

Vom Lernschritt zum Lernbaustein

Überlegungen zur Konzeption und Entwicklung interaktiver Aufgaben

Alfred Schreiber

Bildungswissenschaftliche Hochschule Flensburg, Universität

Dieser Vortrag* wurde auf Einladung der Bildungsinitiative BIG (Bildungswege in der Informationsgesellschaft) am 20. Februar 1997 auf der Tagung „Neue Medien – Neue Aufgaben für die Lehrerbildung“ an der Universität-Gesamthochschule Paderborn gehalten. Vgl. auch die Übersichtsberichte zur Arbeitsgruppe „Didaktik der Mathematik“ sowie die Podiumsdiskussion „Neue Medien – Neue Impulse für Schule und Lehrerbildung?“ vom 21. Februar 1997, abgedruckt in Tulodziecki/Blömeke (1997).

Hochschulen und öffentliche Bildungseinrichtungen sind nicht nur Schauplatz von „Multimedia“, sie beteiligen sich auch zunehmend an der *Realisierung multimedialer Lernanwendungen*. Der folgende Beitrag gilt vor allem der Weiterentwicklung dieser Perspektive. Ich bin davon überzeugt, daß wir dabei der *Interaktivität* der sog. Neuen Medien verstärkte Aufmerksamkeit schenken sollten. Insbesondere setze ich mich dafür ein, das *Konzept des „Frames“* (d.h. des softwaretechnischen Rahmens für eine zu bearbeitende Aufgabe) zu revidieren (hin zum „interaktiven Lernbaustein“) und die Frame-Entwicklung selbst auf ein höheres Niveau zu heben. Meine Betrachtungsweise hat sich während der letzten zehn Jahre in diversen Grundlagen-Entwicklungen zum computerunterstützten Lehren und Lernen und zahlreichen Lernsoftware-Projekten mit Schwerpunkt in der betrieblichen Aus- und Weiterbildung herausgebildet.

* Der hier wiedergegebene Text enthält geringfügige red. Änderungen und einige zusätzliche Literaturhinweise.

Wie "interaktiv" ist Lernsoftware heute?

Obwohl – in einem weiteren Sinne – auch sog. Anwendungsprogramme oder „Werkzeuge“ (vor allem didaktische Werkzeuge) zu Lernzwecken genutzt werden können oder sollen, möchte ich mich hier zunächst einmal auf *Lernsoftware im engeren Sinne* beschränken, d.h. auf Programme, die von sich aus Lernstoff enthalten (oder erzeugen) und ihn auf unterschiedliche Weise vermitteln bzw. durcharbeiten gestatten (z.B. über Such- und Bewegungsmechanismen zugänglich gemachte Informationseinheiten, durch Übungsaufgaben, durch lernzielorientierte Tests, u.a.m).

Seit Beginn der 90er Jahre erlebt der Lernsoftware-Markt einen wahrhaftigen Boom. Dazu haben wesentlich beigetragen:

- grafische Benutzungsoberflächen bei Anwendungen und Betriebssystemen
- die multimediale (auch kontinuierliche Medien einbeziehende) Informationsdarbietung
- Hypertext (in letzter Zeit besonders populär geworden durch das „World Wide Web“)

Sicherlich bilden diese neuen Möglichkeiten eine wertvolle Basis und Bereicherung für die Entwicklung von Lernprogrammen. Darüberhinaus scheint man in ihnen vielerorts auch eine *Steigerung von Interaktivität* zu sehen (manche Autoren sprechen neuerdings gar von „hochinteraktiver“ Lernsoftware).

Ist diese „Hype“ wirklich berechtigt? Ich habe da meine Zweifel. Werfen wir einen kurzen Blick zurück in die frühen Pioniertage des computergestützten Unterrichts (CUU), als noch ein Schreckgespenst namens „Behaviorismus“ umging. Dieses ärgerte Schüler oder zumindest Pädagogen lange Zeit über mit *klein(st)schrittigen* und in hohem Maße *direktiven* Programmen. Deren Lehrinhalt wurde zudem häufig durch vielgeschmähte „*elektronische Blättermaschinen*“ dargeboten.

Das in Abbildung 1 gezeigte Gerät – wie unschwer an der PC-ähnlichen Formgebung zu erkennen ist – kann allerdings nicht aus jener Ära stammen. Es hat denn auch vor nicht allzu langer Zeit das Programmheft einer bekannten Messe geschmückt und soll hier als Anzeichen dafür betrachtet werden, daß das einstige Gespenst einer guten Fee gewichen ist, die auf den Namen „Hypermedia“ hört und auf die heute so manches Hohelied gesungen wird.

Hinzu kommt eine Hintergrundphilosophie (oder -pädagogik), die *Instruktion zugunsten von Konstruktion* aus dem Lerngeschehen zurücknimmt oder gar zu verdrängen sucht (vgl. etwa Schulmeister 1996). Die *instruktionalen Frames*, wie man sie vom CUU (oder CBT) her kennt, erscheinen dadurch noch stärker zurückgedrängt und negativ besetzt als sie es ohnehin schon waren.

Fragen Sie sich aber doch auch einmal umgekehrt: Was ist so interaktiv daran, wenn wir Hypertext-Verknüpfungen verfolgen oder auf ähnliche Weise durch multimedial aufbereitete Doku-

mente „navigieren“? Haben wir es nicht doch wieder mit einer Blättermaschine zu tun – eben nur einer komfortableren und manchmal auch weniger übersichtlichen? – Gewiß, ich kann Sprechtext und Musik zu Gehör bringen, ich kann einen Videoclip oder eine grafische Animation abspielen, anhalten oder verlangsamen. Doch sind dies *im Kern einfachste Interaktionen*: nämlich durch Mausclick veranlaßte Wiedergabe von Text, Bild und Ton.

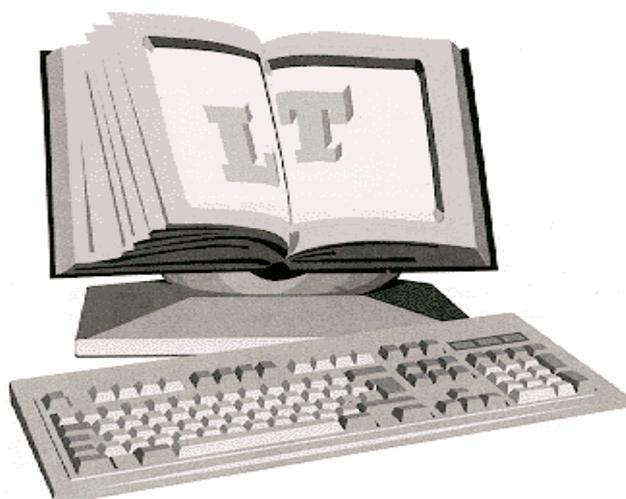


Abbildung 1

Das Navigieren – oder wie es im Internet heißt „Surfen“ – in lexikonähnlichen assoziativen Begriffsnetzwerken mag nützlich für Recherchen oder reine Informationsaufnahme sein, es sollte aber nicht zu dem Glauben verführen, wir hätten es dabei mit einer irgendwie gehirn- oder denkangepaßten Struktur zu tun, die deshalb eine natürliche Präferenz für Lernzwecke genießt.

Im folgenden möchte ich *für den Gebrauch von Frames in Lernsoftware* plädieren. Frame-Orientierung wurde m.E. vorschnell und zu unrecht mit den gerade erwähnten „Gespens-terplagen“ des Behaviorismus in Verbindung gebracht. Darin ist unter anderem *ein* Grund zu sehen, weshalb in heutiger Lernsoftware viel zu selten anspruchsvollere Interaktionsformen bei Aufgaben und Übungen anzutreffen sind und generell das Hauptaugenmerk auf der Präsentation von Information (Hypertext und Multimedia) liegt.

Mein Plädoyer beruht auf einer Revision des Frame-Konzepts und daran geknüpften Überlegungen, wie man – mit vertretbarem Aufwand und unter Ausnutzung vorhandener technischer Möglichkeiten – zu anspruchsvolleren Interaktionen kommen kann. Dabei stellen Aufgaben mit mathematischem Gehalt – das zeigt die an guten Mathematik-Lernprogrammen arme Geschichte des CUU – eine besondere Herausforderung für Fachdidaktiker und Software-Entwickler dar.

Klassische Frame-Struktur und Lerngegenstand

Schauen wir uns zunächst die herkömmliche Frame-Struktur an. Sie stellt einen recht einfachen Kreisprozeß dar (Abbildung 2):

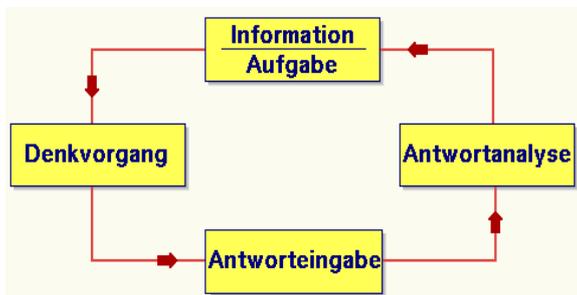


Abbildung 2

Dem Lerner wird zunächst Information dargeboten; dann erhält er die Möglichkeit, ein damit zusammenhängendes Problem zu lösen bzw. eine Frage zu beantworten. Die eingegebene Antwort wird vom System untersucht und ausgewertet. Schließlich erfolgt eine Rückmeldung und ggf. die erneute Darbietung dieser oder einer anderen Aufgabe (eines Problems, einer Situation) usw.

Frames, die diesem Schema unterworfen sind, können sich nach Umfang, Inhalt, Typ und innerem Aufbau immer noch beträchtlich voneinander unterscheiden.

Einige Beispiele:

- Ein Lernschritt nach Skinner ist umfangsmäßig klein, mit hoher Wahrscheinlichkeit erfolgreich abzuarbeiten (dies im Hinblick auf die erwartete Reiz-Reaktions-Verstärkung), verlangt die Konstruktion der Antwort in Form einer Texteingabe und duldet Rückmeldungen wenn überhaupt nur in Form einer knappen Erfolgsquittung.
- Diese streng-behavioristische Auffassung von „Lernen durch Dressur“ war keinesfalls das maßgebliche Vorbild für CUU-Frames. Pressey, Crowder u.a. forderten demgegenüber ein „Lernen durch Einsicht“ und glaubten dies in der Frame-Struktur durch dialogische Anreicherung erzielen zu können. Mit damaligen Mitteln war das allerdings nur bei der einfachen Multiple-Choice-Form möglich. Die Antwortanalyse ist hier sicher, und man kann jeden Antwortwert mit belehrenden Kommentaren verknüpfen und/oder zum Anlaß für Verzweigungen nehmen.

Während Kleinschrittigkeit für sich genommen nicht unbedingt schon ein pädagogisches Übel darstellt – sie kann in bestimmten Situationen sogar durchaus sinnvoll sein – fällt ein anderer Schwachpunkt dieser Art von Frames schon mehr ins Gewicht: der in hohem Maße *indirekte Bezug zum Lerngegenstand*. Er äußert sich darin, daß *die durch den Frame vermittelte Interaktion eigentlich keine auf den Inhalt des Lernschritts bezogene Aktivität* ermöglicht.

Bei einer Durchsicht der in den letzten 10 Jahren hergestellten Lernsoftware-Produkte – in die ich ausdrücklich auftraggeberspezifische Lösungen einbeziehe, die im Durchschnitt von besserer Qualität sind – wird man eine Reihe von Ansätzen erkennen, die diesen Mangel durch entsprechende Aufgabengestaltung zu vermeiden suchen. Die Praxis hat im Großen und Ganzen die zu enge (behavioristische) Auslegung des Frame-Konzepts überwunden und dabei einen direkteren, angemesseneren Bezug zum Lerngegenstand hergestellt. Dies konnte noch ganz *innerhalb* des herkömmlichen Konzepts geschehen, also unter Wahrung des Elementarzyklus von Aufgabe, Lösungseingabe, Analyse und Rückmeldung (vgl. Schanda 1995). Der bessere Gegenstandsbezug macht es häufig erforderlich, auf (zu) einfache generische Strukturen wie Freitexteingabe oder Multiple-Choice zu verzichten.

Eine Revision des Frame-Konzepts

Es liegt nahe, diese faktisch bei vielen Fachautoren und Entwicklern schon stattgefundene Schwerpunktverlagerung im CUU (CBT) auch durch eine konzeptionelle Revision einzuholen, zu unterstützen und voranzutreiben. Dies sollte auf den Ebenen der *didaktischen Gestaltung*, des *formalen Aufbaus* und der *praktischen Nutzung* geschehen:

Ebene I: Didaktische Gestaltung

Unter den vielerlei didaktischen Anforderungen, die immer wieder an Frames bzw. Aufgaben/Problemstellungen im CUU gestellt werden, greife ich hier vier Punkte heraus:

1. Die abverlangte kognitive Leistung muß als solche sinnvoll sein.
2. Inhaltsdarstellung und die daran geknüpfte Aktivität müssen dem fachlichen Gegenstand angemessen sein.
3. Das Problem sollte Transfer oder zumindest Anwendung erfordern.
4. Das Thema der Aufgabe sollte ggf. durch Kontext angereichert werden.

In diesen Punkten erkenne ich substanzielle Bedingungen für die inhaltliche Gestaltung von interaktiv dargebotenen Aufgaben. So leicht sie sich aussprechen und anerkennen lassen, so schwer sind sie umzusetzen und entsprechend selten in der Praxis erfüllt. Aufgabenstellern bzw. Fachautoren und Software-Entwicklern verlangen sie ein hohes Maß an Erfahrung und Können ab.

Ebene II: Formaler Aufbau

Vorbereitend ist das Frame-Konzept inhaltlich und strukturell zu entleeren. Durch diese Abstraktion werden aus Frames so etwas wie *undurchsichtige Items*. Sie können nun einer aufrufenden Komponente unterworfen werden (z.B. einem „Makrotutor“ oder einem assoziativen Netzwerk, s. Abbildung 3). Im Prinzip kann ein so „definierter“ Frame alles enthalten: eine echte Aufgabe, reine Informationsdarbietung, oder auch – durch geeignete Verbindungselemente – Abschnitte von realem Unterricht (wie beim CMI = Computer-Managed Instruction üblich). Aus dieser Sicht bedeutet Frame-Orientierung nichts anderes als das Verfahren, Information durch Gliederung in Untereinheiten zu strukturieren.

Nun ist das Frame-Konzept positiv zu bestimmen. Dies geschieht in zwei Stufen. Auf der ersten Stufe entstehen *elementare Frames*, unterschieden in *informative* und *interaktive*. Ein informativer Frame ist eine selbständige Einheit, in der eine bestimmte Information dargeboten wird. Bei interaktiven Frames ist stets eine Eingabe möglich (z.B. Veränderungen an einem Objekt bzw. Prozeß) und eine Rückmeldung erhältlich (z.B. Information oder Demonstration der Auswirkungen). Ich spreche von einem *Aufgaben-Frame*, wenn die Eingabe analysiert wird und das Ergebnis in die Rückmeldung einfließt. Ein elementarer interaktiver Aufgaben-Frame hat im Prinzip die oben schon beschriebene Form des klassischen CUU-Rahmens. Karl Eckel hat in seinem Werk „Didaktiksprache“ die komplexe innere Struktur – unter dem nicht so glücklichen Begriff „Kleinstunterricht“ – überzeugend und in weitgehender Allgemeinheit dargelegt.

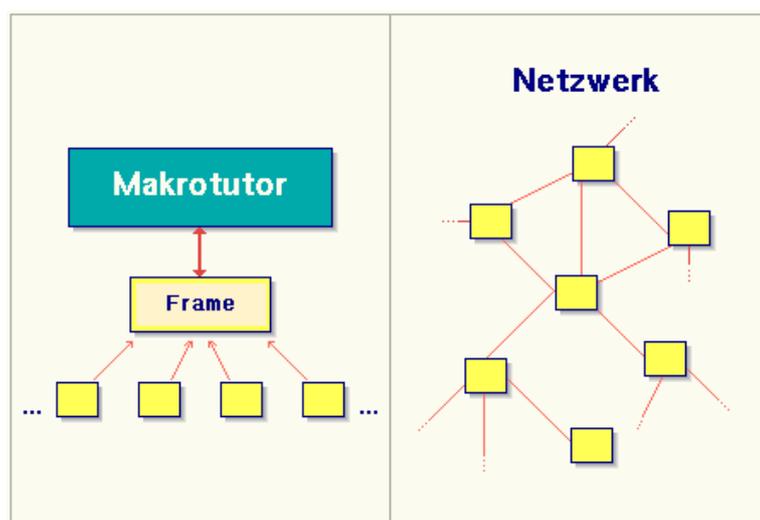


Abbildung 3

Auf der zweiten Stufe sollen „Moleküle“ aus elementaren Frames gebildet werden können. Dabei sind jeweils ein Ein- und Ausgang durch bestimmte (nicht notwendig verschiedene) Teilmodule festzulegen. (Dieser Punkt ist von Bedeutung, wenn der zusammengesetzte Frame als Knoten eines Lernweggraphen betrachtet wird. Komplexionen höherer Stufe sind kaum von Interesse, vielmehr entstehen durch Verknüpfung von Frames oder Ausführung von Frame-Sequenzen größere, i. a. selbständige Lerneinheiten, die gemeinhin als „Kapitel“ oder „Lektionen“ schon ein vollständiges Lernprogramm bilden können.) Ein Frame soll *interaktiv* heißen, wenn er durch mindestens einen interaktiven Elementar-Frame (Basismodul) konstituiert wird. Theoretisch sind allerlei Aufbauvarianten denkbar (Sequenz, Stern usw.).

Ebene III: Praktische Nutzung

Mit den soweit spezifizierten interaktiven Lernbausteinen verfügt man schon über ein Konzept, das den im alten CUU immer wieder (und häufig immer noch) praktizierten Gedanken kleiner Lernschritte überwindet. Ich möchte aber weiter gehen und die Lernbaustein-Idee durch zusätzliche Anforderungen aus der Sicht der praktischen Nutzung profilieren.

Ich denke hier vor allem daran, daß sich ein Lernbaustein in der Welt der Software wie eine *echte Komponente* verhalten sollte, ein Merkmal, das bei Frames in Lernsoftware nach heutigem Stand der Dinge in der Regel nicht anzutreffen ist.

Was ist unter dieser Forderung zu verstehen? Stellen Sie sich einen einfachen Frame in Gestalt einer Rechenaufgabe vor. Typischerweise steckt er in einem größeren Kontext, einem Lernprogramm, das ihn beherbergt und mit den zu seinem Ablauf nötigen Funktionen versorgt. Wenn Sie ihn zu Gesicht bekommen wollen, so müssen Sie sich durch das „Verkehrssystem“ (d.h. die Menüs, Such- und Bewegungsfunktionen) des Lernprogramms zu ihm durcharbeiten.

Der Frame wäre aber flexibler nutzbar, wenn er *als einzelner* und *von außen* abgerufen werden könnte, und das heißt: auch in einer anderen als seiner ursprünglichen Umgebung ausführbar wäre. – In der einfachsten Form kann das so geschehen: Das Programm, das den Frame benutzen will, initialisiert einen Unterprozeß, in dem über eine geeignete Schnittstelle der Frame angefordert und dann separat von „seinem“ Lernprogramm abgewickelt wird (s. Abbildung 4).

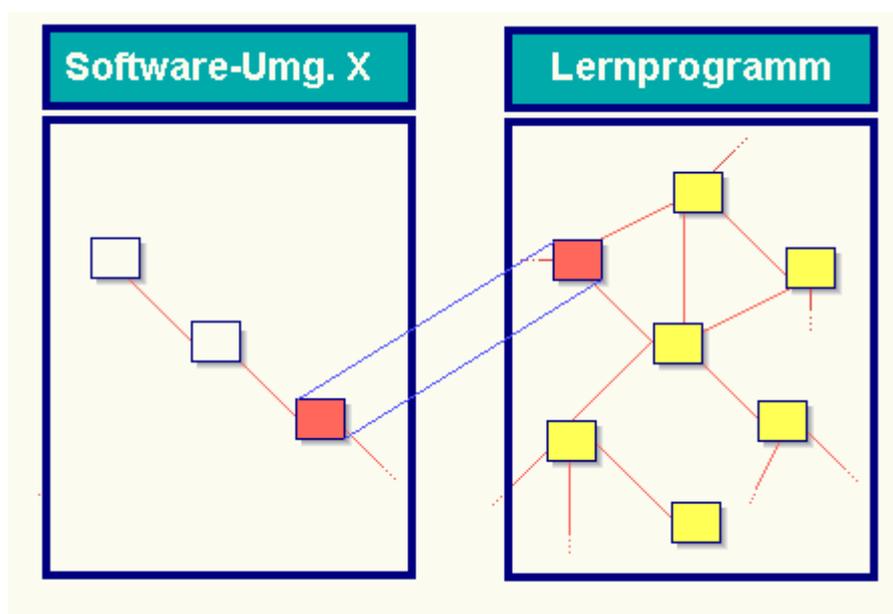


Abbildung 4

Damit so etwas überhaupt funktioniert, müssen sich der Inhalt und die formale Definition des Frames so weit wie irgend möglich von denjenigen Teilen trennen lassen, die für seine Ausführung (und damit die Realisierung der Interaktion) verantwortlich sind.

Das soweit skizzierte Baustein-konzept harmoniert im übrigen gut mit anderen „Paradigmen“. Zum Beispiel könnten fachliche/didaktische Werkzeuge oder auch Hypertext-Programme ihr Angebot durch Inanspruchnahme „externer“ Lernbaustein-Dienste mühelos erweitern (s. Abbildung 5).

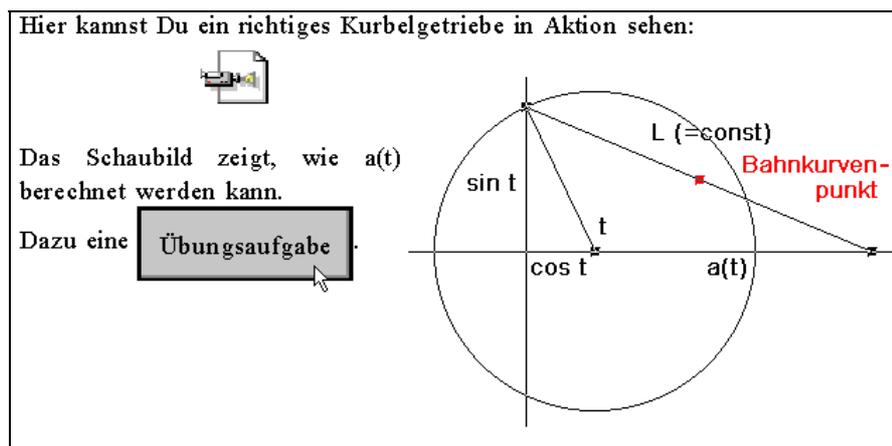


Abbildung 5

Man kann die Idee von Lernbausteinen als Software-Komponenten natürlich noch weitergehender auslegen: Auch über die Grenzen eines Rechners, eines lokalen Netzes oder eines Betriebssystems hinaus macht die *Verteilung* von Lernbausteinen Sinn – vorausgesetzt der hohe technische Aufwand geht einher mit einer entsprechenden didaktischen und fachlichen Qualität sowie einer angemessenen Form der Präsentation und der Interaktion. Klassische Autorenwerkzeuge bleiben hinter

diesen heute z.B. durch die COM-Technologie (COM = *Component Object Model* von Microsoft, vgl. Chappel 1996) gegebenen Möglichkeiten noch zurück. *Java* (von Sun) erscheint diesbezüglich als eine vielversprechende Entwicklerplattform, bietet allerdings (noch) keine CUU-generischen Strukturen (was sich aber durch einen sich allmählich entwickelnden Komponentenmarkt in nicht allzu langer Zeit ändern kann)[†].

Generische Modellierung?

Nachhaltiges Interesse an interaktiven Lernbausteinen in dem hier beschriebenen Sinne wird sicherlich erst durch überzeugende Beispiele geweckt werden. Ich denke hierbei weniger an qualitativ herausragende und dabei gänzlich individuell entwickelte singuläre Beiträge (von denen man natürlich immer lernen kann), sondern an *Entwicklungen auf der Grundlage generischer Strukturen*. Software-Entwickler sollten bei ihrer schwierigen Aufgabe auf ein gewisses Repertoire an fachspezifischen (hier: CUU-spezifischen) Komponenten zurückgreifen können und nicht immer wieder das Rad von neuem erfinden müssen (vgl. Schreiber 1990, 1998).

[†] Inzwischen (Stand Februar 1998) wurden mit dem Java-Autorensystem CBTrainer von TOPIC, Würselen/Aachen, die meisten Aufgabenformen von DUAL (vgl. Schreiber 1992) als plattformübergreifende Bausteine neu entwickelt. Es ist geplant, die dabei als Prototyp entstandene Komponentenpalette für die Portierung eines Aufgaben-Trainers im Bereich Mathematik (Grundstudium Lehramt an der BU Flensburg) einzusetzen.

Natürlich bieten Autorenwerkzeuge wie TOOLBOOK (CBT-Edition, Asymetrix) oder AUTHORWARE (Macromedia) diesbezüglich eine gewisse Palette von Fertigbauteilen, allerdings ohne daß die bisher entwickelten Anforderungen an interaktive Lernbausteine damit zu erfüllen wären. Erschwerend kommt hinzu, daß a) die bereitgestellten Interaktionsformen nicht allgemein genug und b) die jeweiligen proprietären Formate und Plattformen untereinander unverträglich sind.

Ich werde vor diesem Hintergrund weder vor Autorenwerkzeugen warnen noch zu der Aufforderung Zuflucht nehmen, man solle so etwas wie ein CUU-Standardformat in die Welt setzen. Die zuletzt genannte Idee wäre ebenso alt wie illusorisch. Erfolgversprechender erscheint mir der Ansatz zu sein, generische Bausteine für die verschiedenen Bereiche und Ebenen der Frame-Architektur zu entwickeln bzw. im Sinne von Wiederverwendung zu nutzen. Auf diese Weise kann man auch schon teilweise – z.B. unter- oder außerhalb der Formgebung einer Interaktion – von bestimmten Diensten wie der Antwortanalyse oder der Itemsequenzierung profitieren. Ein weiterer Vorteil ist die daraus erwachsende Option auf ein höheres Maß an Bereichs- oder Fachspezifität.

In der Tat liegt in dem zuletzt genannten Aspekt eine empfindliche Schwachstelle generischer Modellierung überhaupt. Mit zunehmender Allgemeinheit schwinden nämlich die Spielräume der fachspezifischen Gestaltung sowie die Möglichkeiten, fachdidaktische Erkenntnisse in Konzeption und Ablauflogik des Frames einfließen zu lassen. Das gilt für Wissensgebiete mit einer semantischen Tiefenstruktur, wie sie die *Mathematik* aufweist, in ganz besonderem Maße.

An dieser Stelle möchte ich exemplarisch auf einige der hierbei zum Vorschein kommenden Probleme und Möglichkeiten anhand der folgenden vier Hauptklassen elementarer (interaktiver) Frames eingehen:

1. Zuordnung
2. Numerische Aufgabe
3. Texteingabe(n)
4. Objekt- und Prozeßmanipulation

Während die Klassen 1-3 in hohem Maße durch generische Strukturen geprägt sind, gilt das genaue Gegenteil für Objektmanipulationen, für die sich kaum logische Typen und ebenso schwer Eingabeformate systematisch beherrschen lassen.

Zuordnungen

Faßt man hierunter auch mehrdeutige partielle Zuordnungen (Relationen), so ergeben sich nach systematischer Analyse 10 mögliche Typen. Herkömmliche CUU-Interaktionen wie Alternative-, Einfach- und Mehrfachauswahl- sowie Anordnungsaufgaben lassen sich leicht hier einordnen. Für die ganze Klasse genügt übrigens ein einziges Antwortanalyse-Verfahren.

Mit Zuordnungs-Frames lassen sich im mathematischen Gebiet nur bescheidene kognitive Leistungen vollbringen. Am ehesten geeignet, weil gegenstandsnah, sind noch solche Aufgaben, in denen es tatsächlich auch um Zuordnungen im mathematischen Sinne geht.

Numerische Aufgaben

Sie gehören seit eh und je zum Standardrepertoire von CUU-Software, werden aber m.E. zu häufig auf die leichte Schulter genommen und entsprechend trivial realisiert. Dabei werfen sie eine Vielzahl interessanter Probleme auf, z.B. die programmgesteuerte Erzeugung numerisch (und ggf. sogar textuell) variabler Fragestellungen, die Konstruktion allgemeiner Prüfschlüssel und Filter für numerische Eingaben, oder die Handhabung numerischer Mehrfacheingaben während einer Aufgabenlösung.

Mit welcher Schwierigkeit allgemeingültige Auswertungsmechanismen verbunden sind, zeigen die Bemühungen um Fehlerdiagnosen in „Intelligenten Tutoriellen Systemen“ zur elementaren Arithmetik; auch die Ideen Eckels (1989) zu einer „regressiven Rechenweganalyse“ sind hier zu nennen.

Eine gewisse Mischung von alledem zeigt ein Beispiel aus einem interaktiven Lernbaustein, der die Ermittlung des Einheitskurses einer Aktie auf der Grundlage eines Makler-Skontros behandelt (Schreiber 1991, s. Abbildung 6):

3.6 ÜBUNGEN: Kursermittlung

Dem amtlichen Makler an der Wertpapierbörse liegen folgende Aufträge für Gestra-Aktien vor:

Skontro			
Käufe		Verkäufe	
Stück	Limit	Stück	Limit
45	billigst	20	bestens
45	137	25	134
50	136	35	135
30	135	40	136
20	134	40	138

Einheitskurs-Berechnung			
Kurs	Nachfr.	Angebot	Umsatz
134	190		
135	170		
136	140		
137	136		
138			

Falsch

Sie müssen feststellen, welche Käufer wären bereit, zu diesem Kurs zu kaufen.
Für die Lösung bitte weiter mit ...

F1 Hilfe F2 Lektionen F3 Abschn F5 Zurück F6 Lexikon F7 Abbruch F8 F9 F10

Abbildung 6

Texteingaben

Die sog. Freitexteingabe war und ist (wegen der Möglichkeit sinnloser oder unvorhergesehener falscher und korrekter Antworten) so etwas wie ein „Angstgegner“ des CUU-Entwicklers. Übrigens ist dieser Frametyp auch bei den Lernern nicht sonderlich beliebt, da man nicht einfach klicken kann, sondern auf der Tastatur tippen muß. Daher wird es allseits begrüßt, wenn die Eingabevielfalt etwas eingeschränkt wird: durch Formatfilter, Verkürzung der Eingabe und vor allem durch Kontextualisierung (was dann zum Lückentext führt). Die Qualität der Interaktion hängt dann aber immer noch entscheidend von der Antwortanalyse ab. Heute darf man einen leistungs-

fähigen Musterabgleich erwarten, z.B. unter Verwendung der gewichteten Levenshtein-Distanz und unter Berücksichtigung phonetischer Äquivalenzen.

Mathematisch gesehen scheint eine Texteingabe nicht viel herzugeben; aber der Schein trügt. Ein interessantes Eingabeobjekt sind Formeln, die als Lösung einer Aufgabe resultieren und die ich hier einmal als linearen Text auffasse. Allein das Problem, eine richtige Formeleingabe auch zu erkennen, macht es notwendig, dem Lernbaustein mathematische Fähigkeiten zu verleihen.

Eine sinnvolle und ökonomische Lösung dieses Problems kann eigentlich nur darin bestehen, sich die Dienste einer geeigneten, bereits verfügbaren Komponente nutzbar zu machen. Am ehesten kommt hier ein CAS-Kern in Frage, der sich vom Lernbaustein aus aktivieren läßt und das Ergebnis des Antwortvergleichs zurückschickt. MATHEMATICA leistet eine solche Kommunikation via MATHLINK auch im entfernten Zugriff (Wolfram 1997).

Objekt- und Prozeßmanipulationen

Im allgemeinen lassen sich mathematische Aktivitäten nicht oder nur mit Mühe durch die eher konventionellen Frameklassen 1-3 abbilden. Handlungen mit Objekten eignen sich besser dazu, bringen allerdings die Schwierigkeit mit sich, daß die Zustände der manipulierten Objekte *kontrolliert* und *interpretiert* werden müssen. Hier ist ein (noch) relativ einfaches geometrisches Beispiel aus dem Lernprogramm „Elly“ von Jürgen Ingold (1994):

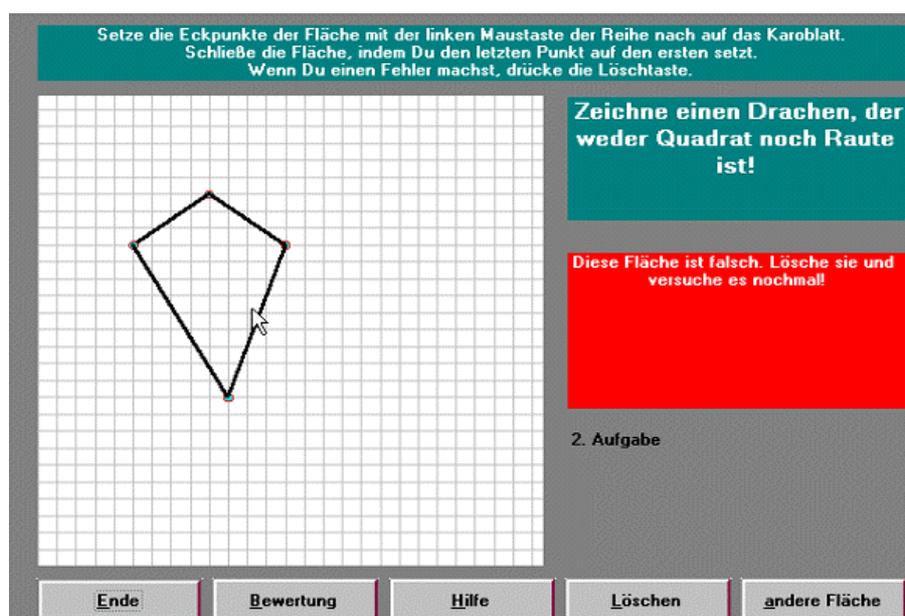


Abbildung 7

Natürlich kann man auf Antwortanalysen verzichten und – was *auch* Sinn macht – ein veränderbares Objekt zum freien oder textbegleiteten Erkunden und Experimentieren darbieten. Von dieser Möglichkeit machen zahlreiche didaktische Werkzeuge zur Visualisierung oder zur Konstruktion Gebrauch.

Gleichwohl scheinen mir evaluierbare Objektmanipulationen – oder hier vorerst einmal bescheidener: *Veränderungen an ebenen Figuren* – eine reizvolle Perspektive für die Entwicklung interaktiver Lernbausteine zu eröffnen. Meine eigenen Untersuchungen zu diesem Thema befinden sich noch im Anfangsstadium, deuten aber schon auf die Möglichkeit hin, wenn schon nicht allgemeine, so doch mindestens gebietspezifische generische Strukturen zu finden.

Literatur

Chappell, D.: ActiveX und OLE verstehen. Ein Leitfaden für Entwickler und Manager. Unterschleißheim: Microsoft Press Deutschland 1996

Eckel, K.: Didaktiksprache. Grundlagen einer strengen Unterrichtswissenschaft. Köln Wien: Böhlau 1989

Ingold, J.: Elly für Windows. 1. Auflage. Berlin: Cornelsen Software 1994

Schanda, F.: Computer-Lernprogramme: wie damit gelernt wird, wie sie entwickelt werden, was sie im Unternehmen leisten. Weinheim; Basel: Beltz 1995

Schreiber, A.: Entwicklung didaktischer Software auf Autorensystembasis. Zentralblatt für Didaktik der Mathematik 22(3), 92-104 (1990)

Schreiber, A. (Wiss. Ltg.): Aktienhandel an der Wertpapierbörse (PC Lernprogramm, Fachautoren: Schick, R., Zabel, O.). Stuttgart: Deutscher Sparkassenverlag 1991

Schreiber, A.: Eine Didaktik-Umgebung für adaptives Lernen (DUAL). Grundlagenstudien aus Kybernetik und Geisteswissenschaft/Humankybernetik 33, 25-31 (1992)

Schreiber, A.: CBT-Anwendungen professionell entwickeln. Ein Leitfaden. Berlin; Heidelberg; New York: Springer 1998 (im Erscheinen)

Schulmeister, R.: Grundlagen hypermedialer Lernsysteme. Theorie – Didaktik – Design. Bonn: Addison-Wesley 1996

Tulodziecki, G., Blömeke, S. (Hrsg.): Neue Medien – neue Aufgaben für die Lehrerbildung. Initiative: BIG – Bildungswege in der Informationsgesellschaft; Tagungsdokumentation. Gütersloh: Verlag Bertelsmann Stiftung 1997

Wolfram, S.: Das Mathematica Buch, 3. Auflage. Bonn, Reading MA: Addison Wesley Longman Verlag GmbH 1997