

Stand: Dezember 1995

# Die Entwicklung computergestützter Lernbausteine für die Anfangssemester im Lehramtstudium Mathematik

Alfred Schreiber

Bildungswissenschaftliche Hochschule Flensburg, Universität

## Zusammenfassung

Es wird über ein Projekt berichtet, dessen Ziel die Entwicklung, Evaluation und Implementierung computergestützter Bausteine ist, mit denen Studienanfänger im Lehramtsfach Mathematik parallel zu Vorlesungen, Übungen sowie der Lektüre von Skripten und Lehrbuchtexten fachliche (oder fachdidaktische) Inhalte durcharbeiten oder nacharbeiten können. Den übergreifenden Rahmen hierzu bildet ein tutorielles System, das als allgemeine (bereichsunspezifische) Schale angelegt ist. Es ermöglicht die Integration unterschiedlicher Tätigkeiten: Üben anhand von Aufgaben (mit adaptiven Lehrstrategien), Selbstkontrolle/Test, Nutzung begriffsvernetzter Fachglossare, Abruf von Skripten, Einsatz mathematischer Werkzeuge (wie DERIVE) u.a.m. Bei der Entwicklung der Studienmaterialien, insbesondere bei der Text- und Aufgabenerstellung, bieten sich gute Gelegenheiten zur Mitwirkung fortgeschrittenerer Studenten. Es werden Forschungsaufgaben bei der Weiterentwicklung des Projekts formuliert.

## 1. Eine revidierte und erweiterte Idee von CAI

Die vorliegende Studie erörtert ein Konzept zu Entwicklung und Einsatz computergestützter Lernbausteine im mathematischen Grundstudium. Es beruht auf einer erweiterten Idee von CAI ("Computer-Assisted Instruction") und zielt auf eine dementsprechend offene und integrative Rahmenstruktur für unterschiedliche Formen mathematikdidaktischer Software.

### *Intelligente tutorielle Systeme*

Zu den schwierigsten einschlägigen Aufgaben gehört die Konstruktion "intelligenter" tutorieller Systeme (ITS, ICAI). Schon die Entwicklung mathematischer Expertenmodule für vergleichsweise eng begrenzte Bereiche wie elementare Termumformungen, quadratische Gleichungen oder Differentiationsregeln ist mit beträchtlichem Aufwand verbunden (Sleman, Brown 1982, Bonnet et al. 1981). Seit den 70er Jahren folgen solche Versuche dem Paradigma, wonach ein ITS einen "idealen Lehrer" abbilden sollte, der den Lehrstoff ebenso gut kennt wie den Schüler und die für ihn angemessenen Lehrhandlungen (Kearsley 1987). Diese ebenso ehrgeizige wie faszinierende Vision hat einige wichtige und interessante Forschungsstudien (jedoch weitaus weniger praktisch bewährte Implementationen) hervorgebracht. Sie muß sich heute der Einsicht anpassen, daß eine Lernermodellierung im ursprünglich intendierten Umfang nicht realistisch ist (Lusti 1992). Zudem bleibt ihre Zielvorstellung auch für sich genommen unbestimmt: Pädagogen könnten sich glücklich schätzen, wenn klar wäre, was eigentlich einen idealen Lehrer ausmacht.

### *Lernerorientierung*

Dieses etwa zeitgleich entwickelte Paradigma stellt den Lernenden und seine Autonomie in den Vordergrund. Der allgegenwärtige Tutor wird aus dem Lerngeschehen verbannt und CAI als instruktionsfrei redefiniert. Diese Idee kommt bei vielen Pädagogen immer wieder

gut an, hätschelt sie doch das Wunschbild eines Schülers, der eigenständig seine Lernprozesse organisiert und selbstverantwortlich über die Inhaltsauswahl, die Bearbeitungstiefe und die Bewertung der Lernvorgänge entscheiden kann. Dabei erhält trotz aller Vielfalt der heute möglichen Darbietungsformen (Video- und Tonausgabe, Grafik, Animation, Hypertext etc.) ein ungeübter Lerner kaum mehr tutorielle Unterstützung als bei der Lektüre eines Buches. Dementsprechend prägt die Buchmetapher auch Entwicklungswerkzeuge (wie Hypercard oder Toolbook) und läßt die Blättermaschine aus den Pioniertagen des CAI in gewandelter Form wiederauferstehen.

### *CAI heute*

ITS und "lernerorientierte" Systeme sind unter anderem als Alternativen zu den frühen Formen des CAI entstanden, die zu einfach und stark von behavioristischen Ideen beeinflusst waren. Seit den 80er Jahren hat sich zunächst die technische Basis des CAI verbessert, dann aber auch der konzeptionelle Rahmen erweitert (Schreiber 1990). Für die heutige Situation sind kennzeichnend: die Verfügbarkeit solider Entwicklungsplattformen, das Vorliegen brauchbarer didaktischer Entwurfshilfen und die Verankerung in einer Vielzahl praktischer Anwendungen vor allem im Bereich der beruflichen Aus- und Weiterbildung. Ich möchte noch einen Schritt weitergehen mit der Überlegung, die neuen aus ITS und Hypermedia gewonnenen Methoden und Techniken in die CAI-Tradition einzubringen und mit dort Bewährtem zu verbinden. Auch auf Seiten der ITS-Forschung erkennt man zunehmend die Vorteile einer solchen Integration (Leutner 1992, Vassileva 1992). Die Synthese unterschiedlicher Paradigmen führt zu Mischformen, die in der Praxis allzu einseitige Ansätze ausgleichen und den Lerner aufgrund des größeren Handlungsfeldes besser motivieren können.

Zu mathematischen Themen wurden CAI-Projekte eher vereinzelt entwickelt. Sie zielen meist auf das Training von Rechenfertigkeiten oder sollen helfen, die begrifflichen Grundlagen und inneren logischen Beziehungen eines Teilgebietes zu festigen. In einer Lernzieltaxonomie sind dies die Stufen am unteren Ende, die natürlich in der Praxis auch ihr Recht haben. Zudem besteht bei dieser Art von Software die Chance, daß lokale Anforderungsprofile, die den Inhalt und die Lehr/Lernorganisation betreffen, in das Endprodukt einfließen. Beispiele hierfür sind GALOIS, ein Algebra-Einführungskurs für Informatiker an der Universität Mailand (Alberti et al. 1988) oder die an der Technischen Universität München entwickelten interaktiven Übungsprogramme zur angewandten Mathematik (Riedmüller 1990). Einen spezifischen, von Studenten gerne genutzten Vorteil bilden dabei die Möglichkeiten von Selbstkontrollen und Tests, die sich in CAI-Systemen gut implementieren lassen (vgl. auch Fey 1991).

## **2. Mathematische Lernbausteine im Grundstudium - Skizze eines Projekts**

Das nun zu umreißende Projekt knüpft an die revidierte und erweiterte Idee von CAI an. Als technische Basis dient die vor diesem Hintergrund konzipierte allgemeine tutorielle Schale DUAL, auf die ich in Abschnitt 3 näher eingehen werde. Im Ergebnis sollen Lernbausteine entstehen, die sich z.B. als Einstiegshilfe für Studienanfänger (in den ersten Semestern und/oder parallel zu Brückenkursen) eignen.

Auf der inhaltlichen Seite bieten sich daher für diese Zielsetzung vor allem Elemente aus folgenden Bereichen an:

- einzelne Themen zur Wiederholung der Schulmathematik, z.B. einfache geometrische oder kombinatorische Grundbegriffe,
- vorbereitendes begriffliches Rahmenwerk ("prerequisites") wie Mengensprache, Abbildungsterminologie etc.
- einfache Routinen und Rechenverfahren, z.B. Lösen von Gleichungen, ggT nach dem euklidischen Algorithmus,
- geeignete Themen der jeweils zugeordneten Veranstaltung, die wiederholt oder vertieft werden sollen.

Aus didaktischer Sicht sind die folgenden Leitgedanken hervorzuheben:

- Einbettbarkeit
- offene und modulare Struktur
- Einbindung externer Ressourcen
- umfassende Unterstützung von Autoren
- flexible Dialogschnittstelle

Die *Einbettbarkeit* von CAI-Bausteinen in eine vorhandene bzw. geplante Unterrichtsorganisation ist eine wesentliche Voraussetzung für ihre Wirksamkeit. Damit definiert sich CAI tendenziell als Ergänzung bzw. Begleitung bestehender Lernformen, eröffnet aber zugleich die Möglichkeit, diese zu restrukturieren und weiterzuentwickeln.

Das setzt eine in mehrfacher Hinsicht *offene und modulare Struktur* voraus. Modularität wird dabei realisiert: a) global durch die weitgehende Trennung von Form und Inhalt, und b) lokal durch Verwendung standardisierter Aufgabenformate. Die Framestruktur für Aufgaben und Informationsdarbietung ist dabei so allgemein gehalten, daß neben klassischen Seitentypen wie Auswahlen, Zuordnungen (vgl. Abb. 1), Lücken oder numerischen Eingaben grundsätzlich auch neue Formate entwickelt und eingesetzt werden können.

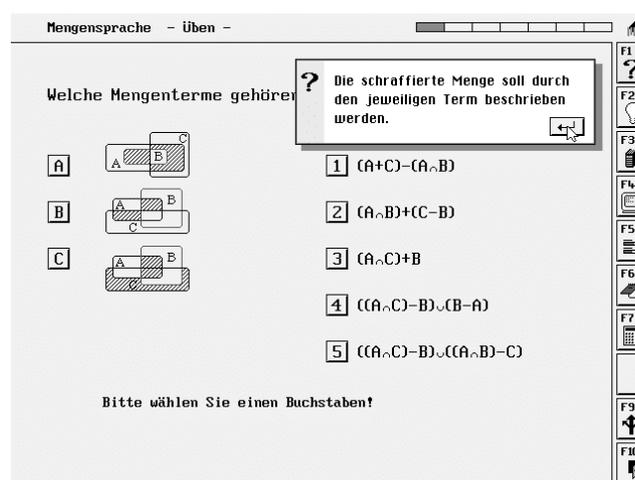


Abb. 1: Zuordnungsaufgabe aus der Mengenlehre mit geöffnetem Hilfenfenster

Letzteres geschieht jeweils abhängig von den Anforderungen in den behandelten Gebieten. Beispiele dieser Art sind freie Formen dynamischer Präsentation, numerische Aufgaben mit variabel generierten Parametern, oder ungebundene Probleme, bei denen Lerner ihr

Ergebnis selbstverantwortlich anhand von Musterlösungen bewerten können. Solange ein neuer CAI-Frametyp nur bestimmte (ziemlich allgemein gehaltene) Protokolle für den Datenaustausch einhält, kann er komplett als externes und separates Programm-Modul entwickelt werden (dieses übernimmt dann aber häufig auch unvermutet schwierige Lasten der Aufgabengenerierung und der Antwortanalyse, vgl.. Maurer et al. 1991). Es kann sich im Extremfall sogar um Problemstellungen oder Abschnitte aus realem Unterricht handeln, deren Bewertung das Programm anschließend von außen (z.B. über eine Betreuungsperson) aufnimmt.- Kurzum, durch das gegenüber klassischem CAI drastisch verallgemeinerte Framekonzept wird tutoriellen Anwendungen ein großes und bei weitem nicht ausgeschöpftes Potential erschlossen.

In engem Zusammenhang damit steht die *Einbindung externer Ressourcen*. In erster Linie handelt es sich dabei um vorhandenes Textmaterial (Skripte zu Vorlesungen, Aufgabensammlungen) sowie fachbezogene Werkzeuge. *Texte* lassen sich in geeigneten Fenstern zugänglich machen. Mit etwas mehr Arbeitsaufwand kann man sie in Fachglossare umbrechen und mit Hypertextreferenzen ausstatten. *Softwarewerkzeuge* dienen zur Visualisierung von Daten und Prozessen, zur geometrischen Konstruktion, numerischen Berechnung oder auch Modellbildung/Simulation. In letzter Zeit spielen, vor allem in der Ausbildung an Hochschulen, Computeralgebra-Systeme eine immer bedeutendere Rolle.

Streng genommen sind Werkzeug-Programme keine tutorielle Software, sie lassen sich aber systematisch als Lehrmittel gebrauchen (z.B. *Mathematica*: Brown, Porta, Uhl 1991) oder didaktisch anreichern (wie etwa in der *Mathcad Education Library* mit Hilfe eines speziellen Autorenwerkzeugs für Mathcad geschehen, vgl. Russell 1993). Wünschenswert (und möglich) ist eine in CAI integrierte Nutzung, die es gestattet, auf extern erzeugte Arbeitsergebnisse innerhalb einer CAI-Seite zurückzugreifen. Für DERIVE, das in letzter Zeit auch für den Mathematikunterricht entdeckt wurde (vgl. Köhler 1991, Kutzler 1995), ist eine solche Anbindung leicht zu verwirklichen.

Ein zentraler Aspekt der Projektentwicklung ist die *Mitarbeit von* (im allgemeinen mehreren) *Autoren*, die die Aufgaben, erläuternden Texte und Grafiken, Fachglossare und Skripte beisteuern. Ihrer umfassenden Unterstützung dient ein als Software-Anwendung realisiertes Beschreibungsmodell (DSKRIPT). Es erlaubt einem Autor, den Inhalt der zu erstellenden Aufgaben und sonstigen Bildschirmseiten auf einfache Weise in Datenbank- und Textdateien zu erfassen und im Prinzip unabhängig von der jeweiligen Plattform zu verwahren und zu pflegen (Schreiber 1994). Hinweise zum praktischen Vorgehen sowie didaktische Grundregeln für CAI werden über eine Dokumentation und, innerhalb der DSKRIPT-Software, gezielt über ein kontextabhängiges Hilfesystem vermittelt.

Auf dieser Basis lassen sich interessierte Studenten vorort in die Entwicklung von Aufgaben und in die Bearbeitung von Skripten etc. einbeziehen. Fortgeschrittenere Beiträge, z.B. die Entwicklung separater fachspezifischer CAI-Frames oder Studien zur Evaluation, können in didaktisch orientierten Examensarbeiten erarbeitet werden.

Als Bindeglied zwischen Lerner und Lehrsystem dient eine *flexible Dialogschnittstelle*. Sie bietet eine Reihe globaler Steuer- und Zusatzfunktionen, die dem Teilnehmer (Lerner) eigene Initiativen im Unterrichtsgeschehen ermöglichen. Zu diesen Möglichkeiten gehören unter anderem:

- *Bewegungen*  
Seitenanwahl und Rucksprung bei der Informationsdarbietung; Rückverfolgung innerhalb von Hypertexten; Themenwahl; Abbruch;
- *Arbeitshilfen*  
persönliche Notizablage zu einzelnen Unterrichtseinheiten; Erstellung bzw. Abruf seitenbezogener Notizen, die allen Programmteilnehmern zugänglich sind; Taschenrechner mit Ergebnisübertragung in numerische Eingaben; Einbindung externer Anwendungen;
- *Abruf von Lernstandsangaben und Ortsanzeigen*
- *diverse Informationen*  
allgemeine und seitenbezogene Hilfen; gestufter Abruf von Lösungen; Fachglossar;
- *Wahl verschiedener Arbeitsformen* in Bezug auf denselben Lehrstoff

Das Programm macht den in den Datenbeständen abgelegten Inhalt auf unterschiedliche Weise zugänglich. In einem Themenmenü können die Lerner das Sachgebiet auswählen oder in einem individuellen Mix zusammenstellen. Anschließend wird die Zugangsweise (Arbeitsform) über *Bearbeitungsoptionen* definiert. Dabei stehen zur Auswahl: reine Informationsdarbietung ("Präsentation") sowie die interaktiven Modi "Üben" (mit entsprechend strukturiertem "Kleinstunterricht"), "Auffrischen" und "Testen". Jede Arbeitsform kann neu begonnen oder fortgesetzt werden. Abhängig vom gewählten Thema und von der gewählten Arbeitsform aktiviert das Programm eine adaptive Lehrstrategie, die auf der Item-Ebene arbeitet und dabei individuelle Lernerprofile erzeugt.

Die soweit geschilderten didaktischen Leitgedanken sollten deutlich machen: Eine directive Steuerung oder gar Gängelung des Lerners durch das Programm wird vermieden, ebenso aber auch der Verzicht auf Instruktion und Lernerfolgskontrollen. Vielmehr können Studenten entscheiden, ob und in welchen Bereichen sie sich der Kontrolle des Tutors unterziehen.

### 3. Das tutorielle Rahmensystem DUAL

Um die geschilderten Erfordernisse zu realisieren, benötigt man ein allgemeines Rahmensystem, das schon "von Haus aus" die Fähigkeiten eines Tutors mitbringt. Das System DUAL (Didaktik-Umgebung für Addaptive Lernprogramme, in der vorliegenden Form seit 1991 zusammen mit T. Merino entwickelt) ist aus der hier dargestellten erweiterten CAI-Konzeption hervorgegangen. Es handelt sich um eine auf einem Standard-PC lauffähige Systemschale, die Inhalt und Ablauflogik (also auch die Unterrichtsmethoden) einer CAI-Anwendung weitgehend separiert und sich mit externen Inhaltsdaten beschicken läßt. Seiner Kennzeichnung als Didaktik-Umgebung entsprechend enthält DUAL ein formales Gerüst zur Einbindung der erforderlichen Unterrichtsbausteine, das sind diverse Lehrstrategien, die ihnen zugeordneten Lernerprofile sowie ein prinzipiell beliebig erweiterbares Repertoire lokaler Interaktionsformen (CAI-Frames) (Schreiber 1992).

In einer Reihe von Projekten der betrieblichen Aus- und Weiterbildung hat sich das System inzwischen praktisch bewährt; als ein Beispiel dieser Kategorie mit vergleichsweise ausführlichem mathematischen Hintergrund sei hier ein Lernprogramm über die Risikoanalyse festverzinslicher Wertpapiere erwähnt (Eller et al. 1994).

Unterrichtsprozesse sind typischerweise zyklisch (sofern der Lehrer das, was der Lernende zum jeweiligen Thema leistet, auswertet und für seine weiteren Initiativen berücksichtigt). Das formale Schema von DUAL wird dem auf zwei Ebenen gerecht:

- *lokal* durch die einzelnen Unterrichtselemente (z.B. Aufgaben), die ein Mikrotutor als Dialog abwickelt;
- *global* durch die Auswahl, Sequenzierung und Verwaltung der Unterrichtselemente über einen Makrotutor.

Der *Mikrotutor* sorgt für die Ausführung der einzelnen Unterrichtsschritte. Er verwendet dabei detailliert ausgearbeitete Interaktionsformate für die CAI-Seitentypen, (z.B. verschiedene Auswahl-, Zuordnungs- und Textkonstruktionstypen, aber auch freiere Formate wie generative Frames, Simulationen, Fremdprogramm-Aufrufe), die sämtlich einem allgemeinem Schema ("Kleinstunterricht" nach Eckel 1989) gehorchen. Ein dem Mikrotutor benannter Frame gibt diesem seinen Typ zu erkennen und verweist gegebenenfalls auf zusätzliche externe Daten (Text, Grafik, Ton, Video), Programme oder auch Subframes, die zu seiner vollständigen Ausführung benötigt werden.

Der *Makrotutor* ist die Instanz, die über den jeweils nächsten Unterrichtsschritt entscheidet, d.h. er benennt dem Mikrotutor den zur Ausführung ausgewählten Frame. Für den Makrotutor sind Frames lediglich abgeschlossene (undurchsichtige) Items, die er über eine interne Liste verwaltet. Dabei realisiert er ein *adaptives Verhalten*, abhängig von der verwendeten Lehrstrategie und dem ihr zugeordneten Lernerprofil.

Für die Konfigurierung der Lehrstrategien bietet DUAL geeigneten Spielraum. Bisher wurden Sequenzierungsverfahren implementiert, die sich vornehmlich für Übungsprogramme eignen. Die Verfahren sind stets in eine Modellfamilie eingebettet. Als solche gelten z.B. Lernkarteien, Zustandsmodelle sowie entscheidungstheoretisch optimierende Methoden. Diese Bausteine sind über eine eigene Makrosprache ansprechbar und liefern so jeweils eine spezielle Strategie. Beispielsweise kann man dem Tutor via Strategie mitteilen, in welchem Zustand ein Item als "gelernt" ausscheiden soll, wie ein Zustand bei auftretendem Fehler zu verändern ist, oder welchen Einfluß Zustände auf die Reihenfolge im ganzen haben. Über diese Parameter ist es schließlich auch möglich, die Wirkung verschiedener Strategien derselben Modellfamilie empirisch zu vergleichen.

#### 4. Zwei zentrale Probleme

Abschließend möchte ich zwei Problemkreise ansprechen, die für die Anwendungspraxis und die CAI-Grundlagenforschung beide zentrale Bedeutung besitzen.

Das erste ist das Problem der *Bereichsspezifität*. Üblicherweise wird auch heute noch die didaktische Struktur eines Lernprogramms vom Autor "handgestrickt". Ein Blick in die Praxis zeigt, daß bei den CAI-Entwicklern meist einfache Rezepte Anwendung finden, weitaus häufiger als adaptive Methoden, die diesen Namen wirklich verdienen. Der Einsatz allgemeiner Rahmenstrukturen kann diese Schwäche ausgleichen, aber eben um den Preis der Abstraktion vom Inhalt der verwalteten Items. Wie universell muß eine tutorielle Schale *mindestens* sein und wie bereichsspezifisch kann sie dabei dann *höchstens noch* sein? Vieles spricht für die Gültigkeit der Regel, daß mit zunehmender Allgemeinheit die

Anwendungsmöglichkeiten fachdidaktischer Erkenntnisse und Prinzipien auf der Makro-Ebene schwinden. Das gilt für Wissensgebiete mit einer semantischen Tiefenstruktur, wie sie die Mathematik aufweist, in ganz besonderem Maße. Daher stellt sich die Frage nach einem geeigneten Bindeglied zwischen Makro- und Mikro-Ebene, mit dem sich eine höhere Fachspezifität unter gleichzeitiger Beibehaltung effizienter allgemeiner Strukturen erzielen läßt.

Der zweite Punkt betrifft vor allem Auftraggeber und Fachautoren von CAI-Produkten. Es ist kaum anzunehmen, daß sich die durchschnittliche Qualität von CAI verbessern wird, wenn nicht in deutlich höherem Maße als bisher die *Probleme der Portierung* gelöst werden. Wer projiziert (und finanziert) gerne ein aufwendiges Programm, von dem niemand mit Bestimmtheit sagen kann, ob es auf der Hardware und mit der Betriebssoftware von morgen noch lauffähig ist? Welcher qualifizierte Fachautor entscheidet sich angesichts der Alternative "Buch oder Computer" für das zweite, wenn sein Werk damit eine voraussichtliche Lebensdauer von 3 bis 5 Jahren kaum überschreiten dürfte? Es ist daher ökonomisch wie sachlich zwingend, daß die für eine CAI-Anwendung grundlegenden Daten (Objekte) eine Existenzform erlangen, die ihr "Überleben" bei allem Wechsel der technischen Rahmenbedingungen sichert. Die derzeit vielfach geübte Entwicklerpraxis wird dem nicht gerecht. Proprietäre Strukturen beherrschen hier seit Jahrzehnten das Feld, auch bei den inzwischen verfügbaren plattformübergreifenden Autorensystemen. Die Zeit scheint gleichwohl reif für andere Lösungen.

## Literatur

Bonnet, A. / Cordier, M.-O. / Kayser, D.: An ICAI System for Teaching Derivatives in Mathematics. In: R. Lewis & D. Tagg (eds.): Computers in Education. North-Holland Publishing Company: IFIP, 1981, S. 135-141.

Brown, D./ Porta, H. / Uhl, J.: Calculus & Mathematica, Part I. Addison-Wesley: Reading (Mass.), 1991.

Dörfler, W. et al. (Hrsg): Computer - Mensch - Mathematik. Beiträge zum 6. Internationalen Symposium für "Didaktik der Mathematik" in Klagenfurt vom 23.-27.9.1990. Hölder-Pichler-Tempski: Wien, Teubner: Stuttgart, 1991.

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

Eller, R. / Merino, T. / Schreiber, A.: Bond Research und Risikomanagement. Moderne Analyse von Zinsinstrumenten (Lernsoftware). Vereinigung für Bankberufsbildung e.V.: Frankfurt am Main, 1994, ATLAS Wirtschaftsverlag: Würzburg, 1994.

Fey, J.: Computers in US Mathematics Education: Recent Research and Development Results. In: (Dörfler et al. 1991), S. 77-101.

Kearsley, G. (ed.): Artificial Intelligence and Instruction. Applications and Methods. Addison-Wesley: Reading (Mass.), 1987.

Köhler, R.: Bericht über einen Unterrichtsversuch zum Einsatz von DERIVE im Analysisunterricht. In: (Dörfler et al. 1991), S. 151-158.

Kutzler, B.: Mathematik unterrichten mit DERIVE. Ein Leitfaden für Lehrer. Addison-Wesley: Bonn, Paris [u.a.], 1995.

Leutner, D.: Adaptive Lehrsysteme. Instruktionspsychologische Grundlagen und experimentelle Analysen. Psychologie-Verlags-Union: Weinheim, 1992.

Lusti, M.: Intelligente tutorielle Systeme. Einführung in wissensbasierte Lernsysteme. Oldenbourg Verlag: München Wien, 1992.

Maurer, H. / Stone, M. G. / Stubenrauch, R. / Gillard, P.: Question/Answer Specification in CAL Tutorials (Automatic Problem Generation Does Not Work). In: (Dörfler et al. 1991), S. 191-197.

Riedmüller, B.: Interaktive Übungsprogramme zur Behandlung von Algorithmen. Entwurf, Entwicklung und Einsatz. Diss. Augsburg 1990.

Russell, H.: Mathcad Education Library: Algebra II. MathSoft, Inc.: Cambridge, 1993.

Schreiber, A.: Entwicklung didaktischer Software auf Autorensystembasis. In: Zentralblatt für Didaktik der Mathematik (ZDM) 1990/H.3, S. 92-104.

Schreiber, A.: Bausteine für Lernprogramme: Beschreibung und Implementierung. In: H.-J. Friemel u.a. (Hrsg.), Forum '90 - Wissenschaft und Technik, Neue Anwendungen mit Hilfe aktueller Computer-Technologien, Trier, 8./9. Oktober 1990. Proceedings, Springer Verlag, Berlin Heidelberg: 1990, S. 333-348.

Schreiber, A.: Eine Didaktik-Umgebung für adaptives Lernen (DUAL). In: Grundlagenstudien aus Kybernetik und Geisteswissenschaft, 33/1 (1992), S. 25-31.

Schreiber, A.: DSKRIPT - Plattformübergreifende Entwicklung von CBT. Modell und praktischer Leitfaden für Fachautoren. BU Flensburg, 1994.

Sleeman, D. / Brown, J.S. (eds.): Intelligent Tutoring Systems. Academic Press Inc., London: 1982.

Vassileva, J.: Dynamic CAL-Courseware Generation within an ITS-Shell Architecture. In: Tomek, I. (ed.): Computer Assisted Learning, 4th International Conference, ICCAL '92 Wolfville, Nova Scotia, Canada, Jun. 1992, Proceedings. Springer Verlag: Berlin Heidelberg, 1992, S. 581-591.