Teamwork: Schönfelder Plus und WordBasic

Maximilian Herberger

Der Multimedia Viewer von Microsoft scheint bereits jetzt zu den Produkten zu gehören, denen dauerhafte Aufmerksamkeit zu widmen sich lohnt. Er ist ein kostengünstiges (knapp unter 1.000,– DM) Instrument für das elektronische Publizieren. Bei der Verteilung damit hergestellten elektronischen Materials fallen keine zusätzlichen Lizenzkosten an, die erforderliche Runtime darf frei weitergegeben werden. Bereits das erklärt zusammen mit der Leistungsfähigkeit des Produkts das Erscheinen zahlreicher Multimedia Viewer-Anwendungen, unter denen der elektronische Schönfelder als juristisch-prominentes Beispiel zu nennen ist (vgl. jur-pc 8/94, S. 2740–2746). Es kommt aber, worauf Dechsling zu Recht hingewiesen hat, noch ein weiteres hinzu: *"Es gibt eine leidlich dokumentierte Programmierschnittstelle, die verlagsspezifische Modifikationen zuläßt" (jur-pc 8/94, S. 2749).* Davon soll im folgenden die Rede sein. Denn der Viewer ist tatsächlich ein erstaunlich offenes und kooperatives Programm. Das hängt teilweise mit seiner Einbettung in die Windows-Architektur zusammen, beruht aber teilweise auch auf einer konzeptuellen Entscheidung von Microsoft, die so weit geht, daß ein Beispielsprogramm in C mitgeliefert wird, das demonstriert, wie man den Viewer nahezu komplett "fernsteuern" kann. Das alles ist begrüßenswert, wenngleich man Dechsling nach einiger Zeit der Arbeit nachfühlen kann, warum er die Programmierschnittstelle als "leidlich dokumentiert" bezeichnet. Hier würde man sich größere Ausführlichkeit aber auch größere Sorgfalt wünschen. Es gibt leider ärgerliche Fehler, die viel Zeit kosten, weil man meistens zuerst annimmt, man selbst habe etwas falsch gemacht.

"Programmierschnittstelle": Nur ein kleiner Stolperstein ...

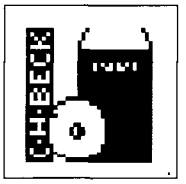
Wer als Anwender von Standardprogrammen von einer "Programmierschnittstelle" hört, wird meist davor zurückschrecken, sich in das Thema zu vertiefen. Suggeriert doch die Bezeichnung, daß man Kenntnisse einer Programmiersprache benötigt, um sich diese Programmierschnittstelle zunutze zu machen. Hat man dann noch einen Blick in den mitgelieferten C-Beispielscode geworfen, erweitert sich der erste Eindruck dahingehend, daß ohne einen C-Kurs weitere Schritte untunlich sind. Daß dem nicht so ist, soll im folgenden gezeigt werden. Natürlich geht es ohne ein wenig Programmieren nicht ab. Aber schon normale Kenntnisse in WordBasic, der zu Word gehörenden Programmiersprache, sind ausreichend, um einige interessante Aktionen in Viewer-Anwendungen anzustoßen (*vgl. zu WordBasic außer den Informationen in der Word-Hilfe noch Russell Borland, Microsoft WordBasic – Handbuch für Programmierer).* Da der elektronische Schönfelder sicherlich die gegenwärtig dominierende juristische Viewer-Anwendung ist, sollen die entsprechenden Möglichkeiten an diesem Beispiel demonstriert werden. Die dabei gewonnenen Erkenntnisse sind aber auf alle Viewer-Anwendungen übertragbar.

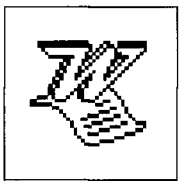
Arbeitsziel: Ein Baukasten

Die nachfolgenden Beispiele beanspruchen nicht, komplette, ausgearbeitete Applikationen darzustellen. Sie verfolgen einzig und allein den Zweck, prinzipiell zu erläutern, wie aus WordBasic heraus einzelne Viewer-Funktionen erreichbar sind, um dadurch eine gegenwärtig vielfach noch bestehende Hürde zu beseitigen. Jeder erfahrene WordBasic-Programmierer wird dann sofort sehen, wo Raum für Verfeinerungen besteht.
Dargestellt werden die verwendeten Viewer-Programmierelemente im Handbuch *"Microsoft Multimedia Viewer Publishing Toolkit – Technical Reference" (im folgenden: TiR),* das man beim Erwerb des Multimedia Viewers erhält.
Die Makros werden auf Diskette der nächsten Ausgabe von jur-pc beigefügt.

Per Doppelklick aus Word zur Norm

In juristischen Texten werden Normen in Bezug genommen. Es wäre für Autor und Leser eine schöne Möglichkeit, wenn durch Doppelklick auf die Normreferenz der entsprechende Schönfelder-Text aufgeblättert würde. Der Makro SCHAUFRUF *(vgl. Kasten 1 auf der folgenden Seite)* realisiert das.
Natürlich könnte man unter Windows etwas ähnliches dadurch erreichen, daß man den Schönfelder während der Textverarbeitung in einem anderen Fenster stehen läßt und dann jeweils dort nachschlägt. Insofern drängt sich der unabweisbare Nutzen dieses Makros nicht unbedingt auf. Er liefert aber, und deswegen steht er (trotz eines sogleich noch zu beschreibenden Mangels an Eleganz) am Anfang, nützliche erste Informationen über die "Fernsteuerung" des Schönfelder mit WordBasic.





Der Makro SCHAUFRUF

Der Schlüssel zum Verständnis des Schönfelder-Aufrufs liegt in der folgenden Funktionsdeklaration:

```
Declare Function
VwrCommand Lib "mva-
pi2.dll"(vwr As Inte-
ger, MVB$, Command$,
optcmd As Integer) As
Integer
```

Diese besagt folgendes:
In der "Dynamic Link Library" (DLL) "MVAPI2.DLL" *(API für 'Application Programming Interface')* wird eine Funktion *VwrCommand* zur Verfügung gestellt, die es erlaubt, Viewer-Befehle anzustoßen. Übergeben werden müssen bei Verwendung der Funktion im Programm die in Klammer stehenden vier Parameter *(vgl. TR 9–21)*.
vwr identifiziert (auch bei mehreren laufenden Viewer-Anwendungen) die Anwendung, die man erreichen will. Verwendet man den Wert *NULL*, wird die angegebene Viewer-Anwendung gestartet.
MVB$ enthält den Namen der .MVB-Datei, die man ansprechen will, im Falle des Schönfelder *SCH.MVB*. Im Beispielsfall befindet sich die Schönfelder CD im CD-ROM-Laufwerk D.
command$ ist der an den Viewer zu übermittelnde Befehl. Im vorliegenden Falle ist das der Sprungbefehl JumpID (abgekürzt als JI), der als Parameter die .MVB-Datei, in die gesprungen werden soll, und die dortige Zielstelle enthält *(vgl. TR 5–57)*.

Um die Zielstelle angeben zu können, muß man wissen, wie im Schönfelder Normdokumente benannt werden. Der Namensstring besteht aus "SCHF" gefolgt von der Gesetzesabkürzung (hier: BGB) und der Paragraphenziffer, jeweils durch Tiefstriche in der aus dem Beispiel ersichtlichen Form verbunden. Besteht das Gesetz aus Artikeln, wird vor die Ziffer ein A gesetzt. Art. 1 GG würde man also mit "SCHF_GG_A1" erreichen. Zwei weitere interessante Sprungtypen seien noch ergänzend genannt:
Der erste verzweigt zu der Stelle in einem Gesetzesinhaltsverzeichnis, an der eine Norm ihren Platz hat: "SCHF_GBO_17IV" etwa würde das Inhaltsverzeichnis der Grundbuchordnung beginnend mit § 17 aufblättern. Der zweite erreicht die den Gesetzen vorangestellten Änderungsverzeichnisse ("SCHF_STVG_AENDERUNGSUEBERSICHT" für die Übersicht zum StVG)

"Beck connectivity"

Das als "Beck connectivity" avisierte Konzept besteht im Kern darin, daß man bei Kenntnis der Dokumentennamen aus einer geeigneten Umgebung dorthin "springen" kann *(vgl. Dechsling, jur-pc 8/94, S. 2749)*. Insofern ist der kleine Beispielsmakro von eben auch eine Veranschaulichung des mit "Beck connectivity" Gemeinten. *optcmd* ist ein Parameter, der darüber entscheidet, wie der Viewer gestartet wird. Gibt man 0 an, erscheint zunächst die Titelseite und dann das Zieldokument. Bei Verwendung von 1 wird sofort zum Zieldokument gesprungen (beim Schönfelder leider mit einem unschönen optischen Effekt in der linken oberen Ecke, weswegen hier 0 gewählt werden sollte).

```
REM SCHAUFRUF
REM Makro ruft eine Vorschrift im Schönfelder auf
REM Im Beispiel ist es § 1 BGB
REM Das schließende einfache Hochkomma muß der senkrechte
REM Strich sein (auf der Taste mit dem #-Zeichen).
Declare Function VwrCommand Lib "MVAPI2.DLL"(vwr As
Integer, MVB$, Command$, optcmd As Integer) As Integer
Sub MAIN
    command$ = "JI('D:\SCH.MVB','SCHF_BGB_1')"
    i =  VwrCommand(NULL, "SCH.MVB", command$, 0)
End Sub
```

Einbindung des Makros in ein Dokument

Um den Makro in einem Dokument zu verwenden, muß man ihn noch einbinden. Dies geschieht über "Einfügen – Feld". Als einzufügendes Objekt wählt man *Makro* und hinterlegt SCHAUFRUF. Das Ergebnis sieht, wenn "Ansicht-Feldfunktionen" aktiviert ist, wie folgt aus:

```
{MAKRO SCHAUFRUF § 1
BGB aufschlagen}
```

"§ 1 BGB aufschlagen" ist der Text, der auf Doppelklick hin den Makro aktiviert. Wenn man will, kann man diesen Text farblich gestalten, so daß er sich als Hypertext-Startstelle vom übrigen Text unterscheidet. Diese Anknüpfung ist auch in den Fußnoten möglich.

Selbstkritik

Die Lösung bietet, wie jeder Kundige sofort erkennt, einen entscheidenden Nachteil: Der Aufruf des Paragraphen ist im Programm enthalten. Man würde bei dieser Strategie also für jeden zitierten Paragraphen einen eigenen Makro benötigen. Eleganter wäre es, wenn der Makro die Referenz im Text auslesen und dann zum Schönfelder verzweigen würde. Eine gewisse Einheitlichkeit bei Normzitaten vorausgesetzt, ist das keine unlösbare Aufgabe. Auf der Grundlage dieser Verfeinerung wäre dann auch ein anspruchsvoller Aktualisierungsmakro folgender Art denkbar: An allen Stellen im Text, an denen Vorschriften wörtlich zitiert werden, wird beim Erscheinen eines neuen Schönfelder der aktuelle Text eingespielt.

Kasten 1

SCHDIREKTKOPIE: Kopieren mit einem Tastendruck

Jeder Schönfelder-Anwender möchte möglichst unaufwendig in der Lage sein, gefundene Texte in eigene Dokumente zu übernehmen. Diesen Wunsch trifft die Schönfelder-Werbung gut (*"Bitteschön, ein Tastendruck genügt"*), nur geht das eben standardmäßig leider nicht. Zwar läßt sich die im Handbuch (und in jur-pc 8/94, S. 2744) beschriebene Prozedur leicht abkürzen, wenn man mit Strg-Einfg den Text direkt in die Zwischenablage befördert. Von dort muß man ihn aber wieder abholen, so daß auf diese Weise das wünschenswerte Ziel "ein Tastendruck" nicht erreichbar ist. Der Makro SCHDIREKTKOPIE *(vgl. Kasten 2)* erreicht es.

Kasten 2

Der Makro setzt voraus, daß der Schönfelder gestartet ist und eine Suche durchgeführt wurde. Benutzt wird dann nur der Viewer-Befehl *CopyTopic()*, der den Inhalt des gerade angezeigten Dokuments in das Clipboard kopiert. Die weitere Zusatzannahme in dem Makro ist die, daß der Zieltext in Fenster 1 steht. Um nun "auf Tastendruck" zu kopieren, ist noch eine kleine Vorbereitung erforderlich. Bekanntlich kann man in Word die Funktionsleiste um weitere Funktionen anreichern. Im Beispiel ist das in der Form geschehen, daß hinter das freundlich lächelnde Mondgesicht in Abb. 1 (auf einem Farbbildschirm ist es gelb) der Makro SCHDIREKTKOPIE gelegt wurde. Wie das Kopieren mit einem Tastendruck abläuft, demonstrieren sodann die Abbildungen 1 und 2.

Abb. 1: Vor der Kopie ...

Der Anschaulichkeit halber wurden hier die beiden Fenster (Textverarbeitung und Schönfelder) nebeneinander plaziert. Die Prozedur funktioniert aber auch, wenn der Schönfelder im Hintergrund steht und die Textverarbeitung ein volles Fenster einnimmt.

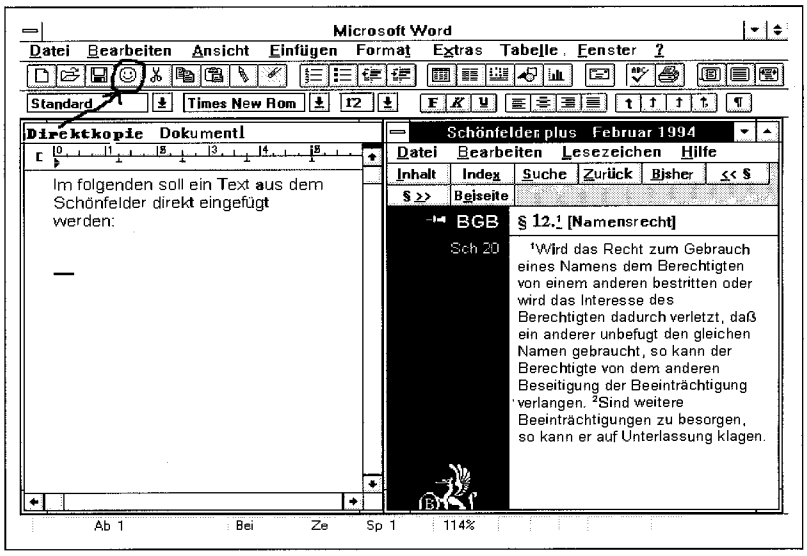
Abb. 2: ... und nach der Kopie.

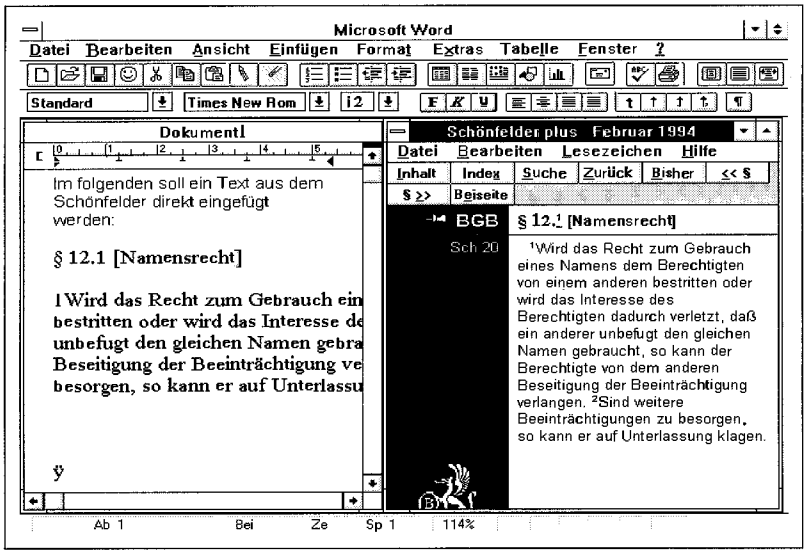
In diesem Beispiel ist ebenfalls für Verfeinerungen Raum. Wer den Copyright-Vermerk schon beim Kopieren entfernen will, kann das durch Anfügen einiger Zeilen in diesem Makro erledigen.

SCHVOLLTEXTSUCHE: Das Original-Such-Interface

Für den Schönfelder wurde ein eigenes Such-Interface geschrieben. Der Makro SCHVOLLTEXTSUCHE *(vgl. Kasten 3 auf der folgenden Seite)* ruft den Schönfelder mit dem Original-Such-Interface auf. Hier werden in der bereits beschriebenen Weise zwei Viewer-Befehle eingesetzt. Der erste "registriert" die Suchfunktion 'SearchDialog' für die Anwendung, der zweite benutzt dann diese Funktion. (Wie man "registriert" und wie man aufruft, ist in der Dokumentation

```
REM SCHDIREKTKOPIE
REM Makro für Direktkopie aus dem Schönfelder in Word
REM ================================================
Declare Function VwrCommand Lib "MVAPI2.DLL"(vwr As
Integer, MVB$,
Command$, optcmd As Integer) As Integer
Sub MAIN
    i = VwrCommand(1, "SCH.MVB", "CopyTopic()", 0)
    Fenster1
    BearbeitenEinfügen
End Sub
```





beim jeweiligen Befehl beschrieben, vgl. hier TR 5–44). Eine derartige "Registrierung" ist immer dann erforderlich, wenn eine Funktion aus einer DLL als Viewer-Befehl eingesetzt werden soll *(vgl. zu den Einzelheiten TR 5–74)*. Beim Registrieren müssen die Parametertypen mit angegeben werden, hier "USU" *(vgl. zur Erläuterung TR 5–75)*.

Das Original-Such-Interface *(vgl. Abb. 3)* hat zwei Vorzüge. *Erstens* ist die Länge der Suchanfrage nicht wie beim Schönfelder-Suchdialog auf 37 Zeichen begrenzt. Stattdessen hat man 255 Zeichen zur Verfügung. (Das Eingabefeld "scrollt" nach rechts.)

Zweitens – und das dürfte der gewichtigere Vorteil sein – kann man sich mit der Option "Suchen in ausgewählten Themengruppen" selbst die Gruppe der Gesetze zusammenstellen, in denen man suchen will. Das ist bei der Schönfelder-Suchfunktion nicht der Fall.

```
REM SCHVOLLTEXTSUCHE
REM Makro zum Aufruf der Viewer Original-Volltextsuche
REM ================================================
Declare Function VwrCommand Lib "mvapi2.dll"(vwr As
Integer,MVB$, Command$, optcmd As Integer) As Integer
Sub MAIN
    htitle = VwrCommand(NULL, "SCH.MVB", "RR('MVFTSUI2',
    'SearchDialog', 'USU')", 0)
    k = VwrCommand(htitle, "SCH.MVB", "FTSEARCH()", 0)
End Sub
```

```
REM SCHSUCHBUTTON
REM Makro zum Einfügen eines Buttons für die Viewer
REM Original-Volltextsuche
REM ================================================
Declare Function VwrCommand Lib "mvapi2.dll"(vwr As
Integer, MVB$, Command$, optcmd As Integer) As Integer
Sub MAIN
    htitle = VwrCommand(NULL, "SCH.MVB", "RR('MVFTSUI2',
    'SearchDialog', 'USU')", 0)
    j = VwrCommand(htitle, "SCH.MVB",
    "InsertButton('SUCHE', '&Original-Suche',
    'ftsearch()',5)", 0)
End Sub
```

```
REM SCHANMERKUNG
REM Makro fügt dem Viewer einen Button hinzu, der den
REM Original-Viewer-
REM Anmerkungsdialog aufruft
REM ================================================
Declare Function VwrCommand Lib "mvapi2.dll"(vwr As
Integer, MVB$, Command$, optcmd As Integer) As Integer
Sub MAIN
    i = VwrCommand(NULL, "SCH.MVB", "CB('ANM', 'A&NM',
    'ANNOTATE()')", 0)
End Sub
```

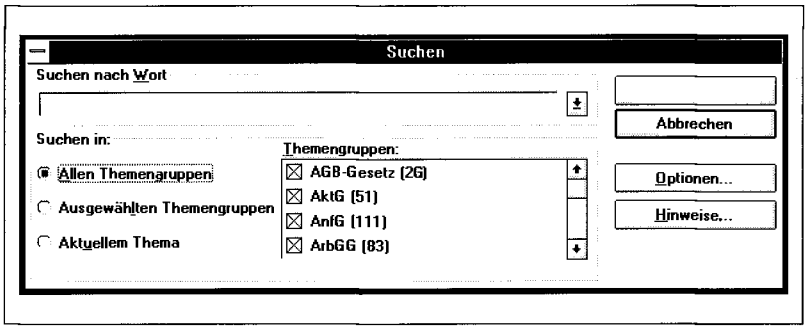


Abb. 3: Das Viewer Original-Such-Interface

SCHSUCHBUTTON: Ein Button für die Originalsuche

Möglicherweise überzeugen die beiden eben genannten Vorzüge der Originalsuche so, daß man diese Suchmöglichkeit im Viewer mit einem eigenen Button zur Verfügung haben möchte. SCHSUCHBUTTON *(vgl. Kasten 4)* zeigt, wie man derartige Zusatzbuttons im Viewer "von außen" anbringen kann.

Bewerkstelligt wird das mit dem Viewer-Befehl *InsertButton*, der vier Parameter übergibt: Einen singulären Namen für den Knopf, den Text (wobei das & vor dem hervorzuhebenden Buchstaben steht), den zu aktivierenden Befehl und die Position des Knopfes *(vgl. TF 5–49)*. Der hinter dem Button liegende Befehl ist der im vorigen Beispiel vorgestellte Volltextsuchbefehl.

Kasten 3

SCHANMERKUNG: Die Original-Viewer-Anmerkungsfunktion

Für den Schönfelder wurde die Anmerkungsfunktion des Viewer durch eine eigene Anmerkungsfunktion ersetzt. Leider ist die Anmerkungslänge dabei auf 255 Zeichen beschränkt worden, eine Längenbegrenzung, die die Original-Viewer-Anmerkungsfunktion nicht kennt (diese ist auf 1969 Zeichen beschränkt). Es könnte also der Wunsch aufkommen, neben der Schönfelder-Anmerkungsfunktion auch die Viewer-Anmerkungsfunktion zur Verfügung zu haben. SCHANMERKUNG *(vgl. Kasten 5)* erlaubt das unter Verwendung der eben erläuterten Technik der Button-Hinzufügung mit einem eigenen Knopf. Das Ergebnis mit geöffnetem Anmerkungsfenster ist aus Abb. 4 (auf der folgenden Seite) ersichtlich.

Abgesehen vom Wegfall der 255 Zeichen-Längenbegrenzung hat

Kasten 4

Kasten 5

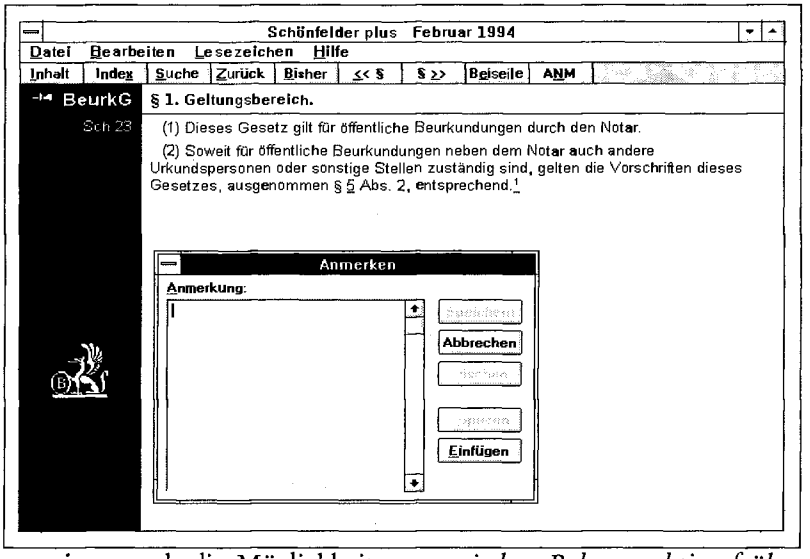


Abb. 4: Original-Viewer-Anmerkungsfenster

man jetzt auch die Möglichkeit, mit "Einfügen" Texte aus dem Clipboard in eigene Anmerkungen zu übertragen.

Die beiden Anmerkungsfunktionen "vertragen" sich miteinander. Man kann auf diese Weise zwischen zwei Anmerkungsinformationsarten differenzieren.

Der guten Nachricht folgt die schlechte: Es ist nicht sicher, daß man mit der Viewer-Originalfunktion geschriebene Anmerkungen auf ein Schönfelder-Update übertragen kann. Das war letzten Endes der Grund für die eigenständige Anmerkungsfunktion im Schönfelder. Da man aber die Schönfelder CD-ROM beim Update behalten darf (und – so das Handbuch S. 3 – *"bei der forensischen Rekonstruktion früherer Rechtszustände"* weiter benutzen soll), ist eine weitere Nutzung von "Original-Anmerkungen" auf jeden Fall sichergestellt. Hat man alte und neue Schönfelder CD-ROM gleichzeitig im Zugriff, könnte man sogar an einen Button in der neuen Version denken, der die Alt-Anmerkungen anzeigt.

An dieser Stelle gilt es, nach dem Eingangslob für die offene Konstruktion des Viewer eine partielle Geheimniskrämerei von Microsoft zu kritisieren, die den Umgang mit Anmerkungen (die in einer .ANN-Datei gespeichert werden) und Lesezeichen (die in einer .BMK-Datei gespeichert werden) unnötig erschwert. In CompuServe hat Microsoft dazu lapidar verlautbart: *"The formats of the .ANN and .BMK files used by the Microsoft Multimedia Viewer version 2.0 are not documented."* Dabei sollte es nicht bleiben.

Ausblick: VisualBasic und C/C++

Auch wenn die vorgestellten Makros nur eine Art Heimwerker-Baukasten darstellen, sind sie – abgesehen von dem mit ihnen erreichbaren Verständnis der "Viewer-Fernsteuerung" – doch bereits geeignet, einige nützliche Kleinigkeiten zu bewirken. Wer eine vollständige eigene Benutzeroberfläche programmieren will, die eigenständig ablauffähig ist, muß sich dann aber mit VisualBasic oder C/C++ vertraut machen. Dabei dürfte der Übergang von WordBasic zu VisualBasic wesentlich leichterfallen als der zu C/C++. Eine erste (leichte) Fortsetzungsübung wäre es, die hier vorgestellten Makros in VisualBasic zu realisieren. Aber davon und von anderen Programmiermöglichkeiten rund um den Viewer soll bei anderer Gelegenheit die Rede sein.

P.S. Ja, Sie haben recht. Es wäre noch gelungener, wenn der oben beschriebene Makro SCHDIREKTKOPIE die bei der Schönfelder-Kopierfunktion wegfallende Gesetzesabkürzung mitliefern würde. Der folgende Makro tut das.

```
REM SCHDIREKTKOPIE1: Direktkopie aus dem Schönfelder in Word mit Übernahme der Gesetzesbezeichnung
REM ================================================================================
Declare Function VwrCommand Lib "mvapi2.dll"(vwr As Integer, MVB$, Command$, optcmd As
Integer) As Integer
Declare Function TitleOpen Lib "MVTITLE2.DLL"(TITLENAME$) As Integer
Declare Function TitleGetInfo Lib "MVTitle2.DLL"(htitle As Integer, ilnfoMsg As Integer,
lparaml
As Long, lparam2$) As Long
Declare Function VwrGetInfo Lib "mvapi2.dll"(iVwr As Integer, ilnfoMsg As Integer, lParaml As
Long, lParam2 As Long) As Long
Declare Function VwrFromMVB Lib "MVAPI2.DLL"(mvb$) As Integer
Sub MAIN
htitle = TitleOpen("D:\SCH.MVB")
j = VwrFromMVB("D:\SCH.MVB")
i = VwrCommand(j, "SCH.MVB", "CopyTopic()", 0)
k = VwrGetInfo(j, 7, NULL, NULL)
topictitle = TitleGetInfo(htitle, 102, k, ueberschrift$)
Fensterl
EinfügenTextmarke .Name = "Zitatanfang"
BearbeitenEinfügen
BearbeitenGeheZu .Ziel = "Zitatanfang"
EndeZeile 1
BearbeitenLöschen
Einfügen ueberschrift$
EndeZeile
EinfügenAbsatz
End Sub
```