

SOPHOS 1.0 – Eine Wissensdatenbank (Teil 1)

Matthias Kraft

sophos (griechisch): geschickt, gewandt, kunstfertig; klug, einsichtsvoll, kundig, schlau, spitzfindig; gelehrt, gebildet, weise.

I. Die Wissensdatenbank SOPHOS

A. Zielsetzung des Programms

Um einen gegebenen Lebenssachverhalt lege artis greifbar zu machen, bedient sich der Jurist im allgemeinen zweier Methoden:

Subsumtion

a) Die originäre Tätigkeit des Juristen im deutschen Rechtskreis ist die Subsumtion. Dabei wird geprüft, ob Tatsachen des jeweiligen Lebenssachverhaltes Tatbestandsmerkmale von Normen oder Definitionen erfüllen (und vice versa).

Falkvergleich

b) Subsidiär hierzu ist die Arbeit mit Hilfe des Vergleichs: Sofern die Subsumtion mangels weiterer Definitionsmöglichkeiten zu einem auftretenden Begriff nicht zu einem endgültigen Ergebnis führt, werden durch Textrecherche vergleichbare Sachverhalte gesucht und vorbestehende Lösungen übernommen oder bewußt verworfen. Eine entsprechende Software sollte bei der Anwendung dieser beiden Arbeitsmethoden unterstützen und so dem Juristen die Möglichkeit bieten, diese sinnvoll und ausgewogen in einem Arbeitsgang zu verknüpfen. Die unter dieser Vorgabe entwickelte Anwenderoberfläche *SOPHOS 1.0* richtet sich im wesentlichen an den Juristen in der Ausbildung. Ohne über spezielle EDV-Kenntnisse zu verfügen, können Tatbestände und Definitionen als logische Strukturen in eine Datenbank eingegeben werden. Der wesentliche Arbeitsvorgang hierbei ist die Umsetzung von Texten, die in einfacher und sinnvoll angeordneter Form eingegeben werden, in ein logisches System. Außerdem können Fundstellen als Volltexte in diese Strukturen eingebunden werden. Die Brücke zwischen beiden Blöcken bilden dabei die juristischen Begriffe. Für welchen dieser Begriffe die Fundstelle eine relevante Aussage macht und wie diese lautet, muß bei der Eingabe in besonderem Maße beachtet werden. Die Eingabe von Daten als Form der Anwendung erfordert eine intensive Beschäftigung mit dem behandelten Stoff und vermittelt eine entsprechend tiefe Einsicht in die jeweiligen Zusammenhänge. Die so gespeicherten logischen Strukturen unterstützen den Anwender bei der Subsumtion. Ausgehend von einem Ursprungsbegriff können diese Strukturen in beliebig tiefen Ebenen aufgerufen werden. Dabei bleibt der Weg zu einer Struktur und zu deren Stellung im System durch die graphische Darstellung jederzeit nachvollziehbar: Man kann sich mit dem Bildschirmfenster über dem Begriffsnetz wie über einer Landkarte bewegen. Existieren zu auftretenden Begriffen keine Strukturen mehr, so recherchiert der Anwender in der eingegebenen Völltextdatei. Zu einem durchgeführten Subsumtionsdialog erstellt das Programm auf Wunsch ein Protokoll in Form eines Gutachtens. In diesem Gutachten werden die logischen Strukturen wieder ausformuliert und relevante Fundstellen vollständig wiedergegeben. Die dieser Oberfläche zu Grunde zu legende Datenbasis soll ein hohes Maß an Allgemeingültigkeit besitzen. Das juristische Wissen muß in einer Form gespeichert werden, die auch andere Methoden der Datenabfrage über andere Anwenderoberflächen ermöglicht. Hauptbeispiel hierfür bildet die dialogische Abfrage der relevanten Begriffe mit gleichzeitigem Aufruf erläuternder Texte.

Adressaten von SOPHOS: Juristen in der Ausbildung.

Im Mittelpunkt des Designs: Die juristischen Begriffe

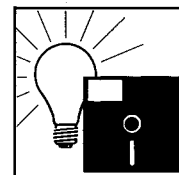
Über dem Begriffsnetz wie über einer Landkarte

Im Folgenden werden die Datenbankstruktur von *SOPHOS* und die Anwenderoberfläche von *SOPHOS 1.0* eingehender beschrieben. Um die Anschaulichkeit zu erhöhen, soll dabei die praktische Anwendung jeweils an diesem Beispiel demonstriert werden: Um seine Schwester zu ärgern stellt sich A vor deren Fenster und ruft „Dein Mann ist soeben schwer verunglückt und liegt im Krankenhaus.“ Dies entsprach nicht der Wahrheit. Die B erschrickt „zu Tode.“ Körperverletzung? Verschärft?

B. Konzeption des Programms

Die grundlegende Konzeption von *SOPHOS* ließe sich am ehesten mit einem Karteikastenprinzip vergleichen. Dem Bearbeiter eines Sachverhaltes seien zwei Karteikästen zur Verfügung gestellt:

Matthias Kraft (Schießgrabenstr. 26, 8900 Augsburg) ist Autor des Programms SOPHOS. Im Rahmen eines Seminars von Prof. Subr (Augsburg) wurden einzelne „Subsumtionsbäume“ in SOPHOS realisiert.



Eine Kartei enthält die Prüfungsschemata. Auf den Karten dieses Kastens sind je ein Oberbegriff und die zugehörige Prüfungsstruktur verzeichnet, die als Folge logischer Verknüpfungen mehrerer Unterbegriffe zu definieren ist. Ein Oberbegriff ist dabei durch genau eine Struktur festgelegt, d.h. es existiert für jeden Begriff nur eine Karte. Der zweite Karteikasten enthält Texte wie Urteile, theoretische Schriften etc. Für jeden Begriff, über den ein Text eine relevante Aussage macht, wird eine gesonderte Karte angelegt. Diese enthält im Titel den jeweiligen Begriff und die Angabe, ob sich der Autor über diesen Begriff positiv, negativ oder neutral – etwa bei Definitionen – äußert. Die Arbeit mit einem derartigen Karteisystem würde dann in etwa so aussehen:

*Der erste
Karteikasten:
Prüfungsschemata*

*Der zweite
Karteikasten:
Texte*

*Die Arbeit mit dem Karteisystem:
Skizze des Ablaufs*

a) Der Benutzer sucht aus dem ersten Karteikasten die seines Erachtens für den bearbeiteten Fall relevanten Begriffe heraus (z.B. Körperverletzung oder verschärfte Körperverletzung vielleicht auch körperliche Mißhandlung, wenn er hier eine Problematik erkennt und direkt in die Spezialabfrage hineinspringen will). Ausgehend von dem hierarchisch höchsten und plausibelsten Begriff (z.B. Körperverletzung) erfolgt dann die Prüfung der Schemata. Man überlegt sich bei jedem verzeichneten Unterbegriff, ob diese Voraussetzung gegeben ist oder nicht. Die Entscheidung wird dokumentiert. Kann oder will man sich nicht festlegen (z.B. körperliche Mißhandlung), sucht man zunächst nach der Schema-Karte des fraglichen Begriffes. Bei der Prüfung dieses Schemas können evtl. wieder problematische Begriffe auftauchen (z.B. üble unangemessene Behandlung) usw.

Sobald sich aus einer Karte eine eindeutige Schlußfolgerung ergibt, wird dieses Ergebnis dokumentiert. Auf der Grundlage aller notierten Angaben wird erneut die vorangegangene Karte geprüft. Am Ende erreicht man so wieder die Ausgangskarte (z.B. Körperverletzung). Auch hier müßte man jetzt zu einem schlüssigen Ergebnis kommen können.

b) Enthält die Schema-Kartei zu einem auftauchenden Begriff keine Karte, so kann man in der Textkartei nach hilfreichen Materialien suchen. Hier stehen zu einem Begriff eventuell auch mehrere Karten zur Verfügung. Findet sich ein Text, in dem die vertretene Meinung plausibel oder der gegebene Sachverhalt in Bezug auf den fraglichen Begriff vergleichbar erscheinen, so wird die auf der Karte angegebene Entscheidung notiert, um sie bei der weiteren Prüfung in der Schema-Kartei anzuwenden.

c) Am Ende legt der Bearbeiter das gefundene Ergebnis anhand der verwendeten Karten (Text und Schema) sowie der notierten Zwischenergebnisse schriftlich nieder.

C. Anwendung des Programms

1. Grundlagen

Die Anwenderoberfläche *SOPHOS* 1.0 wurde teilweise an bekannte Softwareprodukte angeglichen, um den Einstieg zu erleichtern und die Einarbeitungszeit zu verkürzen. Die Bedienung des Programms erfolgt obligatorisch mit der Maus. Nur in unumgänglichen Fällen der Texteingabe wird auf die Tastatur zurückgegriffen. Diese Arbeitsweise wurde gewählt, da sich so Tippfehler bei der Programmbedienung weitgehend vermeiden lassen und da die Einarbeitung mit der Maus dem Anfänger im allgemeinen leichter fällt.

Obligatorisch: Einsatz der Maus

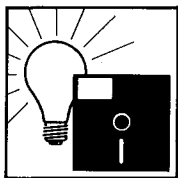
2. Prüfung anhand eingegebener Daten

Im Folgenden wird zunächst vorausgesetzt, daß die Datenbank des Systems schon mit Begriffsschemata und erläuternden Texten ausgestattet ist. Diese Daten können entweder vom Anwender selbst erstellt (s.u. 3) oder von einem Anbieter erworben worden sein.

a) Subsumtion

Dem oben beschriebenen Schemata-Karteikasten entspricht bei *SOPHOS* 1.0 das *Logikfenster*, das zu Beginn einer geplanten Subsumtion mit der Maus angesteuert wird. Daraufhin fragt das Programm einen Startbegriff ab. Dieser bildet den Ausgangspunkt für die gesamte Begriffskaskade, die sich aus dem Prüfungsvorgang – wie schon geschildert – ergeben kann. Das heißt, nur für Begriffe, die auf dieser „Anfangskarte“

*Beginn der Subsumtion:
Ansteuerung des Logikfensters*



erscheinen, können weitere Karten aus der Datenbasis abgerufen werden. Es ist daher gerade für den Anfänger ratsam, einen sehr allgemein gehaltenen Startbegriff zu wählen, um sich eine breite Ausgangsbasis zu schaffen und den Zugriff auf möglichst viele Begriffe und Unterbegriffe zu erlauben. Hier erweist sich die Möglichkeit, daß anstelle von logischen Strukturen auch einfache Inhaltsverzeichnisse auf einer Karte angegeben sein können, als hilfreich. So eröffnet etwa der Startbegriff „Körperverl“ (§ 223 StGB) den Zugang zu beiden Deliktsformen.

Körperverl (§ 223 StGB) (?)
 Körperverl (§ 223I StGB) (?)

verschärfte Körperverl (§ 223II StGB) (?)

Von hier aus kann der Benutzer mit Hilfe der Maus die Prüfungsstruktur des zutreffenden Begriffes aufrufen (also die entsprechende Karte ziehen):

Körperverl (§ 223 StGB) (?)
 Körperverl (§ 223I StGB) (?)
 - Tatbestand (?)
 - Rechtswidrigkeit (?)
 - Schuld (?)
 - Prozeßvoraussetzungen (?)

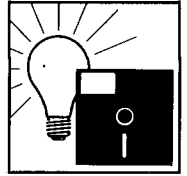
Hier wurde als nächster Schritt die Unterstruktur von Körperverletzung (Begriff Tatbestand) aufgerufen:

Körperverl (§ 223 StGB) (?)
 Körperverl (§ 223I StGB) (?)
 - Tatbestand (?)
 -
 - allgem Hdlg (?)
 - körperl Mißhandig (?)
 oder Gesundhtsbeschädigung (?)
 oder
 - spez Hdlg (?)
 - Speziallösungen (?)
 - ein anderer (?)
 - Kausalität (?)
 - Vorsatz (?)

Der Benutzer kann nun über jeden einzelnen Unterbegriff entscheiden und mit der Maus die jeweiligen Werte, (+) oder (-), eingeben. Das Ergebnis, das sich daraus für diese Struktur ergibt, wird automatisch beim Titelbegriff der Karte angezeigt.

Körperverl. (§ 223 StGB) (?)
 Körperverl. (§ 223I StGB) (?)
 - Tatbestand (?)
 -
 - allgem Hdlg (+)
 - körperl Mißhandig (?)
 oder Gesundhtsbeschädigung (-)
 oder
 - spez Hdlg (-)
 - Speziallösungen (?)
 - ein anderer (+)
 - Kausalität (+)
 - Vorsatz (+)

Die Eingabe der Werte erfolgt lediglich in der gerade aufgerufenen Struktur. Auch von hier aus können wieder tiefere Strukturen angewählt werden. Unklar ist noch die körperliche Mißhandig.



- Tatbestand (-)
 - allgem Hdlg (+)
 - körperl Mißhandig (-)
 - körperl Einwirkg (-)
- oder:
 - seelische Einwirkg (+)
 - wenn h.M. (+)
 - dann
 - mittelb körperl Auswirkg (-)
 - oder unmittelb körperl Auswirkg (-)
 - wenn Mm. (-)
 - dann POSITIV

Jeder einmal aufgerufene Begriff, d.h. jede Titelzeile bleibt sichtbar, so daß der Weg bis zur gerade bearbeiteten Struktur immer erkennbar ist. Der Benutzer kann jederzeit in jede vorhergehende Struktur zurückspringen. Eingegebene Werte werden beim Verlassen einer Struktur gespeichert.

In unserem Beispiel müßte auf der jetzt erreichten Ebene der Begriff körperl Mißhandig mit (-) beurteilt werden, womit sich automatisch für die übergeordneten Strukturen folgendes Bild ergibt:

- Körperverl. (§ 223 StGB) (?)
- Körperverl. (§ 223I StGB) (-)
- verschärfte Körperverl. (§ 223II StGB) (-)

Ist der Anwender nach seiner Prüfung wieder beim Startbegriff angelangt, stehen ihm mehrere Funktionen zur Fortsetzung der Prüfung zur Verfügung:
 Er kann z.B. einen beliebigen neuen Startbegriff wählen und von hieraus die Prüfung fortsetzen. Dabei werden alle Eingaben der vorangegangenen Prüfung beibehalten. Der Benutzer kann sich auch vom Programm Vorschläge für weitere zu prüfende Begriffe machen lassen (Querverweise). Es werden dann alle die Begriffe in einer Liste zur Auswahl gestellt, deren Vorliegen zwar noch unklar mindestens ein Unterbegriff jedoch bereits geklärt ist. Weiterhin gibt es die Möglichkeit die Prüfung gänzlich neu zu beginnen und sowohl die vorherigen Eingaben als auch das Protokoll zu löschen.

b) Textrecherche

Der Zugang zum *Textfenster*, das dem beschriebenen Textkarteikasten entspricht, kann direkt vom *Logikfenster* aus erfolgen. Statt die folgende Struktur eines Begriffes aufrufen zu lassen, kann man via Maus das *Textfenster* öffnen. Der abgefragte Begriff wird damit als Suchbegriff dem *Textfenster* übergeben. In diesem werden dann die Namen aller Texte, die mit dem fraglichen Begriff verknüpft sind, aufgelistet. Gleichzeitig wird der Wert angezeigt, den der Autor des jeweiligen Textes diesem Begriff zuweist. Aus dieser Liste kann der Benutzer den gewünschten Text wählen und zur Ansicht abrufen. Von diesem Text aus kann die Gesamtaufstellung der verfügbaren Texte jederzeit wieder eingesehen werden.

Die im Text vorliegende Entscheidung für oder gegen den Suchbegriff kann per Knopfdruck in das *Logikfenster* übernommen werden.

Für unser Beispiel könnte die Suche nach einem relevanten Text zum Begriff seelische Einwirkung folgendes ergeben:

Textname: Erregung von Ekel

Seelische Einwirkung (+)

A ist vor den Augen der B eine Spinne, was bei dieser Ekel erregt.

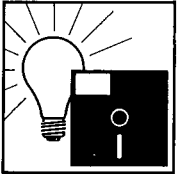
Mm.: Ausreichend für Körperverletzung.

h.M.: Ekelerregung nicht ausreichend. Sie muß etwa mit Magenschmerzen verbunden sein. Daher keine üble unangem Behandig also keine Körperverletzung.

Der „Subsumtionsweg“ bleibt sichtbar

Optionen beim „Neustart“

Übernahme des Textergebnisses in das Logikfenster



„Rohgutachten“ parallel zur
Subsumtion

Bestätigt man per Mausknopf, daß dieser Fall mit dem bearbeiteten vergleichbar ist, wird der zugeordnete Wert für seelische Einwirkung (+) in das aktuelle Prüfungsschema übernommen.

c) Protokollierung

Das Programm registriert jeden Schritt des Prüfungsvorganges in einer Tabelle und stellt aus diesen Informationen auf Anweisung ein Protokoll in Form eines Gutachtens zusammen. Dies geschieht mit Hilfe vorformulierter Standardphrasen, die jedoch über eine Textbausteindatei abgewandelt werden können. Der Text des Gutachtens wird als ASCII-Datei gespeichert und kann so in jedes Textverarbeitungsprogramm eingespeist werden.

Das folgende Beispiel zeigt ein Protokoll mit dem Startbegriff „Strafrecht“:

1. A könnte wegen einfacher Körperverletzung strafbar sein.

Dazu müssen Tatbestand, Rechtswidrigkeit, Schuld und Prozeßvoraussetzungen vorliegen.

1.1. A könnte den Tatbestand verwirklicht haben.

Dazu müssen allgem Handig, körperliche Mißhandig oder Gesundheitsbeschädigung oder spez Handig und Speziallösungen und eine anderer, Kausalität und Vorsatz vorliegen.

...

1.1.1.1. Fraglich ist ob eine seelische Einwirkung vorlag.

Folgender Text ist mit dem Sachverhalt vergleichbar:

A ist vor den Augen der B eine Spinne, was bei dieser Ekel erregt.

Mm.: Ausreichend für Körperverletzung.

h.M.: Ekelerregung nicht ausreichend. Sie muß etwa mit Magenschmerzen verbunden sein. Daher keine üble unangem Behandig also keine Körperverletzung.

...

Eine üble unangemessene Behandig ist nicht gegeben.

Eine körperliche Mißhandig liegt nicht vor.

A hat den Tatbestand nicht verwirklicht.

A ist nicht der Körperverletzung schuldig.

3. Dateneingabe

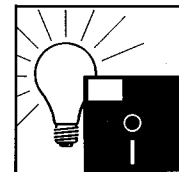
a) Übersetzer

Die Begriffsschemata werden als Texte eingegeben und zwar in einer sinnfälligen Darstellungsform, die etwa dem Bild entspricht, das die Begriffe im Anwendungsmodus (s.o. 3) auf dem Bildschirm ergeben. Der Übersetzer transformiert die als Text eingegebenen Daten in eine (tabellenförmige) Datenbank, die weniger Speicherplatz benötigt und wesentlich schneller lesbar ist. Übersetzt werden können ASCII-Textdateien oder der Speicherinhalt des programmeigenen Texteditors. Dabei muß die Änderungsbefugnis in der Kopfzeile des Code-Textes mit einem Passwort (hier z. B. 1234) nachgewiesen werden.

Logische Verknüpfungen

Neben den drei logischen Standardverknüpfungen UND, ODER und NICHT besitzt das Programm noch zwei weitere Makroverknüpfungen:

Mit der exklusive ODER-Verknüpfung (ENTWEDER-ODER) kann die logische Unvereinbarkeit mehrerer Begriffe definiert werden. Über die WENN-DANN-SONST-Verknüpfung wird die Möglichkeit eröffnet, in Abhängigkeit von bestimmten Merkmalen neue Tatbestandsmerkmale einzuführen. So können in einer Struktur mehrere Lösungswege (etwa aufgrund unterschiedlicher Theorien) vorgezeigt werden. Weiterhin ist eine nahezu beliebig tiefe Verklammerung innerhalb eines Schemas möglich. Zudem werden Informationen über die faktische (im Gegensatz zur logischen) Unvereinbarkeit von Begriffen (Alternative) sowie über Beweislastfragen eingegeben. Die Syntax des Eingabetextes (wie ja auch die optische Aufbereitung der Datenausgabe) ist weitgehend den als „Prüfungsschemata“ bekannten Strukturen angepaßt und deshalb relativ einfach erlernbar. Für unser Beispiel würde ein entsprechender Code-Text etwa so aussehen:



Körperverl. (§ 223I StGB)

- Tatbestand{2231}
-
- allgem_Hdlg{223}
- körperl_MiBhandlg
oder Gesundheitsheschädigung
- oder
- spez_Hdlg{223}
- Speziallösungen{223}
- ein_anderer
- Kausalität{223}
- Vorsatz{223}
- Rechtswidrigkeit{2231}
- Schuld
- Prozeßvoraussetzungen{2231}
- körperl MiBhandlg
- körperl_Einwirkg
- oder
- seelische Einwirkg
- wenn h.M.{seelEinw}
- dann
- mittelh_körperl Auswirkg
- oder unmittelh körperl Auswirkg
- wenn Mm.{seelEinw}
- dann POSITIV

Wie aus dieser Aufstellung ersichtlich, ergeben sich zwei Möglichkeit zur Staffelung von Strukturebenen: entweder ad hoc in mehreren Ebenen verschachtelt (wie bei Tatbestand) oder nacheinander auf gleicher Ebene (wie bei körperlMiBhandlg).

Beide Varianten erlauben es, im Logikfenster jeden untergliederten Begriff auch als selbständigen Oberbegriff abzurufen. Die Wahl der Eingabeform richtet sich jeweils nach der Praktikabilität und Übersichtlichkeit.

Nach diesem Muster läßt sich die vorhandene Datenbank auch verändern und erweitern. Diese Änderungen werden sofort in einer möglicherweise gleichzeitig laufenden Prüfung im Logikfenster berücksichtigt.

Zu der Eingabe der logischen Strukturen ermöglichen es zwei Datenbankbefehle, Textdateien im ASCII-Format (z.B. Fundstellen im Volltext) mit Begriffen aus der Begriffsliste der Logikdatei zu verknüpfen. Dabei können einem Begriff beliebig viele Texte zugeordnet werden. Umgekehrt kann ein Text auf eine unbegrenzte Anzahl von Begriffen verweisen. Gleichzeitig mit der Verknüpfung wird auch dokumentiert, welche Aussage in diesem Text über den entsprechenden Begriff getroffen wird.

Wie auch für die Wissensdatenbanken legt das Programm ein eigenes Inhaltsverzeichnis mit DOS-unabhängigen Trivialnamen an, die eine grobe Orientierung über den Inhalt der Texte und damit eine schnellere Textauswahl wesentlich erleichtern.

b) Texteditor

Der eingebaute Texteditor ist lediglich mit den notwendigsten Funktionen ausgestattet. Er dient vor allem der Korrektur von logischen Strukturen im laufenden Programm. Hierzu ist er mit diversen Zusatzoptionen ausgestattet, die den Zugriff auf die Tastatur verringern und somit zur Vermeidung von Tippfehlern beitragen.

Die Korrektur von logischen Strukturen erfolgt in drei Schritten:

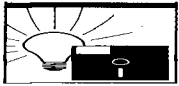
- 1.) Die aktuelle logische Struktur wird vom Logikfenster in eine ASCII-Datei zurückübersetzt und der Texteditor wird geöffnet.
- 2.) Der Quellcode wird im Editor geändert.
- 3.) Der Inhalt des Editors wird übersetzt und das Logikfenster wird aufgerufen. Die vorgenommenen Abwandlungen bilden eine neue Basis für die weitere Prüfung im Logikfenster. Dies erlaubt die sofortige Änderung von Daten während der Prüfung.

Somit ergibt sich eine Unterteilung der Anwenderoberfläche SOPHOS 1.0 in fünf Funktionsblöcke: Logik-, Text- und Protokollfenster dienen dem Abruf der Daten. Das Übersetzungsfenster ermöglicht die Eingabe neuer Daten. Mit Hilfe des Texteditors können bestimmte Datenänderungen vorgenommen und Texte ausgegeben werden,

Einbindung von Textdateien

Korrektur logischer Strukturen

Die fünf Funktionsblöcke von SOPHOS



ohne daß das System verlassen werden muß. Dabei sind Logik-, Text- und Editorfenster bei Bedarf immer auf dem Bildschirm sichtbar. Übersetzer- und Protokollfenster werden als Menüoptionen angewählt.

D. Datenbasis des Programms

1. Aufbau der Datenbasis

Die bisher geschilderte Anwenderoberfläche *SOPHOS* 1.0 basiert auf einer Datenbank, die möglichst vielseitig einsetzbar und allgemein gültig konzipiert sein mußte. Als zentrale Einheit des Programms fungiert dabei der juristische Begriff (z.B. Körperverletzung). Dieser wird in der Datenbank als Boole'sche Variable aufgefaßt. Das bedeutet, er kann die endgültigen Werte wahr und falsch annehmen. In der juristischen Praxis werden diese Wertzuordnungen mit (+) und (-) umschrieben. Alternativ kann noch ein (?) für nicht gesicherte Werte eingeführt werden. Der juristische Begriff stellt, wie gezeigt, den Suchbegriff für Prüfschemata und Texte dar. Zudem ist er wesentlicher Bestandteil der Struktur eines Prüfschemas.

Die von *SOPHOS* verwandten
Tabellen

Aufgrund dieser zentralen Bedeutung werden die Begriffe in einer eigenen Tabelle abgespeichert. Das Programm erzeugt also im wesentlichen drei tabellenförmige Datenbanken. Im Mittelpunkt steht die Liste der Begriffe mit ihren zugeordneten Werten. Diese Auflistung befindet sich in ständigem Dialog mit der Logikdatei:

Einerseits verweist jeder Begriff der Zentraltabelle auf genau einen Schema-Block der Logikdatei. Andererseits werden die darin enthaltenen Unterbegriffe durch einen Rückverweis auf die Begriffsliste definiert, so daß ein logischer Block lediglich Verknüpfungen und Probleme der Beweislast speichert. Gleichzeitig zeigt die Begriffsliste (über eine Zwischendatei vergleichbar einer Fundstellenkartei) auch ständig auf relevante Texte (sequentielle ASCII-Dateien). Dabei können jedem Begriff beliebig viele Texte zugeordnet werden. In der Gegenrichtung weist jeder Text dem Begriff einen Wert zu, so daß auch mit der Textdatei eine Art von Dialog entsteht.

Vergleich mit dem
Karteikastenmodell

Verglichen mit dem Karteikastenmodell formiert sich die Datenbank also einen dritten Karteikasten (Begriffs-Datei), der nur die Titelzeilen der beiden ursprünglichen Kästen enthält und dazu jeweils eine Nummer, die auf eine numerierte Karte in der Schema- oder Text- Kartei verweist. Im Gegenzug wurden in diesen Kästen sämtliche Begriffe durch Zahlen ersetzt, die wiederum auf die Begriffskarten verweisen.

2. Ausbaufähigkeit der Datenbasis

Das so geschaffene System erweist sich als ausgesprochen flexibel, denn sowohl im Bereich der Datenbankedition als auch auf dem Gebiet des Retrievals steht damit eine ganze Palette von Anwendungsbereichen offen:

a) Direkte Datenabfrage

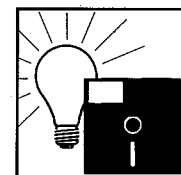
Zur direkten Abfrage der Wissensdaten bieten sich folgende weitere Methoden an: aa) In einem Dialog wird jeder relevante Begriff beim Anwender einzeln abgefragt. Die Formulierung der Frage erfolgt über die Textbausteindatei. Zudem wird sofort auf dem Bildschirm eine Definition aus der Volltextdatenbank abgerufen. Als Antworten kommen neben „ja“ und „nein“ auch die Antworten „unklar“ und „nicht beweisbar“ in Betracht. Es kann also sowohl der Aufruf näherer Informationen angefordert werden als auch nach Beweislastregeln entschieden werden. Eine Ausnahme machen hier lediglich sog. Alternativen-Blöcke, d.h. Gruppen von Begriffen, die sich gegenseitig ausschließen, von denen jedoch ein Begriff zwingend vorliegen muß. Sie werden nicht einzeln sondern als ganzer Block beim Anwender angezeigt. Der Anwender muß sich dann für einen Begriff entscheiden. Regelfall hierfür ist die Auswahl unter mehreren Theorien. Die Entscheidung nach Beweislastregeln entspricht hier der Wahl der herrschenden Meinung.

Beweislastregeln können
berücksichtigt werden

Die Regeln, nach denen das Programm entscheidet, welche Fragen zu stellen sind und in welcher Reihenfolge, können von Anwendung zu Anwendung nach den jeweiligen Bedürfnissen unterschiedlich bestimmt werden. (Eine ähnliche Anwendung enthält bereits *SOPHOS* 1.0 durch den Abfragemodus „Automatik“.) Ziel dieser Anwendung ist wiederum die Erstellung eines Protokolls in Form von Gutachten, Urteil oder anderen Schriftsatzarten.

Abfrage durch unstrukturierte
Eingabe von Begriffen

bb) Eine weitere Abfragemethode ist die unstrukturierte Eingabe beliebiger Begriffe, von denen dem Anwender bekannt ist, ob sie vorliegen oder nicht (kummulatorische



Abfrage). Diese Begriffe könnten anhand einer unsortierten Liste oder über die Tastatur eingegeben werden. Das Programm errechnet dann den Zustand aller weiteren gespeicherten Begriffe und gibt lediglich sich hieraus ergebende Veränderungen aus. Diese Abfragetechnik kann auch als Einstieg in den Dialog verwendet werden. Hier werden dann noch alle diejenigen Begriffe geprüft, die zwar berührt aber noch nicht endgültig entschieden sind. Andererseits ist es auch möglich, dem Programm aufzutragen, alle unklaren Begriffe nach Beweislastregeln zu entscheiden und die sich hieraus ergebenden Veränderungen anzuzeigen.

Letztere Methode ermöglicht es zudem dem Anwender mehrere Lösungswege aufzuzeigen, das Fehlen von Tatbestandsmerkmalen zu signalisieren und somit schnelle Entscheidungen vorzubereiten.

b) Prüfungsmakros

Makros im Zusammenhang mit dieser Datenbank sind gespeicherte Prüfungsvorgänge, die unter bestimmten Namen aufzurufen sind. Zwei Anwendungen für diese Technik seien hier genannt:

aa) Makros können als weitere Komponente der begriffsbezogenen Wissensdaten organisiert werden. Trifft der Anwender bei der Prüfung auf die Frage, ob der Begriff „Sache“ vorliegt, so hat er in *SOPHOS* 1.0 die Möglichkeit, in der Wissens- oder Textdatenbank nach weiteren Informationen zu suchen. Er kann sich mit der Einführung einer Makro-Datei als Makro abgespeichert. Dem durchgeführten und gespeicherten Prüfungsvorgänge bedienen. Möchte er etwa wissen, ob das Einjagen eines Schrecks eine körperliche Mißhandlung i.S. des StGB ist, kann er den oben als Beispiel beschriebenen und als Makro abgelegten Prüfungsweg unter dem Namen „Einjagen eines Schrecks“ aufrufen. Damit wird der gespeicherte Prüfungsvorgang in das Protokoll aufgenommen und das Ergebnis der weiteren Prüfung zugrunde gelegt.

Die Erstellung der Makros erfolgt durch den Benutzer. Sie kann etwa im Dialog automatisiert werden. Es wird dann regelmäßig etwa beim Auftreten des Begriffs Sache nach der Bezeichnung des Objektes gefragt. Ist die Bezeichnung bekannt, so wird das vorhandene Makro übernommen, ist sie unbekannt, so wird weiter geprüft und dabei ein neues Makro angelegt.

bb) Makros können auch im Rahmen eines Lernprogramms Verwendung finden. Hierbei wird ein Sachverhalt vorgegeben und dazu ein oder mehrere Lösungswege vollständig als Makro abgespeichert. Dem Anwender wird zunächst der Sachverhalt ausgedruckt. Er wird dann aufgefordert, den Fall eigenständig durch eine Prüfung am Bildschirm zu lösen. Sein Prüfverhalten wird mit der vorgegebenen Lösung verglichen und während oder am Ende der Prüfung durch das Programm kommentiert.

c) Überbau der Abfrage

Die Struktur der Datenbank und die erwähnten Abfragemethoden dienen zunächst lediglich der Lösung von Detailproblemen eines komplexen Sachverhaltes. So kann beispielsweise bei Körperverletzung lediglich eine Tathandlung eines Täters in Bezug auf ein Opfer geprüft werden. Aus diesen Gründen ist eine Organisation des Datenzugriffs durch ein übergeordnetes Programm angebracht. Hier bieten sich für unterschiedliche Anwendungsziele unterschiedliche Lösungen an:

aa) Wird beim Aufsetzen eines Schriftstücks (Klageschrift, Urteil etc.) lediglich die Lösung eines Detailproblems gefordert, so ist ein Aufruf des Abfrageprogramms unmittelbar aus dem Textverarbeitungsprogramm sinnvoll. Hierzu müßte das Abfrageprogramm speicherresident organisiert sein oder als Fenster einer Multi-Tasking-Oberfläche arbeiten. Das erstellte

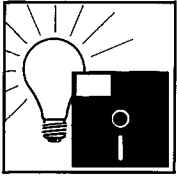
Protokoll kann bereits vom Programm auf die gewählte Schriftsatzform abgestimmt werden. Zudem ist die Eingabe von Parametern denkbar, wie etwa Namen von Beteiligten oder Bezeichnungen von Sachen, die in das Gutachten eingesetzt werden. Das gesamte Protokoll wird dann über eine spezifische Schnittstelle (z.B. ASCII-Datei oder gemeinsamer Papierkorb) in den Text des Textverarbeitungsprogramms eingefügt.

bb) Im Bereich der Verwaltung werden an einem Arbeitsplatz oft viele ähnliche Bescheide gefordert, die lediglich in Einzelproblemen divergieren. Ein übergeordnetes Programm könnte mit Hilfe vorgefertigter Formulare oder Dialoge die Erstellung derartiger Bescheide über Textbausteine nahezu ohne Tastaturzugriff ermöglichen. Dabei wird an rechtlich kritischen Stellen automatisch auf die Wissensdatenbank zugegriffen. Das erstellte Gutachten wird als Bestandteil der Begründung des Bescheides verwendet.

Bereits durchgeführte Prüfungen können als Makros abgelegt werden

Lernen mit Makros

Ideen zur Integration von SOPHOS in die juristische Arbeit



Ein Blick in die Zukunft der
„Expertensysteme“

cc) Das Fernziel von Expertensystemen ist die Lösung komplexer Sachverhalte. Hierzu zwei Stufen eines Überbaus denkbar.

Die erste Stufe dient der Strukturierung eines Sachverhalts durch einen Dialog mit dem Anwender. Hier werden im Benutzerdialog die beteiligten Personen, die streitigen Objekte und die unterschiedlichen historischen Abläufe erfaßt und benannt. Daraufhin wird eine vernünftige Strategie der Abfrage des relevanten Wissens gesucht. Dazu wird der Sachverhalt in möglichst kleine Gruppen aufgeteilt, in der jeweils nur eine Variable (z.B. Täter, Objekt, Opfer etc.) vorkommt. Hiernach wird der Zugriff auf die Wissensbasis gesteuert. Dabei soll das Programm erkennen, welche Daten auch für die folgenden Detailfragen relevant sein können (z.B. gleicher Täter), um diese abzuspeichern und doppelte Abfragen zu vermeiden. Ferner könnte das Programm auf eine weitere Wissensbank mit algebraischer Struktur zugreifen, um auch im Rechtsfolgenbereich (z.B. Schadenshöhe) gespeichertes Wissen anzubieten. Ziel ist es hier, ein komplettes Schriftstück zu entwerfen, in dem alle Fragen durch das Programm im Dialog gelöst wurden.

In einer zweiten Stufe sollen dann Sachverhalte nicht mehr im Anwenderdialog sondern durch Erfassung von Volltexten bearbeitet werden. Der Anwender gibt etwa über einen Scanner einen (weitgehend) vollständigen Sachverhalt ein. Dieser wird durch den Computer gelesen. Ein Programm erfaßt die rechtlich relevanten Tatsachen, strukturiert diese wie in der ersten Stufe und wertet sie wie in der kumulativen Abfrage aus. Lediglich einzelne unbeantwortete Fragen (etwa Entscheidungen in Meinungsstreitigkeiten) werden im Dialog erfaßt oder nach Beweislastregeln gelöst.

Die beschriebenen übergeordneten Programme sind mehr oder weniger leicht zu realisieren. Auch muß von Fall zu Fall entschieden werden, ob sie wiederum nach Art eines Expertensystems mit separat gespeichertem Wissen oder als feste Programme zu organisieren sind. Gerade sehr stereotype Anwendungen (Bescheide) ließen sich etwa auch auf dem Wege des Software-Engineerings anfertigen und pflegen. All diesen Programmen ist jedoch gemeinsam, daß sie zur Abfrage des juristischen Grundwissens auf eine gemeinsame Datenbasis zugreifen können.

Zusammenfassend ist zu sagen:

Dem Anwender werden im Vergleich zum herkömmlichen Karteikastenprinzip folgende Arbeitsgänge durch die derzeit verfügbare Oberfläche *SOPHOS* 1.0 abgenommen: Das Programm sucht die gewünschten Karteikarten (Schema oder Text) automatisch. Das jeweils aktuelle Schema wird selbsttätig geprüft. Die resultierende Entscheidung wird angegeben und, sobald diese Karte abgelegt wird, auch gespeichert. Auf diese Weise werden im Verlauf des Prüfungsvorganges alle bisher erreichten Ergebnisse gesammelt und, soweit relevant, auch angezeigt. Auch die Eintragung der in Texten gefundenen Lösungen geschieht automatisch.

Das System kann den Prüfungsvorgang nach Wunsch mehr oder weniger in seine eigene Regie übernehmen. Es kann einerseits selbständig entscheiden, welche Karten noch benötigt werden und welche nicht und so den Benutzer zu einer optimalen Prüfung zwingen. Andererseits kann z.B. unterbunden werden, daß die gefundenen Ergebnisse notiert werden, was ein Maximum an Prüfungsfreiheit ermöglicht.

Das Programm kann zu jeder Zeit auf Abruf ein gegliedertes Prüfungsgutachten erstellen. Es arbeitet dabei sowohl die bisher verwendeten Schema-Karten als auch die relevanten Texte ein. Darüber hinaus bietet die Datenbasis *SOPHOS* die Gelegenheit auf den Inhalt der Karteikästen auf andere Art und Weise (also über andere Anwenderoberflächen) zuzugreifen. Man ist also mit dem gleichen Datenschatz nicht allein auf die Karteikartenmethodik festgelegt.

(wird fortgesetzt)

Was automatisiert *SOPHOS*?