(planung,standarddruckverlust,SDV),  
  frage_wert(konvektorsystem,einzel_konvektorlaenge_in_meter,EKM),  
  hoch(EKM,3,HOCH),  
  P = HOCH * SDV,  
  format(Ergebnis,"%-0.4",P),!.
```

```
prozedur(rohrsystem,sinnvoller_innenrohrdurchmesser_anschlussleitung__in__mm,[Ergebnis]):-
```

```
frage_wert(indirektes_system,sinnvoller_innenrohrdurchmesser_anschlussleitung__in__mm
,
          SRI),
frage_wert(planung,innendurchmesser,IDAkt),
maximum(IDAkt,SRI,ID),
wandle(Ergebnis,ID),!.
```

```
prozedur(rohrsystem,gesamtanzahl_heizrohre,[Ergebnis]):-
```

```
frage_wert(heizungssystem,auslegungsleistung,POW),
frage_wert(planung,waermeabgabe__pro__einheit,WPE),
frage_wert(planung,anzahl_rohrregister,AR),
frage_wert(planung,heizrohrlaenge__in__meter,RL),
frage_wert(planung,gesamtlaenge_verbindungsrohre__in__meter,GV),
frage_wert(planung,rohranzahl_multiplikator,RAMU),
frage(planung,anbringung,AnbringungSt),
konvert(anb,Anbringung,AnbringungSt),
oertliche_begrenzung_ermitteln(rohr,Anbringung,MIN,MAX),
DUM1 = (RAMU*MAX)/AR,
TH_MAX = trunc(DUM1),
DUM2 = (RAMU*MIN)/AR+1-1E-10,
TH_MIN = trunc(DUM2),
DUM3 = ((POW/WPE)-GV)/RL/AR+1-1E-10,
TH_Anz = trunc(DUM3),
minimum(TH_ANZ,TH_MAX,W1),maximum(W1,TH_MIN,W2),
ANZ = W2 * AR,
wandle(Ergebnis,ANZ),
ergaenze_planung(gesamtanzahl_heizrohre,Ergebnis),
!.
```

```
prozedur(rohrsystem,rohranzahl_multiplikator,[Ergebnis]):-
```

```
pruefe_vater(planung,tischsystem),
frage_wert(planung,heizrohrlaenge__in__meter,HRL),
frage_wert(planung,planungslaenge,PL),
gewaechshauslaenge(GL),gewaechshausbreite(GB),
wegeanteil(WA),tischlaenge(TL),tischanzahl(TA),
TB = (GL * GB - ((WA/100)* GL * GB))/TA/TL,
RAMU = TB * TA * (PL/HRL),
str_real(Ergebnis,RAMU),!.
```

```
prozedur(rohrsystem,rohranzahl_multiplikator,[Ergebnis]):-
```

```
pruefe_vater(planung,binderstielsystem),
frage_wert(planung,heizrohrlaenge__in__meter,HRL),
frage_wert(planung,planungslaenge,PL),
anzahl_binderstiele(AB),
RAMU = (PL/HRL)*AB,
str_real(Ergebnis,RAMU),!.
```

```
prozedur(rohrsystem,rohranzahl_multiplikator,[Ergebnis]):-
```

```
pruefe_vater(planung,vegetationssystem),
frage_wert(planung,heizrohrlaenge__in__meter,HRL),
frage_wert(planung,planungslaenge,PL),
gewaechshauslaenge(GL),gewaechshausbreite(GB),
unbenutzte_flaeche(UF),beetlaenge(BL),
BB = ((GL*GB)-((UF/100)*GL*GB))/BL,
```

```
RAMU = (PL/HRL)*BB,  
str_real(Ergebnis, RAMU), !.
```

```
prozedur(rohrsystem, anzahl_rohrregister, [Ergebnis]) :-  
  pruefe_vater(planung, tischsystem),  
  tischanzahl(TA),  
  wandle(Ergebnis, TA), !.
```

```
prozedur(rohrsystem, anzahl_rohrregister, [Ergebnis]) :-  
  pruefe_vater(planung, binderstielsystem),  
  anzahl_binderstiele(BA),  
  wandle(Ergebnis, BA), !.
```

```
prozedur(rohrsystem, gesamtlaenge_verbindungsrohre__in_meter, [Ergebnis]) :-  
  pruefe_vater(planung, binderstielsystem), !,  
  anzahl_binderstiele(ANZ),  
  GLV = ANZ * 3,  
  wandle(Ergebnis, GLV), !.
```

```
prozedur(rohrsystem, angaben__zur__hydraulischen_dimensionierung, [Ergebnis]) :-  
  frage_wert(indirektes_system, wassermenge__in__m3__pro_stunde, WM),  
  frage_wert(planung, innendurchmesser, ID),  
  frage_wert(planung, delta_t, DT),  
  frage_wert(planung, auslegungsvorlauftemperatur__in_grad_c, VL),  
  frage_wert(planung, maximale_rohrlaenge__in_meter, MRL),  
  frage_wert(rohrsystem, heizrohrlaenge__in_meter, HRLM),  
  frage_wert(rohrsystem, gesamtanzahl_einheiten, GE),  
  frage("planung", material, MAT),  
  frage_wert(MAT, rauhigkeit, Eta_mm),  
  minimum(MRL, GE, MAXLRO),  
  RWM = WM/GE,  
  funktion_rohr_druckverlust(P_mbar, RWM, ID, 1, Eta_mm, DT, VL),  
  hoch(MaxLRO, 3, MaxL),  
  hoch(HRLM, 3, MHRL3),  
  Delta_P = P_mbar*MaxL,  
  Delta_P_1 = P_mbar*MHRL3,  
  format(ST1, "%* Maximal zulässige Länge eines Heizregisters : %-0.0 [m].*  
    Druckverlust bei maximaler Registerlänge von %-0.1 meter: %-0.2 [mbar]*  
    Druckverlust bei Registerlänge von %-0.1 meter: %-0.2 [mbar]", MaxLRO,  
    MAXLRO, Delta_P, HRLM, Delta_P_1),  
  format(ST2, "%* (Druckverluste ohne Formstücke und Druckabgleich)."),  
  ST3 = "%* Der Gesamtdruckverlust steigt mit der dritten Potenz der Länge*  
    des Heizregisters !",  
  ST4 = "%* Werden mehrere Rohre parallel geschaltet, auf evtl. notwendigen*  
    Druckabgleich achten !",  
  ST5 = "%* Der Druckabgleich kann durch den Einbau von Blenden, Schiebern *  
    oder Rohrverengungen erfolgen.",  
  format(Ergebnis, "% % % % %", St1, St2, St3, St4, St5), !.
```

```
prozedur(rohrsystem, gesamtlaenge_verbindungsrohre__in_meter, [Ergebnis]) :-  
  pruefe_vater(planung, tischsystem), !,  
  gewaechshauslaenge(GL), gewaechshausbreite(GB),  
  wegeanteil(WA), tischlaenge(TL), tischanzahl(TA),  
  TB = ((GL*GB)-((WA/100)*GL*GB))/TA/TL,  
  GVR = TB * 2 * TA,
```

```
wandle(Ergebnis,GVR),!.

prozedur(rohrsystem,gesamtlaenge_verbindungsrohre__in_meter,[Ergebnis]):-
  pruefe_vater(planung,vegetationssystem),!,
  gewaechshauslaenge(GL),gewaechshausbreite(GB),
  unbenutzte_flaeche(UF),beetlaenge(BL),
  BB = ((GL*GB)-((UF/100)*GL*GB))/BL,
  GVR = BB * 2,
  wandle(Ergebnis,GVR),!.

prozedur(rohrsystem,maximale_rohrlaenge__in_meter,[Ergebnis]):-
  frage_wert(planung,delta_t,DT),
  frage_wert(planung,innendurchmesser,ID),
  frage_wert(planung,waermeabgabe__pro_einheit,WAE),
  MRL = exp((-0.31952+2.669024*log(ID)+0.041648*DT)*ln(10))/WAE,
  wandle(Ergebnis,MRL),!.

prozedur(rohrsystem,heizrohrlaenge__in_meter,[Ergebnis]):-
  frage_wert("planung",planungslaenge,PL),
  frage_wert("planung",maximale_rohrlaenge__in_meter,MR),
  DUM = PL/MR+1-1E-10,
  P = trunc(DUM),
  HRL = PL/P,
  wandle(Ergebnis,HRL),!.

prozedur(tischsystem,planungslaenge,[Ergebnis]):-
  tischlaenge(TL),
  str_real(Ergebnis,TL),!.

prozedur(vegetationssystem,planungslaenge,[Ergebnis]):-
  beetlaenge(BL),
  str_real(Ergebnis,BL),!.

prozedur(binderstielsystem,planungslaenge,[Ergebnis]):-
  gewaechshauslaenge(SL),
  UBSL = 0,
  L = SL - UBSL,
  str_real(Ergebnis,L),!.

prozedur(tischsystem,anzahl_heizrohre__pro_tisch,[Ergebnis]):-
  frage_wert(planung,gesamtanzahl_heizrohre,GHR),
  tischanzahl(AT),
  DUM = GHR/AT,
  RPT = round(DUM),
  wandle(Ergebnis,RPT),!.

prozedur(an_binderstiel_rohr,anzahl_heizrohre__pro_binderstiel,[Ergebnis]):-
  frage_wert(planung,gesamtanzahl_heizrohre,GHR),
  anzahl_binderstiele(AT),
  DUM = GHR/AT,
  RPT = round(DUM),
  wandle(Ergebnis,RPT),!.

konvert(ort,kultur,"kultur").
konvert(ort,stehwand,"stehwand").
konvert(ort,dachraum,"dachraum").
konvert(ort,boden,"boden").
```



```
konvert(mat,st,"st").
konvert(mat,al,"al").
konvert(mat,pe,"pe").
konvert(mat,pp,"pp").
konvert(gru,rohr,"rohr").
konvert(gru,direkter_heizer,"direkter_heizer").
konvert(gru,luftheizer,"luftheizer").
konvert(gru,konvektor,"konvektor").
konvert(anb,unter_tisch,"unter_tisch").
konvert(anb,neben_tisch,"neben_tisch").
konvert(anb,auf_tisch,"auf_tisch").
konvert(anb,an_binderstielen,"an_binderstielen").
konvert(anb,unter_tisch,"unter_tisch").
konvert(anb,in_kultur,"in_kultur").
```

```
hoch(A,B,C):-!,
    C = exp(B * ln(A)).
```

```
maximum(A,B,A):-
    A >= B,!.
maximum(_,B,B):-!.
minimum(A,B,A):-
    A <= B,!.
minimum(_,B,B):-!.
```

```
fehler(AUS):-
    Aus <= 70,
    exit(99),!.
```

```
fehler(Aus):-!,
    exit(Aus).
```

```
goal
    trap(los,Aus,fehler(Aus)).
```

#### **C.4 Arithmetisches Modell (Simulation) im Bereich Wärmeverbrauchsrechnung**

```
{ WAERME.PAS T.R. }
{ Programm zur Berechnung des Wärmeverbrauchs von Gewächshäusern }

unit waerme;

{$F+,O+}

interface
uses metagraf, typen, allgemei, dbanken, ausgaben;

procedure klima_einlesen(klimazeiger : klimazeigertyp; try_datei : string;
                          witterung : st17);

procedure k_werte_berechnen(klimazeiger: klimazeigertyp;
                             betriebzeiger: betriebzeigertyp);

procedure W_pro_K_aussenflaechen_berechnen(klimazeiger: klimazeigertyp;
                                             betriebzeiger : betriebzeigertyp;
                                             var Aussenflaechenverbrauch:R15);

procedure abschirmungsdaten_ermitteln(betriebzeiger: betriebzeigertyp;
                                        var abschirmung : R15);

procedure norm_leistungen_rechnen(betriebzeiger : betriebzeigertyp;
                                   planungsbasis : basiszeigertyp;
                                   waermeverbr_aussenflaechen,
                                   waermeverbr_abschirmung : R15);

procedure waerme_simulieren(W_pro_K_aussen : R15;
                             betriebzeiger : betriebzeigertyp;
                             regelungszeiger: regelungszeigertyp;
                             klimazeiger:klimazeigertyp);

function heizungsanlagenauslastung(vollbenutzungsstunden:real):string;

implementation

function heizungsanlagenauslastung(vollbenutzungsstunden:real):string;
begin
  if vollbenutzungsstunden > 2500 then
    heizungsanlagenauslastung := 'sehr hohe'
  else if vollbenutzungsstunden > 1500 then
    heizungsanlagenauslastung := 'normale'
  else if vollbenutzungsstunden > 500 then
    heizungsanlagenauslastung := 'niedrige'
  else
    heizungsanlagenauslastung := 'sehr geringe';
end;

procedure K_wert_nach_vw_korrigieren(var K : real; K_4, vw : real);
begin
  K := K_4 + (0.0463 * K_4 * Vw) - (0.1852 * K_4);
end;

procedure energieschirmwirkung (var E : real; K_strich,E_6: real;Abdichtung: st8);
begin
  if Abdichtung = 'gut' then E := E_6 * (K_Strich/ 6.8)
  else if Abdichtung = 'mittel' then E := E_6 * (K_Strich/ 11.05)
```

```
else if Abdichtung = 'schlecht' then E := E_6 * (K_Strich/ 29.43);
end;
```

```
function delta_T_durch_Speicherung(Tagmittel,
                                   Nachtmittel, Maximale_Differenz : real;
                                   ti:shortint;
                                   toh :integer;
                                   Gewaechshausspeicherwert:real) : shortint;
```

```
var tspv : real;
begin
  tspv := ((Tagmittel-Nachtmittel)/Maximale_Differenz) *
          Gewaechshausspeicherwert;
  if tspv < 0 then tspv := 0;
  if tspv >= 25 then
    delta_T_durch_Speicherung := round(tspv)
  else if ((ti-toh) <= tspv) then
    delta_T_durch_Speicherung := round(tspv)
  else if ((ti-toh) > tspv) and ((ti-toh) < 25 ) then
    delta_T_durch_Speicherung := round((tspv*(ti-toh-25))/(tspv-25))
  else
    delta_T_durch_Speicherung := 0;
end;
```

```
procedure klima_einlesen(klimazeiger : klimazeigertyp; try_datei : string;
                          witterung : st17);
```

```
var
df : file;
i : integer;
begin
  arbeitszeile('Ich lade die Klimadaten');
  IF pos('eigene', witterung) > 0 then
    begin
      assign(df, userpfad+'USERKLIM.DAT');
      reset(df,6);
      blockread(df,klimazeiger^.temp,1460);
      blockread(df,klimazeiger^.einstrahl,2920);
      blockread(df,klimazeiger^.wind,1);
      close(df);
    end
  else
    begin
      assign(df,try_datei);
      reset(df,6);
      blockread(df,klimazeiger^.temp,1460);
      blockread(df,klimazeiger^.einstrahl,2920);
      blockread(df,klimazeiger^.wind,1);
      close(df);
      if pos('streng',witterung) > 0 then
        begin
          for i := 1 to 1896 do
            Klimazeiger^.Temp[i] := klimazeiger^.temp[i] -4;
          for i := 8520 to 8760 do
            Klimazeiger^.Temp[i] := klimazeiger^.temp[i] -4;
          end
        end
      else if pos('mild',witterung) > 0 then
        begin
          for i := 1 to 1896 do
            Klimazeiger^.Temp[i] := klimazeiger^.temp[i] +4;
          end
        end
      end
    end
  end;
```

```
    for i := 8520 to 8760 do                { Winter vom 21.12 bis 21.3 }
        Klimazeiger^.Temp[i]              := klimazeiger^.temp[i] +4;
    end;
end;
end;

procedure k_werte_berechnen(klimazeiger: klimazeigertyp;
                             betriebzeiger: betriebzeigertyp);
var i : integer; k1,k2,k_4,e1,e2 : real;
    x : vektortyp;
    dd,ee : file of vektortyp;gg : string;
begin
    assign(dd, userpfad+betriebzeiger^.simulationskennung+'.vek');
    assign(ee, 'zw');
    rewrite(ee);
    reset(dd);
    while not(eof(dd)) do
        begin
            e1 := 0; e2 := 0; k1 := 0; k2 := 0;
            read(dd,x);

            anfrage(x.bedachung1,tabelle[bedach], 'Material', 'K_Wert',gg,k1);

            if leer(x.einsparung1) <> '' then
                anfrage(x.einsparung1,tabelle[bedach], 'Isolierung', 'Einsparung',gg,e1);

            if leer(x.bedachung2) <> '' then
                begin
                    anfrage(x.bedachung2,tabelle[bedach], 'Material', 'K_Wert',gg,k2);
                    if leer(x.einsparung2) <> '' then
                        anfrage(x.einsparung2,tabelle[bedach], 'Isolierung', 'Einsparung',gg,e2);
                end;
            end;

            K_4 := (k1*((100-e1)/100)*(x.prozent1/100)) +
                   (k2*((100-e2)/100)*(x.prozent2/100));

            if x.seg2 = 10 then
                K_wert_nach_vw_korrigieren(x.k_wert,K_4,klimazeiger^.wind)
            else K_wert_nach_vw_korrigieren(x.k_wert,K_4,0);

            write(ee,x);
        end;

        close(dd); close(ee);
        erase(dd); rename(ee,userpfad+betriebzeiger^.simulationskennung+'.vek');
    end;

procedure W_pro_K_aussenflaechen_berechnen(klimazeiger: klimazeigertyp;
                                             betriebzeiger : betriebzeigertyp;
                                             var Aussenflaechenverbrauch:R15);
var i : integer;
    x : vektortyp;
    dd,ee : file of vektortyp;
begin
    assign(dd, userpfad+betriebzeiger^.simulationskennung+'.vek');
    reset(dd);
    for i := 1 to 15 do aussenflaechenverbrauch[i] := 0;
```

```
while not(eof(dd)) do
  begin
    read(dd,x);
    if x.seg2 = 10 then
      aussenflaechenverbrauch[x.seg1-10]:=
        aussenflaechenverbrauch[x.seg1-10] + (x.k_wert* x.reale_flaeche);
    end;
  close(dd);
end;

procedure abschirmungsdaten_ermitteln(betriebzeiger: betriebzeigertyp;
                                       var abschirmung : R15);
var I : integer;
    zuschlag_meter : real; {m}
    dachabzug : array[1..15] of real; { W/K/lfd m !}
    ee : file of vektortyp;
    vektor : vektortyp;
    gg : string;
begin
  assign(ee,userpfad+betriebzeiger^.simulationskennung+'.VEK');
  reset(ee);

  for i := 1 to 15 do
    begin
      abschirmung[i] := 0;
      dachabzug[i] := 0;
    end;

  anfrage('Zuschlag Abschirmung Stehwand (vom Dach)',
          tabelle[technik], 'Name', 'Wert', gg, zuschlag_meter);

  while not(eof(ee)) do
    begin
      read(ee,vektor);
      if vektor.seite2 = 'D' then
        begin
          if
            (vektor.reale_flaeche/betriebzeiger^.abt[vektor.seg1-10].stehwandlaenge)
            > (zuschlag_meter * 2) then
            dachabzug[vektor.seg1-10] := vektor.k_wert * zuschlag_meter
          else
            dachabzug[vektor.seg1-10] :=
              ((vektor.reale_flaeche/betriebzeiger^.abt[vektor.seg1-10].stehwandlaenge)/2)
            * vektor.k_wert;
          end;
        end;
      end;

  reset(ee);
  while not(eof(ee)) do
    begin
      read(ee,vektor);
      if (vektor.seg2 = 10) and (vektor.seite2 <> 'D') then
        abschirmung[vektor.seg1-10] :=
          vektor.k_wert * vektor.reale_flaeche +
          dachabzug[vektor.seg1-10] * vektor.reale_laenge +
          abschirmung[vektor.seg1-10];
      end;
    close(ee);
```

```
end;

procedure reale_innentemperaturen(betriebzeiger : betriebzeigertyp;
                                   regelungszeiger : regelungszeigertyp;
                                   klimazeiger : klimazeigertyp);
type shinttyp = array[1..8760] of shortint;
   inttyp = array[1..8760] of integer;
   singtyp = array[1..8760] of real;
   booltyp = array [1..8760] of boolean;

var doh,dt : file;
    toh : ^inttyp;
    ti : ^shinttyp;
    gg : string;
    ok : boolean;
    i,z,Tag,woche,tsoll,obergrenze,summe,restzeit : integer;
    Tagstunden,Nachtstunden : byte;
begin
new(doh);
new(toh);
new(ti);
assign(dt,arbeitsbereich+'ti_real.int');
rewrite(dt,8760);
assign(doh,arbeitsbereich+'tioh.int');
reset(doh,17520);

for i := 1 to betriebzeiger^.abt_anzahl do
begin
str(i,gg);
arbeitszeile('Ich berechne die Innentemperaturen Segment '+gg);

seek(doh,i-1);
blockread(doh,toh^,1);

if regelungszeiger^.abt[i].ti_steuerung = 'konst' then
for z := 1 to 8760 do
begin
woche := trunc((z/168)+1); if woche = 53 then woche := 52;

if toh^[z] <= regelungszeiger^.abt[i].ti_tag[woche] then
ti^[z] := regelungszeiger^.abt[i].ti_tag[woche]
else if toh^[z] <= regelungszeiger^.abt[i].ti_lueftung[woche] then
ti^[z] := toh^[z]
else if regelungszeiger^.abt[i].ti_lueftung[woche] <=
klimazeiger^.temp[z] then
ti^[z] := klimazeiger^.temp[z]
else
ti^[z] := regelungszeiger^.abt[i].ti_lueftung[woche];
end;

if regelungszeiger^.abt[i].ti_steuerung = 'absen' then
for z := 1 to 8760 do
begin
woche := trunc((z/168)+1); if woche = 53 then woche := 52;

if klimazeiger^.einstrahl[z] > 0 then
begin
if toh^[z] <= regelungszeiger^.abt[i].ti_tag[woche] then
ti^[z] := regelungszeiger^.abt[i].ti_tag[woche]
```

```
else if toh^[z] <= regelungszeiger^.abt[i].ti_lueftung[woche] then
  ti^[z] := toh^[z]
else if regelungszeiger^.abt[i].ti_lueftung[woche] <=
  klimazeiger^.temp[z] then
  ti^[z] := klimazeiger^.temp[z]
else
  ti^[z] := regelungszeiger^.abt[i].ti_lueftung[woche];
end
else
begin
if toh^[z] <= regelungszeiger^.abt[i].ti_nacht[woche] then
  ti^[z] := regelungszeiger^.abt[i].ti_nacht[woche]
else if toh^[z] <= regelungszeiger^.abt[i].ti_lueftung[woche] then
  ti^[z] := toh^[z]
else if regelungszeiger^.abt[i].ti_lueftung[woche] <=
  klimazeiger^.temp[z] then
  ti^[z] := klimazeiger^.temp[z]
else
  ti^[z] := regelungszeiger^.abt[i].ti_lueftung[woche];
end;
end;

if regelungszeiger^.abt[i].ti_steuerung = 'summe' then
begin
  summe := 0; restzeit := 24;

  for z := 2 to 8760 do
  begin
    woche := trunc((z/168)+1); if woche = 53 then woche := 52;

    if (klimazeiger^.einstrahl[z-1] = 0) and
      (klimazeiger^.einstrahl[z] > 0) then
      begin
        Summe := 0;
        Restzeit := 24;
      end;

    if restzeit = 0 then restzeit := 1;
    tsoll := round((regelungszeiger^.abt[i].ti_summe[woche] -
      summe)/Restzeit);

    if tsoll < regelungszeiger^.abt[i].ti_min[woche] then
      tsoll := regelungszeiger^.abt[i].ti_min[woche];

    obergrenze := regelungszeiger^.abt[i].ti_lueftung[woche];
    if obergrenze < klimazeiger^.temp[z] then
      obergrenze := klimazeiger^.temp[z];

    if toh^[z] <= tsoll then
      begin
        ti^[z] := tsoll;
        if ti^[z] > obergrenze then ti^[z] := obergrenze;
      end
    else if toh^[z] <= regelungszeiger^.abt[i].ti_lueftung[woche] then
      ti^[z] := toh^[z]
    else
      ti^[z] := obergrenze;
```



```
        summe := summe + ti^[z];
        restzeit := restzeit-1;

        end;
        ti^[1] := ti^[2];
    end;

    blockwrite(dt,ti^,1);
end;
dispose(ti);
dispose(toh);
close(dt);
close(doh);
end;

procedure heizbedarf(W_pro_K_aussen : R15;
                    betriebzeiger   : betriebzeigertyp;
                    regelungszeiger : regelungszeigertyp;
                    klimazeiger     : klimazeigertyp);
type shinttyp = array[1..8760] of shortint;
   inttyp = array[1..8760] of integer;
   singtyp = array[1..8760] of real;
   booltyp = array [1..8760] of boolean;
var doh,dt,dh,ds,dl,di : file;
    licht_an,schirm_zu : ^booltyp;
    toh : ^inttyp;
    ti : ^shinttyp;
    heiz : ^singtyp;
    gg : string;
    einspar,schirmeinsparung,waerme_verlust,Lichteintrag,
    Tagmittel,Nachtmittel,maximale_differenz,lichtverlust : real;
    ok : boolean;
    i,z,Tag,woche,tsoll,obergrenze,summe,restzeit,schirmnummer,lichtnummer,
    Tagsumme,Nachtsumme: integer;
    Tagstunden,Nachtstunden : byte;
    Gewaechshausspeicherwert : real;
    Speicherabschlag : array[1..365] of shortint;
begin

    new(toh);
    new(ti);
    new(heiz);
    new(schirm_zu);
    new(licht_an);

    assign (dh,'heil.int');
    rewrite(dh,52560);

    assign(dt,arbeitsbereich+'ti_real.int');
    reset(dt,8760);

    assign(doh,arbeitsbereich+'tioh.int');
    reset(doh,17520);

    {----- Basiswert fuer Speicherung laden -----}

    anfrage('maximale Tag-Nacht-Speicherwirkung      ',
            tabelle[technik],'Name','Wert',gg,Gewaechshausspeicherwert);
```

```
{-----}

schirmnummer := 0;
lichtnummer := 0;

for i := 1 to betriebzeiger^.abt_anzahl do
begin
  str(i,gg);
  arbeitszeile('Ich berechne die Innentemperaturen Segment '+gg);

{-----Basisdaten für Energieschirm laden -----}

  if betriebzeiger^.abt[i].schirm_vorhanden then
  begin

    anfrage(betriebzeiger^.abt[i].schirm_material,
            tabelle[einricht], 'Material', 'Einsparung', gg, einspar);

    energieschirmwirkung (schirm_einsparung,
                          W_pro_K_aussen[i]/
                          betriebzeiger^.abt[i].aussenflaeche, einspar,
                          betriebzeiger^.abt[i].schirm_abdichtung);

    waerme_verlust := W_pro_K_aussen[i] *
                      ((100-schirmeinsparung)/100);

    schirmnummer := schirmnummer + 1;
    assign(ds, arbeitsbereich+'schirm.int');
    reset(ds, 8760);
    seek(ds, schirmnummer - 1);
    blockread(ds, schirm_zu^, 1);
    close(ds);
  end
  else
  for z := 1 to 8760 do schirm_zu^[z] := false;

{----- Basisdaten für Belichtung laden-----}

  if betriebzeiger^.abt[i].licht_vorhanden then
  begin
    anfrage('Energieverluste bei Belichtung          ',
            tabelle[technik], 'Name', 'Wert', gg, lichtverlust);

    lichteintrag := betriebzeiger^.abt[i].licht_install_leistung *
                    betriebzeiger^.abt[i].giebellaenge *
                    betriebzeiger^.abt[i].stehwandlaenge *
                    ((100 - lichtverlust)/100);

    lichtnummer := lichtnummer + 1;
    assign(dl, arbeitsbereich+'licht.int');
    reset (dl, 8760);
    seek(dl, lichtnummer-1);
    blockread(dl, licht_an^, 1);
    close(dl);
  end
  else
  for z := 1 to 8760 do licht_an^[z] := false;
```



```
begin
  Nachtmittel := Nachtsumme/Nachtstunden;
  Tagstunden := 1;
  Tagsumme := toh^[z];
  Speicherabschlag[Tag] :=
    delta_T_durch_Speicherung(Tagmittel,Nachtmittel,Maximale_Differenz,
                               ti^[z],toh^[z],Gewachshausspeicherwert);
  inc(Tag);
end
else if (klimazeiger^.einstrahl[z-1] > 0) and
        (klimazeiger^.einstrahl[z] = 0) then
{ Nachtanfang }
begin
  Tagmittel := Tagsumme/Tagstunden;
  Nachtsumme := toh^[z];
  Nachtstunden := 1;
end
else if klimazeiger^.einstrahl[z] <= 0 then
begin
  Nachtsumme := Nachtsumme + toh^[z];
  inc(Nachtstunden);
end
else
begin
  Tagsumme := Tagsumme + toh^[z];
  inc(Tagstunden);
end;
until Tag = 365;
Speicherabschlag[365] := Speicherabschlag[364];

arbeitszeile('Ich berechne Heizbedarf Segment '+gg);

for z := 1 to 8760 do
begin
  if klimazeiger^.einstrahl[z] = 0 then
begin
  if schirm_zu^[z] then
    heiz^[z] := (ti^[z] - toh^[z] - speicherabschlag[trunc(z/24 +
    0.999)]) * waerme_verlust
  else
    heiz^[z] := (ti^[z] - toh^[z] - speicherabschlag[trunc(z/24 +
    0.999)]) * W_pro_K_aussen[i];
  end
end
else
begin
  if schirm_zu^[z] then
    heiz^[z] := (ti^[z] - toh^[z] ) * waerme_verlust
  else
    heiz^[z] := (ti^[z] - toh^[z] )
      * W_pro_K_aussen[i];
  end;
end

  if licht_an^[z] then heiz^[z] := heiz^[z] - lichteintrag;
end;

  blockwrite(dh,heiz^,1);
end;
dispose(heiz);
```

```
dispose(ti);
dispose(toh);
dispose(schirm_zu);
dispose(licht_an);
close(dh);
close(dt);
close(doh);
end;

procedure innenwandbilanz_beruecksichtigen(Betriebzeiger:betriebszeigertyp);
type
singtyp = array[1..8760] of real;

var df,di,dh : file;
    z,i : integer;
    heiz,i_wand_bilanz : ^singtyp;
    gg : string;
begin
new(heiz);
new(i_wand_bilanz);

assign (dh,'heil.int');
reset(dh,52560);

assign (df,'heiz.int');
rewrite(df,52560);

assign(di,'iwand.int');
reset(di,52560);

for i := 1 to betriebszeiger^.abt_anzahl do
begin
str(i,gg);
arbeitszeile('Ich berechne die Innenwandbilanzierung Segment '+gg);

seek(dh,i-1);
blockread(dh,heiz^,1);
seek(di,i-1);
blockread(di,i_wand_bilanz^,1);

for z := 1 to 8760 do
begin
heiz^[z] := heiz^[z] - i_wand_bilanz^[z];
if heiz^[z] < 0 then heiz^[z] := 0;
end;

blockwrite(df,heiz^,1);

end;

dispose(heiz);
dispose(i_wand_bilanz);
close(df);
close(di);
close(dh);
erase(dh);
end;
```



```
schirmnummer := schirmnummer +1;
assign(ds,arbeitsbereich+'schirm.int');
reset(ds,8760);
seek(ds,schirmnummer -1);
blockread(ds,schirm_zu^,1);
close(ds);
end
else
  for z := 1 to 8760 do schirm_zu^[z] := false;
{=Innentemperatur ohne Heizleistung im Segment berechnen ==}

  for z := 1 to 8760 do
    begin
      woche := trunc((z/168)+1); if woche = 53 then woche := 52;

      if schirm_zu^[z] then
        tioh^[z] := round(((klimazeiger^.einstrahl[z] *
                           D_mal_eta_mal_m2)/
                           waerme_verlust)+ klimazeiger^.temp[z])
      else
        tioh^[z] := round(((klimazeiger^.einstrahl[z] * D_mal_eta_mal_m2)
                           /W_pro_K_aussen[i]) + klimazeiger^.temp[z]);

        if (tioh^[z] >= regelungszeiger^.abt[i].ti_lueftung[woche]) and
           (regelungszeiger^.abt[i].ti_lueftung[woche] >= klimazeiger^.temp[z])
        then tioh^[z] := regelungszeiger^.abt[i].ti_lueftung[woche]
        else if (tioh^[z] < regelungszeiger^.abt[i].ti_lueftung[woche]) and
           (regelungszeiger^.abt[i].ti_lueftung[woche] >= klimazeiger^.temp[z])
        then begin end
        else
          tioh^[z] := klimazeiger^.temp[z];
        end;

      seek(doh,i-1);
      blockwrite(doh,tioh^,1);

    end; {Ende for I:= 1 to abt_anzahl}

    dispose(schirm_zu);
    dispose(tioh);
    close(doh);
  end;

procedure innenwaende_bilanzieren(W_pro_K_aussen:R15;betriebzeiger:
betriebszeigertyp);

type
temptyp = array[1..8760] of shortint;
heiztyp = array[1..8760] of real;
var templ,temp2 : ^temptyp;
    i_wand_bilanz : ^heiztyp;
    i : integer; x: vektortyp;
    ee,gg : file; dd : file of vektortyp;    JJJ : string;
```

```
begin
  arbeitszeile('Ich berücksichtige die Wärmebilanz zwischen den einzelnen
  Segmenten');

  new(temp1);
  new(temp2);
  new(i_wand_bilanz);

  assign(dd,userpfad+ betriebszeiger^.simulationskennung+'.vek');
  reset(dd);
  assign(ee,arbeitsbereich+'ti_real.int');
  reset(ee,8760);

  {----- Bilanz = 0 setzen-----}

  for i := 1 to 8760 do i_wand_bilanz^[i] := 0;
  assign(gg,'iwand.int');
  rewrite(gg,52560);
  for i := 1 to betriebszeiger^.abt_anzahl do
    blockwrite(gg,i_wand_bilanz^,1);

  {-----alle Seiten durchgehen-----}

  while not(eof(dd)) do
    begin
      read(dd,x);
      if x.seg2 <> 10 then
        begin
          seek(ee,x.seg1-11);
          blockread(ee,temp1^,1);
          seek(ee,x.seg2-11);
          blockread(ee,temp2^,1);
          seek(gg,x.seg1-11);
          blockread(gg,i_wand_bilanz^,1);

          for i := 1 to 8760 do
            i_wand_bilanz^[i] := i_wand_bilanz^[i] +
              (x.k_wert * x.reale_flaeche *(temp2^[i]-temp1^[i]));

          seek(gg,x.seg1-11);
          blockwrite(gg,i_wand_bilanz^,1);
        end;
      end;

      {-----alle Seiten durchgehen fertig-----}

      dispose(i_wand_bilanz);
      dispose(temp1);
      dispose(temp2);
      close(dd);
      close(ee);
      close(gg);
    end;

  procedure
  belichtungsbedarf_nutzbare_bhkw_abwaerme_ermitteln(betriebszeiger:betriebszeigertyp);
    { ausgabe in belicht.int bhkw.int}
  type singtyp = array[1..8760] of real;
```



```
    booltyp = array[1..8760] of boolean;
var dbe,db,dl : file;ok : boolean;
    i,z : integer;
    th_wirk_grad,el_wirk_grad,energie,energie_bhkw : real;
    bhkw,lichteintrag : ^singtyp;
    licht_an : ^booltyp;gg : string;
begin

new(licht_an);
new(lichteintrag);
new(bhkw);
assign(dl,arbeitsbereich+'licht.int');
reset(dl,8760);
assign(dbe,'belicht.int');
rewrite(dbe,52560);
assign(db,arbeitsbereich+'bhkw.int');
rewrite(db,52560);

if betriebzeiger^.bhkw_einsatz then
begin
    arbeitszeile('Ich berechne mögliche Abwaermenutzung durch BHKW');

    for z:= 1 to 8760 do bhkw^[z] := 0;

    anfrage('BHKW ',tabelle[einricht],'intern','BHKW_Elek',gg,el_wirk_grad);
    anfrage('BHKW ',tabelle[einricht],'intern','BHKW_Therm',gg,th_wirk_grad);
end;

for i := 1 to betriebzeiger^.abt_anzahl do
begin
    if betriebzeiger^.abt[i].licht_vorhanden then
begin
        str(i,gg);
        arbeitszeile('Ich berechne Energieeintrag durch Belichtung in Segment
'+gg);
        Energie := betriebzeiger^.abt[i].licht_install_leistung *
            betriebzeiger^.abt[i].giebellaenge *
            betriebzeiger^.abt[i].stehwandlaenge;

        blockread(dl,licht_an^,1);

        if betriebzeiger^.bhkw_einsatz then
begin
            energie_bhkw := (energie*th_wirk_grad)/el_wirk_grad;
            for z := 1 to 8760 do
begin
                if licht_an^[z] then
begin
                    lichteintrag^[z] := energie;
                    bhkw^[z] := bhkw^[z] + energie_bhkw;
                end
                else lichteintrag^[z] := 0;
            end;
        end
    else
        for z := 1 to 8760 do
begin
            if licht_an^[z] then lichteintrag^[z] := energie
```

```
        else lichteintrag^[z] := 0;
        end;

        blockwrite(dbe,lichteintrag^,1);

        end;
    end;

    blockwrite(db,bhkw^,1);
    dispose(licht_an);
    dispose(lichteintrag);
    dispose(bhkw);
    close(db);
    close(dl);
    close(dbe);

end;

procedure licht_ermitteln(betriebzeiger: betriebzeigertyp;
                        regelung : regelungszeigertyp;
                        klimazeiger: klimazeigertyp);
type ggtyp = array[1..8760] of boolean;
var schirmnummer,i, z,woche,uhrzeit : integer;
    par_faktor,d,durchlass,faktor,summe,kunstlicht_eintrag,
    bel_wirk_grad,str_ausb : real;
    tag,ok,an,aus : boolean;
    lichtzeiger,schirm_zu : ^ggtyp;
    dd,ds : file; gg :string;
begin
    new (lichtzeiger);
    new (schirm_zu);
    assign (dd,arbeitsbereich+'licht.int');
    rewrite (dd, 8760);

    schirmnummer := 0;

    anfrage('Beleuchtungswirkungsgrad           ',
            tabelle[technik],'Name','Wert',gg,bel_wirk_grad);
    anfrage('Umwandlungsfaktor: Globalstr. => PAR ',
            tabelle[technik],'Name','Wert',gg,PAR_faktor);
    anfrage('Strahlungsdurchlaessigkeit Gewaechshaus ',
            tabelle[technik],'Name','Wert',gg,D);

    for i := 1 to betriebzeiger^.abt_anzahl do
        begin
            str(i,gg);
            if betriebzeiger^.abt[i].licht_vorhanden then
                begin
                    anfrage(betriebzeiger^.abt[i].licht_lampenart,
                            tabelle[einricht],'Lampenart','PAR_Ausb',gg,Str_ausb);
                    str(i,gg);
                    arbeitszeile('Ich berechne Energieeintrag durch Belichtung in Segment
'+gg);
                end
            end;
        end;
    end;

    {-----Basisdaten für Energieschirm laden ----}

    if betriebzeiger^.abt[i].schirm_vorhanden then
        begin
```

```
anfrage(betriebzeiger^.abt[i].schirm_material,
        tabelle[einricht], 'Material', 'Durchlass', gg, Durchlass);
schirmnummer := schirmnummer + 1;
assign(ds, arbeitsbereich + 'schirm.int');
reset(ds, 8760);
seek(ds, schirmnummer - 1);
blockread(ds, schirm_zu^, 1);
close(ds);
end;

an := false; Summe := 1000000; tag := false;

kunstlicht_eintrag := betriebzeiger^.abt[i].licht_install_leistung *
                    bel_wirk_grad * str_ausb;

if regelung^.abt[i].licht_steuerung = 'Uhrzeit' then
  for z := 1 to 8760 do
    begin
      woche := trunc((z/168)+1); if woche = 53 then woche := 52;
      uhrzeit := z - (trunc(z/24)*24);
      if uhrzeit = regelung^.abt[i].licht_an[woche] then an := true;
      if uhrzeit = regelung^.abt[i].licht_aus[woche] then an :=
        false;
      lichtzeiger^[z] := an;
    end;
  end;

if regelung^.abt[i].licht_steuerung = 'Lichtsumme' then
  for z := 1 to 8760 do
    begin
      woche := trunc((z/168)+1); if woche = 53 then woche := 52;

      if betriebzeiger^.abt[i].schirm_vorhanden and schirm_zu^[z] then
        Faktor := PAR_faktor * (D/100) * (Durchlass/100)
      else Faktor := Par_faktor * (D/100);

      if tag and (klimazeiger^.einstrahl[z] < 0.001) then Tag := false;
      if (tag = false) and (klimazeiger^.einstrahl[z] > 0.001) then
        begin
          Tag := true;
          Summe := 0;
        end;

      if tag then
        begin
          lichtzeiger^[z] := false;
          summe := summe + klimazeiger^.einstrahl[z] * Faktor;
        end
      else if summe < regelung^.abt[i].licht_lichtsumme [woche] then
        begin
          lichtzeiger^[z] := true;
          Summe := Summe + Kunstlicht_Eintrag;
        end
      else
        lichtzeiger^[z] := false;
      end;
    end;
  end;
  blockwrite(dd, lichtzeiger^, 1);
end;
end;
end;
```

```
close(dd);
dispose(lichtzeiger);
dispose(schirm_zu);
end;

procedure energieschirm_beruecksichtigen(W_pro_K_aussen:R15;betriebzeiger:
betriebszeigertyp);
type heiztyp = array[1..8760] of real;
schirmtyp =array[1..8760] of boolean;
var heiz : ^heiztyp; gg : string;
schirmzeiger :^schirmtyp;
ok : boolean; i,z,schirmnummer : integer;
dd,ee : file; einspar,e : real;
begin
new(heiz);
new(schirmzeiger);
assign(dd,betriebzeiger^.simulationskennung+'.hei');
reset(dd,52560);
assign(ee,arbeitsbereich+'schirm.int');
reset(ee,8760);

schirmnummer := 0;
for i := 1 to betriebszeiger^.abt_anzahl do
begin
if betriebszeiger^.abt[i].schirm_vorhanden then
begin
str(i,gg);
anfrage(betriebszeiger^.abt[i].schirm_material,
tabelle[einricht], 'Material', 'Einsparung',gg,einspar);

energieschirmwirkung (E,W_pro_K_aussen[i],
einspar,betriebszeiger^.abt[i].schirm_abdichtung);
e := (100 -e) / 100;
schirmnummer := schirmnummer +1;
seek(dd,i-1);
seek(ee,schirmnummer -1);
blockread(dd,heiz^,1);
blockread(ee,schirmzeiger^,1);

for z := 1 to 8760 do
if schirmzeiger^[z] then heiz^[z] := heiz^[z] * e;

seek(dd,i-1);
blockwrite(dd,heiz^,1);
end;
end;
dispose(heiz);
dispose(schirmzeiger);
close(dd);
close(ee);
end;

procedure schirm_ermitteln(betriebszeiger: betriebszeigertyp;
regelung : regelungszeigertyp;
klimazeiger: klimazeigertyp);
type ggtyp = array[1..8760] of boolean;
```



```
                                klimazeiger:klimazeigertyp);
begin
  Schirm_ermitteln(betriebzeiger,regelungszeiger,klimazeiger);
    { ausgabe in schirm.int }
  licht_ermitteln(betriebzeiger,regelungszeiger,klimazeiger);
    { ausgabe in licht.int }
  innentemperatur_ohne_Heizung(W_pro_K_aussen,klimazeiger,betriebzeiger,
                                regelungszeiger);
    { ausgabe in ti_oh.int }
  reale_innentemperaturen(betriebzeiger,regelungszeiger,klimazeiger);
    { ausgabe in ti_real.int }
  innenwaende_bilanzieren(W_pro_K_aussen,betriebzeiger);
    { ausgabe in iwand.int }
  heizbedarf(W_pro_K_aussen,betriebzeiger,regelungszeiger,klimazeiger);
    { ausgabe in heil.int }
  innenwandbilanz_beruecksichtigen(betriebzeiger);
    { ausgabe in heiz.int }
  belichtungsbedarf_nutzbare_bhkw_abwaerme_ermitteln(betriebzeiger);
    { ausgabe in belicht.int bhkw.int}
end;

end.
```

```
{ TYPEN.PAS T.R. }
{ Typenvereinbarungen für WAERME.PAS }

{$F+,O+}

unit typen;

interface

uses metagraf;

const Maxseg = 15;

type

st2 = string[2];st3 = string[3];st4 = string[4];st5 = string[5];
st6 = string[6];st8 = string[8];st10 = string[10];st12 = string[12];
st13 = string[13];st15 = string[15];st17 = string[17];st19 = string[19];
st20 = string[20];st25 = string[25];st27 = string[27];st30 = string[30];
st31 = string[31];st35 = string[35];st50 = string[50];st60 = string[60];
st80 = string[80];st150 = string[150];

int15typ = array[1..15] of integer;
R15 = array[1..15] of real;

segmenttyp = array [1..15] of abt_angaben;

const datenpfad : st25 = 'Daten\';
const documentepfad :st25 = 'Docu\';
const userpfad :st25 = 'User\';

type

tagesrealtyp = array[1..365] of real;
tagesrealzeigertyp = ^tagesrealtyp;

jahresintegertyp = array[1..8760] of integer;
jahresintegerzeigertyp = ^jahresIntegertyp;

jahresshortinttyp = array[1..8760] of shortint;
jahresshortintzeigertyp = ^jahresshortinttyp;

jahressingletyp = array [1..8760] of real;
jahressinglezeigertyp = ^jahressingletyp;

shortint52feld = array [1..52]of shortint;
int52feld = array [1..52] of integer;

klimarecord = record
temp : array [1..8760] of shortint;{ Auß.temp. stündlich TRY (grd C) }
einstrahl : array [1..8760] of integer; { diff.+ dir. Einstrahl. TRY (W/m2)}
wind : real; { Jahresdurch. Windgeschw. (m/s) }
end;

klimazeigertyp = ^klimarecord;

vektortyp = record
```

```
seg1,                { innen liegendes Segment + 10 }
seg2 : integer;      { Außensegment +10, wenn=10, dann Außenwand }
seite1,              { s,S = Stehwand, g,G = Giebelwand, D = Dach }
seite2 : char;       { Seitel = Innensegment, Seite2 = Außensegment }
vektorbeginx,        { Vektorkoordinaten, die die Lage der Seite }
vektorbeginy,        { auf dem Screen beschreiben. Bei Dachflaechen}
vektorendex,         { beliebige Vektorkoordinaten. Dachflaechen-}
vektorendey : integer; { koordinaten stehen unter Abteilungstyp.plan}
reale_laenge : real;  { reale Abschnittlänge (m) }
reale_flaeche : real; { reale Länge der Fläche (m2) }

bedachung1 : st19;    { Bedachungsmaterial vgl. BEDACH.DAT }
einsparung1 : st13;   { Einsparungsmaßnahme vgl. BEDACH.DAT }
prozent1 : real;      { Anteil der Fläche mit obigen Materialien (%) }
bedachung2 : st19;    { zweites Bedachungsmaterial vgl. BEDACH.DAT }
einsparung2 : st13;   { Einsparungsmaßnahme für 2 }
prozent2 : real;      { Anteil der Fläche mit zweiten Materialien (%) }
k_wert : real;        { k_strich_wert der Flaeche (W/K/m2) }
end;

abteiltyp = record
  plan                : rect;      { Vektorkoordinaten Dachflaechen }
  waag_ist_giebel     : boolean;
  stehwandhoehe       : real;      { m }
  winkell             : real;      { grad }
  winkel2            : real;      { grad }
  giebellaenge        : real;      { m }
  stehwandlaenge     : real;      { m }
  schiffbreite        : real;      { m }
  aussenflaeche       : real;      { m2 }
  innenwand_vorhanden : boolean;
  aussenabschirmlaenge : real;     { m }
  schirm_vorhanden    : boolean;
  schirm_abdichtung   : st8;
  schirm_material     : st31;
  licht_vorhanden     : boolean;
  licht_install_leistung : real;   { W/m2 }
  licht_lampenart     : string[31]; { vgl. LICHT.DAT }
end;

betriebsart = record

  simulationskennung : string;      { Dateiname ohne Suffix }
  bedachungskennung  : string[30];
  abt_anzahl         : integer;      { Anzahl der Abteilungen }
  abt                 : array [1..Maxseg] of abteiltyp;
  klimazone          : string;
  klimadatei         : string[13];
  norm_aussentemperatur, { grd C }
  extrem_aussentemperatur : integer; { grd C }
  bhkw_einsatz       : boolean;
end;

betriebsart = ^betriebsart;

regelungstyp = record
  ti_kennung : string[9];
```



```
ti_steuerung      : string[5];           { konst,absen,summe }
ti_tag,           { grd C }
ti_nacht,         { grd C }
ti_lueftung,     { grd C }
ti_min           : shortint52feld;      { grd C }
ti_summe         : int52feld;           { grd C h/24h }
schirm_kennung   : string[9];
schirm_steuerung : string[12];         { Uhrzeit,Einstrah.}
schirm_auf,      { Uhrzeit }
schirm_zu        : shortint52feld;     { Uhrzeit }
schirmschaltpunkt_licht : integer;   { W/m2 Par }
schirm_lichtsumme : int52feld;         { Wh/m2/24h PAR }
licht_kennung    : string[9];
licht_steuerung  : string[10];        { Uhrzeit, Lichtsumme }
licht_an,        { Uhrzeit }
licht_aus        : shortint52feld;     { Uhrzeit }
licht_lichtsumme : int52feld;         { Wh/m2/24h }
end;
```

```
klimaregelungtyp = record
  abt : array [1..Maxseg] of regelungstyp;
end;
```

```
regelungszeigertyp = ^klimaregelungtyp;
```

```
wochentyp = record
  menge : array[1..52] of real;        { wöchentliche Werte }
  max_woche,                          { maximaler Wochenwert }
  summe,                               { Summe aller Wochenwerte }
  kosten : real;                       { alle Wochen Gesamtkosten DM }
end;
```

```
feinplanung_ergebnistyp = record
```

```
  gesamtkosten : real;                { Heizung + Belichtung DM/Jahr }
  grundlast,   { notwend. Grundlastträger-Einheiten}
  spitzlast,   { notwend. Spitzenlastträger-Einh. }
  bhkw,        { nutzbare BHKW-Abwärme in kWh }
  licht : wochentyp;                   { notwendiges Kunstlicht kWh }
  heizkosten_seg,
  lichtkosten_seg : array[1..MaxSeg,1..52] of real;
  zusatz_aus_kost,   { Stromzusatzkosten für Energieausbringung DM}
  zusatz_aus_energie, { Stromzusatzmenge für Energiebereitstellung kWh}
  zusatz_berei_energie, { Stromzusatzmenge für Energiebereitstellung kWh}
  zusatz_berei_kost : real; { Stromzusatzkosten für Energiebereitstellung DM}
  licht_verlauf : array [0..23] of integer;
  licht_max : real;   { Maximal erforderliche Lichtleistung}
  licht_stunden : array [0..100] of real;
end;
```

```
analysetyp = record
  regelung : klimaregelungtyp;
  erg      : feinplanung_ergebnistyp;
  betriebsweise,
  grundlasttraeger,
  spitzenlasttraeger : st30;
```

```
grundlasttraegereinheit,  
spitzenlasttraegereinheit : st6;  
grundlastleistung : real;           { W }  
grundlastkosten,  
spitzenlastkosten : real;          { DM/Liefereinheit }  
energietraegerbezeichnung,  
heizungssystembezeichnung,  
elektr_energiebezeichnung : st15;  
heizungssysteme : array[1..15] of st27;  
stromkosten_tag,  
stromkosten_nacht : real;          { DM/kWh }  
strom_tagbeginn,  
strom_nachtbeginn : integer;       { Uhrzeit }  
strom_eigenproduktion : boolean;  
strom_eigenproduktionskosten: real; { DM/kWh }  
witterung :st17;  
end;  
  
analysezeigertyp = ^ analysetyp;  
  
var  
  loesung_vorhanden : boolean;  
  arbeitsbereich   : string;  
  kommandopfad     : string;  
  installationspfad:string;  
  ausgabecode      : word;  
  tempein          : boolean;  
  lichtsummeein   : boolean;  
  
implementation  
  
end.
```

**C.5 Datenbankorientiertes Modell (Standardvorgaben und Struktur) im Bereich Heizenergiebereitstellung**

{ AN\_PREIS.DAT T.R. (erster Teil Daten, zweiter Teil Struktur)

%

Die Anlagenpreise-Datenbank enthält die notwendigen Kapitalaufwendungen, die für die Erstellung der Energiebereitstellung benötigt werden. Sie können die Daten verändern oder Ihre eigene Datenbank erstellen.

#

Preistabellen für die Investitionen bei Heizungsanlagen

5

- Investitionsvolumen Energieversorgung (Anlagen)\*12
- Investitionsvolumen Energieversorgung (Anschlußkosten)\*3
- Investitionsvolumen Energieversorgung (Umstellung)\*4
- Investitionsvolumen Energieversorgung (Zusatzkosten)\*2
- Investitionsvolumen Energieversorgung (Energieausbringung)\*3

@ Anlagenbeschreibung	interne Ab	Pr 1	Pr 2	War	Nu
& Öl-EL	oel	287	77	1	20
& Erdgas	gas	272	59	1	20
& Öl-Gas-Mehrstoffanlage	oel_gas	329	80	2	20
& Kohle	kohle	507	172	2	20
& Holz- oder Strohanlage	holz_stroh	580	214	2	20
& Wärmepumpe (Luft)	waermep_lu	1460	970	2	20
& Wärmepumpe (Oberfl.-Wasser)	waermep_ob	950	578	2	20
& Wärmepumpe (Grundwasser)	waermep_gr	1900	1170	2	20
& hochtemp. Fernwärme	ht_fern	25	5	2	30
& niedrigtemp. Fernwärme	nt_fern	50	10	2	30
& Gas-Luftheizer	gas_lufth	204	154	4	15
& Oel-Luftheizer	oel_lufth	241	176	4	15
& Erdgas	gas_an	25	8	0	15
& hochtemp. Fernwärme	ht_fern_an	50	50	0	45
& niedrigtemp. Fernwärme	nt_fern_an	2500	200	0	45
& von Oel auf Gas	oel_n_gas	63	9.3	1	20
& von Gas nach Oel	gas_n_oel	119	34	1	20
& von Oel nach Mehrstoffanlage	oel_n_mehr	78	11	1	20
& von Gas nach Mehrstoffanlage	gas_n_mehr	143	34	1	20
& Schlummerspeicher	schlummer	30	15	2	15
& Mehrkesselanlage	mehrkessel	57	14	1	20
& Energieausbringung Standard	ht_ausbrin	109	82	3	20
& NT-Energieausbringung	nt_ausbrin	400	350	4	20
& Energieausbringung direkt	di_ausbrin	0	0	0	20

\*Seite 1

Anlagenbeschreibung => Beschreibung des Investitionsobjektes

Es werden mehrere Investitionsobjekte mit ihren dazugehörigen Investitionen dargestellt. Die erste Investitionsspalte gibt Preise für eine 0,2 MW-Anlage an. Die zweite Investitionsspalte bezieht sich auf eine 2,4 MW-Anlage. Mit den beiden Preisen können Sie praktisch eine "Preiskurve" definieren. Die tatsächlich durch das jeweilige Objekt entstehenden Investitionen werden durch Interpolation zwischen den beiden angegebenen Größen ermittelt. Bei Anlagen, die kleiner als 200 W sind, wird das Investitionsvolumen von der 0.2 kW- Spalte zugrunde gelegt. Bei Anlagen, die größer als 2.4 MW sind, wird das Inve-

stitutionsvolumen der Spalte 2.4 MW zugrunde gelegt.

**Quellen:**

Tantau H.J. (1983): Heizungsanlagen im Gartenbau. Stuttgart: Ulmer.

VDI (1980): VDI-Richtlinie 3781, Blatt 4

VDI (1983): VDI-Richtlinie 2067, Blatt 1

VDI (1989): VDI-Richtlinie 2067, Blatt 6

KTBL (1990): KTBL-Taschenbuch Gartenbau. Münster-Hiltrup: Landwirtschaftsverlag.

Bylda (1990): Die Wahl des Grund- und Spitzenlastverhältnisses bei der Auslegung bivalenter Gewächshausheizungsanlagen unter Berücksichtigung verschiedener Energieträger. Diplomarbeit ITG Hannover (unveröffentlicht) Expertenangaben (v.Elsner, Domke, Gabloffsky, Landesgasversorgung Sarstedt, Stadtwerke Hannover)

**\*Seite 2**

interne Ab => für die Prologmodule notwendige interne Abkürzung

**\*Seite 3**

Pr 1 => Kosten bei einer 0.2 MW Anlage [DM/kW inst. Leistung]

Investitionsvolumen (bezogen auf 1kW installierte Leistung), welches entstehen würde bei einer Anlagengröße von 0,2 MegaWatt.

Die Kosten sind Komplettkosten, also inkl. Zubehör, Montage etc. (also auch inkl. Öltank, Schornstein)

Die tatsächlichen Investitionssummen ergeben sich aus der Multiplikation der leistungsbezogenen Investition mit der tatsächlich zu installierenden Leistung.

**\*Seite 4**

Pr 2 => Kosten bei einer 2.4 MW Anlage [DM/kW inst. Leistung]

Investitionsvolumen (bezogen auf 1kW installierte Leistung), welches entstehen würde, bei einer Anlagengröße von 2,4 Mega-Watt.

Die Kosten sind Komplettkosten, also inkl. Zubehör, Montage etc. (also auch inkl. Öltank, Schornstein)

Die tatsächlichen Investitionssummen ergeben sich aus der Multiplikation der leistungsbezogenen Investition mit der tatsächlich zu installierenden Leistung.

**\*Seite 5**

WAR => Jährliche Kosten durch Wartung in Prozent von der Investition

Für jede Heizungsanlage müssen im Laufe des Jahres Wartungs- und Instandhaltungsaufwendungen aufgebracht werden. Der hier angegebene Wert charakterisiert diese Aufwendungen in Prozent vom Anschaffungspreis. Die Daten stammen größtenteils aus den unter der Spalte 'Anlagenbeschreibung' aufgeführten VDI-Richtlinien.

**\*Seite 6**

Nu => Nutzungsdauer [Jahre]

Über die Nutzungsdauer der Anlage kann der Aufwand für die Verzinsung des eingesetzten Kapitals berücksichtigt werden.

Die Standardvorgaben, die hier vorgegeben werden, entstammen größtenteils den unter der Spalte 'Anlagenbeschreibung' aufgeführten VDI-Richtlinien.

\*Seite 7

```
}
(Datei:'AN_PREIS.DAT';
 Bezeichnung : 'ANLAGENPREISE ';
 Varianten : true;
 Variantenschluessel : 'APR';
 Struktur:(

(Name:'Anlage';Typ:'S';Beginn:3;Laenge:30;Mini:0;Maxi:0;
 Sichtbar:[1,2,3,4,5];Veraenderbar:[]),

(Name:'intern';Typ:'R';Beginn:34;Laenge:10;Mini:1;Maxi:20;
 Sichtbar:[];Veraenderbar :[]),

(Name:'Preis_Min';Typ:'R';Beginn:45;Laenge:4;Mini:0;Maxi:9999;
 Sichtbar:[1,2,3,4,5];Veraenderbar:[1,2,3,4,5]),

(Name:'Preis_Max';Typ:'R';Beginn:50;Laenge:4;Mini:0;Maxi:9999;
 Sichtbar:[1,2,3,4,5]; Veraenderbar:[1,2,3,4,5]),

(Name:'Wartung';Typ:'R';Beginn:55;Laenge :3;Mini:0;Maxi:100;
 Sichtbar:[1,3,4,5];Veraenderbar:[1,3,4,5]),

(Name:'Nutzdauer';Typ:'R';Beginn:59;Laenge :2;Mini:1;Maxi:99;
 Sichtbar:[1,2,3,4,5];Veraenderbar:[1,2,3,4,5]),

(Name:'';Typ:'R';Beginn:0;Laenge :0;Mini:0;Maxi:0;
 Sichtbar:[];Veraenderbar:[]),

(Name:'';Typ:'R';Beginn:0;Laenge :0;Mini:0;Maxi:0;
 Sichtbar:[];Veraenderbar:[]),

(Name:'';Typ:'R';Beginn:0;Laenge :0;Mini:0;Maxi:0;
 Sichtbar:[];Veraenderbar:[]),

(Name:'';Typ:'R';Beginn:0;Laenge :0;Mini:0;Maxi:0;
 Sichtbar:[];Veraenderbar:[]),

(Name:'';Typ:'R';Beginn:0;Laenge :0;Mini:0;Maxi:0;
 Sichtbar:[];Veraenderbar:[]),

(Name:'';Typ:'R';Beginn:0;Laenge :0;Mini:0;Maxi:0;
 Sichtbar:[];Veraenderbar:[]),

(Name:'';Typ:'R';Beginn:0;Laenge :0;Mini:0;Maxi:0;
 Sichtbar:[];Veraenderbar:[]),

(Name:'';Typ:'R';Beginn:0;Laenge :0;Mini:0;Maxi:0;
 Sichtbar:[];Veraenderbar:[]),

(Name:'';Typ:'R';Beginn:0;Laenge:0;Mini:1;Maxi:99;
 Sichtbar:[];Veraenderbar:[])))
```

Die vorliegende Arbeit entstand in den Jahren 1988-1991 am Institut für Technik in Gartenbau und Landwirtschaft der Universität Hannover im Rahmen eines vom Bundesministerium für Ernährung, Landwirtschaft und Forsten geförderten Forschungsprojektes.

Herrn Prof. Dr. H.-J. Tantau danke ich für die Überlassung des Themas, für die zahlreichen wertvollen und grundlegenden Anregungen sowie für die vielen kleinen Hinweise 'nebenbei', die häufig die entscheidenden Impulse für weitere Überlegungen waren.

Herrn Prof. Dr. E. Berg danke ich recht herzlich für die Übernahme des Korreferates und für die Diskussionen im Rahmen der Expertensystemgruppe am Fachbereich Gartenbau, die mir den Einstieg in die Expertensystemtechnologie erleichterten.

Herrn Dr.-Ing. B. von Elsner danke ich für die zahlreichen Diskussionen und Anregungen zu meiner Arbeit und für die Unterstützung bei den vielen kleinen aber manchmal entscheidenden Problemen beim täglichen Umgang mit Hard- und Software.

Herrn Prof. Dr.-Ing. Chr. von Zabeltitz danke ich für die großzügige Unterstützung meiner Arbeiten und für die Schaffung einer kreativen Arbeitsatmosphäre am Institut.

Ein Expertensystemprojekt kann nicht ohne Expertenwissen existieren und somit danke ich allen, die mit Ihrem Wissen dazu beigetragen haben, daß diese Arbeit entstehen konnte. Neben den Mitarbeitern und Kollegen des Institutes gilt mein besonderer Dank Dipl.-Ing. O. Domke und Dipl.-Ing. W. Gabloffsky, die wertvolle und hilfreiche Hinweise aus der Beraterpraxis lieferten.

Herrn H.-J. Husmann danke ich für seine kreative Unterstützung bei der Programmentwicklung.







**INSTITUT FÜR TECHNIK IN GARTENBAU  
UND LANDWIRTSCHAFT**  
UNIVERSITÄT HANNOVER

**Thomas Rath**

Einsatz wissensbasierter Systeme zur Modellierung und  
Darstellung von gartenbautechnischem Fachwissen am  
Beispiel des hybriden Expertensystems HORTEX

Dissertation Universität Hannover

Gartenbautechnische Informationen

Heft 34

1992