

Vor- und Nachteile von ToolBook – Ein Erfahrungsbericht¹

Clemens Tobias Steins

Am Anfang stand der Wunsch, eine Windowsapplikation zu schreiben, ohne die notwendigen Kenntnisse in der Windowsprogrammierung erworben zu haben. Als Ausweg bot sich ToolBook an, welches vom Herausgeber treffend als „Windows Construction Set“ bezeichnet wird und einfache Hilfsmittel bietet, Applikationen für die graphische Benutzeroberfläche Windows zu erstellen. Damit soll es sich von anderen Werkzeugen (wie beispielsweise den Programmiersprachen „Turbo Pascal für Windows“ oder „C++ für Windows“ abheben) und die Barriere niederreißen, die bisher von der Erstellung von Windowsapplikationen abhielt².

Arbeitsweise

Die Vorgehensweise bei der Erstellung solcher Anwendungen unter Windows entspricht im ersten Schritt der in einem Grafikprogramm. Wie in diesen werden die gewünschten Schalter, Textfelder oder Grafiken, die im eigenen Programm erscheinen sollen, mittels Maus in jeder beliebigen Größe und an jedem beliebigen Ort plaziert, wobei sich das Windows-übliche Design von Schaltern oder Textfenstern aus Menüs oder einer „Toolbar“ auswählen läßt. Diese dann in kurzer Zeit erstellte Oberfläche mit Leben zu erfüllen, erfordert dann allerdings doch die Kenntnis der jedoch betont einfach gehaltenen Programmiersprache OpenScript, die sich im Aufbau nur

unwesentlich von Basic unterscheidet und zumindest für Anwender, die Programmierkenntnisse haben, leicht zu erlernen ist (Abb. 1).

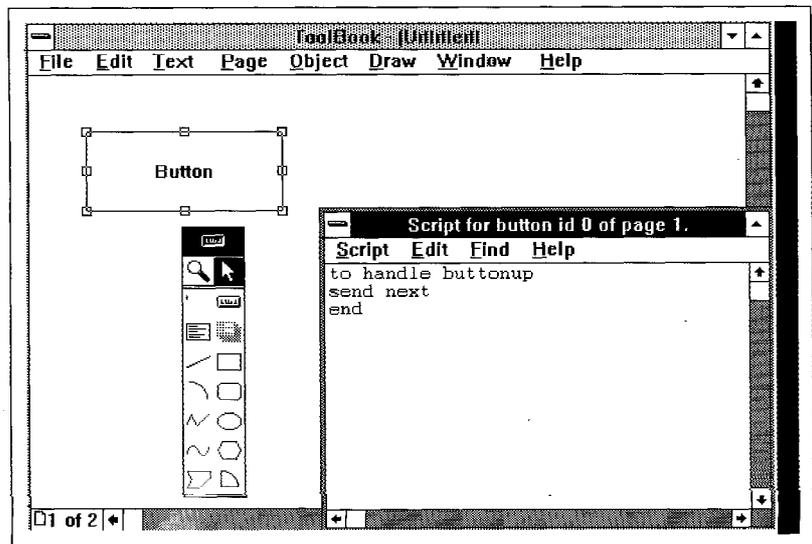
Literaturproblem

Gemessen an diesen Voraussetzungen sind die Hürden, die bestehen, um die Anwendung von ToolBook zu erlernen, größer als erwartet. Der Grund dafür liegt für den, der auf deutsche Erläuterungen angewiesen ist, außerhalb der Software im Angebot an Literatur, die über die Handbücher hinaus in die ToolBook-Handhabung einführen. So sind im Augenblick nur zwei deutschsprachige ToolBook-Handbücher³ auf dem Markt, beide der Gattung „Schnelleinstieg“ zuzuordnen, die ihre Seiten fast ausschließlich auf Installation und das Erstellen der Oberfläche verwenden, ohne auch der Pro-

grammiersprache OpenScript einen ausreichenden Platz einzuräumen. Deutschsprachige Abhilfe schafft da nur das ToolBook-eigene Lernprogramm, das nicht nur didaktisch gut in die ToolBook Programmierung einführt, sondern darüber hinaus, da es selbst in ToolBook geschrieben ist, auch als Quelle für vorgefertigte Skripten gleichsam als Anschauungsobjekt dient. Besteht ein Problem, welches im Lernprogramm augenscheinlich gelöst ist, so wechselt man auch in diesem in den Autorenmodus und kann sich die Problemlösung ansehen. Doch mangelt es dem Lernprogramm an Vollständigkeit, da auch hier Fähigkeiten der höheren ToolBookprogrammierung nicht aufgenommen sind. Auch ist anzumerken, daß nur eine englische Schulversion verfügbar ist, wo beides, Handbuch und Lernprogramm selbstverständlich in englisch vorliegen. Will man ein umfassendes Werk zum Umgang mit ToolBook,

Abb.1:
Schalter mit Script,
das zur nächsten
Seite umblättert.

Clemens Tobias Steins ist Mitarbeiter am Lehrstuhl für Rechtsinformatik der Universität des Saarlandes und Autor des mit ToolBook geschriebenen Lernprogramms eTutor.



¹ Dieser Artikel soll nicht mehr auf die Struktur des Programmes ToolBook eingehen, dazu sei auf jur-pc 11/1992 S. 1846–1854 verwiesen. Hier sollen vielmehr die persönlichen Erfahrungen des Autors bei der Erstellung eines Lernprogramms mit Hilfe von ToolBook wiedergegeben werden.

² ToolBook Companion, S. 3 (Fn.4).

³ Quickstart (Sybex) und Schnelleinstieg (Data Becker).

muß auf die amerikanische Anleitung der Cobb-Group⁴ verwiesen werden, die vollständig und in relativ einfachem Englisch die ToolBookhandhabung erläutert.

Makrorecorder

ToolBook selbst bietet als Alternative zum Erlernen der OpenScript-Programmierung den Makrorecorder⁵. Dieser ist jedoch, wie alle Recorder, nur zu den niedersten Programmierleistungen in der Lage, da bedingte Sprungbefehle und alle

anderen, nicht mit der Maus ausführbaren Programmieroptionen nicht aufzuzeichnen sind. Auch wird der durch den Recorder generierte Code meist überflüssig lang⁶, da vor allem bei Problemen, die sinnvollerweise durch Zählschleifen zu lösen sind, der Makrorecorder den Code entsprechend oft in das Skript schreiben würde.

Tücken des „Objekts“

Hat man besagte Kenntnisse in OpenScript erlangt, wird deut-

lich, daß die ausgeklügelte Struktur von ToolBook auch ihre Tücken hat. Das beginnt bei ärgerlichen Beschränkungen wie der, daß in „Recordfields“ nur 32.000 Zeichen aufgenommen werden können, geht über Schwierigkeiten wie das Importieren von ASCII-Texten, die in Textfenster nur über das Clipboard eingebunden werden können⁷ und dabei so eigenwillig umbrochen werden, daß eine Nachbearbeitung von Hand notwendig wird. Und es findet seinen Gipfel in der Tatsache, daß Handler, also kleine, in OpenScript erstellte Programme, die zum Beispiel auf Aktionen des Benutzers mit der Maus reagieren, in jeder Instanz des Programmes und in jedem Objekt plaziert sein können, und keine Übersicht geboten wird, wo sich Handler befinden, so daß der Benutzer mehr als einmal auf die Suche nach einem Programmteil geht, das die störende Fehlfunktion erzeugt.

Besonders tückisch ist dies, hat man mehrere Objekte zu einer Gruppe zusammengefaßt. Um zu überprüfen, ob den einzelnen Objekten Handler zugeordnet sind und wie diese arbeiten, muß man eine solche Gruppe wieder auflösen, wobei der Handler, der der Gruppe zugeordnet worden ist, zwischengespeichert werden muß, da er bei Auflösen der Gruppe verloren geht (vgl. Abb. 2).

Wunsch: „Handler-Browser“

Dem letzteren, und wohl wichtigsten, Kritikpunkt könnte eine Art „Handler-Browser“ abhelfen, der übersichtlich die benutzten Skripten anzeigt. Eine letzte Grenze bleibt, an die zumindest die Erstellung von

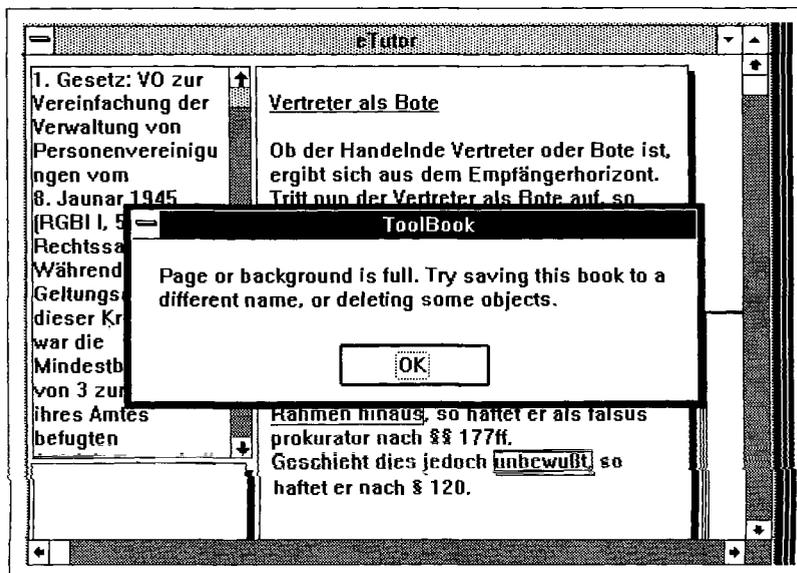
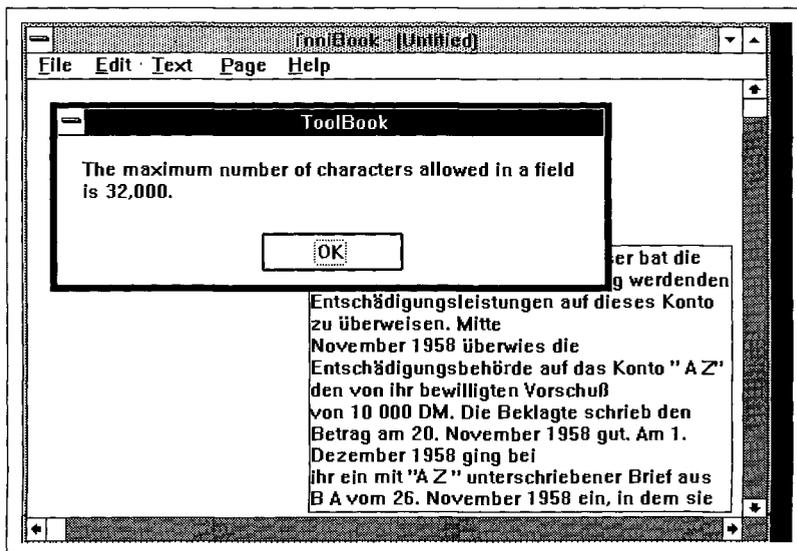
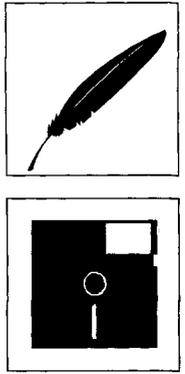


Abb.2: Über das Clipboard eingebundener Text und Fehlermeldung

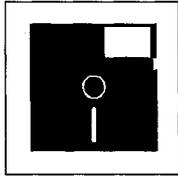
Abb.3: Fehlermeldung bei überfüllter Seite

⁴ The Cobb Group, Joseph R. Pierce, ToolBook Companion, USA 1990.

⁵ ToolBook Companion, S. 277.

⁶ Vgl. Abb. 1.

⁷ Die Option „Importieren...“ im Menü „Datei“ taugt nur für die Einbindung von anderen „ToolBooks“ oder von Datenbanken.



eTutor⁸ gegen Ende ständig stieß. Jede Seite eines ToolBook-Programms kann nur eine endliche Zahl von Objekten und/oder Textinhalt beinhalten, was besonders im juristischen Bereich, man denke an längere Bundesverfassungsgerichtsentscheidungen, störend werden kann (Abb 3, vorhergehende Seite).

Die Versuchung des Unstrukturierten

Über diese einzelnen Kritikpunkte hinaus läßt sich grundsätzlich eine Schwäche bei ToolBook-Programmen entdecken, welche eigentlich schon durch Wirth's Pascal beschrieben und gebannt⁹ schien, die unstrukturierte Programmierung. Wie früher bei Basic verführt ToolBook zum unvorbereiteten „Drauflos-Programmieren“. Disziplinierungsmaßnahmen, wie Variablendeklarationen, beschränken sich in ToolBook auf das nötigste. Auch führt die Objektstruktur hier weniger zur strukturierten Programmierung¹⁰, sondern läßt einen die einzelnen Handler gerade in dem Script anordnen, in dem man ihn aktuell braucht oder in dem man sich jeweils befindet.

All dies soll jedoch nicht in Abrede Stellen, daß ToolBook die Möglichkeiten böte, die Programmiersprache durch globale Funktionen zu verbessern, Objekte zu verändern, oder in entspre-

chenden Instanzen immer wiederkehrende Handler nur einmal anzuordnen. Diese und andere lobenswerte Möglichkeiten werden aber in der Realität zumeist durch mühselige „Techniken“, wie das Kopieren von Script zu Script u. ä.¹¹. Dies liegt meines Erachtens an zwei Gründen: Zum einen ist der Weg zum Wissen um diesen Techniken, wie oben dargelegt, ein dorniger. Zum anderen verliert ToolBook damit seine Vorzüge gegenüber den alternativen Programmiersprachen¹², da dieses Werkzeug nicht gewählt wurde, um sich doch mit den höheren Weihen der Windowsprogrammierung beschäftigen zu müssen. Die hier geforderte Lösung, einfache Benutzbarkeit für den Anwender und Disziplinierung des Anwenders in der Programmierweise gleicht sicherlich der Quadratur des Kreises, es sollte jedoch ein Konzept ersichtlich sein, wo die Programmiersprache ihren Schwerpunkt setzt, und damit ihren Anwenderkreis sieht.

Zielgruppe(n)

Es stellt sich damit die Frage, für welchen Nutzer ToolBook konzipiert ist, den Anwender, der kleine Programme ohne viel Aufwand erstellen will, oder den ambitionierten Anwender, dem gar an der Publizierung seiner Pro-

gramme gelegen ist, und der Optionen, wie die Einbindung von DLL's¹³, den Datenaustausch über DDE¹⁴ und die hohe Anpassungsfähigkeit von ToolBook nutzt. Auf keine der beiden Zielgruppen kann aus der Architektur geschlossen werden. Den professionellen Anwender werden vor allem die Größe von ToolBook-Anwendungen¹⁵, die Tatsache, daß immer die Runtime geladen werden muß und die starre Struktur¹⁶ der Aufteilung in Seiten¹⁷ stören. Der ungeübte Anwender wird dagegen die schnelle und einfache Arbeitsweise mit ToolBook und die Interpreter-Struktur zu schätzen wissen. So ermöglicht die Arbeitsweise learning-by-doing, faßbare Ergebnisse ohne lange und trockene Lernphasen. Diesen Anwender wird jedoch das Preis-Leistungsverhältnis stören, welches sich mit 536,- DM bzw. 890,- DM¹⁸ für ein solches Einsteigerprodukt m.E. nicht die Waage hält.

Der professionelle Nutzer wird ToolBook danach wohl vor allem zur schnellen Erstellung von Oberflächen für den Hausgebrauch nutzen, so zum Beispiel die Schwächen anderer Programmiersprachen im Benutzerschnittstellendesign ausgleichen, indem er auf die oben genannten Funktionen zurückgreift.

Abgestimmt auf die Bedürfnisse der größeren Anwenderschicht von Anfängern in der (Windows-) Programmierung wäre eine

⁸ eTutor ist der Name des Lernprogrammes, bei dessen Erstellung der Autor seine Erfahrungen mit ToolBook gemacht hat. eTutor wird in einer der nächsten Ausgaben von jur-pc vorgestellt werden.

⁹ Rodnay Zaks „Einführung in Pascal und Turbo Pascal“, 29.

¹⁰ Was ja einer ihrer Hauptvorteile ist.

¹¹ Der Autor berichtet dies aus eigener Erfahrung. Es kann nicht ausgeschlossen werden, daß besonders sorgfältige Menschen diese Fehler umgehen können. Die Erfahrungen, die zu der Entwicklung von Pascal geführt haben, scheinen aber darauf hin zu deuten, daß sich der Autor im Durchschnitt befindet.

¹² Turbo Pascal für Windows u. a.

¹³ Dynamic Link Libraries, Programmteile die extern vorliegen, und auf die verschiedene Programme zugreifen können.

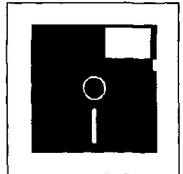
¹⁴ Dynamic Data Exchange, Mittels DDE können Daten aus anderen Windowsanwendungen in ein Programm eingebunden werden. Diese verändern sich dann „dynamisch“ mit.

¹⁵ Stefan Ebeling jur-pc 11/92, 1853.

¹⁶ Stefan Ebeling jur-pc 11/92, 1849.

¹⁷ Siehe auch dazu Stefan Ebeling jur-pc 11/92.

¹⁸ Quelle: Steckenborn Computer.



ToolBook-Version mit kleinerem Funktionsumfang, der wie dargelegt von dieser Anwendergruppe kaum ausgeschöpft wird, und damit schnellerer, kleinerer Runtime und vor allem kostengünstigerem Angebot¹⁹.

ToolBook wofür?

Stellt man die Überlegung an, ob sich die Anschaffung von ToolBook lohnt, so sollte man sein Augenmerk primär auf die Probleme lenken, die man zu lösen gedenkt. Es stellt sich damit vor allem die Frage, wofür ToolBook ein taugliches Werkzeug ist, gefolgt von der Frage, ob es zur Problemlösung nicht bessere Mittel gibt, die gegebenenfalls auch kostengünstiger sind. Das Hauptproblem einer solchen Wertung an dieser Stelle ergibt sich aus dem oben gezogenen Schluß, daß der Hauptanwendungskreis im Bereich der interessierten Anfänger²⁰ zu suchen ist. Diese werden weniger Standardprobleme zu lösen suchen als vielmehr Probleme in Randbereichen, für die der Markt keine zu-

friedenstellenden Angebote bereit hält. Die Vielfalt solcher Probleme erlaubt es nicht, grundsätzlich über den Nutzen von ToolBook zu urteilen.

Das bestätigt auch ein Blick auf die mit „Hypertext“ und „Hypermedia“ bezeichneten Bereiche, auf die hin ToolBook konzipiert ist. Die Vorteile von ToolBook liegen in der Gestaltung der graphischen Benutzeroberfläche. Anwendungen, die man mit ToolBook schreibt, sollten daher ihren Schwerpunkt auch hier setzen. So ist Hypertext zwar eine von ToolBook unterstützte Option, doch ist zu prüfen, ob nicht andere Werkzeuge wie der Microsoft Multimedia Viewer diesbezüglich hilfreicher sind. Dieses Werkzeug ist im Gegensatz zu ToolBook in erster Linie auf Hypertext ausgerichtet und hat bedingt dadurch eine höhere Ablaufgeschwindigkeit. Bezeichnend ist in diesem Zusammenhang, daß die ToolBook-Hilfe, ein Hypertext-System, nicht mit ToolBook selbst erstellt wurde. Absetzen müßte ToolBook sich auch gegen Produkte wie VisualBasic, das nach vergleichbarer Struktur Programme erstellen

läßt, die jedoch kompilierbar und deutlich kleiner sind. Dabei kann schließlich auch der deutliche Preisunterschied eine Rolle spielen²¹.

Resümee

Die Stärken von ToolBook sind die einfache und schnelle Handhabung. Defizite ergeben sich aber in der Wartbarkeit und im Erstellen von professionellen Programmen.

Daher sollte ToolBook vor allem für Problemlösungen eingesetzt werden, deren Wartungsbedarf klein ist und bei denen der Programmumfang überschaubar bleibt. Ob man im Endeffekt ToolBook einsetzt oder zum Beispiel VisualBasic bzw. den Microsoft Multimedia Kit hängt nicht nur vom zu lösenden Problem ab, sondern auch vom persönlichen Eindruck der Programme auf den jeweiligen Nutzer, da jede Arbeitsumgebung zu einem nicht unerheblichen Teil von individuellen Vorlieben geprägt ist.

¹⁹ Stefan Ebeling jur-pc 11/92, 1854.

²⁰ Ebeling zitiert dazu den Begriff des „fortgeschrittenen Nichtprogrammierers“ (jur-pc 11/92, 1854).

²¹ VisualBasic für Windows:
Schulversion: 126,- DM + MwSt
Kommerzielle Version: 277,- DM + MwSt
ToolBook:
Schulversion: 536,- DM + MwSt
Kommerzielle Version: 890,- DM + MwSt
(Quelle: Steckenborn Computer 19.7.1993)