

# dBASE III PLUS

## Relationales Datenbanksystem für ein persönliches Rechts-Informationssystem

Helmut Hoffmann\*

### Teil 1:

- I. Manuelle und elektronische Datensammlungen
- II. dBASE III PLUS — Programmumfang
  - 1. Hardware- und Softwareumgebung · 2. Netzwerk
  - 3. Hilfsprogramme · 4. Programmiersprache
- III. Praktische Arbeit
  - 1. Datenbanken erstellen · 2. Dateistruktur festlegen
  - 3. Sonderfall: MEMO-Felder · 4. Dateistruktur ändern

### Teil 2:

- 5. Dateneingabe · 6. Datenabfrage · 7. Suchzeiten · 8. Sortieren · 9. Indizieren · 10. Literatur
- IV. Programmierung
  - 1. Haupt- und Untermenüs · 2. Datenabfrage · 3. Druckerausgabe
- V. Fazit
  - 1. Lern- und Eingabeaufwand · 2. Konsequenz: Programme und Dateien anbieten · 3. Wünsche des Anwenders
- VI. Ausblick: dBASE IV

### I. Manuelle und elektronische Datensammlungen

Mit herkömmlichen manuellen Mitteln läßt sich ein Informations-System des Juristen nur mit Schwierigkeiten realisieren. Ein wesentliches Problem liegt in der Strukturierung eines Volltextes wie zum Beispiel eines Urteils oder Fachaufsatzes. Viele Kolleginnen und Kollegen legen sich Karteikästen an, um anhand von Stichworten gesammeltes oder selbst erarbeitetes Wissen auch nach längerer Zeit wiederzufinden, wenn die gleiche Fragestellung erneut vorkommt, etwa nach dem Motto: „Das Problem hatte ich schon einmal; um welchen Vorgang handelte es sich?“

Am Beispiel des Urteils des BGH vom 4. November 1987<sup>1</sup> veranschaulicht: Die Entscheidung befaßt sich mit Fragen im Zusammenhang der Sachmängelgewährleistung beim Verkauf eines DV-Systems mit Hard- und Software. Um die Entscheidung bei künftigem Bedarf zuverlässig zu finden, wären Karteikarten zumindest für die Begriffe „Standard-Software“, „Mangel“ und „Einheit von Hardware und Software“ anzulegen, wenn die Kartei nach Stichworten strukturiert ist. Weitere Untergliederungen etwa nach dem Autor, der Fundstelle, dem Datum, dem Vertragstyp oder auch den angewandten gesetzlichen Vorschriften würden zu einer Potenzierung des erforderlichen Aufwands für die Speicherung führen und müssen daher realisti-

scherweise als manuell nicht machbar bezeichnet werden.

Hier setzen moderne elektronische Datenbanksysteme ein, die in gewisser Weise das Karteikartenprinzip für die elektronische Datenverarbeitung benutzen und erweitern. Eine herkömmliche Karteikarte entspricht einem Datensatz. Die Datensätze werden durch Felder strukturiert, die beim Einrichten der Datei festzulegen sind. Wie eine sinnvolle Strukturierung aussehen kann, zeigt Abb. 2, die den Datensatz bezüglich der oben genannten Entscheidung in der Form wiedergibt, die er beim Editieren auf dem Bildschirm hat. Auf die Einzelheiten wird später noch einzugehen sein<sup>2</sup>.

Im Gegensatz zu hierarchischen Datenbanken im Großrechner-Bereich haben sich bei Arbeitsplatz-Computern relationale Systeme durchgesetzt, die eine nahezu beliebige Verquickung von Suchkriterien gestatten, mithin keine Über- und Unterordnungsverhältnisse kennen. So hat zum Beispiel der Verfasser das bereits erwähnte Urteil durch Eingabe der Stichworte „Software“ und „Mangel“<sup>3</sup> in nur einem Suchschritt gefunden<sup>4</sup>.

Der Unterschied zu einer hierarchisch strukturierten Informationssammlung zeigt sich bereits an dieser Stelle: Wäre als oberste Hierarchieebene die Vertragsart zugrunde gelegt und für jeden Vertragstyp eine getrennte Datei angelegt worden<sup>5</sup>, so würde die Ent-

\* Helmut Hoffmann ist Richter am AG Ulm.

1 VIII ZR 314/86 = NJW 1988, 406 = IuR 1988, 16.

2 Kapitel III.2 bezüglich der Dateistruktur sowie Kapitel III.5 bezüglich der in Abbildung 3 zu sehenden veränderten Eingabemaske.

3 Aus einem in Kapitel IV.2 näher beschriebenen Programm.

4 Abb. 4 zeigt die vom Verfasser entwickelte Eingabemaske für allgemeine Datenbank-Recherchen.

5 So strukturiert Zahrt seine Entscheidungssammlung DV-Rechtsprechung, Band 1, 1983, sowie Band 2, 1987.

Wie unflexibel auch elektronische Datenbanksysteme sein können, zeigt Walzl CR 1987, 550 anhand eines auf einem System mittlerer Datentechnik laufenden Programms mit nur 10 erlaubten Suchbegriffen; Walzl nennt leider nicht den Namen des Programms und erläutert auch nicht die Unterschiede des von ihm benutzten Systems zu den in Kapiteln II und IV beschriebenen Abfragemöglichkeiten. Allgemein zu externen und internen juristischen Datenbanken: Tiling/Abel AnwBl 1986, 130.

Ein relationales Konzept mit einer getrennten Suchwortdatei von über 650 Schlüsselwörtern verfolgt die auf nicht mit dem Industriestandard kompatiblen Atari-Rechnern entwickelte vom Carl Heymanns Verlag vertriebene Computer-Datei „BGH-DAT“ unter dem Datenbankprogramm ADIMENS ST. Zu individuellen juristischen Datenbanken neuestens: Eberle CR 1988, 258, 259.

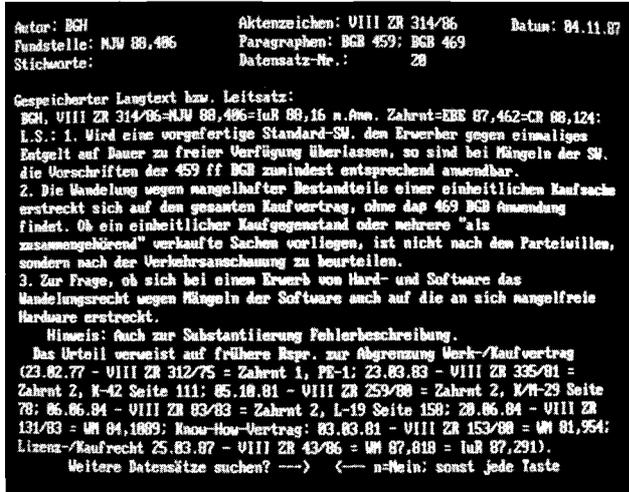


Abb. 1

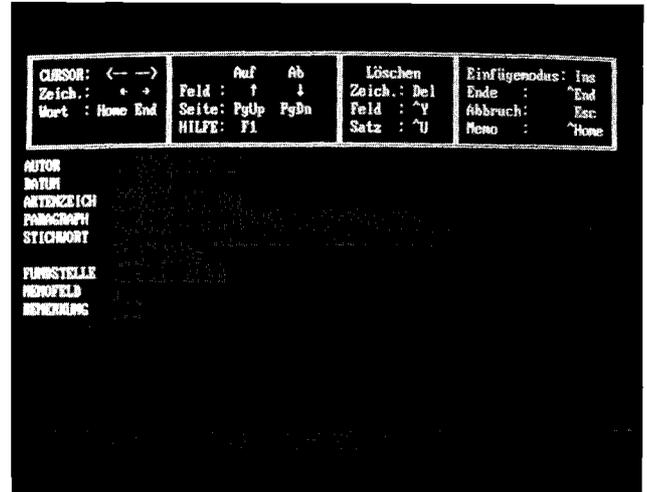


Abb. 2

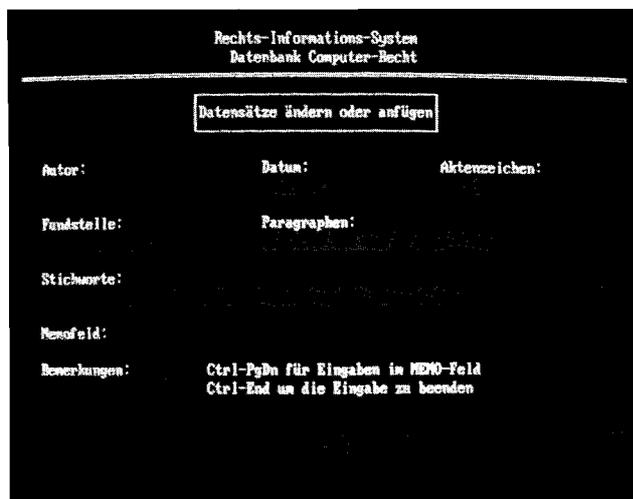


Abb. 3

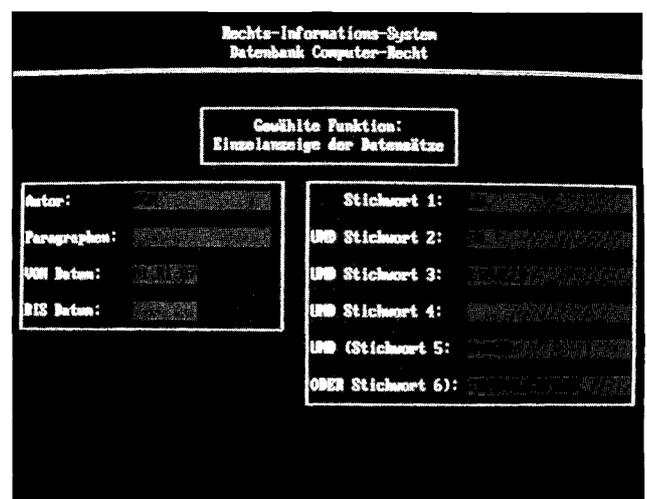


Abb. 4

scheidung bei einer Abfrage in dieser Ebene gefunden werden, nicht aber wäre es zum Beispiel möglich, sich eine Liste aller Fundstellen zu Fragen der Gewährleistung unabhängig vom Vertragstyp zusammenzustellen.

Weil bei Benutzung entsprechend leistungsfähiger Datenbank-Software nach sämtlichen Feldern mit beliebigen logischen Verknüpfungen gesucht werden kann, wäre die Entscheidung auch gefunden worden, wenn das Datum der Urteilsverkündung oder ein bestimmter Zeitraum eingegeben worden wäre. Dieses Prinzip wird mit unterschiedlicher Leistungsfähigkeit von verschiedenen für PCs käuflichen Programmen im Grundsatz immer gleich angewandt. Professionelle Datenbanksysteme sind in der Lage, mehrere Dateien gleichzeitig zu öffnen, und gestatten eine kombinierte Abfrage und Ausgabe. Im kaufmännischen Bereich wird diese Möglichkeit zum Beispiel bei Kunden-, Artikel- und Rechnungsdateien ausgenutzt.

## II. dBASE III PLUS — Programmumfang

### 1. Hardware- und Softwareumgebung

Der Verfasser arbeitet seit längerer Zeit mit dem „Datenbank-Schwergewichtsmeister“<sup>6</sup> dBASE III

PLUS der Firma Ashton-Tate GmbH. Das Programm ist lauffähig auf Rechnern mit dem Betriebssystem PC-DOS 2.0 bzw. MS-DOS 2.1 und höher bei einem Hauptspeicher von mindestens 384 kBYTE RAM und 2 Diskettenlaufwerken. Empfehlenswert sind ein größerer Hauptspeicher und eine Festplatte, ohne die ein größeres Datenbanksystem nicht zu handhaben ist. Gewarnt sei vor der Minimallösung mit zwei Diskettenlaufwerken im bisher noch üblichen 360 kB-Format. Hiermit wird man über das Stadium der Spielerei schwerlich hinauskommen.

Nutzer früherer dBASE-Versionen können ihre Dateien weiterbenutzen. Wer das noch für 8-Bit-Rechner entwickelte dBASE II benutzt hat, kann die Daten über ein mitgeliefertes Konvertierungsprogramm<sup>7</sup> einlesen. dBASE-III-Daten können ohne Konvertierung weiterbenutzt werden. dBASE III PLUS unterscheidet sich von der Vorgängerversion außer der ASSISTent-Benutzeroberfläche durch die Aufnahme neuer Funktionen. In gewissem Umfang sind graphische Darstellungen nunmehr möglich. So lassen sich zum Beispiel einfache und doppelte Rahmen, wie in den Abbildun-

<sup>6</sup> Micro Heft 11/86, Seite 8.

<sup>7</sup> dBASE BRIDGE. Das Programm konvertiert auch Befehle, die in dBASE III anders lauten als in dBASE II.



Abb. 5

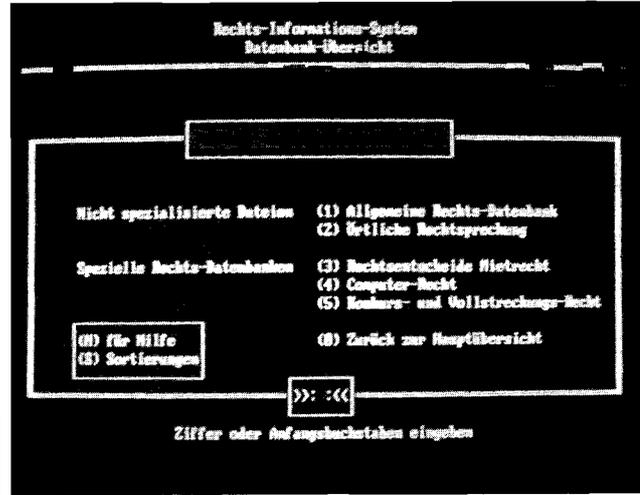


Abb. 7

Rechts-Informationen-System - Datenbank Computer-kecnc		Suche mit folgenden Eingaben:		Seite	1
Autor:	BGH	Autor	Stichworte		24.03.88
Stichworte:	HW.; GEWÄHRL; SW.	Datum		Leitsatz	
Paragrafen:		Altzeichen		und/oder	
Zeitraum vom	01.11.87 bis	Fondstelle		weitere Bemerkungen	
		BGH	If. Mängelrüge:		
		04.11.87	Bestandtheit.		
		VIII ZR 314/86	Gewährleistung.		
		NW 82.406	Einheit von SW. + Standard-SV.		

SGR. VIII ZR 314/86-NW 82.406-IuR 88.16 n.Ann. Zahrat=898 87.462-CR 88.124:  
 1.5.: 1. Wird eine vorgefertigte Standard-SV. dem Erwerber gegen einmaliges Entgelt auf Dauer zu freier Verfügung überlassen, so sind bei Mängeln der SW. die Vorschriften der 459 ff BGB zumindest entsprechend anwendbar.  
 2. Die Wandelung wegen mangelhafter Bestandteile einer einheitlichen Kaufsache erstreckt sich auf den gesamten Kaufvertrag, ohne daß 469 BGB Anwendung findet. Ob ein einheitlicher Kaufgegenstand oder mehrere "als zusammengehörend" verkaufte Sachen vorliegen, ist nicht nach dem Parteivillen, sondern nach der Verkehrsauslegung zu beurteilen.  
 3. Zur Frage, ob sich bei einem Erwerb von Hard- und Software das Wandelungsrecht wegen Mängeln der Software auch auf die an sich mangelfreie Hardware erstreckt.  
 Hinweis: Auch zur Substantiierung Fehlerbeschreibung.  
 Das Urteil verweist auf frühere Expr. zur Abgrenzung Werk-/Kaufvertrag 123.02.77 - VIII ZR 312/75 = Zahrat 1. PZ-1; 23.03.83 - VIII ZR 335/81 = Zahrat 2. R-42 Seite 111; 05.10.81 - VIII ZR 259/80 = Zahrat 2. R/H-29 Seite 78; 06.06.84 - VIII ZR 83/83 = Zahrat 2. S-19 Seite 158; 20.06.84 - VIII ZR 131/83 = WM 84.1089; Know-How-Vertrag: 03.03.81 - VIII ZR 153/80 = WM 81.954; Lizenz-/Kaufrecht 25.03.87 - VIII ZR 43/86 = WM 87.818 = IuR 87.2911.

Abb. 6

gen zu sehen, ohne Schwierigkeiten herstellen<sup>8</sup>. Sortier- und Indexbefehle sind schneller geworden. Eine Reihe von Befehlen ist hinzugekommen.

Farbbildschirme werden unterstützt. Benutzer hochauflösender Bildschirme, die mehr als 80 Zeichen/Zeile und unterschiedliche Schriftarten wie zum Beispiel Kursiv- und Fettdruck direkt darstellen können, werden diese Funktionen bei dBASE vermissen. Spezielle Druckertreiber werden ebenfalls nicht mitgeliefert. Allerdings versteht dBASE die aus dem Druckerhandbuch zu entnehmenden Standard-Druckersteuerbefehle z. B. zum Umschalten auf Schmaldruck, was sich bei der Programmierung von Reports bewährt (Abb. 6).

dBASE III PLUS ist in der Lage, eine Reihe von Fremddateien zu lesen und in Fremdformaten zu schreiben, zum Beispiel ASCII-Texte<sup>9</sup> und Lotus 1-2-3- sowie Multiplan-Dateien. Für Wordstar-Mailmerge-Serienbriefe ist eine Konvertierung ebenfalls möglich.

dBASE II kann maximal 65.535 Datensätze in einer Datei verwalten, dBASE III PLUS nunmehr 1.000.000.000 bei einer unbeschränkten Anzahl von Dateien. Jeder Datensatz kann bis zu 4.000 Zeichen enthalten. Dies zeigt, daß die Dateikapazität praktisch ausschließlich durch die Größe des Massenspeichers begrenzt wird.

Das Programm wird in der neuesten Version seit Sommer 1987 ohne Kopierschutz ausgeliefert; eine im

Sinne der Kunden und rechtmäßigen Besitzer überfällige Entscheidung. Benutzer der früheren Versionen müssen vor einem Backup der Festplatte jedesmal das Programm deinstallieren und danach erneut installieren, weil ansonsten der Kopierschutz das Programm zerstört.

Eine Version auf 3,5-Zoll-Disketten, wie sie von IBM-Systemem der Serie PS/2 und den sogenannten Laptops benötigt werden, ist lieferbar<sup>10</sup>.

### 2. Netzwerk

Einer der wesentlichen Unterschiede zu anderen Produkten und den früheren Programmversionen ist die Möglichkeit des Netzwerkeinsatzes. Der Trend geht im Hardware-Bereich zu vernetzten Systemen. Im Netz sind herkömmliche Programme, die für Einzelplatz-PCs geschrieben wurden, nicht lauffähig.

8 Programme, die diese graphisch orientierten Befehle beinhalten, sind auf älteren Versionen nicht lauffähig, sondern führen in den entsprechenden Befehlszeilen zu Fehlermeldungen.

9 American Standard Code for Information Interchange. Es handelt sich um einen Sieben-Bit-Code, der 128 Standardzeichen sowie 128 Zusatzzeichen definiert, und zwar Buchstaben, Zahlen, Symbole und nichtdruckbare Steuerzeichen.

10 Wegen der größeren Speicherkapazität dieser Disketten paßt das eigentliche Programm im Gegensatz zur Version auf 5,25-Zoll-Disketten auf eine Diskette, was insbesondere bei der Benutzung von Laptops ohne Festplatte vorteilhaft ist.

dBASE III PLUS, in einem LAN<sup>11</sup> eingesetzt, setzt als Arbeitsstationen PCs mit einem Speicher von jeweils mindestens 512 kBYTE RAM voraus. Der File-Server verlangt mindestens 640 kBYTE RAM und natürlich ein Festplattenlaufwerk. Als Betriebssystem sind die neueren Versionen von PC-DOS bzw. MS-DOS ab 3.1 erforderlich. dBASE III PLUS unterstützt die Netzwerke Novell Advanced Netware/86, IBM PC Network Program, IBM Token-Ring und 3Com 3 Plus. Um Datenschutz und Datensicherheit zu gewährleisten, liegen dem Programm zusätzliche Disketten bei, die eine Anmeldung mit Benutzernamen und Paßwort, eine Zugriffskontrolle sowie wahlweise eine Datenverschlüsselung ermöglichen. Diese Zusatzfunktionen sind auch auf Einzelplatz-PCs lauffähig.

### 3. Hilfsprogramme

dBASE gilt als ein zwar mächtiges, aber besonders schwierig zu handhabendes Programm ohne anwenderfreundliche Benutzeroberfläche.

In der neuesten Version ist dieses Urteil nur noch teilweise gerechtfertigt. Denn das Programm erleichtert mit einer wahlweise benutzbaren neuen Benutzeroberfläche namens ASSISTent und den immer beliebter werdenden sogenannten Pull-Down-Menüs ähnlich wie zum Beispiel Framework II<sup>12</sup> vor allem dem DV-Einsteiger das Kennenlernen und Arbeiten mit dem Programm. Mit Hilfe der Cursor-Tasten kann aus der Menüzeile am oberen Bildschirmrand jeweils ein Begriff ausgewählt werden. Es erscheint sodann ein Untermenü, das sich gegebenenfalls durch Anklicken weiterverzweigt. Abb. 5 zeigt den Bildschirm bei Benutzung des ASSISTent, der in der künftigen Version IV durch ein komfortableres System ersetzt werden wird.

Disketten mit Übungsdateien und Musterprogrammen runden das Paket, das der Kunde erwirbt, ab. Geliefert werden neben den Disketten zwei dickleibige Handbücher, die die Möglichkeiten des Programms bis in die Verästelungen aufzeigen. Dem Anfänger wird das Lernen trotz der Übungsbeispiele nicht leicht gemacht: Die Handbücher setzen für den technischen Laien zu viel voraus. Der Anwender hat erhebliche Schwierigkeiten, anhand des Stichwortverzeichnisses Lösungen für seine jeweils aktuellen Probleme zu finden. Die Beispiele kommen aus dem amerikanischen Umfeld und zielen auf den Informationsbedarf des kaufmännischen Anwenders ab.

Neuere Programmversionen erlauben die Einrichtung von sogenannten Screens, also Bildschirmmasken. Sie dienen dazu, bei der Dateneingabe eine graphisch ansprechende Gestaltung der Bildschirmoberfläche zur Verfügung zu haben, aber auch zum Beispiel, Hilfstexte den Datenfeldern beizufügen sowie die Eingabe auf bestimmte Werte zu beschränken, um das Risiko von Eingabefehlern zu reduzieren. Der in Abb. 1 als Ergebnis einer Abfrage dargestellte Datensatz wird in Abb. 2 und 3 im Editier-Modus<sup>13</sup> dargestellt, und zwar zunächst in der Standard-Eingabemaske, die dBASE zur Verfügung stellt, und in dem vom Verfasser entworfenen Screen, der zum einen optisch ansprechen-

der ist, zum anderen Hilfstexte für die Benutzung zur Verfügung stellt.

Screens gestatten es auch, einzelne Datenfelder gegen Überschreiben zu schützen. An einem Beispiel erläutert: Der Verfasser verwaltet sein Konkursreferat unter dBASE. Die Aktenzeichen werden programmgesteuert automatisch in neue Datensätze geschrieben. Weitere Eingaben erfolgen über ein Screen, das dieses Datenfeld vor dem Überschreiben schützt. Da die betreffende Datei nach diesem Feld indiziert ist, beträgt die Suchzeit bei Eingabe des Aktenzeichens unter 1 Sekunde<sup>14</sup>.

Report-Dateien gestatten es, eine optisch ansprechende Gestaltung von gedruckten Abfrageergebnissen zu programmieren. Abb. 6 zeigt den Ausriß der bereits mehrfach erwähnten Datenbank-Abfrage aus einem noch später näher zu erläuternden vom Verfasser geschriebenen Programm. Die bibliographischen Daten sind im Report links untereinander dargestellt; daneben befindet sich die 80-stellige Stichwortleiste, auf 4 Zeilen aufgeteilt. Der übrige Platz der Papierbreite steht für den Ausdruck des für den jeweiligen Datensatz gespeicherten Langtextes zur Verfügung. Solche Darstellungen erfordern im Handbuch nicht beschriebene Programmieretechniken.

Für den Anfänger besonders verwirrend ist eine ganze Anzahl von Fehlern im Handbuch; bei der Beschreibung, welche Tasten jeweils zu betätigen sind, werden häufig die Cursor-Tasten mit der Return-Taste verwechselt; die Erläuterung im Text, welche Informationen auf dem Bildschirm zu sehen sind, stimmen nicht immer mit den im Handbuch abgedruckten Bildschirm-Hardcopies und den tatsächlichen Bildschirminformationen überein — möglicherweise ein Problem der Übersetzung des Programms einerseits und des Handbuchs andererseits.

### 4. Programmiersprache

Bestandteil von dBASE ist eine eigene Programmiersprache. Die hiermit geschriebenen Programme dürfen nicht mit dem eigentlichen Datenbank-Programm verwechselt werden.

Programmierung unter dBASE führt zu einem doppelten Effekt: Zum einen können häufig benötigte Befehlsfolgen, die man auswendig lernen und mit jeweils erheblichem Eingabeaufwand immer wieder in gleichartiger Weise eintippen müßte, gespeichert werden. Zum anderen erlaubt die dBASE-Programmierung eine weitgehend selbst zu gestaltende Benutzeroberfläche.

11 Local Area Network. Lokale Netzwerke sind private schnelle Datenübertragungsnetze. Sie sind herstellerabhängig und erlauben eine Verbindung zu öffentlichen Netzen mit Hilfe eines sog. Gateway.

12 Zu diesem ebenfalls von Ashton-Tate herausgegebenen Programm vgl. Nack, DRiZ 1986, 405 sowie van Raden, IuR 1987, 217, 263.

13 Darstellungsweise während der Eingabe von neuen oder der Änderung bestehender Datensätze.

14 Vgl. zu den Suchzeiten unten Kap. III.7 und Kap. VI, zum Indizieren unten Kap. III.8.

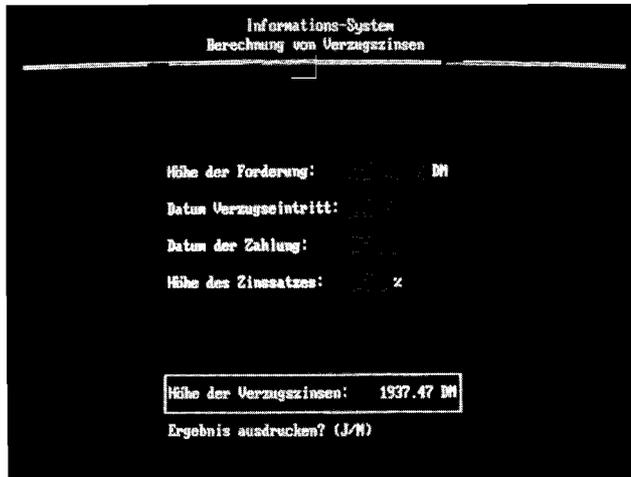


Abb. 8

che auf dem Bildschirm in Form von Haupt- und Untermenüs. Die tägliche Arbeit bei der Benutzung des Datenbankprogramms wird hierdurch entscheidend vereinfacht. Auf die Einzelheiten der Programmierung soll, soweit speziell für den Gebrauch durch den Juristen relevant, in Kapitel IV eingegangen werden.

### III. Praktische Arbeit

#### 1. Datenbanken erstellen

Um eine Datenbank-Organisation vernünftig zu planen, ist zunächst ein Konzept erforderlich, welche Dateien angelegt werden sollen. Mit dem Beispiel des Zettelkastens verglichen: Es dürfte kaum sinnvoll sein, in der gleichen Kartei sowohl Gerichtsentscheidungen wie Telefonnummern oder Adressen von Geschäftspartnern zu speichern. Vorüberlegungen sind auch für eine sachgerechte Entscheidung erforderlich, ob eine einheitliche Adressen-Datei einschließlich Telefonnummern angelegt werden soll oder getrennte Dateien für Adressen und Telefonnummern sinnvoller sind.

Im Bereich des juristischen Informations-Managements erscheint es sinnvoll, zunächst eine allgemeine Rechts-Informationen-Datenbank zu erstellen und in diese sowohl veröffentlichte Urteile aller Gerichtszweige wie Fachliteratur und privat gesammelte nicht veröffentlichte Entscheidungen z. B. der örtlichen Gerichte abzulegen. Falls nur mit Disketten-Laufwerken gearbeitet wird, ist es unbedingt erforderlich, zahlreiche kleinere Dateien anzulegen, da ansonsten sehr schnell die Diskettenkapazität überschritten würde. Einer der Vorteile relationaler Systeme besteht darin, daß nachträgliche Änderungen solcher früh getroffenen Entscheidungen weitgehend möglich sind, weil keine feste hierarchische Struktur vorliegt.

Zu versuchen, mit einem eigenen Rechts-Informationen-System großen Datenbanken wie Juris Konkurrenz zu machen, ist sinnlos. Der Vorteil interner Systeme liegt in der großen Flexibilität. Beim Verfasser hat sich ein Mischsystem zwischen allgemeinen juristischen Datenbanken und spezialisierten Datenbanken

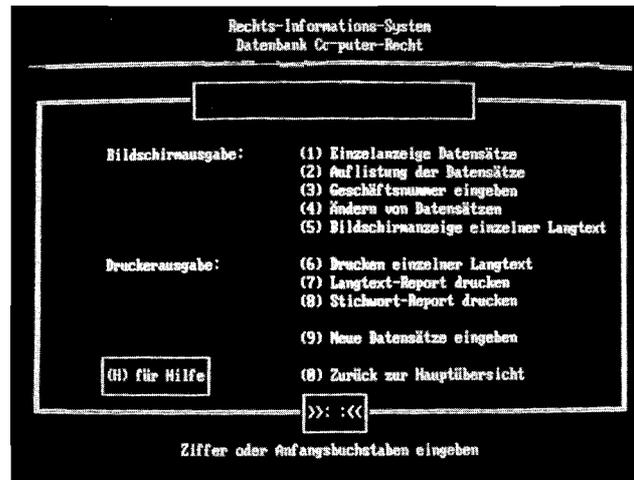


Abb. 9

als zweckmäßig erwiesen. Abb. 7 zeigt das Hauptmenü des Rechts-Informationen-Systems. Neben einer Datenbank mit veröffentlichten Entscheidungen und Sekundärliteratur, die dem Verfasser für die praktische tägliche Arbeit wesentlich erscheinen, ist eine Datenbank mit nicht veröffentlichten Entscheidungen der Gerichte am Ort vorhanden. Hinzu kommen einige spezialisierte Datenbanken. In einer finden sich sämtliche Rechtsentscheide zum Mietrecht, die jemals erlassen worden sind, sowie die wesentlichen sogenannten negativen Rechtsentscheide, jeweils mit ausführlichen Querverweisen zur Entwicklung der Rechtsprechung<sup>15</sup>. In der Datenbank Computer-Recht ist die für dieses Fachgebiet in den verschiedenen Zeitschriften und Büchern publizierte Rechtsprechung und die wesentliche Literatur seit ca. 1980 dokumentiert<sup>16</sup>. Hinzu kommt eine Insolvenzrechts-Datenbank mit im wesentlichen unveröffentlichter örtlicher Rechtsprechung des Verfassers sowie der Beschwerdekammern. Andere spezialisierte Datenbanken schließen sich an. Solche Systeme spezialisierter Rechts-Datenbanken sind nach der Erfahrung des Verfassers von besonderem Interesse auch z. B. für die Rechtsanwälte und sollten deshalb kommerziell zur Verfügung gestellt werden.

#### 2. Dateistruktur festlegen

Die Struktur, also das Gerüst, einer Datei muß stets zuerst festgelegt werden, denn ohne diese ist eine Dateneingabe ausgeschlossen.

Am Beispiel einer allgemeinen juristischen Datenbank erläutert: Mit dem Befehl „CREATE“ und anschließender Eingabe des gewählten Dateinamens, zum Beispiel „JURINFO“, erscheint auf dem Bildschirm eine Eingabemaske, die nach Feldnamen, Feldtyp, Länge und Dezimalstellen<sup>17</sup> fragt. Feldnamen wären beispielsweise Autor, Aktenzeichen, Datum, Fund-

<sup>15</sup> Ca. 320 Entscheidungen.

<sup>16</sup> Ca. 400 Entscheidungen und Aufsätze.

<sup>17</sup> Bei numerischen Feldern.

stelle, Paragraphen, Stichworte. Es sollten leicht merkbare Bezeichnungen gewählt werden, da bei Abfragen in der dBASE-Befehlssprache die Namen derjenigen Felder, in denen gesucht werden soll, einzutippen sind. Abb. 2 zeigt in der linken Spalte die vom Verfasser für die Rechts-Datenbanken gewählten Feldnamen<sup>18</sup>. Mit den in Kapitel II.3 beschriebenen Screens lassen sich in der Dateneingabe-Maske sprachlich gefälligere Begriffe mit Groß- und Kleinschreibung und ohne Längenbeschränkung wählen, vgl. Abb. 3 und unten Kapitel III.5.

dBASE III stellt fünf Feldtypen zur Auswahl. Felder des Typs „ZEICHEN“ sind für alphanumerische Angaben vorgesehen. Hier werden alle üblichen auf der Schreibmaschinentastatur vorhandenen Zeichen akzeptiert.

Felder, in denen arithmetische Operationen vorgenommen werden sollen, müssen den Typ „NUMERISCH“ haben. dBASE III, obwohl eigentlich kein Rechenprogramm, stellt umfangreiche arithmetische Funktionen zur Verfügung, so daß Firmen ihre gesamte Finanz- und Rechnungsverwaltung über dBASE abwickeln können. dBASE rechnet auf mindestens 15 Stellen genau.

Datumfelder erlauben die Durchführung von Kalenderfunktionen. Die Schaltjahre werden vom Programm berücksichtigt. Dies ermöglicht eine korrekte Berechnung der Anzahl der Tage zwischen zwei beliebigen Daten zum Beispiel für die Ermittlung von Verzugszinsen. Eingaben erfolgen in der deutschsprachigen Programmversion nach dem System TT.MM.JJ.

Im Bereich des juristischen Informations-Managements sind die Datumfunktionen praktisch wichtig, um Abfragen zu ermöglichen, die nur Entscheidungen eines genau festzulegenden Zeitraumes betreffen. Die in Abb. 4 vorgestellte Abfragemaske zeigt dies in den beiden Spalten, die fakultativ die Eingabe des Suchzeitraums ermöglichen.

Datums- und numerische Felder werden vom Verfasser z.B. in Rechentools benutzt, die auch unter dBASE, und zwar mit besonders komfortabler Abfragemöglichkeit, realisierbar sind. Eine einfache Eingabe- und Ergebnismaske zur Berechnung von Verzugszinsen zeigt Abb. 8. Das Programm rechnet taggenau die Verzugsdauer auch unter Berücksichtigung eventueller Schaltjahre. Der exakte Zins wird über ein recht kompliziertes, manuell schwer handhabbares System ermittelt. Für jeden Tag des Verzuges in Schaltjahren wird 1/366 des Jahreszinses, in anderen Jahren 1/365 des Jahreszinses errechnet. Dies geschieht dadurch, daß das Programm die Anzahl der Tage vom Verzugseintritt bis zum Ende des Kalenderjahres ermittelt, sodann feststellt, ob es sich um ein Schaltjahr handelt, und entsprechend den Jahreszins des Jahres des Verzugseintritts errechnet. Eine korrespondierende Berechnung wird für die Anzahl der Tage im letzten Jahr des Verzuges angestellt. Eventuelle Zwischenjahre werden mit dem vollen Jahreszins errechnet. Die Summe der drei Werte ergibt das am Bildschirm angezeigte Rechenergebnis. Diese Rechenoperationen sind nur möglich, weil dBASE in Datumfeldern die Differenz zwischen

zwei eingegebenen Datumswerten errechnen kann und auch eine Funktion zur Ermittlung zur Verfügung stellt, ob es sich um ein Schaltjahr handelt oder nicht. Der Eingabe- ist vom Ausgabebereich auf dem Monitor optisch deutlich getrennt. Zusätzlich fragt das Programm, ob die Berechnung auch ausgedruckt werden soll. Solche Berechnungen sind selbstverständlich auch unter Tabellenkalkulations-Programmen leicht realisierbar; die Benutzerführung ist aber mit dBASE-Programmen nach der Erfahrung des Verfassers deutlich besser als mit gängigen Kalkulationsprogrammen.

Logische Felder beanspruchen lediglich den Speicherplatz von 1 Byte. Sie erlauben nur die Eingabe von .T. für „TRUE“ oder .F. für „FALSE“. In einer Rechnungsdatei einer Firma würde in diesem Feld vermerkt werden, ob die Rechnung bezahlt ist oder nicht. Der Verfasser verwendet dieses Feld zum Beispiel, um erkennbar zu machen, ob ein Langtext zum jeweiligen Datensatz gespeichert ist.

Die Feldlänge darf bei alphanumerischen Feldern maximal 254 Zeichen betragen. Dies gestattet die Speicherung auch längerer Stichwort-Eingaben. Die in Abb. 2 zu sehende Struktur der Rechts-Datenbanken des Verfassers weist 80 Zeichen im Stichwort-Feld auf. Dies hat sich bewährt, weil hierdurch bei der Datenausgabe im Rahmen einer Recherche exakt eine Bildschirm-Zeile durch das Stichwortfeld belegt wird, was die Lesbarkeit verbessert (vgl. Abb. 1 und 3).

### 3. Sonderfall: MEMO-Felder

Eine Besonderheit von dBASE III gegenüber dBASE II und anderen Datenbankprogrammen stellen die MEMO-Felder dar. Hier integriert dBASE III eine Textverarbeitung, die es gestattet, Langtexte zu speichern und zum Abruf zur Verfügung zu halten, die zu einem jeweiligen Datensatz in Beziehung stehen.

MEMO-Felder werden in einer getrennten Datei gespeichert. In der Datenbank wird lediglich ein Zeiger auf die Text-Datei abgelegt, nicht etwa der Langtext. Wie Abb. 2 und 3 zeigen, wird deshalb im jeweiligen Datensatz nur der Feldtyp „MEMO“ und nicht der Feldinhalt wiedergegeben. Dateneingabe und -abfrage werden in den nachfolgenden Kapiteln behandelt.

Das dBASE-Textsystem läßt MEMO-Felder von höchstens 5.000 Byte<sup>19</sup> zu. Falls längere Texte, zum Beispiel komplette Urteile, abgelegt werden sollen, kann bei genügend großem Hauptspeicher des PCs ein externes Textprogramm verwendet werden. Hierzu muß die Datei CONFIG.DB um eine Zeile mit dem Befehl WP = <Name des Textsystems> ergänzt werden. In diesem Fall wird beim Aufruf des MEMO-Feldes sofort das Textprogramm geladen.

Der Verfasser benutzt die MEMO-Felder seiner Rechtsprechungs-Dateien vor allem dazu, die veröffentlichten Leitsätze, Inhaltszusammenfassungen, Weiterverweisungen auf andere Fundstellen, veröffentlichte Urteilsanmerkungen und Ähnliches zu spei-

<sup>18</sup> Die Länge der Feldnamen darf 10 Zeichen nicht überschreiten.

<sup>19</sup> Dies entspricht ca. 2-3 Schreibmaschinenseiten.

chern. Nicht realisiert ist dagegen die Dokumentation des gesamten Wortlauts der Entscheidungen, da der Aufwand des Abtippens den Ertrag nicht rechtfertigen würde. Die gespeicherten Kurztexte sowie weiterführenden Hinweise reichen erfahrungsgemäß völlig aus, die Entscheidung zu treffen, ob das Nachlesen der gesamten Entscheidung erforderlich ist oder nicht.

#### 4. Dateistruktur ändern

Eine gewählte Dateistruktur kann jederzeit mit dem Befehl „LIST STRUCTURE“ bzw. nach Betätigung der Funktionstaste 5 auf dem Bildschirm angezeigt oder

auch ausgedruckt werden. Eine Änderung erfolgt mit dem Befehl „MODIFY STRUCTURE“. Es können sowohl die Feldnamen als auch die Typen und Längen geändert werden. Der Dateinhalt bleibt erhalten, wenn die Änderung nur jeweils an einer Stelle erfolgt und sodann abgespeichert wird. Ebenso können weitere Felder hinzugefügt, dazwischengeschoben oder gelöscht werden. Im oberen Drittel des Bildschirms werden entsprechende Hilfen für die Benutzerführung eingeblendet.

Feldverlängerungen führen zu keinem Datenverlust; bei Verkürzungen werden die Sätze hinten entsprechend abgeschnitten. *(wird fortgesetzt)*

## TWAICE - eine Expertensystem-Shell (Teil 2)

Armin Leicht/Volker Siebelink

### E. Knowledge Engineering

#### I. Begriffliches

Als Knowledge Engineering wird der Prozeß bezeichnet, der das Expertenwissen in eine für den Computer verarbeitbare Form überträgt. Die Effizienz eines Expertensystems steigt mit der Anzahl der Regeln und dem Basiswissen. Dabei sind jedoch drei Probleme zu überwinden: das Formulieren des Expertenwissens, das Verarbeiten divergierender Ansichten zu einem Problem und das Übersetzen in eine computerverständliche Sprache. Als Wissensquellen können dabei Experten, Fachbücher, Unterlagen etc. herangezogen werden.

Dem Knowledge Engineer fällt nun die Aufgabe zu, in Interviews mit dem Experten die einzelnen Fakten und die Beziehungen zwischen den einzelnen Informationen herauszufiltern und zu interpretieren, um sie später in die Form von „If-Then-“ (Wenn-Dann-) Regeln zu transformieren. Hieraus können Expertensysteme mit mehreren hundert bis zu mehreren tausend Regeln entstehen. Zu guter Letzt wird zur Herstellung von Prototypen mit dem Experten und den späteren Benutzern Rücksprache gehalten und diese Prototypen in bestehende Software- und Hardware-Umgebungen des Kunden integriert. Hinsichtlich der Terminologie sei noch erwähnt, daß mit dem Experten der Fachmann mit Spezialwissen angesprochen ist und mit Benutzer der Laie, bzw. der nicht so fachspezifisch ausgebildete Benutzer, der das System bei seiner Arbeit einsetzt.

Bei der Entstehung der Wissensbank arbeiten der Experte und der Knowledge Engineer sehr eng zusammen. Nur über die intensive Auseinandersetzung mit der Domäne des Experten kann das Fachwissen in geeigneter Weise aufbereitet werden. Dabei wächst die Wissensbank durch einen Prozeß iterativer Entwicklung, der das Expertensystem in Breite und Tiefe ausbaut<sup>7</sup>.

#### II. Knowledge Engineering bei Nixdorf

##### 1. Das 5-Stufen-Konzept

Die Firma Nixdorf beschreibt das Knowledge Engineering in einem 5-Stufen-Konzept<sup>8</sup>. Dies sei an dieser Stelle kurz zusammengefaßt:

In **Stufe 1** ist es die Aufgabe des Knowledge Engineers bei der Erstellung der externen Wissensbank, das Wissen des Experten zu analysieren und in geeignete Regeln umzusetzen. Hierzu müssen sich sowohl der Knowledge Engineer als auch der Experte mit dem Fachgebiet des jeweils anderen auseinandersetzen und vertraut machen, da sonst mit erheblichen Verständigungsschwierigkeiten zu rechnen ist.

Der Knowledge Engineer wird sich also in die Wissensdomäne des Experten einarbeiten, um einen Überblick in dessen Fachgebiet zu erhalten. Andererseits muß er dem Experten die Grundlagen eines Expertensystems soweit erklären, daß dieser eine Vorstellung davon erhält, in welcher Form sein Wissen gefragt ist. Der Knowledge Engineer wird am besten dem Experten ein lauffähiges Expertensystem mit mindestens einer Wissensbank aus einer allgemein verständlichen Domäne vorführen und ihm so die grundlegenden Prinzipien eines Expertensystems verdeutlichen. Gemeinsam werden beide nun analysieren, welche Probleme von dem neuen Expertensystem gelöst werden sollen, wo das Expertensystem eingesetzt werden soll und wie die Umgebung aussieht, da hiervon das spätere Erscheinungsbild der Dialogkomponente abhängt.

Die Erstellung der externen Wissensbank in **Stufe 2** ist ein Prozeß, der sich nur schrittweise durchführen läßt. In ausgiebigen Gesprächen muß der Knowledge Engineer das Wissen des Experten analysieren und strukturieren. Problematisch hierbei ist, daß es sich bei

<sup>7</sup> hierzu: Nölke in Savory, Künstliche Intelligenz und Expertensysteme, 2. Aufl., Oldenburg 1985, S. 110f.

<sup>8</sup> s. Fn(7), dort S. 112 ff.