

Expertensysteme – eine Einführung –

Klaus Pommerening

12. Dezember 1988

Inhaltsverzeichnis

1 Was ist ein Expertensystem?	2
1.1 Aufgabenstellung	2
1.2 Architektur	2
1.3 Wissensbanken	4
1.4 Ungewisse Regeln	4
1.5 Problemlösungsstrategien	5
1.6 Entwicklungswerkzeuge	6
2 Ein „triviales“ Beispiel	6
3 Militärische Anwendungen	9
3.1 Bisherige Entwicklungen	9
3.2 Drei Projekte	9
3.3 Anforderungen an die Leistungsfähigkeit	10
4 Wie zuverlässig sind Expertensysteme?	11
4.1 Programmierwerkzeuge	11
4.2 Komplexität des Regelsystems	11
4.3 Inkonsistenz in den Regeln	12
4.4 Gewißheitsfaktoren	13
4.5 Expertensysteme und Modellbildung	13
4.6 Was weiß ein Experte?	15
5 Wertung	15
5.1 Die Grenzen der Expertensystem-Technik	15
5.2 Die Gefahren des militärischen Einsatzes	16

1 Was ist ein Expertensystem?

1.1 Aufgabenstellung

Ein Expertensystem ist ein Programm, das Expertenwissen auf einem bestimmten Gebiet maschinell verarbeitet, also einen Experten simuliert. Es soll Probleme aus diesem Gebiet lösen und Handlungsanweisungen geben. Typischerweise denkt man dabei an ein medizinisches Diagnosesystem, das aus den Befunden eines Patienten eine Diagnose herleitet, bei Bedarf aber auch weitere Befunde anfordert und schließlich Therapieempfehlungen gibt.

Expertensysteme werden zum Bereich der „künstlichen Intelligenz“ gezählt. Sie sollen Probleme verstehen, analysieren und lösen können, selbst bei unvollständigem oder diffusen Wissen, das nur in Form von Faustregeln oder vermutlichen Schlußfolgerungen vorliegt und nicht in Form von eindeutigen Daten und empirisch abgesicherten Naturgesetzen.

Anwenden sollen sich Expertensysteme für folgende Arten von Aufgaben lassen [4]:

Interpretation: Ableitung von Situationsbeschreibungen aus Daten.

Vorhersage: Ableitung wahrscheinlicher Konsequenzen aus gegebenen Situationen.

Diagnose: Ableitung von Systemfehlern aus Beobachtungen.

Design: Konfiguration von Objekten unter gegebenen Bedingungen.

Planung: Erzeugung und Bewertung einer Folge von Aktionen zur Erreichung eines Zielzustands.

Überwachung: Vergleich von Soll- und Istwerten.

Fehlerbehebung: Empfehlung von Korrekturen bei Fehlfunktionen.

Reparatur: Ausführung eines Plans zur Umsetzung von Empfehlungen.

Ausbildung: Diagnose und Verbesserung des Verhaltens des Lernenden.

Steuerung: Interpretation, Voraussage, Korrektur und Überwachung des Systemverhaltens.

Nach Aussagen ihrer Verfechter sollen Expertensysteme die Grenzen sprengen, die der strengen formalen Modellbildung gesetzt sind; sie sollen auch Ergebnisse für Probleme erzielen können, für die es keine Algorithmen gibt [4, Chapter 1].

1.2 Architektur

Expertensysteme haben einen typischen Aufbau, siehe Tabelle.

Der Datenbereich, meistens Wissensbasis oder Wissensbank genannt, besteht aus

Anwender		Experte
Konsultations- Komponente	Erklärungs- komponente	Wissenserwerbs- komponente
Problemlösungskomponente (Inferenzmaschine)		
fallspezifisches Faktenwissen	Zwischen- und Endergebnisse	bereichsspezifisches Expertenwissen

- dem Expertenwissen, das bei der Programmerstellung aus einem oder mehreren Experten gezapft wird und sich während einer Konsultation (Programmanwendung) nicht ändert,
- dem Faktenwissen, das der Anwender während der Konsultation eingibt und das das zu lösende Problem beschreibt,
- den Ergebnissen, die während der Konsultation durch die Problemlösungskomponente hergeleitet werden.

Der funktionale Bereich des Programms, Steuersystem oder Expertensystemschale (shell) genannt, besteht aus

- der Wissenserwerbskomponente, die nur bei der Erstellung oder Erweiterung des bereichsspezifischen Teils der Wissensbasis durch den Experten verwendet wird,
- der Konsultationskomponente, die sowohl die Benutzerschnittstelle, also die Steuerung des Dialogs mit dem Benutzer, als auch gegebenenfalls die automatische Meßdatenerfassung enthält,
- der Problemlösungskomponente, in der die Strategien zur Auswertung des Wissens und zur Lösungssuche festgelegt sind,
- der Erklärungskomponente, über die der Anwender bei Bedarf Erklärungen über die Arbeit des Expertensystems, insbesondere Begründungen für gefundene Lösungen erhält.

Ist das System geschickt entworfen, so ist die Schale von der Wissensbasis unabhängig. Darauf beruhen die auf dem Markt angebotenen Expertensystem-Entwicklungsumgebungen; sie bestehen aus einer „leeren“ Schale, in die nur noch die Wissensbasis eingefüllt werden muß, und schon hat man ein lauffertiges Programm. Ganz so einfach ist das aber doch nicht, denn wie wir noch sehen werden, besteht ein erheblicher Teil des Erstellungsaufwands daraus, die Inferenzmaschine zu effektivem Vorgehen zu bringen.

Den Fortschritt, den die „Erfindung“ der Expertensysteme bedeutet, kann man am folgenden groben Schema erkennen:

Klassische Programme enthalten Programmtext und Daten.

Dateiverwaltungs-/Datenbanksysteme: Trennung der Daten vom Programmtext und Ablage in einer „Datenbank“.

Expertensysteme: Herauslösung von Entscheidungen/Schlußfolgerungen aus dem Programmtext und Ablage in einer „Wissensbank“.

Dadurch wird die Programmerstellung vor allem für Aufgaben vereinfacht, die viele Verzweigungen enthalten, wie die bereits genannten. Softwaretechnisch gesehen sind Expertensysteme also einfach nur Programme mit einer bestimmten, für manche Aufgaben besonders geeigneten Systemarchitektur.

1.3 Wissensbanken

Die Grundvoraussetzung für die Funktionsfähigkeit eines Expertensystems ist die formale („maschinenlesbare“) Darstellung des Wissens. Dafür gibt es verschiedene Ansätze. Hier soll nur der wichtigste behandelt werden – Regeln. Regeln haben die Form

WENN(*Bedingung*)*DANN*(*Folgerung*).

Beispiel aus der Medizin:

IF Stuhlbeschaffenheit = ‘reiswasserartig’
THEN Infektion = ‘Cholera’

Eine solche Regel wird von der Problemlösungskomponente ausgewertet, sobald ihre Bedingung als wahr erkannt ist und sie aufgrund der Festsetzungen in der Problemlösungsstrategie an der Reihe ist. Die Folgerung wird dann der Wissensbasis, Abteilung Zwischen- und Endergebnisse, hinzugefügt. Das ist das ganze Geheimnis hinter der Aussage „Ein Expertensystem kann selbständig aus bekanntem Wissen neues herleiten.“

Die Wissensbasis enthält außer Regeln auch Fakten, z.B. „Eine Rakete ist eine Waffe.“ Je nach Programmiersprache oder Entwicklungsumgebung werden sie als Regeln mit „leerer“ (also stets wahrer) Voraussetzung oder als möglicher Wert eines Parameters in der Wissensbasis abgelegt, z.B.

PARAMETER Waffe
TAKEN FROM (... , Rakete, ...)

Die Wissensdarstellung in Regeln ist sehr flexibel, leicht auswertbar – und unübersichtlich. Eine grundsätzlich andere Darstellung des Wissens benutzt hierarchische Tabellen („frames“). Diese ist viel übersichtlicher, aber viel schwerer auszuwerten und spielt in der Praxis bisher eine geringere Rolle. Die im folgenden vorgestellten Probleme bestehen dabei im wesentlichen genauso.

1.4 Ungewisse Regeln

Expertensysteme werden dafür gerühmt, daß sie auch vages, ungenaues Wissen, Indizien, Verdachtsmomente verarbeiten können und damit zu gültigen Schlußfolgerungen kommen. Die übliche Methode, dies zu erreichen, ist die Verwendung von Gewißheitskoeffizienten in den Regeln. Das kann zum Beispiel so aussehen:

IF Stuhlbeschaffenheit = ‘himbeergeleeartig’
THEN THERE IS 0.4 EVIDENCE
THAT Infektion = ‘Amöbenruhr’

Die Probleme dieses Verfahrens sind:

- Welche Größe soll man einem Gewißheitskoeffizienten sinnvollerweise geben?
- Wie soll man bei mehreren Indizien für eine Hypothese die Gewißheiten kombinieren? Z.B. werden im allgemeinen zwei schwache Indizien zusammen schon als recht starkes Indiz angesehen.
- Wie soll man Gewißheiten über längere Schlußketten fortpflanzen? Z.B. gelte A mit Gewißheit 0.8, aus A folge B mit Gewißheit 0.9, aus B wieder C mit Gewißheit 0.6. Wie gewiß ist C dann?

Es gibt verschiedene theoretische Ansätze.

Bedingte Wahrscheinlichkeiten. Man spricht auch vom Bayesschen Ansatz. Hier hat man einen klaren theoretischen Hintergrund für Kombination und Fortpflanzung; die Größe der Koeffizienten wird durch empirische Beobachtung geschätzt. Dieses saubere Konzept ist in der Praxis kaum brauchbar. Erstens gibt es oft keine verwertbaren Beobachtungen. Zweitens ist die Kombination von Wahrscheinlichkeiten unbefriedigend; sie verleiht dem Zusammentreffen verschiedener Indizien zu wenig Gewicht, und fehlende Evidenz für eine Aussage wird automatisch als Evidenz für das Gegenteil genommen.

Evidenzfunktionen nach Dempster/Shافر. Hier gibt es sinnvolle und konsistente Kombinations- und Fortpflanzungsregeln; das Problem der sinnvollen Definition der Größen wird offen gelassen. [10]

Fuzzy-Mengenlehre. Hier wird das Problem der Definition der Größen so gelöst, daß man statt numerischer Werte verbale Beschreibungen wie „some evidence“, „starker Verdacht“, ... benützt, die im Prinzip nur anordenbar sein müssen [1]. Um zu Kombinations- und Fortpflanzungsregeln zu kommen, müssen diese Werte aber programmintern wieder auf Zahlen abgebildet werden.

Ein Beispiel dafür könnte so aussehen:

```
IF Stuhlbeschaffenheit = 'himbeergeleeartig'  
  THEN THERE IS SOME EVIDENCE  
    THAT Infektion = 'Amöbenruhr'
```

wobei der Zusatz SOME den Faktor 0.4 bedeutet. Diese Regel ist also zu der obigen völlig gleichwertig.

1.5 Problemlösungsstrategien

Grundsätzlich unterscheidet man zwischen Vorwärts- und Rückwärtsverkettung von Regeln. Bei der Vorwärtsverkettung werden nach jeder neugewonnenen Erkenntnis alle ausführbaren Regeln angewendet, um wiederum neue Erkenntnisse zu gewinnen. Der Vorteil ist die Vollständigkeit der hergeleiteten Folgerungen, der Nachteil der hohe Zeitaufwand. Bei der Rückwärtsverkettung geht man zielgerichtet vor. Es werden Regeln gesucht, aus denen sich eine zu beweisende Hypothese herleiten läßt. Hat eine solche Regel eine noch nicht als wahr erkannte

Voraussetzung, so wird diese rekursiv als vorübergehendes neues Beweisziel behandelt usw. In der Praxis werden die beiden Verkettungsrichtungen oft mit dem Ziel größtmöglicher Effektivität kombiniert.

Bei umfangreichen Wissensbanken ist es im allgemeinen viel zu aufwendig, den Bedingungsteil aller Regeln auf Erfülltsein zu testen. Vielmehr wird in jedem Schritt eine bestimmte Auswahl untersucht. Es kann auch vorkommen, daß mehrere Regeln anwendbar sind, aber nur eine benötigt wird oder sich die Regeln widersprechen (Beispiele später). Für diesen Fall muß festgelegt sein, in welcher Reihenfolge die Regeln anzuwenden sind. Die einfachste Strategie wäre, immer die zuerst gefundene Regel anzuwenden. Entschieden werden muß auch, ob eine Regel nur einmal oder immer wieder anzuwenden sein soll (Gefahr von Endlosschleifen).

Obwohl man mit der Verwendung von Regeln hofft, vom prozeduralen Programmieren weg-, zum „deklarativen“ Programmieren hinzukommen, ist man also doch, vor allem aus Gründen der Effizienz, gezwungen, sich sehr sorgfältige Überlegungen über die Abarbeitungsreihenfolge der Regeln zu machen.

1.6 Entwicklungswerkzeuge

Für die Entwicklung von Expertensystemen gibt es verschiedene Hilfsmittel:

Klassische Programmiersprachen - Prozedurale Allzwecksprachen von Assembler oder FORTRAN bis ADA oder MODULA-2.

KI-Sprachen - Vor allem LISP und, moderner und besser geeignet, PROLOG.

Werkzeugkisten - Pakete von fertigen Prozeduren zum Einbau in eigene Programme („Tools“).

Entwicklungsumgebungen - Die schon erwähnten „Shells“.

Die folgende Tabelle vergleicht die verschiedenen Entwicklungshilfsmittel.

Phase	A	KI-Sprache	Tools	Shells
Problemanalyse	+	+	+	+
Spezifikation	+	+	+	+
Entwurf	+	+	erleichtert	wenig Aufwand
Implementation	+	erleichtert	reduziert	wenig Aufwand
Test	+	erleichtert	reduziert	wenig Aufwand

Dabei bedeutet A eine klassische höhere Programmiersprache, und + einen hohen Aufwand. Natürlich sinkt die Flexibilität der Programmierung und die Abarbeitungsgeschwindigkeit des fertigen Programms mit steigendem Entwicklungskomfort.

2 Ein „triviales“ Beispiel

Um die Tücken von Expertensystemen zu zeigen, will ich ein Demonstrationsbeispiel aus einem gängigen Entwicklungssystem vorführen. Es geht nicht darum, ein bestimmtes Produkt schlecht zu machen, sondern nur darum, die Fallen bei der Entwicklung von Expertensystemen zu verdeutlichen. Das Beispiel selbst ist trivial und erhebt keinen Anspruch, ernstgenommen zu werden.

Hi. I am your animal identification system.
Please describe one of the following animals.
The animals that I have knowledge about are:

albatross
penguin
ostrich
zebra
giraffe
tiger
cheetah

Animal appearance:
(Choose any number of the following:)

x___ has feathers
___ has fur
x___ has hooves
___ has long neck
___ has claws

What does animal's mother produce?
(Choose one of the following:)

x___ lays eggs
___ gives milk

How does the animal move about?
(Choose any number of the following:)

x___ flies
___ flies well
___ swims
___ runs or walks

Animal color and markings:
(Choose any number of the following:)

---- tawny
---- black stripes
---- dark spots
x___ black and white

The animal's identity is albatross.

---- How ----

I assigned value to the animal's identity of IDENTITY by

1. Rule ALBATROSS1 which states that

if ANIMAL CLASS is 'bird' and
ANIMAL LOCOMOTION is ('flies','flies well')
then there is 0.9 evidence that the animal's identity = 'albatross'.

As a result of this rule
the animal's identity has been assigned =
'albatross' (.9).

2. Rule ALBATROSS2 which states that

if ANIMAL CLASS is 'bird' and
ANIMAL LOCOMOTION is ('flies','flies well') and
ANIMAL COLOR is 'black and white'
then the animal's identity = 'albatross'.

As a result of this rule
the animal's identity has been assigned = 'albatross' (1).

Wir halten fest: Das Expertensystem hat in einer unvorhergesehenen Situation (es sollte etwa einen Flugdrachen identifizieren) eine Angabe („hat Hufe“) einfach ignoriert und einen Fehlschluß gezogen.

3 Militrische Anwendungen

3.1 Bisherige Entwicklungen

Die Arbeit [5] gibt einen berblick ber bereits existierende militrische Expertensysteme. Wegen der zu vermutenden Geheimhaltung ist natrlich nicht bekannt, ob dies die wichtigsten sind und ob sie berhaupt verwendet werden.

PLAN: Untersttzung der Luftverkehrskontrolle in der Nhe von Flugzeugtrgern.

HASP/SIAP: Aufspren und Identifizieren von Schiffen durch Sonarsignale.

TATR: Taktische Luftzielaufnahme.

I&W: Analyse strategischer Indikatoren und Warnmeldungen.

Ferner soll es Expertensysteme zur Satellitenkontrolle und zur Analyse der Kommunikation auf einem taktischen Schlachtfeld geben.

Exemplarisch soll HASP/SIAP kurz behandelt werden (nach [5]). „Das System hat die Aufgabe, in einem bestimmten Gebiet Schiffe zu entdecken, zu lokalisieren und den Schiffstyp zu identifizieren. Zu diesem Zweck werden auf einem Teilgebiet des Meeresbodens Untersee-Mikrophone angebracht, die auftretende Gerusche aufnehmen und zu einer zentralen Stelle weiterleiten. Hier werden mit Hilfe dieser Sonogramme und nach Ausscheiden von Strgeruschen die Schiffstypen identifiziert.“ Ein typisches Beispiel fur eine Regel in diesem System ist

```
IF Quelle ist kurzlich verschwunden
AND hnliche Quelle wurde
    auf anderer Frequenz festgestellt
AND die Quellenorte sind eng beieinander
THEN Es handelt sich um dieselbe Quelle mit
    einer Wahrscheinlichkeit von 0.3
```

Hier werden also Regeln mit Ungewiheitskoeffizienten verwendet.

3.2 Drei Projekte

Fur die SCI (strategic computing initiative) wurden drei konkrete Projekte als Ziele formuliert. Sie sollen eine breite technologische Basis fur weiterreichende KI-Anwendungen schaffen und so die militrische und wirtschaftliche Vormachtstellung der USA festigen. Als KI-Technik werden insbesondere die Expertensysteme gefordert. Die Projekte sind:

Fahrerloses Landfahrzeug. Erkennung und Interpretation der Umgebung; Planen und Schlieen.

Automatischer Pilotenassistent. Auswahl und Vorverarbeitung von Informationen fur den Piloten; berwachung der Flugsteuerungssysteme; Wissen ber Flugzeuge, Wahrnehmungen und Taktik.

Seegestütztes Kampfführungssystem für Flugzeugträgerverbände.

Planen und Überwachen der Aktionen militärischer Systeme auf dem Gefechtsfeld; Einschätzung der Gefechtslage; Handlungsempfehlungen.

Weitere KI-Techniken, die hierbei weiterentwickelt und verwendet werden sollen, sind Bilderkennung, Verarbeitung natürlicher Sprache und Robotik. Die gewonnenen Erkenntnisse sollen die Basis sein für weitere Entwicklungen von autonomen Systemen („mensenleerer Krieg“), Frühwarnsystemen, Kampfführungssystemen (battle management systems - Expertensysteme als neue Feldherren) mit noch höheren Ansprüchen bis hin zum „Krieg der Sterne“ (SDI).

3.3 Anforderungen an die Leistungsfähigkeit

Bisherige Expertensysteme enthalten höchstens einige 100 Regeln und verarbeiten bis zu 200 Regeln pro Sekunde. Für die drei bei SCI geplanten Projekte sind bis zu 20000 Regeln geplant mit einer Verarbeitungsgeschwindigkeit von bis zu 10000 Regeln pro Sekunde – also knapp die hundertfache Leistung, wie es scheint. Doch der Schein trügt. Es ist zu bemerken, daß die Komplexität des Regelwerks kombinatorisch explodiert, auf jeden Fall weit überproportional zunimmt. Gibt es 20000 Regeln und werden pro Sekunde 10000 verarbeitet, so bedeutet das nicht, daß das Programm in 2 Sekunden fertig ist. Sondern die Lösungsstrategie erfordert, daß mit den neu gewonnenen Zwischenergebnissen die gesamten Regeln immer und immer wieder durchsucht werden, bis eine Lösung kombiniert ist. Je größer die Wissensbasis ist, desto öfter muß auch jede einzelne Regel geprüft werden.

Die zu entwickelnden Expertensysteme sollen weitgehend „Echtzeit-Expertensysteme“ und zum Teil verteilte Expertensysteme sein. Sie haben die Aufgaben

- Regeln für den Eingriff (rules of engagement), Doktrin der Schlachtenführung – Wann werden welche Waffen wo eingesetzt? Welche Ressourcen werden verteidigt oder aufgegeben? (z.B. die Bundesrepublik Deutschland)
- Einschätzung der Art eines Angriffs.
- Einschätzung der Situation – Zustand der Feindseligkeiten, Status der Ressourcen, Behandlung von Unsicherheiten bei teilweise zerstörten oder nicht unterscheidbaren Zielen.
- Vereinfachung von Entscheidungen.

Zum letzten Punkt zwei Zitate:

- „Es ist von keinem Nutzen, (dem Präsidenten) einen Raum voller Statusanzeigen zu geben und zu sagen ‘Hier ist es, Chef, treffen Sie eine Entscheidung’. Es muß auf eine Skala heruntergekocht werden – zum Beispiel grün, gelb oder rot – und er kann aufgrund des Zeigerausschlags entscheiden, was er tun sollte.“ (R.C.Aldridge, ehemaliger Direktor für Wehrforschung und -entwicklung, zitiert nach [7].)

- „Die Einflußmöglichkeit durch höhere Befehlsstäbe wird auf einfache Entscheidungen begrenzt werden, einschließlich der Entscheidung, eine automatische Antwort auf einen Angriff durch eine andere zu ersetzen. Um die höheren Befehlsstäbe in diesen Entscheidungen zu unterstützen, muß diese Funktion (rules of engagement; R.K.-S.) eine Auswahl, Aggregation und Synthese von Informationen so bereitstellen, daß nur das relevante Material für kritische Entscheidungen dargestellt wird. Die Befugnis, Atomwaffen einzusetzen, ist in dieser Funktion enthalten.“ (James Fallow, zitiert nach [7].)

Wenn ich das richtig verstehe, ist gemeint: Der General gibt ein Ziel ein (Sibirien befreien), das Programm entscheidet, daß dazu Atomwaffen nötig sind.

4 Wie zuverlässig sind Expertensysteme?

4.1 Programmierwerkzeuge

Die drei Musterprojekte von SCI sollen in LISP implementiert werden. Es ist bemerkenswert, daß das Department of Defense hier von seiner Lieblings-Programmiersprache ADA abrückt. Das dürfte seinen Grund in der erzielbaren Rechengeschwindigkeit haben, besonders da es spezielle LISP-Rechner gibt. Auf der anderen Seite wird die Zuverlässigkeit der damit erstellten Systeme stark vermindert. LISP-Programme sind alles andere als leicht zu durchschauen, kaum zu überarbeiten und nicht so gut gegen Fehler gesichert wie Programme in modernen Programmiersprachen, die die Prinzipien des Software-Engineering unterstützen. LISP ist die Abkürzung für „List Processing Language“, wird aber von Spöttern auch als „Lots of Indecipherable Silly Parentheses“ (haufenweise unleserliche blöde Klammern) bezeichnet. Das strukturierte Programmieren wird von PROLOG besser unterstützt; aber auch dafür gelten natürlich die prinzipiellen Grenzen, die in den nächsten Abschnitten gesteckt werden.

4.2 Komplexität des Regelsystems

So wie in der Frühzeit des Programmierens etwas kompliziertere Programme in „Spaghetti-Code“ ausarteten, so ist es bei der Erstellung von Regelbasen leicht, „Regelspaghetti“ zu erzeugen, d.h., ein unentwirrbares komplexes logisches Geflecht, das jeden Änderungsversuch mit unerwarteten und oft nur schwer zu entdeckenden Nebenwirkungen bestraft. Komplexe Systeme tendieren zur Unbeherrschbarkeit und entfalten so quasi ein Eigenleben, das zunehmend außer Kontrolle gerät. Das vorgeführte Beispiel zur Tierbestimmung machte schon deutlich, daß intuitiv einfach erfaßbare Situationen selbst bei Beschränkung auf lächerlich wenige Entscheidungsmöglichkeiten eine recht große Zahl von Regeln erfordern und Raum für Fehler bieten.

Nun läßt sich diese Kritik sicher mit Recht abschwächen. Man kann selbstverständlich auch bei der Erstellung von Expertensystemen sorgfältig und verantwortungsbewußt vorgehen. Auch Regelwerke kann man strukturieren und modularisieren und dadurch ihre Komplexität merklich reduzieren, Tests und Wartbarkeit erleichtern usw. Prinzipiell gilt aber auch hier, daß man die Zuverlässigkeit des Systems nur beschränkt erhöhen und Fehlerfreiheit nie errei-

chen kann. Es ist der übliche Kampf mit einem linearen Schwert gegen eine exponentielle Hydra.

Für militärische Anwendungen bringt die große Komplexität der Softwaresysteme eine weitere Gefahr mit sich, die der Sabotage. Schließlich gibt es ja Viren, die lange Zeit unauffällig in großen Systemen schlummern und nur im Ernstfall ihre Wirkung entfalten. „Was ist mit den Tausenden von Menschen, die am Aufbau und der Wartung solcher Systeme beteiligt sind? Wie stellen wir sicher, daß wir ihnen vertrauen können, daß sich nicht ‘zwanghafte Programmierer’ mit einem Todestrieb in diesen Systemen verewigen? Welche polizeistaatlichen Maßnahmen sind erforderlich, um ein solches komplexes und verteiltes System beispielsweise vor Hackern zu schützen?“ [7]

4.3 Inkonsistenz in den Regeln

Expertensysteme im heutigen Verständnis folgen nicht immer den Gesetzen der Logik. Obwohl die beiden Regeln

```
IF aa = TRUE THEN bb = TRUE
IF bb = FALSE THEN aa = FALSE
```

logisch völlig äquivalent sind, wirken sie ganz unterschiedlich. Die erste Regel wird angewendet bzw. ausgeblendet, sobald *aa* einen Wert bekommen hat, die zweite, wenn *bb* einen Wert hat. Oder: *aa*, *bb* und *cc* seien drei logische Parameter. Folgende drei Regeln seien in der Wissensbasis enthalten:

```
IF aa = TRUE THEN bb = TRUE
IF bb = TRUE THEN cc = TRUE
IF aa = TRUE THEN cc = FALSE
```

In einer typischen Anwendung sei gerade *aa* als wahr erkannt worden. Dann sind die Ergebnisse

$$aa = TRUE, bb = TRUE, cc = FALSE.$$

Die Erklärung dafür ist, daß nach Wertzuweisung an *aa* die erste und dritte Regel angewendet werden können; die zweite Regel wird dann nicht mehr beachtet, weil *cc* schon einen Wert hat. Die logische Lösung, daß *aa* falsch sein muß, wird nicht entdeckt.

Die Ergebnisse hängen also offensichtlich von der Reihenfolge der Regeln ab, obwohl die Reihenfolge bei diesen logischen Schlüssen keine Rolle spielen darf. Zwar wird das nichtprozedurale Programmieren angestrebt, der Entwickler muß aber den Ablauf der Prozeduren genau überwachen. Das macht er durch die Problemlösungsstrategien - hoffentlich behält er dabei den Überblick! Denn jede Anweisung oder Regel zur Steuerung der Abarbeitungsreihenfolge erhöht wieder die Komplexität des Programms.

Wie sehr man durch Änderung einer Auswertungsreihenfolge Ergebnisse manipulieren kann, zeigt in einem anderen Zusammenhang der Artikel [9].

Das Regelsystem kann also unentdeckte logische Widersprüche enthalten, die durch kein denkbare Verifikationsverfahren zu entdecken sind. Natürlich wirken solche Widersprüche zunächst einmal konstruiert. Sie können aber erstens versteckt sein, z.B. in einem umfangreichen Regelwerk oder in Werten, die den Parametern erst beim Programmablauf zugewiesen worden sind. Zweitens können sie in der Praxis leicht auftreten, wenn ein Expertensystem Regeln von mehreren unabhängigen Experten enthält.

4.4 Gewißheitsfaktoren

Das Konzept der Gewißheit (Certainty/Evidence) ist sehr verwaschen und muß grundsätzlich scharf kritisiert werden. Keiner der vorhandenen Ansätze ergibt eine Lösung, die ein Mathematiker gutheißen kann.

Die Zuordnung von Gewißheitsfaktoren durch den Benutzer nach Augenmaß ist sehr ungenau. Die numerische Stabilität der Folgerungen wird an keiner Stelle untersucht. Erfahrungen mit Formeln machen hier sehr skeptisch. Jeder, der ernsthaft Mathematik anwendet, weiß, wie anfällig Formeln für kleine Manipulationen an Parametern sind - oft lassen sich damit weitgehend beliebige Ergebnisse erzielen. Auch die Reihenfolge der Abarbeitung einer Formel kann das Ergebnis kraß beeinflussen. Schlimm ist, daß bei Expertensystemen die Formeln sogar noch auf schwer durchschaubare Weise in Regeln versteckt sind. Auf Manipulationsmöglichkeiten mit der Reihenfolge von Regeln wurde schon hingewiesen; durch die Zuordnung von Gewißheitsfaktoren werden diese noch erweitert.

Hier gilt in besonderem Maße das Urteil von Parnas: „Heuristische Methoden liefern keine Systeme, auf die man sich verlassen kann“ [8].

4.5 Expertensysteme und Modellbildung

Meiner Meinung nach sind Expertensysteme bisher nicht genügend unter erkenntnistheoretischen Gesichtspunkten betrachtet worden. Sie sind logische, formale Modelle eines Stücks der Wirklichkeit; soweit sie Gewißheitskoeffizienten verwenden, sind sie auch mathematische Modelle. Der Entwickler eines Expertensystems erstellt also implizit ein mathematisch-logisches Modell des betreffenden Bereichs. Das ist ein sehr problematisches Unterfangen, das stets große Sorgfalt und gründliche theoretische und experimentelle Absicherung erfordert.

Man unterscheidet zwischen exakten, „harten“ Modellen, wie sie in den exakten Naturwissenschaften vorkommen, und qualitativen, „weichen“ Modellen, z.B. in biologischen, wirtschafts- oder sozialwissenschaftlichen Anwendungen. Hier sind oft schon die zugrundeliegenden Größen ungenau und umstritten und prinzipiell nicht durch Beobachtung festzulegen, die Beziehungen und Wirkungen äußerst kompliziert und unübersichtlich. Was ist zum Beispiel die Inflationsrate? Was der Wechselwirkungskoeffizient zweier Tierarten? Dagegen sind z.B. beim Raumflug sehr genaue Voraussagen ferner Landpunkte üblich. Die Rechtfertigung für ein Modell, ob hart oder weich, kann letztlich immer nur der Erfolg sein, d.h., daß seine Aussagen mit der Wirklichkeit übereinstimmen. Selbst in der Physik, wo die Modelle oft überwältigend genau sind, kann man nicht darauf verzichten, die Theorie durch Experimente zu überprüfen. Bei weichen Modellen sind qualitative Prognosen das beste, was man erreichen kann, also etwa Aussagen über Gleichgewichte, Trends, Einflüsse von Umgebungsänderungen. Solche qualitativen Beschreibungen sind vielleicht nur grobe Näherungen der Wirklichkeit, aber doch oft genauer und deutlicher als rein verbale Beschreibungen. Jedoch haben auch weiche Modelle nur im Rahmen einer konsistenten Theorie einen Sinn. Und einen Wert haben qualitative Prognosen auch nur, wenn man mit dem Modell „spielen“ kann, d.h., wenn man ungenaue Größen variieren und sich so davon überzeugen kann, welchen Einfluß die Ungenauigkeit auf die qualitative Prognose hat.

Noch darunter sind auf einer dritten Stufe die Expertensysteme angesiedelt,

wenn man die Kriterien ihrer Verfechter ernst nimmt. Sie verwenden als theoretische Grundlage der Modellbildung Heuristiken, Faustregeln, vages Wissen. Das „Modell“ wird so lange gefittet, bis ein paar vorgegebene Testfälle einwandfrei löst. Diese vorgeschlagene Entwicklungsmethode für Expertensysteme ist ein inkohärentes Vorgehen mit zusammengeschnitzten Methoden und somit das miserabelste Vorgehen zur Modellbildung, das überhaupt denkbar ist. Positive Ergebnisse in komplexen Testfällen besagen nichts mangels Theorie. Hat das System auch in einem anderen komplexen Fall Erfolg? In einem „Alltagsfall“? Es ist ein gewaltiger Irrtum zu glauben, „Ich stecke vages Wissen und unzuverlässige Daten in den Computer, die Ergebnisse werden dann schon exakt sein“. Unscharfe Problemgebiete werden nicht schon allein dadurch beherrschbar, daß man sämtliche Ansprüche an Strenge, Konsistenz und Validität der Modellbildung fallen läßt.

Parnas bezeichnet als „KI-2“ die „Verwendung einer bestimmten Reihe von Programmier-techniken, die als heuristische bzw. regelbezogene Programmierung bezeichnet werden“, also im wesentlichen die Expertensystemtechnik, und sagt dazu:

„Ich halte die in der Definition KI-2 genannten Ansätze für gefährlich und finde einen großen Teil der Arbeiten auf diesem Gebiet irreführend. Die Regeln, die anhand von Untersuchungen des menschlichen Problemlösungsverhaltens aufgestellt werden, erweisen sich als inkonsequent, unvollständig und ungenau. Heuristische Programme werden durch praktisches Herumprobieren entwickelt, wobei eine neue Regel immer dann hinzugefügt wird, wenn ein Fall auftritt, der von den bisherigen Regeln nicht behandelt wird. Mit dieser Vorgehensweise entstehen meistens Programme, deren Verhalten schlecht zu verstehen und schwer voraussehbar ist. Bei KI-2-Forschern gilt diese evolutionäre Vorgehensweise bei der Programmierung als selbstverständlich und richtig. Ich habe zu solchen Programmen noch weniger Vertrauen als zu unstrukturierten herkömmlichen Programmen. Man weiß nie wann das Programm abstürzen wird.

Gelegentlich habe ich die Behauptungen eines KI-2-Forschers unter die Lupe nehmen müssen, und ich bin jedesmal enttäuscht gewesen. Bei näherer Betrachtung stellte sich heraus, daß die Heuristik eine kleine Anzahl von eindeutigen Fällen behandelte, aber nicht allgemein anwendbar war. Der Autor konnte für die Fälle, die vom Programm richtig behandelt wurden, ein spektakuläres Verhalten vorweisen. Die anderen Fälle sollten nach seiner Auffassung zukünftige Forscher beschäftigen. Tatsache ist, daß die verwendeten Techniken oft nicht allgemein anwendbar sind, und das verbesserte Programm kommt nie zustande.“

Expertensysteme können also aus mindestens vier Gründen nicht einmal weiche Modelle sein:

- Es gibt keine exakten Methoden zur Wissensdarstellung.
- Dem Wissensgebiet liegt keine sorgfältig ausgearbeitete Theorie zugrunde.
- Das Gesamtsystem läßt sich nicht experimentell absichern.
- Ein Variieren der Modellparameter ist kaum möglich. Wie soll man das in Regeln abgebildete Wissen ein wenig variieren? Selbst die Variation der Gewißheitskoeffizienten verbietet sich in einem großen Regelwerk schon allein wegen ihrer schieren Menge.

Bezeichnenderweise sind die bisher in der Praxis erfolgreichen Expertensysteme gar keine solchen im Sinne der Definition, sondern Modelle eines „harten Wissensbereichs“ (z.B. Rezepturen zur Mischung von Farben), die nur mit Hilfe der Expertensystemtechnik als Programme implementiert worden sind.

Eine gewissen Wert haben „vage“ Expertensysteme in der Hand eines Fachmannes aber doch – sie können ihm Fingerzeige, Ideen, Assoziationen liefern, auch wenn die „Ergebnisse“ sie für sich alleine genommen ohne jede Aussagekraft sind. Sie müssen vom Fachmann geeignet interpretiert und völlig unabhängig von diesem „Modell“ überprüft und durchdacht werden. Und schließlich hängen die Anforderungen an die Validität eines Modells natürlich von der Anwendung ab. Wenn man mit Hilfe eines Expertensystems eine Ölquelle findet, verzeiht man ihm gerne 20 Fehldiagnosen – die Zahl der erfolgreichen Probebohrungen ist dadurch vielleicht allemal reduziert. In der Medizin, in Kernkraftwerken oder für militärische Anwendungen gelten aber ganz andere Zuverlässigkeitsanforderungen.

4.6 Was weiß ein Experte?

Ein Expertensystem soll das Wissen eines oder mehrerer Experten des jeweiligen Gebietes enthalten. Aber wer ist Experte? Wo hört sein tatsächliches Wissen auf, wo fängt seine persönliche Meinung an? Und gerade die ist ja im Expertensystem auch gefragt. Aber da scheiden sich schnell die Geister. Wer hat noch nie von einem Streit zwischen Experten gehört? Wer hat noch nie in zwei Lehrbüchern zum gleichen Thema gegenteilige Behauptungen entdeckt? Welche Superexperte soll dann entscheiden?

Ein großes Problem ist auch, wie weit sich Expertenwissen überhaupt formalisieren läßt. Ein Computerprogramm kann nur seine vorgefertigten Schlüsse ziehen, der menschliche Experte hat dagegen fünf Sinne und einen Verstand und setzt diese auch ein. Insbesondere wenn eine Auflösung von Konflikten gesucht wird, ist Kreativität nötig, und da versagt jedes Programm.

Und wenn es gar keinen Experten gibt? „SDI wirft . . . Probleme auf, für deren Lösung wir zur Zeit keine menschlichen Experten haben. Oder gibt es heute Menschen, die bei der Betrachtung von im Flug befindlichen Raketen mit großer Sicherheit und Zuverlässigkeit echte Gefechtsköpfe von Attrappen unterscheiden können?“ [8]. Besonders wenn es um die Festlegung von Gewißheitskoeffizienten nach Erfahrungswerten oder Augenmaß geht, wer soll diese Erfahrung und dieses Augenmaß haben?

5 Wertung

5.1 Die Grenzen der Expertensystem-Technik

In keinem anderen Gebiet der Wissenschaft sind durch überzogene Werbeaussagen sowohl aus der Industrie als auch von Forschern so unrealistische Erwartungen in der Öffentlichkeit geweckt worden wie im Gebiet der „künstlichen Intelligenz“. Das betrifft auch die Erwartungen an die Leistungsfähigkeit von Expertensystemen, von denen man einige Wunder erhofft. Illusionen sind aber eine miserable Voraussetzung für eine seriöse Wissenschaftspolitik. DARPA glaubt: „Die Methoden zur Identifizierung und Mechanisierung von praktischem Wis-

sen, gesundem Menschenverstand und Fachwissen sind ausgereift und finden jetzt breite Anwendung“ (zitiert nach [5]).

Expertensysteme als standardisierte Software-Entwicklungsmethode können die Fehleranfälligkeit der Programmentwicklung reduzieren. Wie wir gesehen haben, ist die Reduktion aber sehr begrenzt und kann mit der Zunahme der Komplexität bei Vergrößerung des Programms nicht mithalten. Alle bekannten Probleme mit der quasi naturgesetzlichen Unzuverlässigkeit von Software bleiben bestehen; sie werden sogar wesentlich verschärft, wenn man Expertensysteme im eigentlichen Sinne zur Verarbeitung vagen Wissens einsetzt.

Expertensysteme sind wegen ihrer Problemlösungsstrategie für Echtzeitanwendungen nicht geeignet. Dieses Problem läßt sich durch Trimmen der Problemlösungskomponente verringern, indem man Abkürzungen der Suchstrategie einbaut. Dadurch erhöht sich aber die Komplexität des Steuermechanismus überproportional, und das bedeutet eine signifikante Abnahme der Zuverlässigkeit. Da die Komplexität auf kombinatorischer Explosion beruht, kann auch eine Weiterentwicklung der Hardwarefähigkeiten dieses Problem prinzipiell nicht lösen.

Eine der wesentlichen Anforderungen an Expertensysteme ist die Durchschaubarkeit der Entscheidungsfindung mit Hilfe der Erklärungskomponente. Sie ist unter Echtzeitbedingungen nicht zu realisieren. Wenn dem Präsidenten im Ernstfall als Unterstützung eine Ampel mit den Farben grün, gelb oder rot angeboten wird, hat er, um zu einer verantwortungsbewußte Entscheidung zu kommen, wohl kaum noch Zeit, sich stundenlang durch die Erklärungskomponente zu hangeln.

Wird in einem Expertensystem vages Wissen verarbeitet, d.h. Faustregeln oder Ungewißheitskoeffizienten, so ist es nicht mehr als Modell der Wirklichkeit brauchbar; es kann also nicht selbständig gültige Schlüsse herleiten, sondern nur Hinweise auf mögliche Folgerungen geben. Es kann also den menschlichen Experten nicht ersetzen, sondern nur unterstützen. Insbesondere ist ein solches Expertensystem in einem „autonomen System“ grundsätzlich nicht zu gebrauchen.

Ein Expertensystem kann nur auf Situationen reagieren, die sein Entwickler zumindest implizit vorgesehen hat. Ein unvorhergesehenes Ereignis setzt es sofort schachmatt; das geben sogar die Verfechter zu [4, Abschnitt 2.4]. In solchen Situationen hilft nur noch Kreativität oder Verantwortungsbewußtsein, und daß man diese maschinengerecht formalisieren kann, hat noch keiner der Reklamekünstler der „künstlichen Intelligenz“ glaubwürdig behauptet.

Wissen kann nur soweit maschinell verarbeitet werden, wie es sich formal, z.B. in Regeln und Fakten, organisieren läßt. Die Bemühungen um die vollständige Formalisierung des menschlichen Wissens haben eine Jahrhunderte alte Tradition, ohne daß eine Lösung wesentlich näher gekommen wäre.

5.2 Die Gefahren des militärischen Einsatzes

Ich will die Gefahren speziell des militärischen Einsatzes noch einmal in Form von Thesen verdeutlichen. Da es sich dabei selbstverständlich um persönliche Wertungen ohne absoluten Gewißheitsanspruch handelt, müssen diese Thesen in sich nicht notwendig konsistent sein; vielmehr folgen sie der Devise „Erstens klappt es sowieso nicht; wenn es aber doch klappen sollte, dann taugt es nichts“.

- Expertensysteme können das nicht leisten, was für die geplanten militärischen Anwendungen, insbesondere im Rahmen von SDI oder SCI, von ihnen gefordert wird. Die Unterstützung der Forschung auf diesem Gebiet bedeutet Verschwendung von Mitteln, die für die Lösung dringender Menschheitsprobleme gebraucht werden.
- Sollten Expertensysteme aber doch das eine oder andere Teilproblem lösen, so treten alle Vorbehalte hinsichtlich der Zuverlässigkeit von Software in Kraft:
 - Unvermeidbarkeit von Programmfehlern,
 - unvollständige Testmöglichkeiten,
 - unbeherrschbare Komplexität,
 - begrenzter Anwendungsbereich,
 - fragwürdige Aussagekraft der Ergebnisse,

und der Anwender wird in seiner moralischen Pflicht zur Entscheidung überfordert, weil er die Empfehlungen des Systems unter Zeitdruck akzeptieren muß.

- Sollte es Fortschritte auf dem Gebiet der Expertensysteme geben, werden diese mit ziemlicher Sicherheit geheimgehalten. Dadurch wird die Wissenschaft immer mehr der öffentlichen Kontrolle entzogen; Wissenschaftler, die unabhängig auf diesem Gebiet forschen, geraten unter Druck zur Geheimhaltung oder Mitarbeit, werden durch Geheimdienste erpreßbar. Der Weg zum militärischen Überwachungsstaat ist kurz. Schließlich wird verhindert, daß die neuen Erkenntnisse zur Lösung ziviler Probleme verwendet werden. Diese Gefahren bestehen nicht nur bei tatsächlichen, sondern auch bei vermeintlichen oder erhofften Fortschritten.

Literatur

- [1] K.-P.Adlassnig: Ein einfaches Modell zur medizinischen Diagnostik mit fuzzy Teilmengen. *EDV in Medizin und Biologie* 13(1982), 12-16.
- [2] J.Bickenbach, R.Keil-Slawik, M.Löwe, R.Wilhelm (Hrsg.): *Militarisierte Informatik*. Schriftenreihe Wissenschaft und Frieden, Berlin 1985.
- [3] K.H.Böker(Hrsg.): *Verantwortung des Informatikers – militärische Anwendungen der Informatik*. Dokumentation eines Seminars an der Universität-Gesamthochschule Paderborn 1987.
- [4] F.Hayes-Roth, D.A.Waterman, D.B.Lenat (ed): *Building Expert Systems*. Addison-Wesley, Reading Mass. 1983.
- [5] F.Höfting, S.Rudolph: Expertensysteme – Feldherren der Zukunft? In [3].
- [6] IBM: *Expert System Development Environment – User Guide*. SH20-9608-1, Second Edition (July 1986).
- [7] R.Keil-Slawik: „Intelligente“ Kriegsführung ohne Menschen? In [2].
- [8] D.L.Parnas: *Software Wars*. Kursbuch 83 (1986), 49-69.
- [9] D.G.Saari: Symmetry, Voting, and Social Choice. *The Mathematical Intelligencer* 10 (1988), 32-42.
- [10] G.Shafer: *A Mathematical Theory of Evidence*. Princeton Univ. Press 1976.
- [11] R.Wilhelm: Strategic Computing Initiative: Ein kurzer Überblick. In [2].