

# ObjectVision Workshop:

## Automatische Tenorierung der vorläufigen Vollstreckbarkeit

Matthias Kraft

Die vorläufige Vollstreckbarkeit von Urteilen gehört zu den typischerweise automatisierbaren Vorgängen auf allen Arbeitsplätzen von Richtern. Sie muß regelmäßig tenoriert und begründet werden. Das zugrundeliegende Regelwerk ist nicht sehr komplex im Vergleich zu anderen Problemen, kann jedoch auch nicht als trivial bezeichnet werden. Dieses Thema bietet sich also für eine Demonstration der Arbeitstechnik mit einem Werkzeug wie ObjectVision an. Die folgenden Ausführungen sollen dabei kein Patentrezept bieten, sondern lediglich exemplarisch den Arbeitsgang eines Projektes beschreiben.

### Aufarbeitung der normativen Grundlagen

Die Übertragung eines juristischen Regelwerkes in die EDV setzt immer ein intensives Studium der gesetzlichen Grundlagen voraus. Bei der Erstellung einer ObjectVision-Applikation werden bereits anfangs folgende Fragen im wesentlichen durch eine Analyse der Normen zu beantworten sein:

- Welches Ziel wird mit der Anwendung verfolgt? Soll das Endprodukt – wie wohl meistens – ein Schriftsatz sein oder reicht das reine Rechenergebnis aus? Kommt es in dem Schriftsatz etwa bei der Begründung einer Entscheidung auch auf den Rechenweg an?
- Inwieweit lassen sich die für die Erreichung des Ziels notwendigen Informationen aus anderen Daten berechnen? Welche Regeln stellt das Gesetz hierfür zur Verfügung? Dabei sind unter Berechnung auch logische Entscheidungen zu verstehen. Zudem ist zu klären, ob eine Berechnung nur einen vom Benutzer änderbaren Vorschlag oder eine bindende Entscheidung darstellt.
- Welche Informationen werden insgesamt vom System benötigt und in welcher Form? Welche Daten muß der Benutzer sehen? Können einzelne Daten in unterschiedlichen

Formularen zusammengefaßt werden? Welche dieser Daten sind schon an anderer Stelle im System erfaßt oder lassen sich leicht aus solchen Daten errechnen.

Praktisch ist zu Anfang eines Projektes die elektronische Erfassung und Aufarbeitung des Gesetzestextes. Dabei kann unter Umständen eine Übernahme der Normen aus der juris-Normdatenbank in Betracht kommen. Der Gesetzestext kann dann z. B. in eine Hypertext-Umgebung wie WinHelp eingegeben werden. Bereits die Verschlagwortung und die Verknüpfung der einzelnen Normen macht deren genaue Analyse unumgänglich und führt oft zu interessanten Erkenntnissen. Zudem liegt der Text dann elektronisch vor, was spätestens zur Erstellung eines Hilfesystems für die geplante Anwendung von Nutzen sein kann.

### Zielvorgabe und notwendiger Input

In der Praxis wird man zu Beginn der Arbeit ObjectVision starten und ein neues leeres Formular im Autorenmodus öffnen. Dies ermöglicht ein spielerisches Ausprobieren der Überlegungen. Oft zeigt erst das Erproben der Algorithmen weitere juristische Probleme der Materie.

Im Beispielfall der vorläufigen Vollstreckung ist Endziel der Anwendung die Generierung ei-

nes Tenors sowie einer Begründungspassage im Urteil. Technisch gesehen wird also ein Teil eines Text(verarbeitungs)dokuments erzeugt. Endergebnis und Entscheidungsweg müssen demnach in irgendeiner Form der Textverarbeitung (in diesem Falle WinWord) zugänglich gemacht werden. Dieses Endziel sollte man bei der Arbeit im Auge behalten.

Betrachtet man die einschlägigen Vorschriften §§ 708 und 709 ZPO, so stellt man fest, daß zur Tenorierung folgende Informationen benötigt werden:

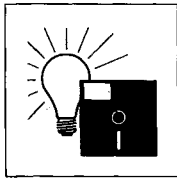
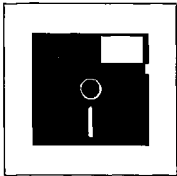
- Ist das Urteil für Kläger und/oder Beklagten vorläufig vollstreckbar?
- In welcher Höhe muß der Kläger/Beklagte Sicherheit leisten?
- Hat der Beklagte/Kläger die Möglichkeit, die Vollstreckung durch Sicherheitsleistung abzuwenden?

Es empfiehlt sich, zu jeder dieser Fragen ein entsprechendes Feld in dem offenen Formular einzurichten. Dabei werden die Sicherheitsleistungen als Eingabefelder mit Währungsformat definiert. Für die Frage der vorläufigen Vollstreckbarkeit wird ein Feld des Typs „Wahr/Falsch“ benötigt (Abb. 1, auf der Folgeseite).

### Berechnungsbäume

Anschließend folgen die Überlegungen zur Berechnung dieser Ergebnisfelder:

Matthias Kraft  
ist Assistent am  
Lehrstuhl für  
Rechtsinformatik  
der Universität  
des Saarlandes  
und Autor  
des Programms  
SOPHOS



Urteile sind generell für vorläufig vollstreckbar zu erklären gem. §§ 708, 709 ZPO. Eine Ausnahme ergibt sich aus § 712 I S. 2 ZPO. Dem Benutzer muß demnach die Möglichkeit eingeräumt werden, eine Entscheidung gem. § 712 ZPO zu fällen. Eine Analyse der Norm läßt erkennen, daß sich in den zwei Absätzen mit jeweils zwei Sätzen insgesamt 5 Entscheidungsvarianten verbergen, die sich im Prinzip gegenseitig ausschließen. (4 Sätze + die Möglichkeit, daß die Norm nicht

einschlägig ist). Zudem kann die Norm gegenüber beiden Parteien wirksam sein. Zur Darstellung dieser Alternativen werden entsprechend zwei sogenannte *Markierungsfelder* verwendet. In ihnen können die einzelnen Varianten vom Anwender alternativ durch „Ankreuzen“ ausgewählt werden. Da die vorläufige Vollstreckbarkeit von keinen weiteren Informationen abhängt, können nun die Regeln für die Entscheidung der Vollstreckbarkeit implementiert werden. Hierzu

erstellt man einen sog. Berechnungsbaum für das zu berechnende Feld. Abbildung 2 zeigt einen derartigen Berechnungsbaum für das Feld „Das Urteil ist vorläufig vollstreckbar (KL)“.

Der Baum ist wie folgt zu interpretieren: Hat das Feld „§ 712 Abwendungsbefugnis d Bkl“ den Wert „Abs 2 S 2“ hat, so erhält das zu berechnende Feld den Wert „Nein“, sonst erhält es den Wert „Ja“ (Abb. 2).

Als nächstes muß entschieden werden, ob der jeweilige Vollstreckungsgläubiger eine Sicherheit zu leisten hat: Generell ist vor der Vollstreckung gem. § 709 ZPO eine Sicherheit zu leisten. Ausnahmen ergeben sich in den Fällen der §§ 708 und 710 ZPO. Der Anwender muß also die Möglichkeit erhalten einzugeben, ob eine der Ausnahmen vorliegt. Dabei sollte beachtet werden, daß im Falle des § 708 ZPO mehrere Alternativen möglich sind, die sich zumindest bei späteren Entscheidungen etwa wegen § 711 ZPO unterschiedlich auswirken können. Hier wird also ein entsprechendes Markierungsfeld nötig sein, das die Auswahl der einzelnen Nummern ermöglicht. Zu bedenken ist allerdings, daß § 708 Nr. 11 ZPO nicht im Widerspruch zum tatsächlichen Streitwert oder den sich hieraus ergebenden Gebühren stehen darf. Die Anwendung könnte statt dessen aus dem Streitwert, und dem Prozeßergebnis selbst eine Entscheidung bezüglich des § 708 Nr. 11 ZPO treffen, sofern nicht eine der übrigen Nummern des § 708 ZPO einschlägig ist. Am einfachsten ist es hierzu, die Nr. 11 in einem getrennten Feld zu behandeln. Die Abbildung 3 stellt einen Vorschlag für die Anlage der Felder dar.

Nach der Implementierung der Felder kann der Berechnungsbaum für die Frage der Sicherheitsleistung erstellt werden. Einen Vorschlag hierzu zeigt die Abbildung 4.

Der Baum enthält dabei eine Besonderheit, da die Entscheidung über das Ob der Sicherheitsleistung und über die Höhe kombiniert

erstellt man einen sog. Berechnungsbaum für die Frage der Sicherheitsleistung erstellt werden. Einen Vorschlag hierzu zeigt die Abbildung 4.

Der Baum enthält dabei eine Besonderheit, da die Entscheidung über das Ob der Sicherheitsleistung und über die Höhe kombiniert

Abb. 1: Ansicht des ersten Fensters des neuen Formulars in der Arbeitsebene „Formular“

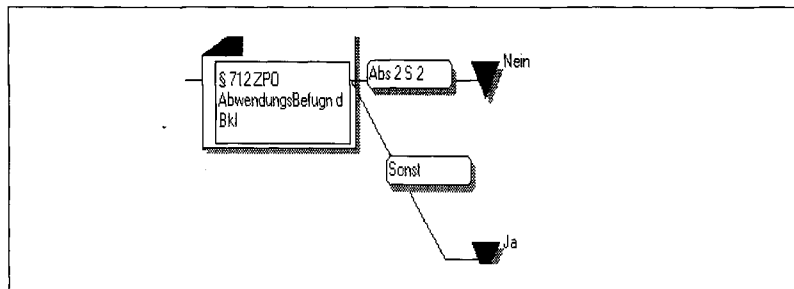


Abb. 2: Berechnungsbaum für „Das Urteil ist vorläufig vollstreckbar (KL)“

Abb. 3: Felder zur Darstellung der Problematik des § 708 ZPO

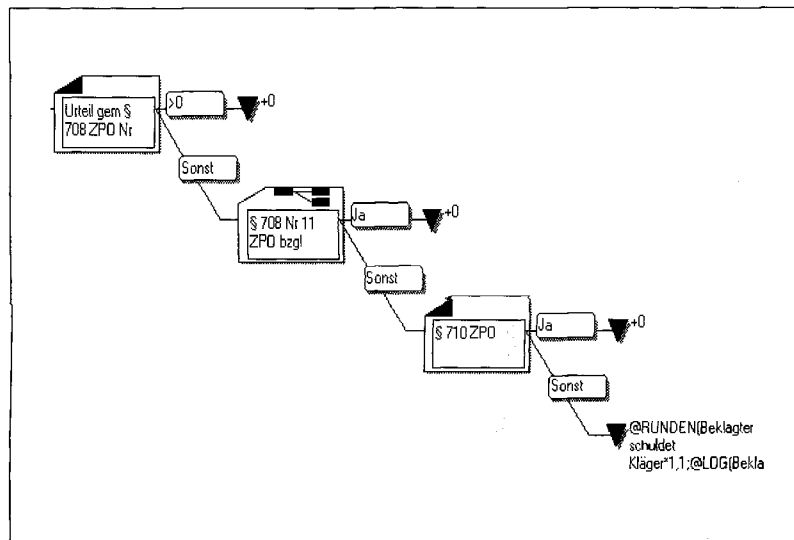


Abb. 4: Berechnungsbaum für die Höhe der notwendigen Sicherheitsleistung des Klägers. Sie ist „0“, wenn keine Sicherheit zu leisten ist. Der Regelfall wird immer durch die Bedingung „Sonst“ dargestellt.

nirt werden. Im folgenden wird auf die Berechnung der Höhe der Sicherheitsleistung näher eingegangen. Die Sicherheitsleistung muß im Prinzip den Betrag abdecken, den der Vollstreckungsschuldner bei einem Wegfall der vorläufigen Vollstreckbarkeit vom Vollstreckungsgläubiger verlangen kann.

Rechtliche Grundlage für die Rückabwicklung sowie eventuelle Schadensersatzforderungen ist § 717 ZPO. In der Praxis werden regelmäßig der Wert der zugeprochenen Hauptforderung, die an den Schuldner zu zahlenden Prozeßkosten sowie ein Sicherheitszuschlag angesetzt. Letzterer wird meist über den Daumen gepeilt, so daß sich eine runde Summe ergibt.

Mathematisch läßt sich diese Überlegung ausdrücken wie in Listing 1 gezeigt.

Dabei stellt die Funktion „Daumen“ ein Problem dar, da sie im Funktionsumfang von ObjectVision nicht enthalten ist. An ihrer Stelle soll die Funktion „@runden“ verwendet werden. Die Genauigkeit der Rundung soll durch die Anzahl der Stellen der Zahl bestimmt werden. Die Stellenzahl ihrerseits wird mathematisch durch den (ganzzahligen) Logarithmus der Zahl ausgedrückt.

Das ergibt für die Höhe der Sicherheitsleistung die Beispielformel in Listing 2.

Zur Berechnung der Sicherheitsleistung fehlen demnach Angaben darüber, ob und zu welcher Leistung der Beklagte verpflichtet wurde, sowie über die Höhe der Prozeßkosten, die er an den Beklagten zu leisten hat (und umgekehrt). Um dem Anwender möglichst viel Arbeit abzunehmen, soll die Anwendung Hilfen insbesondere zur Berechnung der Gerichts- und Anwaltsgebühren bereitstellen.

Benötigt werden folgende Informationen:

- anwaltliche Vertretung der Parteien
- Höhe des Streitwertes
- Prozeßergebnis, d. h. Obsiegen des Klägers.

Die entsprechenden Felder werden zunächst im Formular angelegt. Danach werden die Berechnungsbäume entwickelt.

Die Berechnung einer Gebühr nach GKG bzw. BRAGO könnte in einem ObjectVision-Berechnungsbaum vorgenommen werden. Dieser Baum ist jedoch recht komplex (Abb. 5).

## Einbinden von DLLs

Da die Berechnung derartiger Gebühren immer wieder in juristischen Anwendungen auftaucht, stellt sich die Frage nach einer ökonomischeren Lösung. Hierzu bietet ObjectVision die Möglichkeit, sog. DLLs in die Applikationen zu integrieren.

DLLs sind Windows-spezifische Bibliotheken von Funktionen, die dynamisch in andere Programme eingebunden werden können. Die „GEBUEHR.DLL“ z. B. wurde am Lehrstuhl für Rechtsinformatik in Saarbrücken

mit TurboPascal für Windows entwickelt. Sie enthält Funktionen zur Berechnung von diversen Gebühren, wobei die Gebührentabellen sehr einfach zu pflegen sind. (Andere DLLs mit juristischen Funktionen wurden beispielsweise von V. Nilgens entwickelt und stehen in der jur-pc Mailbox zur Verfügung; vgl. zu den DLLs noch Nilgens, jur-pc 1992, S. 1762-1763.)

Die benötigten Funktionen einer DLL müssen beim Öffnen des Formulars dynamisch eingebunden werden. Das erfolgt durch einen sogenannten *Ereignisbaum*. Die ganze Applikation, ein Feld, ein Formular oder jedes andere Objekt kann einen derartigen Baum besitzen. Er ähnelt dem Berechnungsbaum, führt jedoch nur beim Auftreten von definierten Ereignissen wie Mausclicks, Tastenschlüsseln etc. bestimmte Aktionen aus. In diesem Fall reagiert der Ereignisbaum der gesamten Applikation auf das Ereignis „Öffnen“ mit den Funktionen in Listing 3.

```
Sicherheitsleistung = Daumen ((O+P)*1,1)
O: Obsiegen
P: Prozeßkostenanteil
```

Listing 1

```
Sicherheitsleistung = @runden(O+R*1,1;@log((O+R)*1,1)-2)
O: Obsiegen
P: Prozeßkostenanteil
```

Listing 2

```
EINBINDEN („@GKG“; "BB"; "Streitw"; "gebuehr.dll"; "GKG"; 0)
EINBINDEN („@BRAGO“; "BB"; "Streitw"; "gebuehr.dll"; "BRAGO"; 0)
```

Listing 3

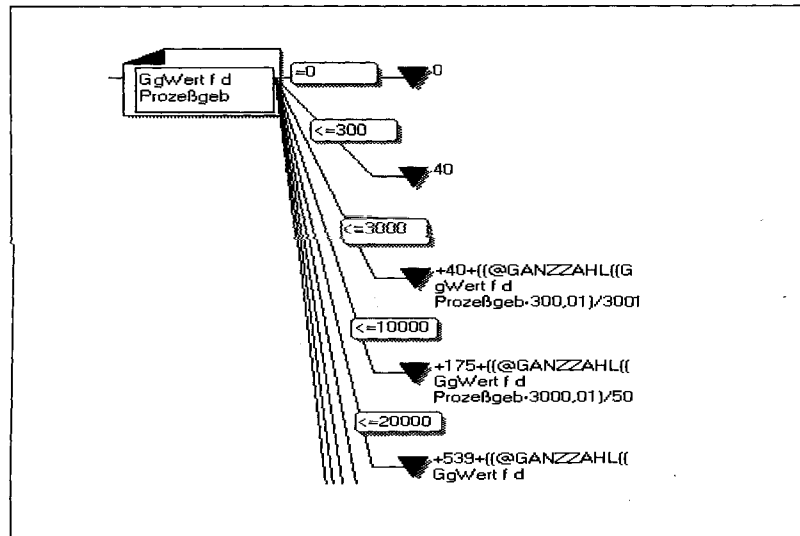
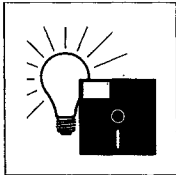
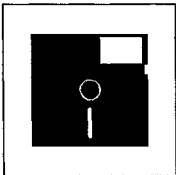


Abb. 5:  
Ausschnitt aus einem Berechnungsbaum für Gerichts- bzw. Anwaltsgebühren



Ab jetzt stehen in der Applikation die beiden Funktionen @GKG(Streitw) und @BRAGO(Streitw) sowie alle anderen ObjectVision-Funktionen zur Verfügung. Sie berechnen aus Streitw die entsprechenden GKG- und BRAGO-Gebühren. Mit diesen neuen Funktionen lassen sich die benötigten Berechnungen leichter ausführen. Die folgenden Schritte sollen nun nicht mehr so detailliert beschrieben werden, da sich die dargestellte Arbeitsweise lediglich wiederholt. Exemplarisch herausgegriffen sei noch der Problembereich der Aufhebung der Kosten:

In § 92 II ZPO wird dem Gericht die Möglichkeit eingeräumt, die Kosten des Rechtsstreits gegeneinander aufzuheben, wenn das Prozeßergebnis in etwa ausgeglichen ist. In diesem Fall hätte jeder Beteiligte die eigenen An-

waltskosten selbst zu tragen, was wiederum Auswirkungen auf die Höhe der vollstreckbaren Forderung hat. Da es sich um eine Kann-Vorschrift handelt, muß die Applikation den von ihr berechneten Entscheidungsvorschlag zur Disposition stellen. Dies erfolgt mit einem Wahr/Falsch-Feld „Aufhebung der Kosten“, in dem der Vorschlag angezeigt wird und geändert werden kann. Alle Felder, die von dieser Entscheidung abhängen, müssen im Berechnungsbaum eine entsprechende Verzweigung vorsehen.

### Import von vorhandenen Daten

Nach der Anlage aller notwendigen Felder und der Fertigstellung der Berechnungsbäume sollte die

Anwendung zunächst gründlich getestet werden. Zu diesem Zeitpunkt arbeitet sie noch lokal, d. h., sie hat keinen Kontakt zu externen Informationsquellen. Nun ist es Zeit, Möglichkeiten einzubauen, bereits erfasste Daten aus anderen Quellen zu übernehmen. Im Testversuch existierte beispielsweise eine in Superbase implementierte Testdatenbank mit – rudimentären – Verfahrensdaten (Abb. 6). Nachdem die vorläufige Vollstreckbarkeit naturgemäß erst gegen Ende eines Falls eine Rolle spielt, sind in dieser Datenbank Informationen wie Streitwert oder anwaltliche Vertretung der Parteien bereits vorhanden. Mit Hilfe der Option „Verbindungen ...“ werden deshalb DDE-Verknüpfungen zu den entsprechenden Datenbankfeldern angelegt (Abb. 7).

Zur Bedienung der Datenbank wird ein eigenes Fenster eingerichtet, in dem der Benutzer die wesentlichen Inhalte der Datenbank angezeigt erhält und in ihr mit Hilfe von Schaltern blättern kann. Komplexere Suchen müssen allerdings im Datenbankprogramm selbst durchgeführt werden. Zur Konsolidierung des Datenbankinhaltes sollte vorgesehen werden, daß Datenänderungen etwa des Streitwertes im Formular vorgenommen werden können und automatisch mit der Datenbank abgeglichen werden. Hierzu löst das geänderte Feld nach der Eingabe das Ereignis „Ändern“ aus. Eine Funktion des Ereignisbaumes des Feldes reagiert auf dieses Ereignis, indem sie den neuen Wert an die Datenbank via DDE zurückschickt.

AZ 2C1/91 Angelegt 28.Apr.1977

Name Meyer J. Lehmann

Richter Streng  Einzelrichter

Streitwert 2312,45  Anwalt  Kl  Bkl

Bemerkung Familiensache mit Tücken

Beteiligung als	Name	Vorname
KL	Meyerin	Hedwig
BKL	Lehmann	Markus
RAKL	Schlaumeyer	K.
RABKL	Pechvogel	R.

Abb. 6: Maske der verwendeten Testdatenbank

**DDE-Verbindung**

Verbindung

Anwendung

Dokument

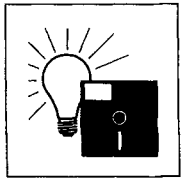
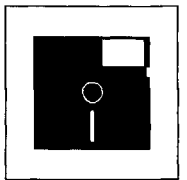
Name	Verbunden mit
Kl_m_Anw	Kläger anwaltlich vert
Bkl_m_Anw	Beklagter anwaltlich v
AZ	AZ
Streitwert	Streitwert
Name	Sache

OK  
 Hilfe  
 Abbruch

Abb. 7: ObjectVision-Dialogbox zur Anlage von DDE-Verknüpfungen

### Verbindung zur Textverarbeitung

Zur Weiterverarbeitung sollen die Daten wie anfangs postuliert an die Textverarbeitung (hier WinWord) übermittelt werden. Dies erfolgt regelmäßig über eine oder mehrere DDE-Verbindun-



gen. (Selbstverständlich ließe sich auch eine ASCII-Datei erzeugen, die WinWord hinterher importiert. Diese Vorgehensweise entspricht allerdings nicht mehr dem Stand der Technik.) Die Übermittlung der Daten via DDE kann auf verschiedene Weise erfolgen:

- Das Textverarbeitungsprogramm kann die Daten von Object Vision anfordern.
- ObjectVision kann die Daten von sich aus übermitteln.
- ObjectVision kann anhand von Vorgaben Makros in der Textverarbeitung auslösen, die beispielsweise bestimmte Textbausteine einfügen.

Der Einsatz dieser unterschiedlichen Mittel hängt ganz von dem Einzelfall ab und ist oft reine Geschmackssache. Man sollte jedoch immer darauf achten, daß jede Applikation genau die Aufgaben übernimmt, für die sie bestimmt ist. So sollten etwa längere Textpassagen in der Textverarbeitung verwaltet werden und nicht in ObjectVision.

Im Falle der Tenorierung der vorläufigen Vollstreckbarkeit, ergibt sich eine begrenzte Anzahl von Formulierungen. Sie werden als Textbausteine in WinWord gepflegt. Die Entscheidung, welche Textbausteine im konkreten Fall einzusetzen sind, fällt dagegen in ObjectVision, da die hier möglichen Berechnungsbäume geradezu geschaffen sind, für derartige Entscheidungsvorgänge. Für die Übertragung des Textbausteinennamens sind zwei Wege denkbar:

- Das Formular löst das Makro „BearbeitenTextbaustein“ in Word aus und fügt so den Textbaustein selbst ein.
- WinWord steuert die Ergebnisübernahme mittels Makro. Hierbei wird der Textbausteinname in einem definierten ObjectVision-Feld abgefragt und der entsprechende Baustein in das Dokument eingefügt.

Letztere Variante ist im Endeffekt einfacher und verdient den Vorzug. In der Testanwendung wird die eigentliche Tenorierung der Vollstreckbarkeit in drei Bau-

steingruppen vorgenommen, die modular hintereinander gesetzt werden. Zunächst wird über die vorläufige Vollstreckbarkeit generell entschieden, danach werden getrennt Sicherheitsleistung des Klägers sowie Abwendungsbefugnis des Beklagten und umgekehrt entschieden. Die Abbildung 8 zeigt den Berechnungsbaum, der den Namen des ersten Bausteins bestimmt.

Ein WinWord-Makro, das den Inhalt des Feldes mit diesem Namen abfragt und als Textbausteinnamen interpretiert, zeigt Listing 4.

Ein derartiges WinWord-Makro kann zum einen durch einen Schalter im ObjectVision-Formular angestoßen werden, so daß

der Benutzer hierzu das Formular nicht verlassen muß. Zum anderen kann der Aufruf von ObjectVision und der anschließende Ablauf des Makros auch durch einen Menübefehl in WinWord initiiert werden. So kann die automatische Entscheidung der Tenorierung leicht in den Prozeß der Abfassung des Urteils integriert werden.

Werden für den Tenor Zahlenwerte aus dem Formular benötigt, etwa die Höhe der notwendigen Sicherheitsleistung, so erfolgt die Übernahme der Werte aus dem Formular durch ein entsprechendes WinWord DDE-Feld. Es kann als Platzhalter direkt im Textbaustein platziert werden (Listing 5).

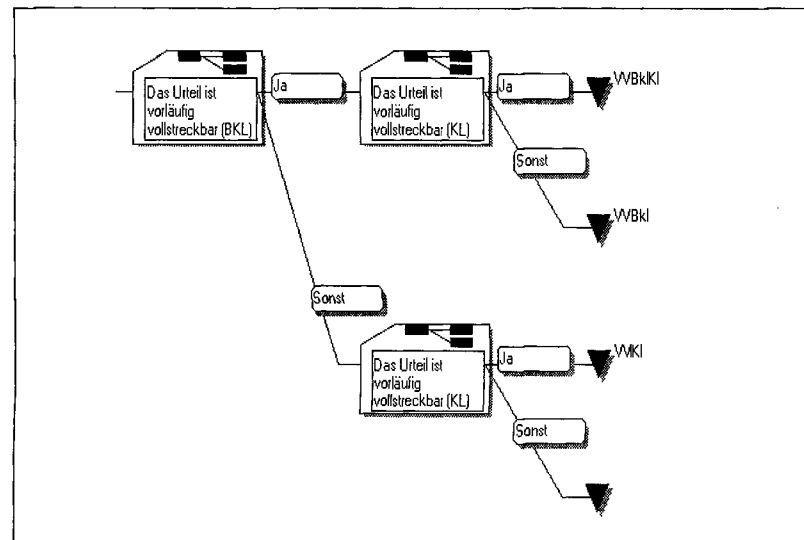


Abb. 8: Berechnungsbaum zur Ermittlung des Textbausteinennamens, der die vorläufige Vollstreckbarkeit generell tenoriert.

```

Sub MAIN
' Öffnen der DDE-Verbindung
DDEKanal = DDEInitiate("Vision", "zpo-vv2.ovd")
' ...möglichen Fehler abfangen...
' springen zur Position im Dokument
BearbeitenGeheZu "TenorVV"
' Abfragen des Textbausteinennamens
Bst$ = DDERequest$(DDEKanal, "BstTenorVV")
' Einfügen des Textbausteins
if Bst$ <> "" Then
    BearbeitenTextbaustein Bst$, 1
End If
' ...Weitere Textbausteine einfügen...
' DDE-Verbindung wieder Schließen
DDETerminate DDEKanal
End Sub

```

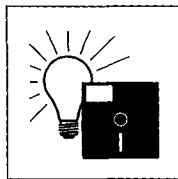
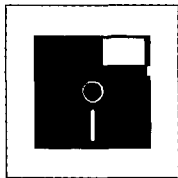
Listing 4

```

Der Kläger hat vor der Vollstreckung eine Sicherheit in Höhe
von DM {DDE VISION ZPO-VV2.OVD "Sicherheitsleistung KL" \*
FormatVerbinden} zu leisten.

```

Listing 5



## Feinarbeit an der Benutzerschnittstelle

Nach der Implementierung der Funktionalität kann an die Optimierung der Benutzerschnittstelle gegangen werden. Die Genauigkeit und Sorgfalt, die hierauf verwendet werden, hängen insbesondere von der Frage ab, ob auch Dritte die Anwendung benutzen sollen bzw. ob die Anwendung gar kommerziell vertrieben werden soll. Der „Einzelkämpfer“ etwa bei Gericht sollte dabei bedenken, daß ein derartiges Programm auch für den Nachfolger eine geballte Ladung an Wissen parat hält.

Zunächst sind die bis zu diesem Zeitpunkt angelegten Felder in funktionalen Einheiten, d. h. in einzelnen Formularen zu gruppieren. Danach kann an die graphische Feinarbeit gegangen werden. Schriftarten und Umrahmungen der Felder sind zu wählen. Lücken können mit Mustern ausgefüllt werden. Oft ist ein Erklärungstext recht hilfreich. Dieser kann in Abhängigkeit von vorangegangenen Eingaben oder anderen Parametern variieren. Weiterhin können einzelne Felder gegen Änderung gesichert werden.

Diese Möglichkeit wird man etwa dann ausnützen, wenn eine Entscheidung des Programms bindend ist. Auch sollten Felder, die lediglich Informationen aus anderen Datenquellen enthalten (z. B. das Aktenzeichen aus der

Datenbank) nicht vom Endanwender verändert werden können.

Ein wesentlicher Teil der Schnittstellengestaltung ist die Implementierung von kontextsensitiver Hilfe. Hierzu bietet ObjectVision selbst eine entsprechende Option. Sie erlaubt es, für jedes Feld einen Hilfetext zu hinterlegen, den der Benutzer mit <F1> abrufen kann. Im vorliegenden Fall bietet sich die Eingabe der ohnehin schon elektronisch erfaßten Normen für die jeweiligen Felder an.

Zusätzlich ist – allerdings nicht ohne einige Mühen – auch die Einbindung von WinHelp-Titeln möglich. Da diese Funktionalität nicht unmittelbar von ObjectVision unterstützt wird, wird hierzu die Funktion „@WinHelp(Datei;Thema)“ aus einer DLL eingebunden. Im Ereignisbaum der Applikation wird auf das Ereignis „Strg+H“ hin je nach bearbeitetem Feld („@AktFeldName“) ein entsprechendes Hilfethema aufgerufen. Vorteile der Hilfe durch WinHelp sind die Hypertext- und Suchkomponenten des Systems. Zudem kann von mehreren Applikationen etwa auf eine gemeinsame Gesetzesdatenbank zugegriffen werden.

## Ein Wort zum Schluß

Der Leser wird gemerkt haben, daß sich auch durch ObjectVision nicht alle Probleme wie von selbst lösen. Die Entwick-

lung derartiger Anwendungen bindet viel Zeit und Energie. Dennoch sprechen einige Punkte für diese Technik:

- Es ist tatsächlich in gewisser Rahmen möglich, ohne Programmierkenntnisse Rechen- und Entscheidungsprobleme an einem individuellen Arbeitsplatz zu lösen.
- Der Systembetreuer einer Arbeitseinheit in Betrieben, Behörden oder Kanzleien hat eine sehr schnelle Möglichkeit, auf individuelle Anforderungen einzelner Arbeitsplätze einzugehen.
- Entscheidungen des Systems bleiben auch für den Endnutzer immer transparent.
- Für Hobby-Programmierer dürfte die Einbindung von DLLs interessant sein. Hierdurch kann er sich ohne lange Einarbeitung in die Windows-Programmierung über seine alten Algorithmen eine ansprechende Oberfläche legen.
- Die Integration in eine vorhandene Arbeitsumgebung ist verhältnismäßig leicht zu realisieren.

Bei der Einführung von ObjectVision etwa bei Gericht sollte darauf geachtet werden, daß die gesamte Organisation ein derartiges Werkzeug verträgt. Wenn sich nämlich jeder Benutzer auch in die Entwicklung der ObjectVision-Applikationen einarbeiten muß, etwa weil keine Logistik zur Erstellung der Formulare vorhanden ist, kann ein derartiges Werkzeug letztendlich auch sehr kontraproduktiv wirken.



# JURISTISCHE SOFTWARE VON A - Z

**Software, CD-ROM's, Hardware, Zubehör. Alles aus einer Hand!**

bitte Spezialkatalog anfordern bei:  
 soft-use Computerprogramme  
 Benner, Kralls & Trossen GbR  
 im Mühlberg 39, 5230 Altenkirchen  
 Tel.: 02681/70468 - Fax: 02662/5795