

Juristische „Expertensysteme“ – Gedanken zwischen Theorie und Praxis

Andreas Günther

Teil 3: Anmerkungen zur Realisierung eines subsumtionsunterstützenden Dialogsystems

III. Die Programmierwerkzeuge

1. Aufgabenstellung

2. Einordnung – State of the art

a. Erste Konzepte

b. Expert System Shells

c. Autorensysteme

Benutzerfreundliche Autorensprachen, die sich speziell zum Aufbau von Dialogsystemen eignen, werden ebenfalls seit längerem kommerziell entwickelt und vertrieben. Hier sind z.B. Systeme wie SEF³² und E_T_AS³³ einzuordnen³⁴. Auch das Standardprogramm HyperCard³⁵ läßt sich zur Erstellung von Dialogsystemen einsetzen, geht aber in seinen Anwendungsmöglichkeiten weit darüber hinaus und ist als sog. „Bildschirm-Informationssystem“ kein Autorensystem im klassischen Sinne. Seit einiger Zeit entsteht nun eine neue Generation juristischer Entwicklungssysteme, die die Eigenschaften von Autoren-, Lern- und Datenbanksystemen in sich vereinigen wollen (so insb. JURBASE³⁶ und JULE³⁷). Als (wenn auch nicht typischer) Vertreter der Kategorie der Autorensysteme stand dem Verfasser eine Version des in Tübingen entwickelten DIALTUE(2) zur Verfügung.

3. XPRO und DIALTUE(2) – Zwei „Philosophien“ im Vergleich

Im folgenden soll nun näher auf die Eigenschaften beider Tools und die Erfahrungen, die

der Verfasser in Bezug auf die Eignung beider „Denkzeuge“ zur Erstellung von Dialogprogrammen gemacht hat, eingegangen werden.

a. XPRO

XPRO³⁸ soll ein Werkzeug zur Entwicklung von Expertensystemen (bzw. dem, was man z.Zt. unter „Expertensystemen“ versteht), also ein Expert System Shell, sein. Das Ziel von XPRO ist hauptsächlich didaktischer Art: es soll die grundlegenden Eigenschaften von Expertensystemen und Shells praktisch aufzeigen und dadurch, daß es kostenlos erhältlich ist, vor allem Schulen, Universitäten und anderen Ausbildungseinrichtungen den Zugang zu diesem Bereich der Softwaretechnologie eröffnen.

Großer Wert wurde bei der Entwicklung von XPRO darauf gelegt, die innere Struktur von Expertensystemen zu simulieren; ansatzweise sind also die heute für ein Expertensystem charakteristischen Elemente Erklärungskomponente, Interferenzkomponente, Benutzeroberfläche, Datenbasis und Wissens-

nicht möglich.⁴⁰ Der Benutzer lernt also durch die Arbeit mit

32 Expert-Teach-Autorensystem; es ist 1987 von Kurzbach & Rehr (Erlangen) entwickelt worden (beschrieben in H. Fiedler/G. Oppenhorst, Anm. 27, S. 175ff).

33 Einen Überblick über weitere Autorensprachen bieten H. Brenner, Anm. 4, S. 75ff (98ff); H. Fiedler/G. Oppenhorst, Anm. 27, S. 161ff; vgl. auch S. Grundmann, Anm. 8, S. 580ff; R. Schulmeister, Autorensysteme und Alternativen, in C. Eberle (Hrsg.), Informationstechnik in der Juristenausbildung, 1989, S. 161ff.

34 Es wurde von Bill Atkinson insb. für den Apple Macintosh entwickelt und beruht auf dem sog. Hypertext-Ansatz (hierzu vgl. H. Poxrucker, HyperCard – das neue Universalprogramm für den Macintosh, IuR 1988, S. 266ff, 322ff; A. Vichr, Universalprogramm Hypercard, NJW-CoR 2/1989, S. 19ff; H. Fiedler/G. Oppenhorst, Anm. 27, S. 180ff).

35 R. Halter (Autor des Programms) beschreibt JURBASE in Jura 1989, S. 161f.

36 Hinweise auf JULE finden sich im jur-PC Newsletter 1989, S. 32f und bei M. Herberger, Anm. 27, S. 83ff (109f).

37 XPRO ist eine Entwicklung von Ralph Langner (Langner Expertensysteme, Hamburg); es kann kostenlos benutzt, kopiert und weitergegeben werden und steht für den „Download“ in der Mailbox zur Verfügung.

38 Zu den einzelnen Eigenschaften des Systems siehe das XPRO-Handbuch von R. Langner, das sich auf der Systemdiskette als XPRO.DOC befindet; vgl. ferner zu XPRO: H. Fiedler/G. Oppenhorst, Anm. 27, S. 206f; R. Fischer, Anm. 27, S. 159ff; Computer Persönlich Nr.20 – 14.9.1988, S. 118; jur-PC Newsletter 1989, S. 63ff.

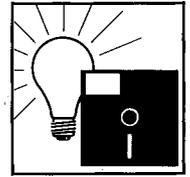
39 Zu den Suchstrategien R. Langner, im XPRO-Handbuch S. 28f.

40 Vgl. hierzu T. Gordon/G. Quirchmayr, Anm. 29, S. 123ff (124ff); L. Philipps, Simulationen und Spiele als juristische Lernprogramme, in Seegers/Haft (Hrsg.), Rechtsinformatik in den achtziger Jahren (ARI Bd. 20), 1984, S. 219ff (228f).

Andreas Günther hat im Rahmen des Seminars „Juristische Expertensysteme“ (Prof. Brehm Bayreuth) einen Dialog zum Thema „Vertragsschluß“ entworfen, der in der Mailbox verfügbar ist.

Architektur von XPRO

basis vorhanden.³⁹ Die Wissensbasis ist grundsätzlich regelorientiert, d.h. sie beruht im Ansatz auf logischer Programmierung. Die verwendete Suchstrategie ist das Backward-Chaining; Forward-Chaining ist



XPRO Schlußfolgerungen auf der Basis der Prädikatenlogik kennen, wie sie für deklarative Programmiersprachen (wie z.B. Prolog) kennzeichnend sind. Allerdings muß er sich trotzdem Gedanken über den prozeduralen Ablauf des Programms machen, damit sinnvolle Ergebnisse erzielt werden. Dies liegt unter anderem daran, daß eine strikte Trennung von Wissensbasis, Interferenzkomponente und Benutzeroberfläche nicht konsequent durchgehalten wird. Eine Stärke des Systems ist, daß es sehr einfach zu handhaben und die Programmierung schnell erlernbar ist. Es wird seinem Ziel, Studenten als Experten in die Rolle des Knowledge Engineers zu versetzen, durchaus gerecht, da

Studenten als „Knowledge Engineer“

es die Eigenschaften der zur Zeit verwendeten Expert System Shells hinreichend einfach abbildet und simuliert. Erwähnt werden soll hier nur die vorgesehene Trace-Funktion und die Möglichkeit, während des Dialoges „Warum“-Fragen zu stellen und dadurch ohne Eingriff in den logischen Ablauf der Konsultation die gerade untersuchten Regeln zu erhalten.

Durch die regelbasierte Programmierung bietet das System dem juristischen Benutzer die Möglichkeit, abstrakte Rechtsätze in Regeln darzustellen. Eine rein deklarative Logik läßt sich zwar bei der Konzeption eines subsumtionsunterstützten Dialoges nicht durchhalten, vom Ansatz her regt das System den Benutzer jedoch dazu an, sich über die Regel-Ausnahme-Struktur des Gesetzes Gedanken zu machen und darauf aufbauend, die Regeln möglichst

Nachdenken über Regel und Ausnahme

dogmatisch sauber zu formulieren und auch Beweislastfragen mit zu berücksichtigen.⁴¹ Nach den Erfahrungen des Autors sind auch nachträgliche Änderungen und Erweiterungen der Regelbasis relativ problemlos möglich. Durch eine sinnvolle Bezeichnung der logischen Objekte und Werte lassen sich die Regeln in ihrer Formulierung der natürlichen Sprache annähern und so gestalten, daß sie auch für einen anderen Systembenutzer als den ursprünglichen Programmautor schnell verständlich werden.

Als eine entscheidende Schwäche der Shell stellte sich die fehlende Unterstützung bei der Bildung von logischen Modulen heraus. Unterprogramme im herkömmlichen Sinne sind einer regelorientierten Logik zwar grundsätzlich fremd; die durch die einfache

Nicht möglich: (Echte) Module

Ausgestaltung der logischen Programmiermöglichkeiten bedingte, gänzlich fehlende Erleichterung einer gezielten Modularisierung (abgesehen von der Möglichkeit, mehrere Wissensbasen zusammen zu laden) über die Bildung von Regeln und rein optische Gestaltung des Listings hinaus, erweist sich jedoch bei der Erstellung von umfangreichen Dialogen als äußerst hinderlich und setzt insbesondere der Zusammenschaltung von ver-

schiedenen Regelwerken enge Grenzen.

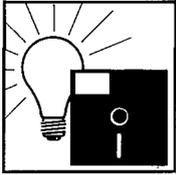
Dadurch, daß das Shell primär dazu dient, dem „Wissensautor“ die innere Struktur eines Expertensystems aufzuzeigen, sind die Gestaltungsmöglichkeiten der Benutzeroberfläche ausgesprochen mager ausgefallen. Die Gestaltung von Frage- und Antworttexten ist sehr unkomfortabel und unflexibel; ein Dialog läßt sich nur schwer ansprechend realisieren. Bei der Benutzung der fertigen Dialogsysteme stellte es sich ferner als hinderlich heraus, daß es nicht möglich ist, einzelne Fragen und Antworten innerhalb eines Dialoges rückgängig zu machen. Erfolgte eine falsche Eingabe oder will man einen einzelnen Zweig des „Dialogbaumes“ zurückverfolgen und noch einmal anders bearbeiten, so muß man den Dialog beenden und von vorne beginnen. Daran ändert auch die Tatsache nichts, daß es möglich ist, das Ziel des Dialoges bei der Programmierung der Wissensbasis offen zu lassen und am Anfang des Dialoges zu definieren.

Alles in allem ist aber festzuhalten, daß XPRO sein didaktisches Ziel erreicht. Es soll nicht ein möglichst komfortables Werkzeug zur Erstellung von Dialogsystemen sein, sondern

Summa summarum: XPRO ist didaktisch nützlich

seinem Benutzer einen Einblick in die Welt der regelbasierten

⁴¹ Mit XPRO de luxe bietet Langner nun ein immer noch preisgünstiges Expert System Shell größerer Leistungsfähigkeit an.



Wissenserfassung vermitteln.⁴² Gerade auch in der Anwendung auf juristische Fragestellungen stößt der „studentische Wissens-acquisiteur“ schnell auf die Grenzen und Probleme der modernen KI-Technologie, die auch umfangreichere Shells nur schwerlich überwinden dürften.

b. DIALTUE(2)

DIALTUE (2.1)⁴³ wird hier zwar als Autorensystem betrachtet, im Grunde hat das Tübinger Dialogverfahren aber mit den herkömmlichen Autorensprachen nicht viel mehr gemein als die Zielsetzung⁴⁴; es soll auch dem programmieretechnisch „unbelasteten“ Autor ermöglichen, relativ einfach und komfortabel Computerdialogprogramme, sei es als Lehr- und Lernsysteme oder als DSS, zu entwerfen.

Herkömmliche Autorensprachen bieten dem Autor eine hohe Benutzeroberfläche, die speziell auf die komfortable Gestaltung der einzelnen Dialogkomponenten, auf die Probleme der Antwortprüfung und u.U. auf die Beurteilung des Lernerfolges zugeschnitten ist. Die Systeme ermöglichen zwar Verzweigungen und Sprünge nach Antwortanalysen oder nach Menüauswahl innerhalb der prozeduralen Programmabläufe und fördern dadurch die Vernetzung von Strukturen; die eigentliche Adressverwaltung bei Sprungbefehlen muß der Autor allerdings selber vornehmen.⁴⁵ Diese Arbeit nimmt das Tübin-

Wissensbäume ab. Es fördert somit die Ablage von Wissen in vermaschten Strukturen. Hintergrundgedanke war es, sich möglichst eng an der Organisation von Wissen im menschlichen Kopf zu orientieren und sich der Denkweise des Juristen, Strukturen zu bilden, anzupassen. Der Autor muß sein Wissen in Textteile zerlegen und diese Textteile werden im Computer in einer Baumstruktur abgelegt, wobei jeder Textteil ein sogenannter Knoten innerhalb der aus mehreren Schichten bestehenden Baumstruktur ist.⁴⁶ Das so in Form von automatisch verwalteten Baumstrukturen niedergelegte Wissen läßt sich ferner in Regeln umwandeln und für Expert System Shells nutzbar machen, DIALTUE(2) kann

DIALTUE(2): Mögliche Schnittstelle zu Expertensystemen

daher sogar als Schnittstelle zu Expertensystemen eingesetzt werden.⁴⁷

Ausgesprochen angenehm ist die Arbeit mit DIALTUE(2) dadurch, daß der Autor durch die Vorgabe der Baumstrukturen ohne Inhalt in hierarchischer Form von all den syntaktischen Problemen befreit wird, die die Verwendung einer Programmiersprache, sei sie prozedural oder deklarativ und von einer Expert System Shell umgeben, mit sich bringt; er muß sich nur an der Gestaltung des Dialoges orientieren.

Dieser Vorteil wird allerdings dadurch erkauft, daß der Autor einen relativ exakten Programmablaufplan erstellen muß. Denn zum einen bedingt die Erstellung einer Baumstruktur eine strenge Disziplin und zum anderen erfordert der spezielle

Notwendig: Exakter Ablaufplan

Aufbau von DIALTUE(2) eine genaue Protokollierung der einzelnen Schritte, um nicht durch die Vielzahl der Schichten und Knoten den Überblick zu verlieren.⁴⁸ Das Wissen muß im voraus sehr weit durchstrukturiert sein, und die Erstellung eines weitverzweigten Netzes erlaubt ein unmittelbares Programmie-

42 DIALTUE (1.0) wurde im Herbst 1982 von Manfred Gerblinger am Lehrstuhl von Prof. Dr. F. Haft in Tübingen entwickelt und als Version 1.1 von Dr. Gerhard Ringwald für den Einsatz auf PCs angepaßt. Dr. Ringwald entwickelte und programmierte dann 1985 eine wesentlich veränderte und benutzerfreundlichere Version 2.0 (H. Cames, Nonverbale Expertensysteme – Der Ansatz Tübinger Dialog, in G. Ringwald (Hrsg.), Perspektiven formaler Methoden im Recht (NMIR Bd. II), 1986, S. 109ff (114, 117f); zur Entstehung des Programms auch F. Haft, Anm. 8, S. 148ff (151f)). Mit geringen Modifikationen wird DIALTUE(2) seit 1988 von der Firma USEWARE GmbH (7049 Steinenbronn) vertrieben.

43 Von seinen Programmierern wird es sogar eher als das Gegenteil der klassischen Autorensysteme betrachtet (G. Ringwald/J. Sulz, Automatische Datenverarbeitung im Recht, stud.iur. 1988, S. 82ff (84)).

44 H. Brenner, Anm. 4, S. 75ff (98, 103).

45 Eine genaue Beschreibung der Funktionsweise von DIALTUE(2) findet sich bei: A. Leicht, „Expertensysteme“ im Unterricht – das Tübinger Dialogverfahren DIALTUE, IuR 1987, S. 201ff; G. Ringwald, Das Tübinger Dialogverfahren (DIALTUE(2)) – eine Schnittstelle zu Expertensystemen, in Erdmann/Fiedler/Haft/Traummüller (Hrsg.), Computergestützte Juristische Expertensysteme (NMIR Bd. I), 1986, S. 111ff; H. Cames, Anm. 43, S. 109ff (117ff); vgl. auch H. Fiedler/G. Oppenhorst, Anm. 27, S. 184f.

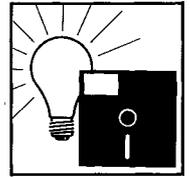
46 Hierzu G. Ringwald, Anm. 46, S. 111ff; H. Cames, Anm. 43, S. 109ff (121f).

47 A. Leicht, Anm. 46, S. 201ff (202).

48 Ein exemplarisches Beispiel einer solchen Übersicht zum Programmablauf findet sich bei A. Leicht, Anm. 46, S. 201ff (203).

Automatische Verwaltung hierarchischer Wissensbäume

ger Dialogverfahren dem Benutzer durch die automatische Verwaltung der hierarchischen



ren am Computer nicht. Es fehlen leider Hilfen, diesen Entwurf am Bildschirm vorzunehmen, ja selbst die Darstellung des Inhalts der kompletten Baumstruktur als Ausdruck läßt insoweit zu wünschen übrig, als die eigentliche Zeichnung der Struktur noch von Hand erfolgen muß.⁴⁹ Nachteilig ist ferner, daß eine einmal aufgebaute Struktur nur sehr schwer veränderbar ist. Nachträgliches Einfügen von Knoten oder die Veränderung ganzer Teile der Struktur ist kaum möglich und macht oft eine komplette Neuprogrammierung notwendig. Auch lassen die Gestaltungsmöglichkeiten innerhalb der Textfenster noch zu wünschen übrig, sowohl durch deren Begrenztheit als auch durch einen dem Wordprocessor gewöhnten Benutzer recht unkomfortabel erscheinenden Umgang mit den Texten. Hier wäre eine Kombination mit den graphischen Möglichkeiten der herkömmlichen Autorensprachen wünschenswert.⁵⁰

Eine Schwäche des Systems ist es zwar, daß der Benutzer eines Dialoges stets im allerersten Textteil einsteigen und sich über die entsprechenden Antworten an das ihn interessierende Problem heranarbeiten muß, denn eine Abkürzung der Beweisführung durch die Bildung von sog. Metaregeln, die über Regeln

Nicht verfügbar: Metaregeln

oder Fakten möglich wäre, kennt das System nicht.⁵¹ Der Benutzer kann allerdings innerhalb des Dialoges einzelne Antworten wieder rückgängig machen und so ganze Zweige zurückverfolgen. Für die Erstellung des vorliegenden Dialoges erwies sich die relative Starrheit der Bindung der Textteile unter-

einander und die Tatsache, daß das System keine Fakten kennt, nicht als nachteilig; dies mag in anderen Fällen allerdings zu Problemen führen.

Zusammenfassend läßt sich – ohne der anschließenden Gegenüberstellung vorzugreifen – feststellen, daß DIALTUE(2) zwar weiterhin verbesserungsfähig, aber auch verbesserungswürdig ist.⁵² Das Konzept eines

Summa summarum: DIALTUE(2) verdient Aufmerksamkeit

leeren hierarchischen Baumes eignet sich gut für den Aufbau großer und stark vernetzter Strukturen, wie sie Dialogprogramme erfordern, ohne in die Tiefen der prozeduralen oder deklarativen Programmierung hinabsteigen zu müssen.

c. Vergleich

Ein selbständiger Ausschnitt des Dialogsystems⁵³ wurde in XPRO und in DIALTUE(2) realisiert, so daß ein direkter Vergleich möglich ist.⁵⁴

Hierarchische Wissensstrukturen lassen sich sowohl in Regeln erfassen, als auch in Form einer automatisch verwalteten Baumstruktur. Die entscheidenden Unterschiede der beiden „Philosophien“ liegen dabei in der

Regeln vs. Baumstruktur

Verweisungstechnik und der Strukturierung des im Dialog enthaltenen juristischen Wissens. In XPRO wird die Ver-

weisung durch die Gleichheit von logischen Objekten und deren Werten bewirkt und so der Aufruf von Untersätzen zur Aufschlüsselung von Tatbestandsmerkmalen, die in einer Hauptregel stehen, erreicht. In DIALTUE(2) geschieht dies durch die Angabe im System, an welche Stelle der Dialoghierarchie das System springen soll, wenn der Benutzer eine bestimmte Antwort eingibt. Hier muß also für jede Antwort eine Verweisung im Dialog vorgesehen sein, wohingegen in XPRO diese Verweisung nicht explizit geschieht, sondern durch die regelorientierte, logische Programmierung der Wissensbasis erreicht wird. Der Arbeitsaufwand angesichts der Potenzierung von möglichen Antworten bei der Programmierung größerer Rechtsgebiete ist bei der Verwendung von DIALTUE(2) aber nur scheinbar größer, da die Verweisung innerhalb der Baumstruktur im

49 So auch H. Brenner, Anm. 4, S. 75ff (104).

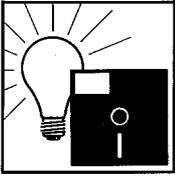
50 G. Ringwald, Anm. 46, S. 111ff (115).

51 Zu denkbaren Weiterentwicklungen G. Ringwald/J. Sulz, Anm. 44, S. 82ff (84); angeregt durch die Überlegungen von Ringwald zur automatischen Subsumtion in einer juristischen Dialogsprache und durch sein Tübinger Dialogverfahren wurde z.B. die Entwicklung von KONDIAL (Konstanzer Dialog Autoren und Lernsystem) durch G. Spieß an der Universität Konstanz; es dient jedoch z.Zt. ausschließlich zu Ausbildungs- und Anschauungszwecken und will nicht mit professionellen Systemen konkurrieren (H. Fiedler/G. Oppenhorst, Anm. 27, S. 185).

52 Hierfür bot sich die Frage an, ob eine Willenserklärung vorliegt, die abgegeben worden und dem Empfänger zugegangen ist; dieser Teil ging dann zweimal – als Angebot und Annahme jeweils leicht modifiziert – in den Dialog zum Vertragsschluß ein.

53 Vgl. S. Grundmann, Anm. 6, S. 175ff (183ff), der Prolog und das Tübinger Dialogverfahren gegenüberstellt.

54 S. Grundmann (ebenda), der von einem erheblich größeren Arbeitsaufwand ausgeht, lag bei seinem Vergleich nur DIALTUE Version 1.1 vor, bei der tatsächlich noch alle Verweisungen über Schlüsselnummern in einer Steuerleiste von Hand vorgenommen werden mußten.



Normalfall automatisch vorgenommen wird und nur die „außerordentlichen“ Querverbindungen manuell erstellt werden müssen.⁵⁵

Der Verfasser empfand diese grundsätzlich automatische Verwaltung der Verweisung als angenehm. Denn die Konzeption der impliziten Verweisungen wird in XPRO dadurch erschwert, daß sich der Autor trotz Regelorientierung Gedanken über den prozeduralen Ablauf machen muß, um in der Struktur des Programmes versteckte Verweisungsfehler aufzuspüren. Diese Fehlerquelle wird im teils automatisch, teils manuell erstellten Netz von expliziten Verweisungen durch

strukturiert werden und die Struktur der Rechtssätze läßt sich direkt aus den Regeln wiedererkennen. Diese Möglichkeit

„Widerspiegelung“ der Struktur – ja oder nein?

bietet das Tübinger Dialogverfahren nicht. Dies muß allerdings kein entscheidender Nachteil sein. Denn der Programmator sollte sich natürlich im Vorfeld der Konzeption des Dialoges genau über die Struktur der behandelten Rechtsgebiete im klaren sein. Er muß dann allerdings einen Schritt weitergehen und kann nicht die Struktur als solche eingeben, sondern muß sich an dem eigentlichen Prozeß der Subsumtion orientieren und diesen den einzelnen Tatbestandsmerkmalen entsprechend in seiner Baumstruktur abbilden; er programmiert den Dialog (der normalerweise als Entscheidungsfindung in seinem Kopf abläuft) praktisch direkt und nicht über den Umweg der zugrundeliegenden Regeln. Die Ablage von Wissen mittels des Tübinger Dialogverfahrens ist also insofern tatsächlich „expertennäher“, als sie sich z.T. eher an den eigentlichen Gedankengängen im Kopf eines geübten Juristen orientiert und ihn nicht zwingt, Regeln zu bilden, die er schon so weit verinnerlicht hat, daß er ihnen, wenn überhaupt⁵⁶, nur noch unbewußt folgt.⁵⁷

Für den Benutzer eines Dialogprogramms, das mit einem der beiden Systeme erstellt wurde, ist das Ergebnis – abgesehen von konkreten Bedienungsunterschieden bei der Antworteingabe – jeweils identisch, beide grundlegenden Unterschiede in der Philosophie machen sich nur bei der Programmkonzeption für den Autor bemerkbar.

Im vorliegenden Fall empfand der Verfasser die Erstellung mit DIALTUE(2) als angenehmer. Dies mag daran liegen, daß der Subsumtionsprozeß im Falle der Willenserklärung schon so weit verinnerlicht ist, daß nicht mehr explizit auf die entsprechenden Regelformulierungen zurückgegriffen wird, sondern das interne „Frage-Antwort-Spiel“ im (ldausurmäßigen) Gutachtenstil eher automatisch abläuft. Dies kann sich anders darstellen, wenn es darum geht, kompliziertere Regel-Ausnahme-Zusammenhänge von Normen in einem Dialog zu erfassen. Dann mag es durchaus sinnvoll sein, den Schritt der Regelbildung bei der Konzeption des Dialoges nicht zu überspringen, sondern in das Programm eingehen zu lassen. Der Dialog dient dann zur Unterstützung oder Übung des eigentlichen Subsumtionsprozesses, wohingegen das Programm selber dazu verwendet werden kann, die Struktur der behandelten Regelungen aufzuzeigen.

Eine allgemeingültige Aussage über die Eignung der Systeme

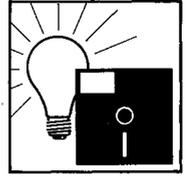
Verwaltung von Verweisungen

DIALTUE(2) vermieden. Gerade in umfangreichen Baumstrukturen besteht damit nicht die Gefahr, daß das System einer Verweisung folgt, an die der Ersteller nicht gedacht hat. Ein entscheidender Nachteil ist allerdings, daß der Autor aus den oben genannten Gründen die eigentliche Baumstruktur von Hand mitprotokollieren muß, da sich die Verweisungen nicht durch einen simplen Ausdruck von Wenn-Dann-Regeln, sondern nur in einer graphischen Baumstruktur übersichtlich darstellen lassen. Ein weiterer grundlegender Unterschied ist, daß in XPRO die Struktur der abgebildeten Rechtsvorschriften selber mit in das Programm eingeht. Die Strukturen des Gesetzes, d.h. Rechtssätze als allgemeine Regeln und Ausnahmen, lassen sich direkt in den Regeln der Wissensbasis abbilden. So kann das Programm von Hauptbegriffen zu Unterbegriffen, von Unterbegriffen zu noch tiefer liegenden Unterbegriffen hin

55 Hier sei noch einmal auf H.L. Dreyfus/S.E. Dreyfus (Mind over Machine, s. Teil 1, Anm. 15 und 34) verwiesen, die den regelbasierten Ansatz grundsätzlich verwerfen. Expertenverhalten ließe sich nicht durch das Merkmal der Regelkompetenz erklären. Vielmehr sei ein Experte dadurch gekennzeichnet, daß er über eine „effizient organisierte Bibliothek von Spezialfällen verfüge“, in die die konkrete Situationswahrnehmung mittels eines „holistischen Ähnlichkeitsvergleichs“ zur Entscheidungsfindung eingeordnet werde (vgl. M. Drücker, Anmerkungen zu Dreyfus' „Mind over Machine“, IuR 1987, S. 205ff). Mag diese Kritik an den Grundannahmen der KI-Forschung noch so berechtigt sein, auf der Ebene, um die es hier geht, zeigt die praktische Erfahrung, daß auch regelbasierte Dialogsysteme eine nützliche Hilfe auf dem Weg zum Experten (vgl. das Dreyfus'sche Skill-Modell) sein können.

56 Die Bezeichnung von DIALTUE(2) als Schnittstelle zu Expertensystemen (G. Ringwald, Anm. 46, S. 111ff) ist insofern durchaus gerechtfertigt.

57 Vgl. G. Ringwald/J. Sulz, Anm. 44, S. 82ff.



unter Berücksichtigung dieser beiden grundlegenden Unterschiede läßt sich also nicht

Worauf kommt es letztlich an?

treffen. Eine Entscheidung hängt letztlich von folgenden Punkten ab: ob das zu programmierende Rechtsgebiet sinnvoller Regelbildung zugänglich ist, inwieweit diese Regeln u.U. im Gesetz schon vorgegeben sind und nur noch abgebildet werden müssen, und inwieweit der Autor den im Dialog zu durchlaufenden Subsumtionsprozeß schon selber verinnerlicht hat und die Regelbildung für ihn eher ein Schritt zurück wäre. Neben den grundsätzlich verschiedenen Ansätzen sei noch auf Gesichtspunkte hingewiesen, die nicht durch die „Philosophie“ der Werkzeuge bedingt sind, sondern in deren konkreter Ausgestaltung begründet liegen und sich bei der Verwendung komfortablerer Expert System Shells anders darstellen. Keine großen Unterschiede zeigen sich in den Möglichkeiten der konkreten Gestaltung des Dialoges; hier bieten beide Systeme dem Autor keine besonders komfortablen Werkzeuge (im Gegensatz zu den herkömmlichen Autorenspra-

Zusatzkriterien (außerhalb der „Werkzeug-Philosophie“)

chen). Auch die Beschränkung auf feste Antwortvorgaben, die z.B. durch sog. String-Analysen überwunden werden könnte⁵⁸,

ist beiden Systemen eigen. Eine gezielte Modularisierung, hier zum Beispiel im Hinblick auf die an mehreren Stellen vorkommende Prüfung einer Willenserklärung, ist in DIALTUE(2), bedingt durch die Gesamtkonzeption, noch weniger möglich als in XPRO. Ein weiterer Nachteil des Tübinger Dialogverfahrens ist das Fehlen von Fakten zur Beweisabkürzung und der Zwang, den Dialog immer komplett abarbeiten zu müssen und nicht in der Mitte einsteigen zu können. XPRO hingegen kennt Fakten und gestattet es dem Benutzer, am Beginn des Dialoges Zwischenziele frei zu definieren. Diese Nachteile für den Dialogbenutzer von DIALTUE(2) werden allerdings dadurch aufgewogen, daß er im Fragebaum hin und herwandern kann, also in der Lage ist, Antwortwege zurückzuverfolgen und neu zu beschreiten. Dies ist insbesondere zur Verdeutlichung komplexer Zusammenhänge bei der Benutzung des Dialogsystems von großem Wert. Für den Ersteller des Computerdialoges bietet DIALTUE(2) den Vorteil, daß es menuegesteuert und somit sehr einfach und übersichtlich zu bedienen ist. Es

Austesten beim Entwickeln: Leichter mit DIALTUE

nimmt dem Benutzer die gesamte Dateiverwaltung ab und ermöglicht ferner ein schnelles und problemloses Austesten der bisher erstellten hierarchischen Strukturen während der eigentlichen Entwicklung. Zur Programmierung in XPRO bedarf es hingegen noch eines gesonderten Text-Editors; durch das somit notwendige

„Rapid Prototyping“: Leichter in XPRO

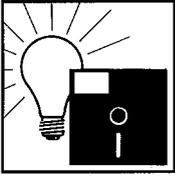
„Hin- und Herladen“ wird die Fehlersuche und Korrektur relativ mühsam und zeitaufwendig. Allerdings gestattet XPRO ein nachträgliches Erweitern und Umstellen der Dialogstrukturen und erlaubt dem Autor eher einmal ein „Drauflosprogrammieren“ oder „rapid prototyping“ als DIALTUE(2), abgesehen von den oben schon erwähnten Vorteilen in der Dokumentation der eigentlichen Dialogstruktur, die in DIALTUE(2) von Hand vorgenommen werden muß.

Ein umfassendes, abschließendes Urteil läßt sich nicht fällen. DIALTUE(2) mag sich zwar eher an den absoluten Computereinrichter, der Lehr- und Lernprogramme schreiben will, wohingegen XPRO auf den „Vorwissenden“, der an den praktischen Anwendungen moderner KI-Technologie interessiert ist, abzielt. Für die Erstellung von Dialogprogrammen sind beide insgesamt gleich gut bzw. gleich schlecht geeignet; für welche „Philosophie“ sich ein Jurist, der subsumtionsunterstützende Dialogprogramme entwerfen will, zweckmäßigerweise entscheidet, sollte – neben Vorwissen und persönlicher Vorliebe – nach den Erfahrungen des Verfassers nicht unerheblich von der Strukturierung des zu erfassenden Rechtsgebietes abhängen.

4. Anforderungen an eine juristische Dialogsprache

Ein Patentrezept für ein Werkzeug zur Entwicklung subsumtionsunterstützender Dialogsysteme gibt es sicher nicht. Die hier wiedergegebenen Erfahrun-

⁵⁸ G. Ringwald, Automatisierte Subsumtion, DSWR 1984, S. 29ff (32).



gen haben gezeigt, daß unterschiedliche Ansätze für unterschiedliche Themengebiete und unterschiedliche Autorengruppen durchaus ihre Berechtigung haben.

Allgemein läßt sich jedoch feststellen, daß entsprechende „Sprachen“ einfach und schnell erlernbar sein müssen. Juristen verfügen oft nicht über ausgefeilte Programmierfertigkeiten, Programmierern hingegen fehlen die juristischen Kenntnisse. Die Beschreibung des Dialogs muß also ohne größeres EDV-Wissen möglich sein.⁵⁹ Daß dies auch im Wege einer regelbasierten Programmierung möglich ist, hat XPRO grundsätzlich gezeigt. Voraussetzung dafür ist allerdings, daß ein bequemes, möglichst an der Umgangssprache orientiertes Formulieren von Regeln und Ausnahmen, eine Definition von Objekten und die Verfolgung von Schlußketten möglich sind.⁶⁰ Ferner sollte großer Wert auf die Flexibilität des Systems gelegt werden, um es offen für spätere Veränderungen, Verbesserungen und Anpassungen des Lösungsweges zu halten.⁶¹

Hinzukommen muß die Möglichkeit, die Dialog- und Benutzerführung möglichst ansprechend gestalten zu können. Sinnvolle Tools müssen es gestatten, schnell und komfortabel Dialogwindows aufzubauen und längere Texte, die ursprünglich aus externen Quellen stammen, zu übernehmen. Einen

chen (z.B. in juris oder BGH-DAT) vorformuliert und in seinen Dialog einbindet. Er könnte dann dem Benutzer an der für das Problem entscheidenden Stelle auf Knopfdruck das relevante Dokumentenmaterial zur Verfügung stellen, ohne daß es der Mühe einer eigenen Recherche bedürfte.⁶²

Die Entwicklungsmöglichkeiten subsumtionsunterstützender Dialogsysteme profitieren nicht zuletzt von den Errungenschaften, die auf anderen Gebieten erzielt werden. Insbesondere von der Entwicklung zunehmend natürlichsprachlicherer Systeme und der Forschung auf dem Gebiet der höheren juristischen Methodenlehre im Rahmen der Rechtsinformatik dürften in Zukunft entscheidende Impulse ausgehen.

(Der Beitrag wird fortgesetzt.)

Was der Jurist sich wünschen würde ...

weiteren Schritt in Richtung auf aktive Informationssysteme würde es bedeuten, wenn der direkte Anschluß an Dokumentationssysteme möglich wäre. Denkbar ist, daß der Autor des Dialogprogramms an bestimmten Stellen Datenbankrecher-

59 R. Traunmüller, Anm. 14, S. 22If (227).

60 Zur Problematik und Unterscheidung von „offenen“ und „geschlossenen“ Systemen G. Ringwald, Perspektiven formaler Methoden im Recht, in Ringwald (Hrsg.), Perspektiven formaler Methoden im Recht (NMIR Bd. II), 1986, S. 9ff (28ff).

61 G. Ringwald/J. Sulz, Anm. 44, S. 82ff (84).