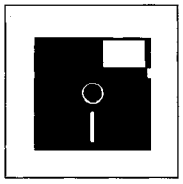


Toolbook – ooP für jedermann?

Stefan Ebeling

1. Programmierwerkzeuge der vierten Generation

Die neue Zeit ...

Bald wird der interessierte Anwender nicht mehr auf den Kauf fertiger Programme angewiesen sein oder sich mühsam umfangreiche Programmierkenntnisse aneignen müssen. Denn seit einigen Jahren existiert eine neue Generation von Programmen. Der Trend, der bei Excel und WinWord mit schon recht ausgefeilten Makrosprachen begann, setzt sich bei Programmen wie Microsofts jüngstem Produkt „Visual Basic"[1], „knowledge pro"[2] oder „Toolbook"[3] fort. Mit einfachen Mitteln werden Oberflächen erstellt. Die zugehörigen Funktionen können dann von jedem Interessierten nach einigen wenigen Tagen Lernarbeit ebenfalls selbst erstellt werden. Vorbei sind die Zeiten, in denen mühsam alle Einzelheiten eines Fensters in einer prozeduralen Programmiersprache beschrieben werden mußten. Heute werden nur noch „Objekte" erzeugt, denen dann „Eigenschaften" zugewiesen werden.
Selbst der professionelle Entwickler sieht sich mittlerweile immer mehr von derartigen Programmen unterstützt[4]. Die Euphorie geht soweit, daß teilweise das Ende konventioneller Programmierung für Windows-Applikationen vorausgesagt wird[5].
Soweit zur Theorie. Was hat es nun aber auf sich mit dieser besonderen Art der Programmierung, die auch als „objektorientiert" bezeichnet wird? Nach einem kurzen Überblick über die Grundprinzipien objektorientierter Programmierung (ooP), soll am Beispiel Toolbook (TBK) ein interessanter Vertreter dieser Programmgattung vorgestellt werden.

1.1 ooP – das Zauberwort

Objektorientiert ...

Objektorientiert – dieser Begriff beschäftigt seit einiger Zeit in vielerlei Zusammenhängen den interessierten aber auch den weniger interessierten Leser von Fachzeitschriften. Er wird allerdings in solcher Bedeutungsvielfalt gebraucht, daß es selbst Experten schwerfällt, eine exakte Definition zu geben. Kaum eine Software, die nicht demnächst objektorientiert wird oder es in Wahrheit längst ist, wie man uns eifrig versichert[6]. Die Verwirrung wird auch dadurch nicht geringer, daß sich Objektorientiertheit nicht bloß auf die Programmierung als solche beschränkt. Vielmehr erfaßt sie als Philosophie sämtliche Bereiche der Softwareentwicklung[7], will sich teilweise sogar als Synonym für eine Geisteshaltung verstanden wissen, die sich Unternehmen nicht nur für die Datenverarbeitung zu eigen machen sollten[8]. Der Begriff wird in so unterschiedlichen Zusammenhängen wie Programmiersprachen, Datenbanken, wissensbasierten Systemen, künstlicher Intelligenz und Softwaredesign gebraucht[9].

Stefan Ebeling ist Rechtsanwalt in Braunschweig.

---

1 Erhältlich bei Microsoft, München zum Einzelhandelspreis von ca. 600,– DM; ausführliche Beschreibung dazu z. B. von Michael Tischer in: MS-Anwenderjournal 4/91, S. 18 ff.; Anwendungsbeispiele in MS System-Journal Nr. 7/8 1991, S. 6 ff., und Nr. 11/12 1991, S. 35 ff.

2 Version 2.0 Hersteller: Knowledge Garden Inc; Einzelhandelspreis ca. 950,– DM; erhältlich bei ExperTeam, Dortmund.

3 Einzelhandelspreis ca. 700,– DM (deutsche Version ca. 1300,– DM); erhältlich bei Softline, Oberkirch/ADI, Karlsruhe/Access Computer, München.

4 Erwähnt seien schlagwortartig nur folgende Produkte: GFA-Basic für Windows/Turbo-Pascal für Windows/Quick-C/WinBasic/Object Script/Object Vision/Actor/Plus für PC unter Windows und für den Mac, das sogar datenkompatibel zu mit Hypercard erzeugten Mac-Stacks ist.

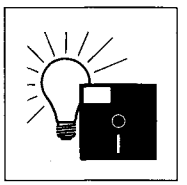
5 Schraudolph in: WIN, 11/91, S. 168 ff., „Countdown für das Ende des SDK".

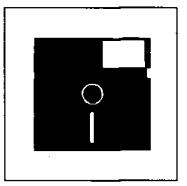
6 Peltzer in „Computerwoche", Nr. 33 aus 1991, S. 25 ff., „Noch gibt es Hürden auf der Prachtstraße zum SW-Entwurf", S. 25.

7 Als Beispiele seien genannt:
- Systementwurf: dazu Meik in CW 91, Heft 33, S. 36, „Objektorientierter Entwurf eines funktionalen Systems";
- Organisation: dazu Moser in CW 91, Heft 33, S. 28, „Weniger Reibungsverluste durch objektorientierte Organisation";
- Analyse;
- Design.

8 Moser in CW 91, Heft 33, S. 28, „Weniger Reibungsverluste durch objektorientierte Organisation".

9 Klas in „Computer Persönlich", Heft 25 aus 1991, S. 40, „Objektorientierte Datenbanken".





Dabei soll durch einen objektorientierten Ansatz der Computer endlich zu dem werden, was sich alle schon immer gewünscht haben: Mit wenigen Handgriffen wird er mühelos in jede beliebige Maschine verwandelt; der Computer programmiert sich selbst, wir sagen ihm nur noch wie[10].

Es scheint also auf jeden Fall der Mühe wert, sich zumindest einen Überblick über die in allernächster Zeit anstehende (oder bereits stattgefundene?) Revolution in der Softwareentwicklung zu verschaffen.

Software-Generationen 1 bis 4

Nach den Zeiten des binären Codierens (1. Generation) folgte die Phase der Assemblerprogrammierung, die dem Programmierer immerhin schon die Verwendung von Symbolen ermöglichte (2. Generation). Darauf folgte dann die softwaretechnische Gegenwart mit den bekannten Hochsprachen wie Basic, Pascal, C usw. (3. Generation). Seit einigen Jahren ist aber auch im Bereich der Programmierung die oben beschriebene Zukunft Gegenwart geworden. Die Zeit der „Fourth Generation Languages", kurz „4 GL" ist angebrochen. Diese Sprachen haben einen größeren Umfang, sind komplexer als die bekannten Hochsprachen und bieten mehr als nur die Grundelemente einer Programmfunktion. Eine Untermenge dieser 4-GL-Sprachen stellen die sogenannten objektorientierten Sprachen dar. Hierbei ist wiederum zwischen den eigentlichen (neuen) Sprachen der vierten Generation, die pur objektorientiert sind, wie beispielsweise „Smalltalk" und „Eiffel", und solchen, die um eine objektorientierte Komponente erweiterte (alte) Sprachen sind, wie objektorientiertes Pascal öder C++, zu unterscheiden.

1.2 Was ist ooP?

Windows als Beispiel

Als Grund-Beispiel mag die weitverbreitete Oberfläche Windows dienen. Das Hauptprogramm Windows stellt allen Unterprogrammen („Windows-Applikationen") vordefinierte Objekte wie Schaltflächen, Fenster, Dialogboxen usw. zur Verfügung. Soll aus einer Applikation heraus unter Windows beispielsweise ein Fenster geöffnet werden, muß sich der Programmierer keine Gedanken um die programmtechnische Realisierung machen, die nötigen Schritte zur Erzeugung und Anzeige eines Bildschirmfensters sind als Methoden des Objekts „window" im Programmcode von Windows bereits festgelegt, es genügt also, Windows mitzuteilen, es habe ein Objekt mit dem Namen „Fenster" darzustellen. Dieses Objekt wird sich dann mittels der vordefinierten Methode(n) selber darstellen. Objektorientiertes Programmieren ist nun nichts anderes als das Hantieren mit derartigen Objekten.

Objekte und Methoden

Zu jedem Objekt werden Methoden festgelegt, die definieren, wie das Objekt auf einen Anstoß von außen, ein Ereignis – die Übersendung von Botschaften – reagieren soll. Die Summe dieser vordefinierten Verhaltensweisen eines Objekts bildet seine Eigenschaften.

Da alle möglichen Ereignisse vom Programmierer mit im einzelnen vordefinierten Verhaltensweisen (Methoden) bedacht sind, besitzt ein Objekt einen ganzen Satz von derartigen festgelegten Verhaltensweisen. Beispielsweise soll ein Window nicht nur in der Höhe, sondern auch in der Breite verändert werden können, oder es soll gar minimiert oder maximiert werden, ein Fenster soll die Eigenschaft „aktiv" oder „inaktiv" annehmen oder eine bestimmte Größe oder Position auf dem Monitor haben.

Exempel: Meister und Lehrling

Vergleichbar ist dies mit der Kommunikation zwischen zwei Personen: Das Objekt „Meister" weist das Objekt „Lehrling" an, die Aufgabe „Reinigen der Werkbänke" zu erledigen. Dem Objekt „Lehrling" wird mitgeteilt (Botschaft), was zu tun ist (Ereignis). Das „Wie" bestimmt der Lehrling dagegen selbst unter Benutzung seiner Erfahrung (Methode/Eigenschaft des Objekts „Lehrling"). Diese Vorgehensweise hat den auf der Hand liegenden Vorteil, daß nicht jedesmal die komplette Anweisung zur Vorgehensweise bei der Reinigung von Werkbänken zwischen den Objekten ausgetauscht werden muß. Es genügt vielmehr, dies einmal festzulegen. Jeder der sich dann dieser Methode bedienen möchte, braucht dem Lehrling nur noch die Mitteilung zu machen, er möge die Werkbänke reinigen, ohne sich um das „Wie" Gedanken machen zu müssen.

1.2.1. Vererbung/Polymorphismus

Polymorphes „Erbrecht"

Der Clou bei der objektorientierten Programmierung ist – wie oben schon angedeutet – nun, daß der Programmierer die notwendigen Objekte nicht alle neu erzeugen, also jedesmal das Rad neu erfinden muß[11]. Entweder werden ihm die Objekte fertig zur Verfügung

[10] Zitiert nach Peltzer in „Computerwoche", Nr. 33 1991, S. 25 ff., „Noch gibt es Hürden auf der Prachtstraße zur Softwareentwicklung".

[11] Der alte Entwicklertraum von modular aufgebautem, wiederverwendbarem Code könnte damit wahr werden.

gestellt, oder er kann ein neues Objekt durch Kopieren eines vorhandenen Objekts erzeugen. Diesen Vorgang der Übertragung aller Eigenschaften eines Objekts auf ein anderes wird als Vererbung bezeichnet.
Soll das durch Vererbung entstandene, neue Objekt nun weitere odere teilweise andere Eigenschaften als der „Erblasser" haben, kann unter Beibehaltung der übrigen, weiterhin gewünschten Eigenschaften modifiziert werden. Die einzelnen Objekte sind also polymorph.[12]

1.2.2. Datenkapselung

Ein weiterer wesentlicher Unterschied zur herkömmlichen Programmierung läßt sich unter dem Stichwort „Datenkapselung" erfassen. Mit objektorientierten Programmiersprachen werden Programmcode und Daten zu festen Einheiten, eben den „Objekten" zusammengefaßt und nicht wie in herkömmlich erstellten Programmen, getrennt verwaltet.

1.2.3. Schnittstellenkonsistenz

Das dritte wesentliche Konzept objektorientierter Programmierung ist eigentlich eine notwendige Folge aus der Erstellung von Objekten durch Vererbung von Eigenschaften: Schnittstellenkonsistenz. Durch Definition allgemein akzeptierter Schnittstellen zwischen einzelnen Modulen und Objekten eines Programms besteht die Möglichkeit, daß andere Programme diese Objekte an der definierten Schnittstelle ansprechen und damit deren Funktionalität nützen können, ohne im einzelnen wissen zu müssen, wie ein Programm intern gestaltet ist.

„Nextstep"

Nach diesem Prinzip ist beispielsweise auch die (wieder einmal) revolutionäre neue Software von Steve Jobs für seine Next-Rechner, die Software „Nextstep", aufgebaut: Dieses aus über einhundert in sich geschlossenen Funktionsmodulen mit spezifischen Verhaltensmustern (Objekten) aufgebaute Programm erlaubt vielfältige Kombinationen dieser Module zu immer neuen Programmen[13].

3 Beispiele: 15 Minuten

Soviel zur Theorie. Aber keine Angst, so schwierig, wie es sich anhört, ist die Angelegenheit gar nicht. Schon an den folgenden Beispielen wird deutlich werden, das die objektorientierte Programmierung vieles leichter macht. So können innerhalb von wenigen Stunden ohne tiefgreifende Kenntnisse der bekanntermaßen aufwendigen Windows-Programmierung einfache Applikationen erstellt werden, für die man mit dem Software Developement Kit (SDK) sonst Tage oder Wochen benötigt hätte. Außerdem muß das teure SDK nicht gekauft werden. Die Erstellung der drei folgenden Beispiele dauerte insgesamt 15 Minuten. Sie führen zwar jeweils „nur" zur Darstellung eines einzelnen Fensters ohne jede weitergehende Funktionalität auf dem Bildschirm, doch wird der enorme Vorteil derartiger Werkzeuge deutlich, wenn man sich vergegenwärtigt, daß schon diese Aufgabe z. B. mit dem SDK von Microsoft beinahe 100 Programmzeilen (und jede Menge Know-how) erfordert (Abb. 1).

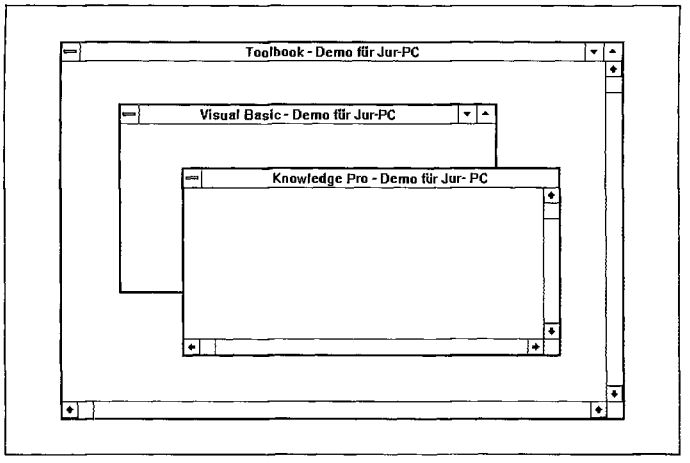


Abb. 1: Drei Fenster erzeugt mit knowledge pro für windows[14]*, Visual Basic*[15] *und Toolbook*[16]

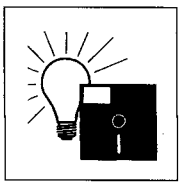
[12] Polymorphismus = Vielgestaltigkeit.

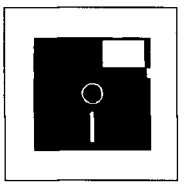
[13] Nach „Wink aus Kalifornien" Wirtschaftswoche, Heft 11 aus 1992, S. 81 (82).

[14] Es kann auch ein kompiliertes File erzeugt werden; zum Ablauf ist aber immer zumindest eine RunTime-Version von kp erforderlich.

[15] Visual Basic bietet ein voll funktionsfähiges Fenster als sogenanntes Formular an. Lediglich der Text der Titelleiste muß noch als Eigenschaft des Objekts „caption" definiert werden. Codeeingabe ist bis zu diesem Punkt nicht erforderlich. Es kann dann auch ein lauffähiges (*.exe-)File erzeugt werden. Zum Ablauf ist nur die (lizenzfrei erhältliche) Funktionssammlung „vbrun.dll" erforderlich.

[16] Beim Programmstart wird ein fertiges Fenster als erste Seite eines sogenannten books zur Verfügung gestellt. Um ein den anderen Beispielen vergleichbares Fenster zu erzeugen, muß lediglich noch der Text der Titelleiste festgelegt werden und die standardmäßig mit angezeigte Menüleiste mit dem Befehl „hide menubar" ausgeblendet werden.





2. Toolbook

Stellvertretend für die neue Generation objektorientierter Programme soll im folgenden das Leistungsspektrum des Programmpakets „Toolbook" von Asymetrix vorgestellt werden. Dabei erhebt dieser Beitrag keinen Anspruch auf vollständige Beschreibung aller Programmfeatures[17]; es sollen hier nur die Möglichkeiten, die dieses und ähnliche Werkzeuge bieten, vorgestellt werden.[18]

„Bücher ..."

Mit TBK werden unter Windows Programme für Windows – sogenannte books – erzeugt. Ein PC, DOS, Windows und die entsprechende Hardwareausstattung sind also Voraussetzung für den Einsatz von TBK. Eine Version für Rechner, die mit dem Betriebssystem OS/2 ausgerüstet sind, ist ebenfalls erhältlich. Laut Herstellerangaben soll die Portierung in die jeweils andere Welt gut funktionieren.

... und noch einiges dazu.

Mitgeliefert wird weiter eine ganze Reihe fertiger und weiterverwendbarer Objekte, Applikationen und Vorlagen (natürlich mit TBK erstellt). Die Palette reicht vom daybook, einem kompletten Kalender mit ToDo-Listenverwaltung, über eine Einführung in Animationstechniken bis zu Hinweisen für die Benutzung von Windows-Bibliotheken, einem Taschenrechner und einer eigenen Oberfläche zum Anlegen und Verwalten von books. Natürlich steht auch ein mit TBK realisiertes Lernprogramm zur Verfügung.

Eine Erzeugung kompilierter Files, die allein ablauffähig sind, ist nicht möglich. Jedes book ist nur zusammen mit der Vollversion von TBK oder auch zusammen mit der Runtime-Version lauffähig. Eine (fast) kostenlose Runtime-Version von TBK, die es ermöglicht, von anderen mit TBK erstellte Programme (books) zu benutzen, ist aber auch hierzulande in diversen Mailboxen und bei Shareware-Versendern erhältlich.

Spiritus rector: Paul Allen

Entwickelt wurde TBK von ehemaligen Microsoft-Mitarbeitern. Vater der Idee und Firmenchef der Herstellerfirma Asymetrix ist der Mitbegründer von Microsoft, Paul Allen. Diese Konstellation bietet wesentliche Vorteile im Markt.

Multimedia Resource Kit

Da die Verbreitung des Produkts offensichtlich von Microsoft gefördert wird[19], existiert insbesondere im amerikanischen Raum bereits eine große Anzahl von Toolbook-Applikationen. Die Verfestigung der Zusammenarbeit zeigt sich auch darin, daß seit kurzem ein Multimedia Resource Kit (MMRK)[20], das eine Sammlung von Programmerweiterungen zur Steuerung von Hard- und Software unter Microsofts „Multimedia Extension" zur Verfügung stellt, erhältlich ist. Als weitere interessante Neuerung ist seit einiger Zeit unter dem Namen „Convert-It" ein Programm zur Konvertierung von hypercard-stacks auf tbk-Format erhältlich.[21]

Mit dabei: Ein Editor

Neben einigen Werkzeugen zur Erzeugung und Bearbeitung von grafischen Elementen gehört ein mit allen Standardfunktionen (außer Gliederungsmodus) ausgestatteter Texteditor zum Lieferumfang. TBK bietet auch eine Schnittstelle zur Integration selbstgeschriebener DLLs. Damit ist sichergestellt, daß jede fehlende Funktion selbst geschrieben und eingebunden werden kann. Der Lieferumfang wird abgerundet durch ein sehr ansprechend und übersichtlich gestaltetes Handbuch und einen weiteren Band zur Einführung in die mitgelieferte Programmiersprache „Open Script". Für den, der kommerziell Applikationen erstellen und vertreiben möchte, ist[22] ein „author's resource kit" erhältlich. Der Käufer erwirbt neben dem Recht, selbsterstellte books lizenzfrei weiterzugeben, einen Script Remover, der zum Verschlüsseln des Source-Codes dient, und einen „searcher", der die Textsuche über alle Scripts aller Objekte in einem book ermöglicht. Außerdem erhält der zukünftige Autor eine umfangreiche Sammlung von DLLs[23].

---

17 Eine umfassende Darstellung findet sich z. B. bei Tischer, „QuickStart-Toolbook", Sybex-Verlag 1991.

18 Der Beschreibung wurde die englische Fassung des Programms in der Version 1.5 zugrunde gelegt. Dementsprechend werden zur Beschreibung der Funktionen ausschließlich die im Programm verwendeten englischen Begriffe benutzt. Dies gilt auch, soweit eindeutig und einfach übersetzbare Begriffe, wie z. B. „book", in Rede stehen.

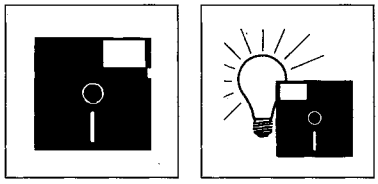
19 In den USA wird seit einiger Zeit mit jeder Windows-Version die RunTime-Version von TBK, mit deren Hilfe der Anwender vorgefertigte books benutzen kann, kostenlos mitgeliefert.

20 Zum Preis von ca. 1.000,– DM bei den oben (Fn. 1) genannten Distributoren.

21 „Convert.It" ist zum Preis von ca. 600,– DM ebenfalls bei den oben (Fn. 1) genannten Distributoren erhältlich.

22 Bei den in Fn. 1 genannten Distributoren.

23 dynamic link libraries – in C geschriebene Funktionen, die (dynamisch) in ein Programm eingebunden werden können. Vorteil: kleinere Programme, da Funktionen je nach Bedarf dynamisch gebunden oder gelöst werden.



Ein neues Buch schreiben ...

2.1 Books und Fields

Eine mit TBK erstellte Anwendung nennt sich book und besteht aus einer einzigen Datei mit der Extension „*.tbk". Jedes book besteht aus Objekten. Diese Objekte können Felder zur Aufnahme von Daten, Schlüsselwörtern, Grafiken oder Schaltflächen sein.

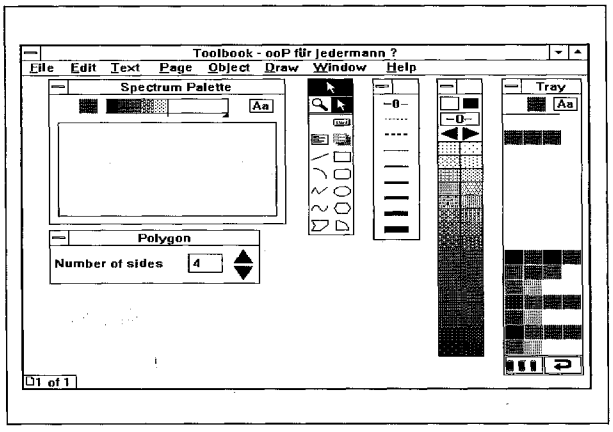


Abb. 2: Die Werkzeugpaletten von TBK

Die Erstellung eines neuen books wird durch Aufruf des Befehls „file/new" angestoßen. Auf dem Monitor erscheint dann ein neues, leeres book, auf page 1 aufgeschlagen. Mit Hilfe des Text- und/oder des Grafikeditors kann dieses book nun gefüllt werden. Mit Hilfe von Werkzeugpaletten werden durch Klicken und Ziehen mit der Maus Objekte plaziert und manipuliert (Abb. 2). Jedes book besteht dabei aus mindestens einem background (Hintergrund) und (begrenzt durch den verfügbaren Arbeitsspeicher) beliebig vielen pages. Objekte, die auf mehreren oder allen pages eines books erscheinen sollen, werden auf dem background plaziert.

Die pages kann man sich als durchsichtige Folien, die den Blick auf den background freigeben, vorstellen. Weiter enthält jedes book eine oder mehrere pages, die ihrerseits wieder andere Objekte wie fields, record-fields (Felder zur Aufnahme von Daten), buttons oder grafische Elemente aufnehmen.

Die Teile eines Buches

Die Anzahl der möglichen pages eines books wird nur durch die Speicherkapazität der Festplatte begrenzt.[24] Auf jeder page können beliebig viele und beliebig lange fields erzeugt werden. Paßt eine page dann nicht mehr komplett auf eine Monitorseite, ergänzt TBK die Darstellung automatisch um horizontale und/oder vertikale Bildlaufleisten.

Lesen kann man jeweils nur eine Seite.

Von allen pages eines books kann – hier liegt eine Parallele zum Printmedium – jeweils nur eine auf dem Bildschirm angezeigt werden. Abhängig vom verfügbaren Speicherplatz können aber auch mehrere books gleichzeitig geöffnet sein[25]. Über diesen Umweg können dann auch zwei oder mehr pages gleichzeitig am Bildschirm angezeigt werden. Außerdem bietet sich so die Möglichkeit, komplexe, häufig verwendete Funktionen in sogenannten Systembooks, auf die andere books zugreifen können, abzulegen. Man vermeidet so Redundanzen, die sich durch Wiederholung des gleichen Codes in vielen books ergäben.

2.2 ooP mit OpenScript

Für den, der TBK individuell anpassen will oder Vorstellungen hat, die sich mit den Standardvorgaben nicht verwirklichen lassen, stellt TBK neben der Schnittstelle zum Einbinden von DLLs eine ausgereifte Programmiersprache namens „OpenScript" zur Verfügung. Die Synatx der Sprache ist dabei nicht wie in herkömmlichen Sprachen kryptisch; sie ist vielmehr so angelegt, daß sich die Codezeilen wie natürlich-sprachliche englischsprachige Sätze lesen lassen. So bewirkt zum Beispiel der Befehl: „to handle ButtonDown goto page 2 in this Book", daß TBK dann, wenn ein bestimmter button niedergedrückt wird, die zweite page des aktuellen books anzeigt.

Rücksichtnahme auf „Nichtprogrammierer"

Die üblichen Kontrollstrukturen[26] und Befehle zur Manipulation von Variablen sind natürlich ebenfalls vorhanden. Um Nichtprogrammierern den Umgang mit der Sprache zu erleichtern, kennt openscript[27] keine unterschiedlichen Variablentypen. Auch die Möglichkeit, Daten mit anderen Programmen über eine DDE-Schnittstelle auszutauschen, wurde realisiert. Zum Kennenlernen der Sprache oder für den, der ohne sich einarbeiten zu müssen, einfache Vorgänge automatisieren möchte, steht auch ein Macrorecorder zur Verfügung. Wie sieht nun die ooP mit TBK aus?

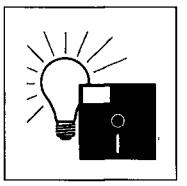
---

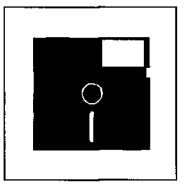
[24] Wie für alle Windows-Anwendungen genügt zwar auf dem Papier bereits ein AT mit 1 MB Hauptspeicher, Freude kommt aber frühestens auf einem 386er mit mindestens 4 MB Hauptspeicher und mindestens 30 MB großer Festplatte bzw. entsprechend freiem Platz auf einer größeren Platte auf.

[25] Dies wird dadurch ermöglicht, daß gleichzeitig mehrere Instanzen von TBK in den Arbeitsspeicher geladen werden können.

[26] Befehle zur Ausführung von Schleifen und Abfrage von if-Bedingungen.

[27] Wie dies beispielsweise in Basic der Fall ist.





2.2.1. properties (Eigenschaft/Attribut)

Es existieren viele vordefinierte Objekte wie Record-Felder, Felder, hotwords, buttons. Pages, backgrounds und books selber gelten ebenfalls als Objekte. Wer kein für seine Belange passendes Objekt findet, kann durch Modifikation eines vorhandenen ein seinen Bedürfnissen entsprechendes erzeugen.

„Eigentumstheorie"

Jedem Objekt werden ein Name sowie Attribute, in TBK heißen sie „property", zugeordnet. Eine property bestimmt Aussehen und Darstellungsweise eines Objekts. Auch der Inhalt eines Eingabefeldes stellt beispielsweise nichts anderes als ein property dieses Eingabefeldes dar. Die properties kann der author weitgehend selbst festlegen. Zu jedem Objekt existieren aber auch Voreinstellungen, da ein Objekt zumindest hinsichtlich des eigenen Aussehens auch dann Eigenschaften benötigt, wenn der author diese nicht explizit festlegt. Beispielsweise soll das Aussehen eines buttons oder die Position und Größe eines Objekts auf dem Bildschirm auch dann feststehen, wenn sich der author hierüber keine Gedanken gemacht hat. Die properties jedes Objekts können in openscript abgefragt und auch geändert werden.

2.2.2. scripts (Methoden)

Steuerung durch Ereignisse

Den Objekten können außerdem sogenannte scripts zugeordnet werden. Hier werden die Anweisungen festgelegt, mit deren Hilfe der author bestimmt, wie dieses Objekt auf ein bestimmtes Ereignis reagieren soll. Die scripts sind wiederum in sogenannte handler[28] unterteilt. Ein handler bildet einen Abschnitt im Code und enthält open-script-Anweisungen für ein bestimmtes Ereignis. Ereignisse werden in openscript „event" genannt. Zu einem Objekt können auch mehrere handler definiert werden, mit deren Hilfe das Objekt dann unterschiedlich auf verschiedene events reagieren kann.

2.2.3. messages und events

Events stellen das Kernstück im Ablauf eines mit TBK erzeugten Programms dar. Die Erstellung einer Applikation ist nichts anderes als die Festlegung der Reaktion der Objekte auf durch events erzeugte messages. Als event wird alles „was passiert", wie etwa das Bewegen der Maus, die Aktivierung einer Menüoption, das Betätigen eines buttons usw., angesehen. Man spricht daher auch von einem „event-driven-system". Die messages sind mit Prozeduraufrufen in herkömmlichen Programmiersprachen vergleichbar.

Die Objektorientiertheit wird hier besonders deutlich: Anstatt eine bestimmte Prozedur selber aufzurufen, wird einem Objekt, das die Prozedur beinhaltet, eine message zugesandt, auf die dieses dann mit der Ausführung des entsprechenden handlers im script reagiert.

2.3 Hypertext-Autoren- und Lesesystem

Für Juristen mit Textinteresse

Den für den textorientiert arbeitenden Juristen interessantesten Ausschnitt aus der breiten Palette von TBK stellen sicherlich die Möglichkeiten zur Erstellung und Benutzung von Hypertext dar. In bislang einzigartiger Weise wurden Autoren- und Lesesystem für Hypertextanwendungen verknüpft. Mit TBK erhält man ein Management- und Herstellungssystem für Hypertext, das auch ein Volltextretrieval anbietet.

2.3.1. reader- und author-level

TBK kann in zwei verschiedenen Modi betrieben werden: dem reader- oder dem author-Modus. Die Umschaltung von einem in den anderen Modus erfolgt durch bloßes Betätigen der <F3>-Taste. Nur im author-Modus sind die Objekte modifizierbar.

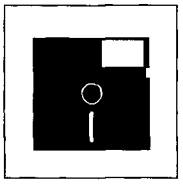
Autor = Programmierer

Der author ist in TBK vergleichbar dem Programmierer. Er kann auf alle Objekte ändernd zugreifen und Inhalte eingeben. Der reader kann dagegen weder die Grundstrukturen verändern, noch den Inhalt der vorgegebenen pages ändern. Er ist also auf die Tätigkeit eines Lesers beschränkt. Ihm kann lediglich die Möglichkeit eingeräumt werden, Text in Textfelder einzutragen. Der Wechsel vom reader- in den author-Modus kann durch ein Paßwort gesperrt werden.

2.3.2. Datenimport

Wie bereits oben dargestellt, stellt TBK Textfelder zur Verfügung. Eine Variante dieser Textfelder stellen die sogenannten record-fields dar. Mit Hilfe dieser Felder werden in TBK Datenbank-Funktionen realisiert. Die record-fields selbst liegen immer auf einem background, während die Feldinhalte (die Daten selbst) auf der jeweiligen page im Vordergrund abgespeichert sind. So wird ein aus mehreren Feldern bestehender background erstellt, der

[28] In der ooP werden sie auch „Methode" genannt.



in seiner Gesamtheit einer Tabelle entspricht, wobei jede beschriebene page einen eigenen Tabelleneintrag enthält[29]. Auf jeder page kann also ein einzelner Datensatz eingetragen werden.

Texteingabe auch über die Zwischenablage

Zum Import von Text stehen verschiedene Möglichkeiten zur Verfügung: Die Texteingabe kann einmal mittels des eingebauten Editors erfolgen. Der Editor verfügt über sämtliche Standardfunktionen wie Ausschneiden und Einfügen, Blöcke markieren usw.
Weiterhin kann die Texteingabe aber auch windowstypisch über die Zwischenablage erfolgen. Auf diesem Weg besteht also auch die Möglichkeit, Text aus unter Windows aufgerufenen DOS-Anwendungen in TBK einzubringen.

Auch möglich: Formatierter Import

Schließlich besteht auch die Möglichkeit, Datenbankdateien im ASCII-Format[30] in record-fields einzulesen. Dabei ist es nicht erforderlich, vor dem Import durch Anlage entsprechender Felder eine Datenbankstruktur vorzugeben.
Diese Arbeit übernimmt TBK selber. Durch ein frei definierbares Trennzeichen unterteilte Texte werden von TBK beim Import in der Weise interpretiert, daß für jedes Feld auf dem background ein neues record-field und für jeden Datensatz eine neue page angelegt wird. Praktisch ist dabei, daß bestimmte Bereiche eines Textes in Anführungszeichen gesetzt werden können, die dann beim Import nicht berücksichtigt werden. Zum Import von Datendateien, die mit dem häufig verwendeten Datenbankprogramm DBase erstellt wurden, wird ein spezielles book, das diese Aufgabe einfach und komfortabel erledigt, mitgeliefert.

2.3.3. Datenexport

Die eben beschriebenen Funktionen ermöglichen natürlich in umgekehrter Richtung auch den Export von Text. Eine oder mehrere pages eines books können ausgedruckt werden, wobei die gesamte Palette der von Windows unterstützten Drucker zur Verfügung steht. Außerdem kann auch als sogenannter report eine Liste aller Inhalte eines bestimmten Felds ausgegeben werden, das kann wahlweise jeweils auch in das Windows-Clipboard erfolgen.

2.3.4. Hypertexterstellung und -verwaltung

Hypertext = „Feste Verdrahtung zwischen Texten"

Darüber, was Hypertext im allgemeinen oder im juristischen Arbeitsfeld sein kann oder ist, soll an dieser Stelle nicht reflektiert werden.[31] Sehr plastisch und für die meisten Aspekte völlig ausreichend ist die Beschreibung als 'feste Verdrahtung zwischen Texten': Ein „vgl. dazu oben S. 345" läßt sich als Hypertext-Link verstehen[32]. Entscheidend ist immer, daß mindestens zwei Texte existieren, zwischen denen inhaltlich, nicht aber räumlich ein direkter Zusammenhang besteht. Die Herstellung einer Verbindung ist auch im Printmedium möglich und ist gerade bei juristischen Texten traditionell. Wird die Verbindung mittels Hypertext hergestellt, kann diese jedoch wesentlich schneller als über Fußnoten, bei denen u. U. mehrmaliges Blättern nötig ist, verfolgt werden.

„links" und „hotwords"

Derartige Verbindungen zwischen zwei Textelementen werden in TBK durch „links" realisiert. Befindet man sich im author-Modus, wird eine Verbindung zwischen einem Wort oder einer Gruppe von Worten dadurch hergestellt, daß diese als „hotword" gekennzeichnet werden und dann diesem hotword ein Zielpunkt zugewiesen wird. Dazu wird die page, auf der sich der in Bezug zu nehmende Text befindet, aufgeschlagen. Am Zielpunkt angekommen, wird nur noch festgelegt, ob der Link uni- oder bidirektional sein soll. Bidirektional meint, daß TBK zusätzlich zum link auf der in Bezug genommen page einen button anlegt, durch dessen Betätigung man an den Ausgangspunkt des links zurückkehrt.
Der als hotword definierte Text unterscheidet sich vom übrigen durch eine andere (blaue) Farbe oder dadurch, daß der Mauszeiger beim Überrollen des Begriffs die Form ändert.
Dieser Philosophie folgend, kann neben der herkömmlichen linearen Vorgehensweise von jeder page eines books zu einer beliebigen anderen page im selben book oder auch in anderen books verzweigt werden. Nach der Erzeugung eines links kann durch Betätigen der <F3>-Taste die Umschaltung vom author- in den reader-Modus erfolgen, d. h., es kann sofort die Wirkung in der Zielumgebung getestet werden.

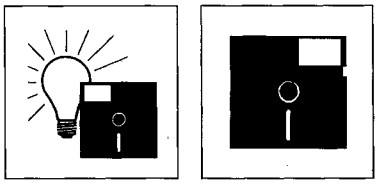
---

29 Shaw in Microsoft System Journal, 3/4 1991, S. 38 (40).

30 Aus beinahe jedem DB-Programm exportierbares Format.

31 Für die zahlreichen Definitions- und Beschreibungsversuche sei exemplarisch verwiesen auf Krüger, „Hypertext für Juristen – Grundlagen und Probleme", jur-pc 3/92, S. 1497 ff (ausführlich); Seidensticker in: „Information Management mit Hypermediakonzepten", S. 50; Hamburg '90; Ebeling, „ArBIS – Experten im Hyperraum", jur-pc 9/91, S. 1237 (1240).

32 M. Herberger in jur-pc CD-ROM Digest 92, S. 62 f.



Möglich auch: Animierte Grafiken

2.4 Oberflächen- und Grafikmanipulation

Nur kurz sei noch auf die ebenfalls umfangreich vorhandenen Werkzeuge zur grafischen Manipulation hingewiesen. Über eine einfach zu bedienende Importfunktion können Grafiken diverser Formate auf einzelnen pages plaziert werden. Animierte Grafik (bewegter Ball o. ä) ist bei entsprechend leistungsfähiger Hardware ebenfalls realisierbar. Die Oberfläche eines books kann völlig frei definiert werden. Menüleisten, Menüobjekte und Menüpunkte können beliebig erzeugt und verändert werden.

2.5 Kritik

Größenprobleme

Wo viel Licht ist, ist bekanntlich auch Schatten. Auch bei TBK ist dies, was hier nicht verschwiegen werden soll, nicht anders. Ein potentielles Problem ergibt sich aus der oben erwähnten gemeinsamen Abspeicherung von Daten und Programmcode. Durch dieses Vorgehen entstehen schnell größere books, die ihrerseits erhöhte Anforderungen an den Arbeitsspeicher stellen. Die für die Darstellung der oben gennanten Beispiele erforderlichen Dateien belegen folgenden Arbeitsspeicher: knowledge pro = eine Datei 150 KB; Visual Basic = zwei Dateien mit zusammen 390 KB; TBK = eine Datei mit 12.646 KB. Das Laden eines books dauert ob dieser Technik im Vergleich lange. Ist ein book einmal geladen, verlaufen Recherchen in diesem book aber recht schnell.

Grafik: Kleinster gemeinsamer Nenner

Eine dynamische Anpassung an verschiedene Grafikkarten ist nicht möglich, da Felder und Grafiken nicht automatisch skalierbar sind. Will man Anwendungen einem breiten Kreis zugänglich machen, muß daher der kleinste gemeinsame Nenner[33] gewählt werden. Dies erklärt auch, warum die Standard-Größenvorgabe für books auf VGA-Monitoren nicht den ganzen Bildschirm ausfüllt, sondern mit einem recht breiten umlaufenden Rand dargestellt wird.

Weniger schön ist auch, daß beim Import Zeichenformatierungen wie fett, kursiv und unterstrichen verlorengehen.[34] Sollen die Formatierungen erhalten bleiben, ist der Autor auf den Weg über das Windows-Clipboard angewiesen.

Warum nicht Klassen von Objekten?

In einigen Bereichen ist auch die Umsetzung objektorientierter Vorgehensweisen noch nicht besonders glücklich gelöst. So findet zwar eine Datenkapselung statt, und Schnittstellenkonsistenz ist, wenn auch nur rudimentär, über DDE-Fähigkeit vorhanden. Schwer wiegt jedoch eine andere konzeptionelle Ungereimtheit hinsichdich der praktikablen Vererbung von Objekteigenschaften: So ist es nicht möglich, Klassen von Objekten zu bilden, um so, quasi automatisch, alle Eigenschaften auf neue Objekte zu vererben, indem ein Objekt als einer Klasse zugehörig definiert wird. Der Vererbung ähnliche Effekte können nur über die Plazierung von Objekten auf backgrounds, die dann auf allen Vordergrundseiten sichtbar sind, erzielt werden.

Schnell wird der Code unübersiehtlich.

Die Programmierung größerer Systeme wird, da der Code nicht zusammenhängend vorliegt, sondern sich auf die jeweiligen Objekte verteilt, schnell unübersichtlich. So ist es nicht möglich, den Code für alle Objekte eines books auf einmal zu bearbeiten. Ist beispielsweise ein Objekt durch Verdoppelung eines anderen erzeugt worden und sollen nun bestimmte Eigenschaften sowohl des Ursprungs- als auch des neuen Objekts geändert werden, müssen beide Objekte erst geöffnet und dann einzeln geändert werden. Daß das Problem auch bei Asymetrix gesehen wird, ist daran zu erkennen, daß daß author resource kit eine Suchfunktion, die das Auffinden bestimmter strings im Code über alle Objekte eines books[35] ermöglicht, enthält.

2.6 Anwendungsbereich

Welchen konkreten Nutzen kann TBK nun dem interessierten Juristen bieten?

Nutzen für Juristen: Umgang mit unstrukturierten Informationen

Eines der bislang ungelösten Probleme ist die Handhabung und Verwaltung von unsystematisch, nicht nach bestimmten Regeln klassifizierbaren Informationen. Wohl jeder dürfte die mehr oder weniger unzureichenden und aufwendigen Versuche, Ordnung in eigene, in unstrukturierter und unsystematischer Form vorliegende Informationen zu bringen, kennen.[36] Ein Ansatz mit TBK ist denkbar. So können auf einer großen Anzahl pages diverse Informationen wie Entscheidungen, Adressen, Notizen, ToDo-Listen usw. festgehalten

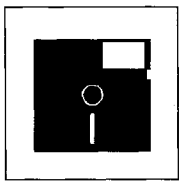
---

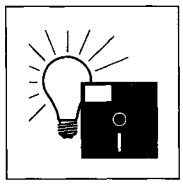
33 Im Moment wohl noch EGA-Grafik.

34 Dies ist allerdings bei Konkurrenzprodukten nicht anders.

35 Und sogar über mehrere books hinweg.

36 Eben dieses Probleme versuchen diverse PIM-Systeme, wie z. B. Agenda, ERNA oder Packrat, zu lösen.





Volltextsuche

werden, wobei inhaltlich zusammenhängende Informationen durch Hypertext verbunden werden. Auch bei der Erstellung einer Materialsammlung[37] kann TBK gute Dienste leisten. Um nicht die Übersicht zu verlieren, ist es möglich, über alle Felder eines books eine Volltextsuche zu starten. Eine Indexierung der abgelegten Daten erfolgt zwar nicht. Dies muß, solange nicht eine Suche über mehrere books stattfindet, jedoch kein Nachteil sein, da beim Start eines books jeweils das ganze book geladen wird. Die Suche nach in diesem book vorhandenen Daten geht dann ausgesprochen schnell vor sich. Der Autor hat Erfahrung mit einem book in der Größe von ca. 3 MB (entsprach 600 pages) gesammelt. Auf einem 20 MHz 386er mit 10 MB Hauptspeicher traten keine nennenswerten Verzögerungen auf. Dabei stellt TBK ein Volltextretrieval, das man entweder auf eine page, auf alle pages (exclusive backgrounds) oder nur auf bestimmte Datensatzfelder begrenzen kann, zur Verfügung.

Ein Wermutstropfen ...

Ein Wermutstropfen sei jedoch nicht verschwiegen. Zur umfassenden Verwaltung von Hypertext wäre es unbedingt erforderlich, daß das Werkzeug beim Löschen eines Textes, auf den in einem anderen Bezug genommen wird, zumindest einen Warnhinweis ausgibt. Eine derartige Warnung erfolgt jedoch leider nicht; es kann also u. U. zu Inkonsistenzen kommen. Eine Warnung erfolgt nur, wenn der Text, der einen link auf einen anderen Text enthält, gelöscht werden soll. Solange dieses Problem nicht gelöst ist, sind Anwendungen mit Schwerpunkt Hypertext sicherlich Grenzen gezogen.

Unbestreitbar innovativ

Unbestreitbar ist TBK Vertreter eines innovativen Produktkonzepts. Problemlos lassen sich mit diesem Werkzeug sehr ansprechende Oberflächen gestalten, wobei in der jetzigen Fassung der Schwerpunkt sicherlich auf grafisch anspruchsvoller Darstellung und Multimedia-Anwendungen liegt. Zur reinen Texterstellung und Verwaltung bieten sich dagegen andere Programme, wie z. B. Guide für Windows[38] oder das hinlänglich bekannte 1st Card, eher an. TBK ist dort in seinem Element, wo viel Wert auf ansprechende Darstellung gelegt wird. Denkbare Anwendungsbereiche sind die Verwaltung von Plänen aller Art, die Verwendung als Trainingsinstrument für interaktive Lernsysteme, Lernpakete und Courseware. Gut vorstellbar sind ebenfalls die Erstellung von Betriebsanleitungen und Online-Referenzen mit TBK, Besucherinformationssystem, Preisliste mit dahinterliegenden Artikelbildern, Kataloge auf Diskette, die Verwendung als Front-End für Datenbanken usw.

2.7 Fazit

Große Wirkung mit einfachen Mitteln

Der große Vorteil von TBK besteht darin, daß es mit relativ einfachen Mitteln möglich ist, eine den eigenen Bedürfnissen exakt entsprechende Umgebung zu erzeugen. Dabei kann auch auf diverse bereits vorgefertigte Anwendungen wie „daybook" u. ä. aufgebaut werden. Hier stellt sich dann allerdings im Einzelfall die Frage, bis zu welchem Punkt der Anwender sinnvoll selber entwickeln kann.

Wie stets bei Juristen: Es kommt darauf an.

Die im Titel aufgeworfene Frage läßt sich daher – wie üblich – mit einem klaren: „Es kommt darauf an!" beantworten. Sicher dürfte es auch dem interessierten „Normaljuristen" nicht möglich sein, nach zwei Stunden Einarbeitungszeit neue umfangreiche Programme zu erstellen. Trotzdem bietet TBK aber einige reizvolle Möglichkeiten, die sich bereits nach kurzer Einarbeitungszeit erschließen. Denn die richtige Lösung sind oft nicht Programme mit Dutzenden von Funktionen (von denen nur die 10 wichtigsten überhaupt bekannt sind und genutzt werden), sondern ein exakt auf die persönliche Situation zugeschnittener kleiner nützlicher Helfer.

Der Preis: 700,– bis 1.300,– DM

Dem, der sich z. B. eine maßgeschneiderte eigene Arbeitsumgebung zur Integration diverser Applikationen[39] entwickeln möchte[40], bietet TBK viele kreative Möglichkeiten. Ob diese Möglichkeiten allerdings eine Anschaffung zum Preis von 700,– bzw. 1.300,– DM für die deutsche Version zu rechtfertigen vermögen, muß der Entscheidung des einzelnen überlassen bleiben.

Toolbook – ooP für jeden, der Freude an anspruchsvoller Grafik hat und dem der Spieltrieb noch nicht gänzlich abhanden gekommen ist; in diesem Rahmen bietet TBK bereits dem „fortgeschrittenen Nichtprogrammierer"[41] diverse erstaunliche Möglichkeiten.

---

37 Für diesen Aufsatz.

38 Beschreibung z. B. Fieger in „Computer Persönlich", Heft 25/90, S. 25 f.

39 Wie beispielsweise zur Integration der diversen nützlichen Berechnungswerkzeuge von Herrn Nilgens aus Köln, die u. a. in der jur-pc Mailbox erhältlich sind.

40 Der Aufruf anderer Programme aus TBK heraus ist möglich.

41 Begriff von Shaw in MSJ, 3/4 '91, S. 38.