

Visuelle Entwicklungsumgebungen und KnowledgePro Windows (Teil 2)

Peter Ebenhoch

Backward Chaining

3.4 Entwerfen von Expertensystemen

KnowledgePro ermöglicht auch die Erstellung von Expertensystemen (XPS)⁴⁶. Dieses Potential entspringt direkt der Sprachstruktur: Da Topics nicht deklariert werden müssen und weil das System in einer vorgegebenen Reihenfolge die Topic-Hierarchie absucht, können Regeln aufgestellt werden, deren einzelne Bedingungen wiederum Topics enthalten können (Backward-Chaining).

Wird das System gestartet und eine Variable abgefragt, die noch nicht initialisiert wurde, so wird das entsprechende Topic aufgerufen, welches wiederum noch nicht initialisierte Topics oder eine Frage an den Anwender oder einen Zugriff auf eine Datenbank enthalten kann.

Beteiligtenrücktritt nach Haft

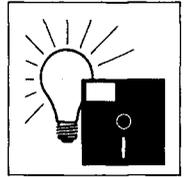
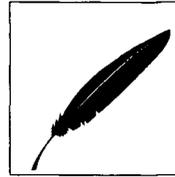
Zwei Beispiele für den Aufbau eines Expertensystems liegen bei. Besonders einfach ist es, einen vorgegebenen Entscheidungsbaum in ein Expertensystem umzusetzen. Im folgenden soll dies anhand einer Grafik aus Haft, Strafrecht Allgemeiner Teil, illustriert werden:

```
(*direkte Umsetzung der Struktur des Beteiligtenrücktritts (B=Beteiligter) nach Haft, Strafrecht AT, 4. Auflage, S 240*)
'Beteiligtenrücktritt'().
(*Startet durch Aufruf des folgenden topics:*)
topic 'Beteiligtenrücktritt'.
(*Rücksetzen der Wissensbasis, das heißt der Topics:*)
reset ([Ergebnis, 'Tat vollendet', 'unabhängige Vollendung', 'ernsthafte
Verhinderungsbemühung', 'Vollendungsverhinderung']).
say ('Ergebnis:', ?Ergebnis).
(*das Ergebnis soll angezeigt werden; weil es natürlich noch nicht
feststeht, das Topic noch nicht initialisiert wurde, wird danach gesucht:
Backward-Chaining setzt ein*)
ask ('Noch eine Abfrage?', WeiterMachen, [Ja, Nein]).
if ?WeiterMachen = Ja then Beteiligtenrücktritt()
(*erneuter Start, nach dem ersten Durchgang*)
    else close_window() and exit(). (*Nicht WeiterMachen*)

topic Ergebnis().
if ?'Tat vollendet' is Ja
    and ?'unabhängige Vollendung' is Nein
        then Ergebnis is 'strafbar, Vollendung'.
(*dasselbe gilt hier und bei den folgenden Regeln: Mangels Wertzuweisung
wird nach dem jeweiligen Topic gesucht - siehe unten - und dieses
evaluiert*)
if ?'Tat vollendet' is Ja
    and ?'unabhängige Vollendung' is Ja
    and ?'ernsthafte Verhinderungsbemühung' is Nein
        then Ergebnis is 'strafbar, Versuch'.
if ?'Tat vollendet' is Ja
    and ?'unabhängige Vollendung' is Ja
    and ?'ernsthafte Verhinderungsbemühung' is Ja
        then Ergebnis is Rücktritt.
if ?'Tat vollendet' is Nein
    and ?'Vollendungsverhinderung' is Nein
    and ?'ernsthafte Verhinderungsbemühung' is Nein
        then Ergebnis is 'strafbar, Versuch'.
```

Peter Ebenhoch, Studium der Rechtswissenschaft in Wien. Gegenwärtig Arbeit auf den Gebieten Rechtstheorie und der Rechtsinformatik.

⁴⁶ dazu vgl zB. Jim Steinhart, Expert Systems Explained, Canadian Datasystems, April 4/1990, S. 30-33



```

if ?'Tat vollendet' is Nein
  and ?'Vollendungsverhinderung' is Nein
  and ?'ernsthafte Verhinderungsbemühung' is Ja
  then Ergebnis is Rücktritt.
if ?'Tat vollendet' is Nein
  and ?Vollendungsverhinderung is Ja
  then Ergebnis is Rücktritt.
end. (*Ergebnis*)

topic 'Tat vollendet'.
  ask ('Wurde die Tat vollendet?', 'Tat vollendet',
    [Ja, Nein]).
(*Benutzer wird gefragt und mit diesem Wert oben weitergeprüft*)
end.

topic 'unabhängige Vollendung'.
  ask (' War die Vollendung unabhängig vom Versuchsbeitrag
    des B?', 'unabhängige Vollendung', [Ja, Nein]).
end.

topic 'Vollendungsverhinderung'.
  ask (' Hat B die Vollendung verhindert?',
    'Vollendungsverhinderung', [Ja, Nein]).
end.

topic 'ernsthafte Verhinderungsbemühung'.
  ask (' Hat B sich ernsthaft bemüht, die#n Vollendung zu
    verhindern?', 'ernsthafte Verhinderungsbemühung',
    [Ja, Nein]).
end.
end. (*Beteiligtenrücktritt*)

```

Jede Verzweigung im System stellt ein Topic dar, *alle* möglichen Ablaufwege im Diagramm müssen in Regeln gefaßt werden (bei einem größeren System werden die Regeln zweckmäßigerweise in Subtopics als Untereinheiten zusammengefaßt).

Dank der flexiblen Topicstrukturen lassen sich nicht nur solche regelorientierte Backward-Chaining Systeme sondern auch Forward-Chaining Modelle verwirklichen⁴⁷; wie das Handbuch anmerkt, ist es oft zielführender auf solche (starrten) Regeln ganz zu verzichten, und das menschliche Vorgehen bei einer Problemlösung direkt in Topics abzubilden.

Forward Chaining

3.5 Verwendung als Autorensystem

Auch die Eignung zur Verwendung eines VDE's als Autorensystem hängt einerseits von den oben kurz angerissenen konzeptuellen Vorgaben, andererseits vom vorgesehenen Verwendungszweck (wie zB. Präsentation, Animation) und von den verfügbaren Ressourcen ab⁴⁸. Allgemein soll ein Autorensystem die rasche Erstellung von Anwendungen ermöglichen, die interaktiv Wissensinhalte und Wissenstrukturen darstellen und vermitteln können. Kennzeichnend bzw. wünschenswert sind deshalb⁴⁹:

1. Überblicksfunktionen (Browser)
2. grafische Darstellungsmöglichkeiten (Bitmaps)
3. einfache Auswertung von Reaktionen des Anwenders
4. Unterstützung von verschiedenen Ebenen des „Authoring“
5. Standard-Sprach-Elemente (wie Bedingungen, Schleifen etc.)

⁴⁷ Ein Beispiel für ein objektorientiertes forward-chaining Expertensystem findet sich im KPWin++-Hilfssystem.

⁴⁸ auch für viele Autorensysteme gilt leider, daß das Ergebnis nur eine eingeschränkte Funktionalität aufweist, bzw. langsam und schwerfällig ist; vgl zB. zu Compel: Bernd Steinbrink, c't 10/1993, S. 174

⁴⁹ der Kriterienkatalog entstammt: Brian L. Dear, HyperCard. What Is It?, BYTE 8/1988, Macintosh Supplement, S. 71-74, S. 71

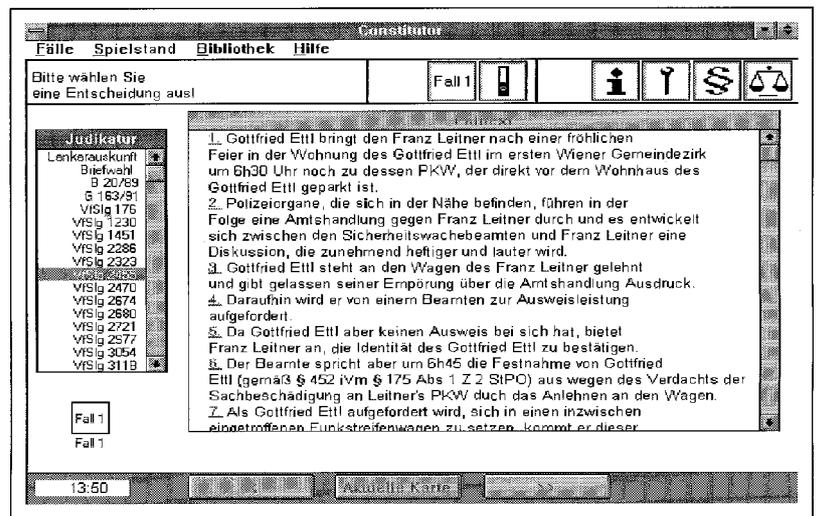
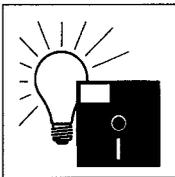


Abb. 8:
Zugriff auf die
Rechtsprechungs-
bibliothek von „Con-
stitutor“ über ein Li-
stenfenster.

Bilanz zum Thema
„Autorensystem“

- 6. Funktionen zur Dateimanipulation
 - 7. Möglichkeit zur Einbindung von anderen, herkömmlichen Sprachen (wie C++, Pascal etc.)
 - 8. Einbindung von Audio und Video⁵⁰
- KPWin unterstützt direkt die Punkte 2, 3, 5, 6, 7; die restlichen Punkte können durch entsprechende Topics leicht nach Wunsch gestaltet werden³⁰: eine einfache Überblicksfunktion läßt sich zum Beispiel durch die Anzeige der Indexeinträge in einem Listenfenster erzielen, durch eine sortierte Liste entsteht ein Glossar; oder es wird der Quellcode des mitgelieferten Library-Browser an die eigenen Vorstellungen angepaßt; Grafikanimationen sind inzwischen direkt durch Grafikbefehle umsetzbar, nicht direkt unterstützt werden aber bewegte Bild-Animationen; diese müßten eigenhändig implementiert werden. Windows Multimedia Extensions können über die jeweilige DLL angesprochen werden.

4 Handhabung, Schattenseiten, Preis und Support

4.1 Handhabung

Im Grunde einfach ...

Trotz der zahlreichen dargestellten Features und der Mächtigkeit der Sprache ist KPWin eine sehr einfache Sprache mit eigenem Charakter.

... aber Lernen tut doch not.

Voraussetzung für die Verwendung einer Sprache ist aber, daß man bereit ist, sie sich auch anzueignen. Geht es nicht nur um das Zeichnen der Oberfläche, sondern um das Erstellen einer reagierenden Anwendung, so dürfte einem dies aber (derzeit) keine noch so visuelle Umgebung vollständig abnehmen⁵¹.

Klammern
gewöhnungsbedürftig

Für Einsteiger, die schon mit anderen Programmier- oder Makrosprachen beschlagen sind, mögen anfangs die Klammern ungewohnt erscheinen. Dank der zahlreichen Defaults und der eingängigen Befehlsbezeichnungen ist diese Hürde aber schnell genommen. Die Fertigstellung einer Anwendung erfordert dann den Blick ins Handbuch oder die Verwendung der On-Line-Hilfe.

„literarisches Programmieren“

Die freie Wahl von Topic-Namen als Variablen in Verbindung mit den jedem englischsprechenden eingängigen Befehlen erlaubt das, was – in Anlehnung an *Donald Knuth* – als „literarisches Programmieren“ bezeichnet werden könnte: das Umsetzen von Ideen in Code direkt aus dem Gedanken heraus. Zusammen mit den Kommentaren können so Programme als sauber strukturierte luzide Texte entstehen!

Kleine „intelligente“ Funktionen

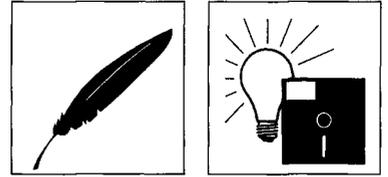
Gerade das Implementieren von kleinen „intelligenten“ Funktionen, die zB. eine Benützer-eingabe prüfen oder Teile aus einer Datei lesen, geschieht regelmäßig verblüffend schnell und ohne überflüssigen Programmieraufwand.

Nötig:
Konzeptuelle Planung

Bei großen Applikationen entsteht mehr Code, der auch bei KnowledgePro gut durchdacht und getestet sein will. Dabei bezieht sich das Durchdenken hier primär auf das Konzept selber, das umgesetzt werden soll, und nicht auf programmiertechnische Probleme wie

⁵⁰ Dieser letzte Punkt hat unter anderem durch die technische Entwicklung („Multimedia“) in letzter Zeit an Bedeutung gewonnen: Manche Autorensysteme wie MacroMind Director oder MediaBlitz (siehe Fußnote 11) bieten ganze Misch- und Schneidepulte zur Audio- und Videomanipulation an. vgl Steinbrink, c't 10/1993, S. 174

⁵¹ Zum beschränkten Nutzen eines Makro-Recorders: Steins, a. a. O., S. 2247



Speicherverwaltung, falsch gesetzte Pointer oder auf das Überwinden oder Umgehen von strukturellen Einschränkungen der Umgebung.

Die Speicherverwaltung von KPWin funktioniert problemlos: Wenngleich laut Handbuch die Verwendung lokaler Topics die Ausführungsgeschwindigkeit beschleunigt und Speicherplatz spart, so verhielt sich eine erste Entwicklungsversion von Constitutor, die fast ausschließlich globale Topics enthielt, völlig unproblematisch, sowohl bezüglich der Performance als auch bezüglich des Speichermanagements.

Die Lesbarkeit des Codes ist dank der einfachen Befehlsbezeichnungen prinzipiell sehr gut, hängt aber in weiterer Folge von der Vorgangsweise des Anwenders ab: Einerseits läßt sich durch die Verwendung beliebiger Topicnamen die Lesbarkeit noch weiter steigern, andererseits aber kann durch schlampige Benennung und überflüssig-exzessive Verschachtelung auch die leichte Lesbarkeit der Befehle vereitelt und nicht mehr Nachvollziehbares geschaffen werden.

Problemlose Speicherverwaltung

Gut lesbarer Code

4.2 Schattenseiten

4.2.1 Performance

KPWin nimmt nicht nur wenig Platz auf der Festplatte in Anspruch, sondern gibt sich notfalls auch mit einem schnellen 386 SX sowie 2 MB RAM zufrieden. Ärgerlich dabei: Der selber mit KPWin geschriebene Designer läßt aufgrund seiner Komplexität doch den Wunsch nach einem möglichst schnellen Rechner aufkommen⁵².

Etwas schneller ist doch besser.

Ein IBM-kompatibler 386er, mit 33 oder 40 MHz getaktet und mit 4 MB Hauptspeicher genügt aber auch in dieser Hinsicht⁵³, wobei auch hier die Regel gilt, daß unter Windows eine schnellere Maschine und mehr Hauptspeicher nie schaden (sieht man von der eigenen Brieftasche ab).

Die Ladezeit der Programme, die durch die Syntaxprüfung bedingt ist, steigt natürlich mit deren Umfang an; vor allem kurz vor Fertigstellung einer größeren Applikation mit Einbindung aller benötigten Programmteile entstehen dadurch – abhängig von der Prozessorleistung mitunter störende – Wartezeiten. Diesem Umstand kann aber durch modulare Programmierung dadurch entgegengewirkt werden, daß schon fertiggestellte Teile eigenständig kompiliert und mit `include` eingebunden werden; die Ladezeit eines kompilierten Teiles, aus dem die Kommentare entfernt und der Syntax vom Interpreter schon geprüft wurde, ist nämlich wesentlich kürzer.

Ladezeiten

4.2.2 Grafik und Animationen

Die früher fehlenden Grafikfunktionen^{40 41} wurden von KnowledgeGarden seit der Version 2.3 ebenso ergänzt, wie der für manche professionellen Applikationen zu langsamen, sonst aber für eine Interpretersprache mehr als ausreichenden Ausführungsgeschwindigkeit inzwischen durch KPWin++ abgeholfen werden kann.

Mithilfe dieser Grafikfunktionen und mit den Befehlen zum Verschieben und Verändern von Bitmaps können auch einfache Animationen gestaltet werden. Verglichen mit ToolBook mangelt es KPWin aber an weitergehenden Animationsfähigkeiten. Wohl nur fortgeschrittene Anwender werden solche Befehle selber schreiben wollen (daß dies mit KPWin möglich wäre, beweist nicht zuletzt der KPWin Designer).

Nur einfache Animationssequenzen möglich

4.2.3 Dokumentation und Lernkurve

Die Muttersprache des Computers ist Englisch und KPWin ist nur in einer englischen Version erhältlich: Da die Fonts und die direkt angesprochenen Dialogfenster aber von Windows verwaltet werden, sind bezüglich Umlaute und verwendeter Menüs (zB. `file_menu` als Datei-Öffnen) keine Probleme gegeben.

„Die Muttersprache des Computers ist Englisch ...“

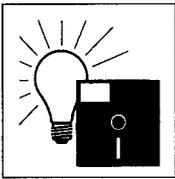
Für manche abschreckender wirkt vermutlich die Lektüre der englischsprachigen Handbücher. Das mitgelieferte einführende Programmierhandbuch und das Referenzhandbuch sind aber in einem nicht-technischen, verständlichen Stil geschrieben und decken alle Aspekte ab; leider sind sie manchmal aber etwas knapp geraten und nicht ganz fehlerfrei.

Da zudem die beispielhafte Entwicklung einer kompletten Anwendung fehlt, fühlt sich vor allem der Neueinsteiger mit der vorhandenen knappen „Quick-Start“ Broschüre alleinge-

Etwas wenig Propädeutik ...

⁵² Da KPWin++ (siehe 5.2) aber die Umwandlung jedes KPWin-Programmes in ein C++-Exe-Programm ermöglicht, dürfte auch der Designer in folgenden KPWin-Versionen als wesentlich schnelleres C++-Programm enthalten sein.

⁵³ Constitutor wurde auf einem 386er-Rechner mit 40 Megahertz und 4 Megabyte Hauptspeicher sowie auf einem 486er mit 33 MHz und 8 Megabyte RAM entwickelt.



... kompensiert durch die simple Syntax.

„If you are in doubt, try it out.“

Exzellenter Support

Lizenzfreie Runtime

dBASE-Kit, SQL-Kit und 14 Formate

Precompiler für C++

lassen, zumal er sich auch das Studium des Quellcodes der beigelegten Programme nicht zutrauen mag. Da auch ein einführendes Lernprogramm sowie zusätzliche Literatur fehlen, setzt das Erlernen von KPWin deshalb ein gewisses Maß an Eigeninitiative voraus; ohne diese dürfte aber auch ein anderes VDE nicht zum gewünschten Ergebnis führen⁵⁴.

Dieses Manko wird zudem durch die simple Syntax der Sprache, die zahlreichen fehlermindernden Default-Einstellungen und durch die Möglichkeit des raschen Austestens neuer Funktionen weitgehend entschärft. Wie Diskussionen im Forum auf CompuServe belegen, ist das Problem auch dem Hersteller KnowledgeGarden bekannt, der für die nächsten Updates Abhilfe in Form von mehr Dokumentation und umfangreicheren Beispielen versprochen und für KPWin++ (siehe 5.2) auch schon gehalten hat.

Eine – mehr psychologische – Desorientierung kann auch Anwender mit Vorerfahrung in einer anderen Programmier- oder Makrosprache ereilen: Es fällt eingangs schwer zu glauben, daß Programmieren so einfach sein kann! Letztendlich kann dagegen wohl nur der Hinweis aus dem Handbuch wiederholt werden: If you are in doubt, try it out.

4.3. Support und Preis

Hervorzuheben ist noch der exzellente Support durch KnowledgeGarden bzw. durch seine Distributoren BoldBuyer\$ und MIK/Zürich (für Österreich und die Schweiz), die – ob über das CompuServe Forum (Go Knowledge) oder telefonisch – auch bei Anfängerproblemen weiterhelfen und mit ergänzenden Tips aufwarten konnten.

Angesichts der lizenzfreien Weitergabemöglichkeit des Runtime-Moduls ist der Preis von KnowledgePro Windows^{7 55} verglichen mit anderen Autorensystemen als günstig (vgl. zB. die Gegenüberstellung in¹¹), verglichen mit anderer Software zur Erstellung von Expertensystemen sogar als ausgesprochen niedrig zu bezeichnen.

5 Zusatzprodukte

5.1 Datenbanken

Taucht in einer datenintensiven Applikation der Wunsch nach einer „echten“ Datenbank auf, so kann entweder auf das dBase-Kit⁵⁶, das mit einer Topic-Bibliothek an dBase angelehnte KPWin-typische Befehle zur Einbindung von dBase- und Clipper-Dateien zur Verfügung stellt, oder aber auf das darüberhinausgehende SQL-Kit zurückgegriffen werden. Das SQL-Kit erlaubt ebenfalls die Einbindung von KPWin-Topics, die über eine von Pionier Q+E lizenzierte Bibliothek den Zugriff auf und das Schreiben in insgesamt 14 verschiedenen Datenbankformaten (ua. dBase, Novell Btrieve, Oracle, Informix etc.) erlauben. Auch hier darf das damit erstellte Programm lizenzfrei weitergegeben werden.

5.2 KPWin++

Den Vorteilen von Interpretersprachen (direkte Ablauffähigkeit ohne Umweg über die Kompilierung) steht als Nachteil vor allem die geringere Geschwindigkeit gegenüber. Seit Ende 1992 ist nun unter der Bezeichnung KPWin++ ein Precompiler für KPWin erhältlich, der den Quellcode eines KPWin-Programmes direkt in C++-Quellcode umwandelt und dadurch das (diesfalls echte) Kompilieren⁵⁷ in eine ausführbare C++-Exe-Datei ermöglicht.

Das hört sich nicht nur faszinierend an, es ist auch faszinierend: Der vorherige KnowledgePro-Code kommentiert plötzlich den C++-Code! Das rasch erstellte KPWin-Programm liegt nach kurzer Umwandlung in C++ vor: Sogenannte „Mannjahre“, die ansonsten für C++-Programme zu veranschlagen sind, reduzieren sich auf Tage. Die erstellte Exe-Datei bleibt auf eine 490 kB große DLL angewiesen, die das KPWin-API enthält. Im Unterschied zu Visual Basic wird die erstellte Exe-Datei aber nicht mehr interpretiert; die Geschwindigkeitszuwächse sind entsprechend beträchtlich: Je nachdem, ob primär die Grafik-Ausgabe oder interne Rechenleistung in Beschlag genommen werden, verkürzt sich die Ausführungsgeschwindigkeit gegenüber der – schon nicht langsamen – Runtime-Version um um den Faktor 2 bis 20. Das so erstellte Programm läßt selbst auf langsamen Rechnern keine Wünsche mehr offen.

⁵⁴ zu den Hürden bei ToolBook Steins, a. a. O., S. 2246

⁵⁵ KPWin kostet 1080,- DM, für Hochschulen bzw. Hochschulangehörige 840,- DM.

⁵⁶ Karl Sarnov, Zwei drauf. KnowledgePro (Windows): als dBase Frontend und in der Version 2.0. c't 3/1992, S. 88.

⁵⁷ Unterstützt werden Microsoft Visual C++ 1.0 und Borland C++ 3.1.

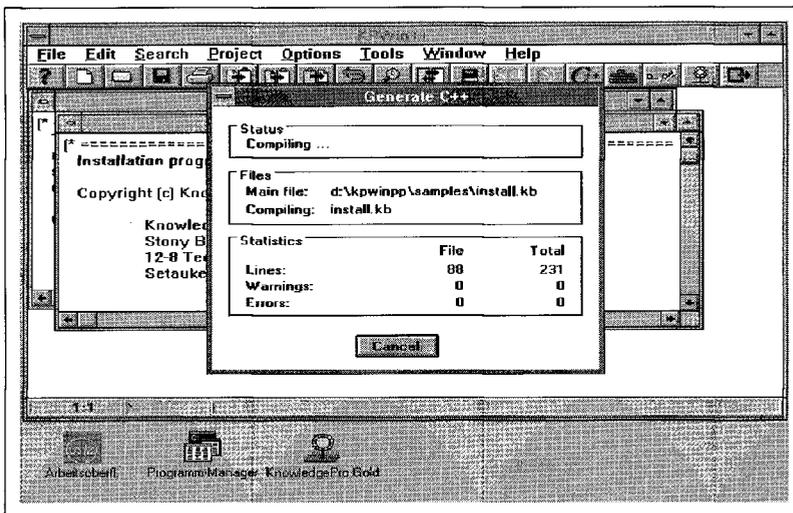
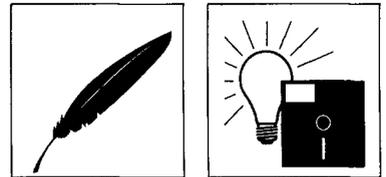


Abb. 9:
Generieren von
C++-Code mit
KPWin++.

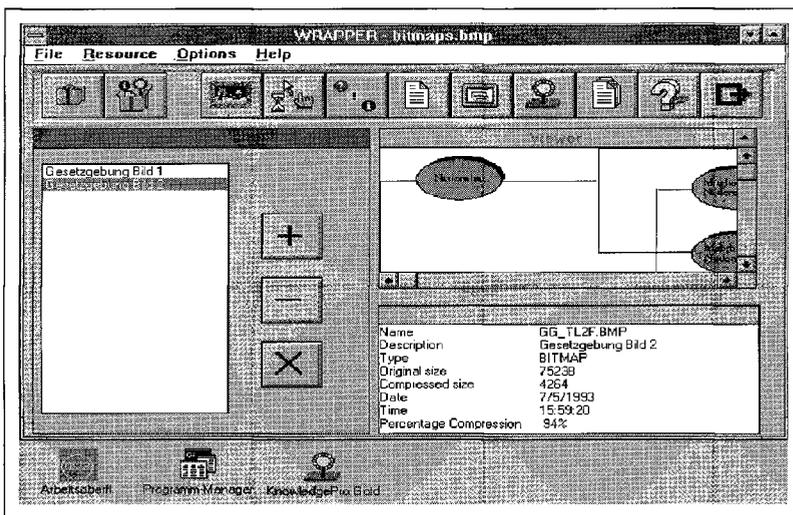


Abb. 10:
Die Oberfläche des
„Wrapper“-Zusatz-
tools mit einem
komprimierten Bit-
map.

5.3 Wrap

VDE's und Autorensysteme leiden oft darunter, daß der gebotene Komfort zu Lasten der Schnelligkeit und der Ressourcen des Endprodukts zur Verfügung gestellt wird.

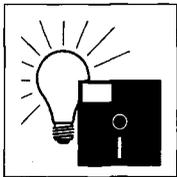
Vor allem die Verwendung von farbigen Bitmaps, nimmt ungeheuer Speicherplatz in Anspruch: so beansprucht ein farbiges Bitmap (256 Farben) von 640 mal 480 Pixel alleine 300 Kilobyte! Eine Anwendung, die mehrere Megabyte nur für die Verwaltung der Bitmaps benötigt, erscheint aber nicht besonders erstrebenswert ... Abhilfe schafft diesbezüglich ein Komprimierungsprogramm namens WRAP, das von KnowledgeGarden als DLL angeboten wird und das Online-Komprimieren und – Dekomprimieren von Texten, Bitmaps, Icons, Programmteilen etc. ermöglicht. Die Verwendung von WRAP ist mit KPWin besonders einfach, weil eine Bibliothek mit neuen Topics eingebunden werden kann, und diese dann an Stelle der bekannten KnowledgePro Befehle verwendet werden können. WRAP kann aber von jeder Sprache verwendet werden, die DLL's ansprechen kann (ua. ToolBook, Visual Basic; C++, Pascal, ...).

WRAP erlaubt weiters, die Ressourcen beliebig miteinander zu kombinieren und in einer Datei abzuspeichern; jede archivierte Ressource kann mit einer bis zu 32 Zeichen langen Bezeichnung versehen werden. Das „natürlichsprachige“ Programmieren wird dadurch noch weiter ermöglicht: anstatt ein Icon mit seinem Dateinamen bezeichnen zu müssen (`load_icon ('xy.ico')`) kann er mit einer passenderen Bezeichnung im Archiv abgelegt und dann direkt mit dem Befehl `wrap_load_icon (ARCHIV, '32 Zeichen lange Ikonbezeichnung')` in den Speicher geladen werden.

Der Komprimierungsfaktor erreicht bei Texten (Gesetzen) ca. 50%; Bitmaps und Icons werden sogar auf weniger als 10% der ursprünglichen Größe zusammengepackt.

Dadurch, daß auf das Archiv direkt mit Hilfe der Bezeichnung der Ressource zugegriffen und auch nur die gewünschte Ressource dekomprimiert wird, verlangsamen sich die Ladezeiten nur unwesentlich (sie können sich wegen der Verkürzung der Zugriffszeit sogar verringern) und kann auf die Indexierung von Texten verzichtet werden.

*Komprimierung von Bitmaps,
Texten etc.*



Durch die Möglichkeit der Bezeichnung der Ressourcen ergeben sich auch datenbankähnliche Verwendungsmöglichkeiten: Jeder Paragraph, Artikel oder jedes Urteil kann unter seiner Bezeichnung in einer Archivdatei abgelegt werden. Zum Retrieval wird dann nur mehr diese Bezeichnung benötigt, zB.: `wrap_load_text (Gesetz, 'Artikel 117')`.

Die Ladezeiten hängen primär von der Größe des komprimierten Eintrags ab, die Gesamtgröße der Archivdatei ist hingegen praktisch belanglos (wie ein Versuch mit einer 1 Megabyte großen, auf die Hälfte komprimierten Datei, die 180 Gerichtsentscheidungen enthielt, ergab).

*Auch lizenzfrei:
WRAP-DLL*

Wird ein Programm mit dem KPWin-Runtime-Modul und der – ebenfalls keiner Lizenzgebühren unterliegenden – WRAP-DLL weitergegeben, so ist der Zugriff auf die komprimierten Ressourcen nur mehr im Rahmen der im Programm implementierten WRAP-Funktionen möglich; dieser Verschlüsselungseffekt mag auch nicht immer ungelegen sein ...

WRAP wird zusammen mit einem KPWin-Programm auch von KnowledgeGarden selber zur Installation von WRAP verwendet; dieses Installationsprogramm liegt ebenfalls im Quellcode bei.

6 Zukunftsmusik

KnowledgePro Windows entstand aus KnowledgePro DOS und ist schon für Microsoft Windows/NT erhältlich.

Weitere Plattformen

KnowledgeGarden Inc. arbeitet derzeit an einem umfangreichen Update der DOS-Version und an KnowledgePro UNIX. Noch in diesem Jahr soll außerdem die Portierung auf Apple MacIntosh und IBM OS/2 erfolgen.

7 Resümee

KnowledgePro Windows ist in der Version 2.3 eine ausgereifte, einzigartige Entwicklungsumgebung^{22 58 59}, die ein ungeheures Potential in sich birgt:

*Einzigartige
Entwicklungsumgebung ...*

Als High-Level Sprache der vierten Generation ermöglicht KPWin sowohl einen leichten Einstieg in die Windows-Programmierung, als auch fortgeschrittenen Anwendern oder professionellen Entwicklern umfangreiche Möglichkeiten intelligenter Programmierung.

Durch die zahlreichen Funktionen zur Listenverarbeitung und zur Textmanipulation und –präsentation ist KPWin im juristischen Bereich sowohl prädestiniert zur Erstellung von Lernsoftware, Information-Retrieval-Software und Hypertextsoftware⁶⁰ als auch zum Aufbau von Expertensystemen, also genau in den für die Rechtsinformatik zukunftsweisenden Bereichen^{61 62}.

Wenngleich andere VDE's durch die Verfolgung eines eingängigen Konzeptes oder eine betont visuelle Ausrichtung schneller zu optisch ansprechenden Einstiegsergebnissen führen mögen, so beschränkt sich KPWin nicht auf diese und ermöglicht auch die einfache Erstellung umfangreicher, komplexer Applikationen sowie die weitere Vertiefung in spezielle Gebiete (wie Hypermedia, OOP, XPS) oder in die Windows-Programmierung allgemein. Die verfügbaren Zusatzprodukte SQL-Kit, KPWin++ und WRAP erlauben auch die rasche Umsetzung professioneller Vorhaben.

... und doch: Die Qual der Wahl.

Der engagierte private Anwender mit juristischem Background, der eigene Ideen umsetzen will, wird den günstigen Preis von Visual Basic und das einfachere Konzept sowie die weitergehenden Animationsfähigkeiten von ToolBook gegen die Vorteile der eleganten, topic-orientierten Strukturierungsmöglichkeiten (mit Regeln und Objekten), der integrierten Text- und Hypertextfunktionen, und des weiteren Horizontes, den KPWin bietet, abwägen müssen.

Wenn KnowledgePro Windows für weitere grafische Benutzeroberflächen zur Verfügung steht, so wird es, um das eingangs gesetzte Zitat von *Ted Nelson* abzuwandeln, auf allen wichtigen Plattformen einfacher werden, einen Computer zu programmieren, als ihn nicht zu programmieren.

⁵⁸ Sarnov, Mehrkämpfer, c't 10/1991, S. 146

⁵⁹ Richard Hale Shaw, PC Magazine, Januar 1991, KnowledgePro Windows, Best of 1990, S. 199: „By putting the power of Windows development in the hands of nonprogrammers, we may look back on the arrival of KnowledgePro Windows as the beginning of the real Windows revolution.“

⁶⁰ zur Benutzerinteraktivität von Dokumenten und KnowledgePro vgl. Philip C. Murray in: Documentation Goes Digital, BYTE 7/1993, S. 126.

⁶¹ Sartor, Artificial Intelligence and Law, CompLex 1/93, Norwegian Research Center for Computers and Law, S. 91 ff.

⁶² Haft, Verhandeln, Die Alternative zum Rechtsstreit, C. H. Beck 1992, S. 242