

Praktische RUU-Entwicklung mit dem Autorensystem TOPIC

Alfred Schreiber

(deutsche Fassung eines Beitrags zur Fourth Prague Conference on Educational Cybernetics, 1991)

Das Ziel meiner knappen Ausführungen besteht darin, die Praxis der didaktischen Programmierung auf der Basis des Autorensystems TOPIC in einigen ausgewählten Schwerpunkten zu illustrieren. Es versteht sich von selbst, daß damit eine Einführung in TOPIC nicht geleistet werden kann. Ich kann vor allem Hinweise geben, die dann im Detail noch auszuführen wären.

Im einzelnen sollen folgende Themen angesprochen werden:

1. Kurzcharakterisierung von TOPIC
2. Sprachelemente der Erweiterungsstufe
3. Makro-Bibliotheken, Kursumgebungen
4. Prozesse unter der "Zeitlupe"
5. Projekte mit TOPIC
6. FrameMaker, Einstieg in den Rechnerunterstützten Unterricht (RUU)

Insbesondere werde ich nach Möglichkeit einen neueren Stand (TOPIC 3.0) berücksichtigen, der im Frühjahr 1990 veröffentlicht wurde. Die aktuelle Version umfaßt gebündelt praktisch sämtliche Komponenten der Erweiterungsstufe (sog. Boxen) und wartet mit neuen Möglichkeiten in den Bereichen der Grafik, Speicherverwaltung und Programmieretechnik auf.

1. Kurzcharakteristik von TOPIC

TOPIC ist ein Software-Werkzeug für den professionellen Entwickler von RUU. Als solches wird es derzeit vor allem an einer Reihe bundesrepublikanischer Hochschulen und Bildungseinrichtungen sowie in privaten Firmen für die Produktion von Unterrichtsprogrammen eingesetzt.

Das System gliedert sich in einen Sprachteil, die Autorensprache CCL (Courseware Construction Language), und einen menügesteuerten Bereich, in dem alle diejenigen Werkzeug- und Dienstprogramme (einschließlich fremder Software) aufrufbar sind, die der RUU-Autor für seine Arbeit benötigt. CCL ist im Kern eine Interpretersprache, deren Programme direkt (z.B. vom Editor aus) ausgeführt und ausgetestet werden können. Fertige (ausgetestete) Programme werden üblicherweise kompiliert (Umwandlung in einen intermediären Objektcode) und sind dann für den Endnutzer über ein spezielles Laufzeitmodul ausführbar.

Das Sprachkonzept läßt sich in aller Kürze wie folgt kennzeichnen: CCL gliedert sich in eine Grundstufe und eine Erweiterungsstufe. Auf beiden Stufen findet der RUU-Autor spezialisierte und mächtige Befehle (bzw. Befehlsstrukturen) für die Präsentation, die Interaktion und die Schnittstellensteuerung im Lernprogramm, also für alle bei der Unterrichtsentwicklung wesentlichen Bereiche.

Ein Prinzip, mit dem effektiv programmiert werden kann, ist hier die Manipulation geeigneter Datenobjekte.

Hierzu zählen z.B.: Bilder(stapel), Menüs, Tonfolgen, Lexika usw. Diese Gebilde lassen sich mit geeigneten Editoren unmittelbar erstellen. Im Lernprogramm werden sie zum Leben erweckt durch spezielle Befehle: z.B. bewirkt ein Befehl, daß ein Stapel aus Phasenbildern (einer Bewegung) entlang einer bestimmten Bahn über den Bildschirm geschoben wird und dabei die Bewegung kontinuierlich (nach dem "Filmprinzip") abläuft. Ein anderer Befehl öffnet ein Fenster und realisiert ein Glossar mit Fachbegriffen, und dgl. mehr.

Zwei wichtige Punkte sind zu erwähnen:

- CCL besitzt, sozusagen "nach unten hin", eine Schnittstelle zu Pascal (Turbo-Pascal 3.0), über die jedes CCL-Programm Daten mit einem externen Pascal-Programm austauschen kann. Dies gestattet Erweiterungen jeder Art oder auch das Anschließen sehr spezieller Simulationen.
- Es besteht die Möglichkeit (und bei fortgeschrittener Programmierung auch die Notwendigkeit) zur Bildung von (prozeduralen parametrisierten) Makros, sozusagen einem Ausbau "nach oben hin". Diese Makros lassen sich in eigenen Dateien außerhalb des aufrufenden Programms verwalten und speicherresident aktivieren.

Der erste Punkt hat es im Laufe der TOPIC-Entwicklung ermöglicht, das System schrittweise immer neuen Aufgaben und Erfordernissen anzupassen. Auf diese Weise ist die sog. Erweiterungsstufe entstanden.

Der zweite Punkt zielt in seinem Ausbaustadium auf eine Art allgemeiner Lehrsystementwicklung als Schaffung integrierter Kursumgebungen und Kursschalen für RUU.

Auf beides werde ich im folgenden etwas näher eingehen.

2. Sprachelemente der Erweiterungsstufe

Das Konzept der "Boxen" besteht darin, daß ein in Pascal geschriebenes Programm in das zu entwickelnde Lernprogramm eingebunden werden kann. Dies erfolgt nicht über den gewöhnlichen "user exit" (Ausgang für Anwendungen), sondern über einer eigene Schnittstelle, die den Datenaustausch mit dem externen Programm (im Sinne einer Überlagerung) gestattet.

Die häufigste Anwendung dieses Konzepts besteht in speziellen Erweiterungen des Sprachrahmens. Das heißt: Pascal-Prozeduren lassen sich neuen Befehlen auf der CCL-Ebene unterlegen. Auf diese Weise erstellt man Befehlssätze für die Videoanmsteuerung ebenso wie für die Kontrolle einer Sprachkarte usw. Eine solche Box kann maximal 18,5 KByte groß sein (die Pascal-Bibliothek nicht mitgerechnet), und insgesamt bis zu 15 solcher Module lassen sich speicherresident in der Laufzeit des Lernprogramms laden.

Zu den wichtigsten Boxen der derzeitigen Erweiterungsstufe von TOPIC gehören die folgenden:

- Funktionen-Box (Funktionen, Felder, Stack, FOR-Schleife)
- Record-Box (Kurshierarchie, Meßgrößen, edierbare Eingabe)
- Lexikon-Box (Lexikon-Editor, Lexikon, erweiterter Text)
- Effekt-Box (weitere Bildschirmseiten, Fensterformen, Bewegungen und Überblenden, komplexe Animation, "Multitasking")

Im Laufe der Zeit wurden eine Reihe weiterer Box-Entwicklungen durchgeführt, von denen hier nur einige wenige Beispiele zu nennen sind:

- Taschenrechner

- Itemsequenzierung nach dem Lernkartei-Prinzip (externe Lehrstrategie, die auch auf einer Datenbank operieren kann)
- spezielle Zusatz-Boxen (projektspezifisch, oder für besondere Peripherie: Video, berührungsempfindlicher Bildschirm, Sprach-Karte usw.)

Natürlich stellt sich bei reichlichem Gebrauch externer Module das Problem der Arbeitsspeichergröße. Die neue TOPIC-Version stellt dazu ein virtuelles Speichermodell zu Verfügung. Dieses Modell erlaubt die Nutzung von max. 64 MegaByte Arbeitsspeicher (nach Prioritäten konfigurierbar als RAM, EMS oder Platte und in 15 Kombinationen aus diesen).

Bemerkung zur Grafik: In punkto Grafik wird nach neuem Stand ein vereinheitlichendes Treiberkonzept praktiziert. Damit wird nicht nur die Entwicklung für gängige (und darüberhinausgehende) Standards erleichtert (Text, CGA, Hercules, EGA, VGA bis 1024x786 Punkte, VGA mit 256 Farben u.a.m.). Der Änderungsaufwand bei der Umstellung von Betriebsarten (z.B. von EGA auf Hercules) ist minimal bis null.

3. Makro-Bibliotheken, Kursumgebungen

Makrostrukturen sind eine fundamentale Voraussetzung für die Erstellung komplexerer Funktionseinheiten, z.B. Anzeige und Definition eines Funktionstastenmenüs, oder: Ausgabe von Hilfen und Lösungen zu einer Aufgabe und dgl. mehr.

Es genügt dazu im allgemeinen keineswegs, einfach irgendwelche nützlichen Makros zu sammeln. Wesentlich ist vielmehr, Funktionssysteme von miteinander arbeitenden Makros zu entwickeln, die dafür sorgen, daß sich ein großer Teil der Ablaufroutinen im Lernprogramm im Hintergrund abspielt; z.B.:

- Verwaltung der Lernerdaten,
- innerer und äußerer "Seiten"-Aufbau (allg. Layout)
- Bewegungsoptionen und Restituierung des Status quo,
- Menüführung,
- ein Teil der Lernprozeßkontrolle, u.a.m.

Nicht wenige TOPIC-Anwender hatten anfänglich mit diesen Dingen ihre Schwierigkeiten. Häufig war nicht wirklich klar, was eigentlich "hinter den Kulissen" des Kurses zu erledigen ist. Es konnte sich zudem erst allmählich ein adäquater "Programmierstil" herauskristallisieren, der diesen Aufgaben gewachsen war. Dabei ist eines gewiß klar geworden: RUU-Entwicklung ist etwas anderes als Entwicklung von Anwendersoftware (und schon gar nicht einfach ein Spezialfall) - diese Lehre muß jeder erfahren, der mit herkömmlichen Techniken (gar von BASIC her anerworbenen) diese Probleme zu lösen versucht.

Schließlich wurde von der TOPIC-Entwicklergruppe das Muster einer Makrobibliothek vorgelegt (TL: TOPIC-Library), die als Quelltext zusammen mit einer Dokumentation dem Unterrichtsautor hilft, in die grundlegenden Techniken der Lehrsystementwicklung einzusteigen, eigene Änderungen zu erproben und auch zu ganz anderen Möglichkeiten vorzustoßen.

Inzwischen liegen mannigfache Erfahrungen in der Bibliotheks- bzw. Schalen-Entwicklung vor (zu nicht geringem Teil in Zusammenhang mit speziellen Vorgaben gewonnen, vgl. Abschnitt 5 und 6).

4. Prozesse unter der "Zeitlupe"

Ein neues Funktionselement der praktischen Arbeit ist die TRACE-Box. Mit ihr kann sich der Entwickler in

ein laufendes Programm einschalten, dort beliebige Teilprozesse herausgreifen und ablaufen lassen und verändern. Die Änderungen werden selbsttätig im Quelltext realisiert.

Vor allen Dingen hilft hierbei eine automatische Justierung und Übertragung aller geometrischen Parameter (Koordinaten, Längen) in den TOPIC-Befehlen (Grundstufe, Erweiterungsstufe, Standardbibliothek und selbst definierte Makros). Das heißt z.B. konkret: Ein direkt (mit Maus oder Pfeiltasten) gerahmtes Fenster auf dem Bildschirm wird - für den Autor unmittelbar im TRACE-Box-Fenster sichtbar - auf Wunsch sofort in die zugehörige Befehlszeile eingetragen.

5. Projekte mit TOPIC

TOPIC hat sich zumindest in dem Sinne praktisch bewährt, als eine Reihe von Projekten auf seiner Grundlage entwickelt und erfolgreich zum Abschluß gebracht wurden. Hier kann ich nur eine (keineswegs repräsentative) Auswahl nennen:

- "Beleihungswertermittlung I/II" für den Deutschen Sparkassen- und Giroverband, Bonn, (TOPIC);
- Lernprogramme "Der BV-Computer" für die Bayerische Vereinsbank, München, sowie "EASY" für die Erste Österreichische Sparcasse (INSYS);
- Rollenspiel "Schiffahrt Anno 1778" für das Landesmuseum für Technik und Arbeit, Mannheim (INSYS);
- aktuelles Projekt mit TOPIC: "Aktienhandel an der Wertpapierbörse" für den Deutschen Sparkassen- und Giroverband, Bonn;
- zahlreiche weitere Projekte anderer Firmen und Bildungseinrichtungen, z.B. FrameMaker (Pädagogische Hochschule Flensburg, ECOL Entwicklungsinstitut für Computerunterstütztes Lernen e.V., Würselen).

6. FrameMaker, Einstieg in RUU

Ziel des zuletzt erwähnten Vorhabens war (seit 1987) die Entwicklung eines Modells für Erstanwender rechnerunterstützter Lehrverfahren. Das dabei entstandene Ergebnis, FrameMaker, soll helfen, angehende Autoren mit den Grundstrukturen des RUU vertraut zu machen. Ausgangspunkt waren dabei die einerseits wachsenden Qualitätsansprüche an das "neue Medium" Unterrichtssoftware und der andererseits immer noch erhebliche Abstand zwischen den Ideen (Wunschvorstellungen) über RUU und der Realität von Entwicklung und pädagogischem Einsatz. Häufig sind meiner Erfahrung nach auch schon bei weniger ehrgeizigen RUU-Vorhaben die produktionstechnischen oder die ökonomischen Möglichkeiten rasch erschöpft.

In dieser Situation galt es, dem Erstanwender ein geeignetes Modell an die Hand zu geben, das bereits nach kürzester Einarbeitung die Umsetzung eigener Konzepte in ablauffähige Programme gestattet. Damit dient FrameMaker als eine Art "Brücken"-Werkzeug für diejenigen pädagogischen Autoren, die zwar RUU-Eigenentwicklungen wünschen und benötigen, jedoch im ersten Schritt die dazu erforderlichen Ressourcen (Zeit, Ausstattung, Programmierkenntnisse) nicht in ausreichendem Maße mitbringen. In der Ausbildung von RUU-Autoren wird damit ein Anfang geleistet.

Als Modell mit Strukturvorgaben reduziert FrameMaker den Zeitaufwand für RUU-Entwicklungen nicht weniger drastisch als den Aufwand und die Möglichkeiten der "freien" Programmierung (innerhalb des Modells). Außerdem verringert sich dabei auch der "Beschreibungs"-Abstand zwischen Autorenvorlage und Lernprogramm: ganze Substrukturen lassen sich vom Autor direkt mit jeweils einem Befehlswort aktivieren, z.B.: Verwaltung von Lernständen, Positions- und Statuskonservierung, Aufruf von Wörterbüchern, Auswahlantwort-Seiten sowie Seiten für freie Texteingabe, u.a.m. Bei der Nutzung des TOPIC-

Autorensystems können später vom Autor solche Makro"ebenen" selbst entworfen und geschaffen werden.

In einem weiteren Schritt wurde auf der Grundlage des FrameMaker-Modells eine Programm-Schale für "Selbst-Test-Hilfe-Umgebungen" (Bezeichnung nach Haeffner) erstellt und zur Produktion von Programmen (Begleitung von Übungen im Fach Mathematik) angewendet. Studierende und Lehrkräfte können hier geeignetes didaktisches Material für RUU aufbereiten.

Hinweis

Für eine vom Verfasser in den Gesamtzusammenhang der Entwicklung didaktischer Software eingebettete Erörterung von RUU-Entwicklungswerkzeugen sei abschließend hingewiesen auf "Zentralblatt für Didaktik der Mathematik", Jg. 1990, Heft 3 (Analyseenteil).
