

Auszug aus:

**B. Koerber u. I.-R. Peters**

# **Informatische Bildung in Deutschland**

**Perspektiven für das 21. Jahrhundert**

Berlin 1998, LOG IN Verlag. S. 127-155

Peter Gorny

## **Didaktisches Design telematik-gestützter Lernsoftware**

### **1. Vom Rechner zum Kommunikator**

Informations- und Kommunikationstechniken wachsen in immer größeren Maße zusammen: Von dieser Hypothese ausgehend werde ich die Konsequenzen für das Lehren und Lernen aufzeigen, wenn sich der »Rechner« zum »Kommunikator« wandelt. In diesem Beitrag erläutere ich die erwarteten oder schon eingetretenen Änderungen, bewerte sie an Hand von Kriterien der menschlichen Kommunikationsbedürfnisse, bringe sie in methodische Kategorien und ziehe daraus Schlüsse für das Design von Unterrichtssoftware.

Einer der wichtigsten Unterschiede zwischen telematik-gestützter und »normaler« Unterrichtssoftware ist der klare Vorteil der Unterstützung von Kommunikation mit dem Lehrer – und manchmal mit anderen Lernern, während herkömmliche Unterrichtssoftware nur eine asynchrone Kommunikation mit dem Programmentwickler darstellt.

Es ist hier nicht der Platz, einen vollständigen Überblick über alle Effekte der Telematik auf alle Formen von Lehren und Lernen zu geben. Obwohl die obige These weit gefaßt ist, werde ich keine Langzeitprognosen und Zukunftsvisionen abgeben, sondern mich auf die Möglichkeiten konzentrieren, die schon heute oder in Kürze an Schulen und Universitäten zur Verfügung stehen werden. Ich werde also nichts z.B. über Videokonferenzen, Breitbandkanäle oder voll multimediale Netzwerkanwendungen auf Hochgeschwindigkeitsleitungen sagen, da diese Techniken wegen des erforderlichen großen Aufwands nur in Sondersituationen einsetzbar sind.

Ein weiterer Punkt ist der Entwicklungsaufwand, der für Lernsoftware betrieben werden muß: Gerade die Software, die den Lernern ohne Lehreraufsicht in die Hand gegeben werden soll, muß inhaltlich und technisch so perfekt sein, daß Schwächen des Produkts vom Lerner allein überwunden werden können. Daher resultiert die Forderung: Das didaktische Design der Lernsoftware aller Kategorien, die in unserem Zusammenhang diskutiert werden, muß den Kriterien der

Softwareentwicklung ebenso genügen wie den pädagogischen Kriterien, die auch für anderes Unterrichtsmaterial gelten.

## **2. Didaktische Überlegungen zum Lehren und Lernen mit Unterrichtssoftware**

### **Didaktische Paradigma**

Zur Zeit gibt es nur rudimentäre Methodiken zum Einsatz von Unterrichtssoftware beim Lehren an Schulen und Universitäten. Während die Diskussion an Schulen einige Fortschritte erzielt hat, muß angemerkt werden, daß medienorientierte Methodik praktisch kein Thema unter akademischen Lehrkräften ist. Stattdessen scheint ein breiter Konsens über einige Prinzipien zu herrschen, die Thissen so zusammenfaßt (Thissen, 1997):

»[...]

- Der Lernstoff ist grundsätzlich vermittelbar.
- Der Lehrer / Experte weiß, was der Lerner in Zukunft wissen und deshalb lernen soll. Er weiß, was der Lerner braucht.
- Der Lehrer kennt in etwa den Lernprozeß des Lerners und kann ihn steuern.
- Es gibt eine optimale Stoffvermittlung.
- Wissen läßt sich mit Hilfe der Sprache (Schriftsprache / Bildsprache) vom Lehrer auf den Lerner übertragen.
- Aufgabe des Lehrers ist es, Antworten zu geben.
- Aufgabe des Schülers ist es, den Lernstoff mehr oder weniger passiv aufzunehmen und in seinem Gedächtnis abzuspeichern. Auf diese Weise eignet er sich das Wissen des Lehrers nach und nach an.

[...]«

Thissen kennzeichnet damit die Prämissen der »Nürnberger Trichter-Didaktik«, die er auch »Mythen über das Lehren und Lernen« nennt. Stattdessen plädiert er für eine didaktische Umsetzung eines »konstruktivistischen Lernmodells« (vgl. Grudin, 1993), das aus früheren kognitionspsychologischen Modellen entwickelt wurde, um die

Einflußfaktoren, die die Lernbereitschaft kennzeichnen, einbeziehen zu können. Seine Schlußfolgerungen sind:

»[...]

- Lernen ist ein aktiver Prozeß der Wissenskonstruktion [...] immer nur in Verbindung mit bereits vorhandenem Wissen. [...] Der Lerner muß beim Wissensaufbau aktiv sein, er muß Fragen stellen und sich mit dem angebotenen Material auf seine Weise beschäftigen können. [...]
- Lernen ist eine individuelle Konstruktion eines menschlichen Geistes. Aus diesem Grund gibt es so viele eigene und unvorhersehbare Lernwege wie es Lerner gibt.
- Wissen ist nicht vermittelbar. Dem Lehrer ist es unmöglich, seine Kenntnisse dem Lerner direkt weiterzugeben. Vielmehr hilft er dem Lerner durch sein Tun, durch Hinweise, Fragen und Informationen, selbst Wissen zu konstruieren. [...]
- Es kommt zunächst einmal darauf an, die richtigen Fragen im Lerner zu wecken. Erst wenn echte Fragen im Lerner geweckt sind, setzt sich der Lernprozeß von selbst in Gang. [...]
- Lernen heißt, mentale, kognitive Landkarten zu konstruieren, die immer mehr detailliert und verfeinert werden.

[...]«

### **Einflußfaktoren auf den Entwurf von Lehr/Lernprozessen**

In einem groben Modell (nach Meyer, 1991) können wir die wichtigsten Faktoren, die den Entwurf von Lehr- und Lernprozessen beeinflussen, identifizieren (s. Abbildung 1): Lernziele, Lerninhalte, Unterrichtsmethoden und organisatorische Rahmenbedingungen.

In unserer Diskussion werden wir dieses Modell dazu benutzen, den Entwurfsprozess zu strukturieren, weil es leicht mit den Vorgehensweisen der Entwicklung von Unterrichtssoftware kombiniert werden kann.

Für unsere Zwecke kategorisieren wir weiterhin Unterrichtssoftware auf einer Skala (Abbildung 2) vom selbstorganisierten offenen Lernen bis zum lehrerzentrierten Unterricht, der das Lernen praktisch automatisch zur Folge hat.

Peter Gorny

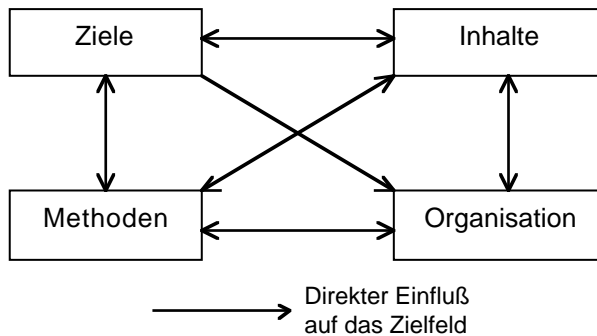


Abbildung 1: Einflußfaktoren beim Pädagogischen Entwurf von Unterricht und Unterrichtsmaterialien (nach H. Meyer, 1991)

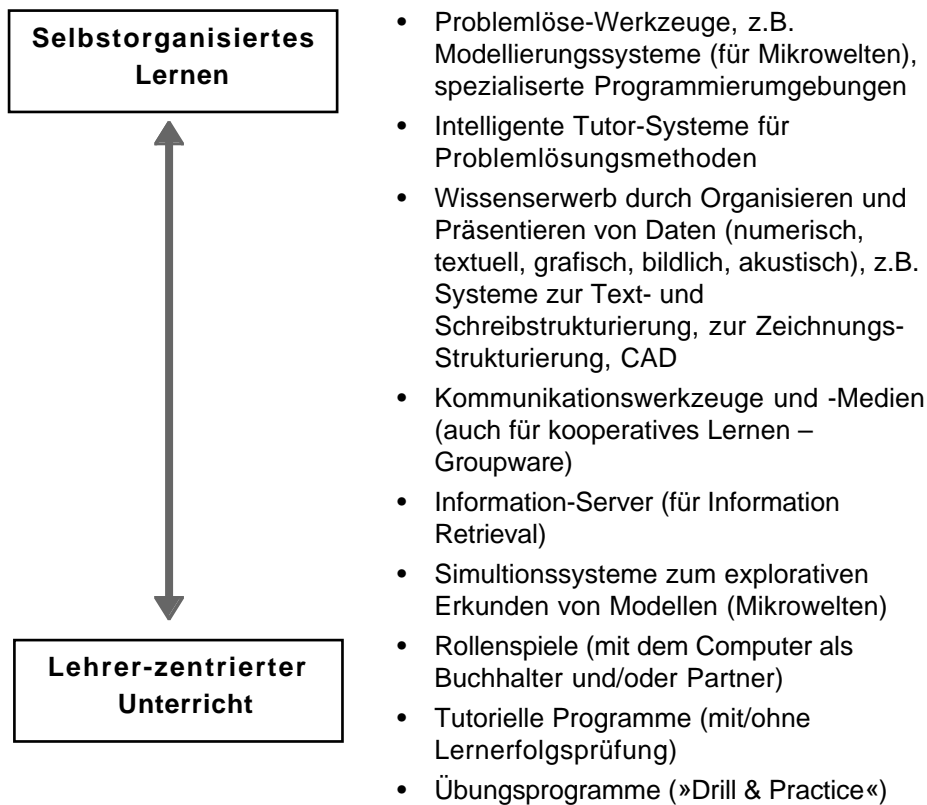


Abbildung 2: Kategorien von Unterrichts- und Lernsoftware, geordnet nach den Möglichkeiten zum selbstorganisierten Lernen bzw. vom Lehrer inhaltlich und organisatorisch bestimmtem Lernen (»lehrer-zentrierten Lernen«)

### **Zum Aufbau dieses Beitrags**

Im nächsten Kapitel »Technische Möglichkeiten des Internet« werden wir die Möglichkeiten der Telematik untersuchen: erst zu Kommunikationszwecken im Unterricht, dann unter den besonderen Bedingungen in Lernsituationen. Das 4. Kapitel »Szenarien für computerunterstützte Kommunikation im Unterricht« ist der Beschreibung einiger Unterrichtsszenarien gewidmet. Die Methoden der Softwareergonomie für die Entwicklung von benutzergerechter Software diskutieren wir in Kapitel »Softwareergonomie – State of the Art der Entwicklung von Benutzungsoberflächen für Büroanwendungen«. Diese Methoden werden mit didaktischen Anforderungen kombiniert, so daß wir einige Hinweise für den Entwurf von Lernsoftware in Kapitel 6 »Verknüpfung didaktischer und softwareergonomischer Methoden für das Software-design« entwickeln können, die an einem Beispiel verdeutlicht werden sollen.

## **3. Technische Möglichkeiten des Internet**

### **Die gegenwärtige Internet-Anbindung deutscher Schulen und Hochschulen**

Während es bei deutschen Lehrern noch nicht allzu verbreitet ist, privat einen Computer zu besitzen, hatte Umfragen zufolge bereits 1993 jeder zweite Schüler der 7. Klasse in Deutschland zu Hause Zugang zu einem Computer, und insgesamt 70 % aller Schüler haben irgend eine Art von Computerzugang (Rohe-Krebeck, 1993). Der Prozentsatz von computernutzenden Lehrern in Schulen der Sekundarstufe II ist höher, aber immer noch weit entfernt von 100 %. Schätzungen zufolge hat jede Sekundarschule in Deutschland mindestens einen Computerraum mit im Mittel 12 PCs, die meist lokal vernetzt sind. Während es eine 100-%ige Internetanbindung von Universitäten und Fachhochschulen gibt (obwohl deshalb aber längst noch nicht alle Mitarbeiter einen direkten Internetzugang haben), gibt es nach den Unterlagen die im Deutschen Bildungsserver (DBS, 1997) gespeichert sind, nur ein paar tausend deutsche Schulen mit direkten oder indirekten Anschlüssen an WANs (Wide Area Networks).

### Technische Möglichkeiten für Kommunikationsprozesse

Die technischen Kommunikationsmöglichkeiten beeinflussen in grossem Maße die organisatorischen Möglichkeiten und damit auch die Inhalte und Methoden des Lehrens (siehe Abbildung 1). Das Internet bietet Kommunikationsmöglichkeiten, die in den Metaphern traditioneller Kommunikationsformen wie Briefwechsel (E-Mail), Anschlagbretter (Bulletin Boards, Newsgroups) und Büchereien (Information Server, Informationssysteme) beschrieben werden können.

Heute relevante technische Plattformen für asynchrone Kommunikationsprozesse sind:

- Electronic Mail (Transferprotokoll: smtp)
- Newsgroups (Transferprotokoll: nntp)
- WorldWideWeb – verteiltes hypertextuell verknüpftes Multimedia-system (Transferprotokoll: http) mit den Eigenschaften:
  - à interaktiver Zugriff und Navigation
  - à internet-basiert
  - à multimedial
  - à Hyperlinks zwischen den Dokumenten (Zeiger auf Positionen in Dokumenten)
- HyperWave (Hyper-G) – Erweiterung der WWW-Funktionalität durch
  - à verbesserte Navigation
  - à interaktive Anmerkungen zu Dokumenten durch die Benutzer
- Erweiterung des WWW durch Java-Applets (und verschiedene »plug-ins«) mit den Möglichkeiten:
  - à automatisches Downloading von Java-Programmen
  - à interaktive Benutzung von Java-Programmen

Weitere Möglichkeiten des Internets, auf die hier nicht weiter eingegangen wird, sind:

- FTP (Datentransfer von/auf andere Computer über das Internet)
- Chat (online-Austausch von kurzen Texten)
- Audio- / Videokonferenzen, Tele-Vorlesungen

(Während alle anderen Formen von den normalen Internetprotokollen unterstützt werden, benötigen Audio/Videokonferenzen spezielle Transferprotokolle, die ununterbrochenen Datentransfer gewährleisten.)

### »Virtuelle« Briefe, Anschlagbretter und Bibliotheken

Mit den obengenannten technischen Systemen können die Funktionen der asynchronen Kommunikation mit Text- und Bilddarstellung sowie Tonausgabe betrieben werden:

- Funktion »*Virtuelle Briefe*« (E-Mail)

Diese Funktion erlaubt das Versenden einer Nachricht an einen Empfänger, die dort mit einer gewissen Zeitverzögerung eintrifft (d.h. Sekunden, Minuten oder Stunden später). Eine derartige textuelle Kommunikation basiert auf Zeichencodes wie dem ASCII-Code. Ein direktes Versenden von Bild- oder Audiodaten ist nicht möglich; vielmehr müssen diese erst in eine (unlesbare) ASCII-Zeichenfolge codiert werden, die dann wie eine normale Textdatei verschickt werden kann. Viele Mailprogramme erlauben es, solche Nicht-Textdateien (codiert und komprimiert) an eine Mail anzuhängen, so daß es dem Empfänger möglich ist, nach Dekomprimierung und -kodierung die Datei normal zu benutzen. (Diese Technik wird auch eingesetzt, um Textdateien mit Zeichen zu verschicken, die nicht im Standard-ASCII-Code vorgesehen sind, z.B. nationale Sonderzeichen.)

Der Empfänger einer Nachricht muß nicht durch eine Einzeladresse bestimmt werden, es kann auch die Adresse eines »Listservers« sein, der sie dann an alle Adressen einer Verteilerliste weiterreicht.

- Funktion »*Virtuelle Anschlagbretter*« (Newsgroups, WWW)

In Analogie zum Aushang an einem Anschlagbrett wurden die »Postings« in Newsgroups eingeführt, weil der Benutzer eine Nachricht erst durch Abholen vom Brett (Downloading vom Server) erhält – im Gegensatz zu Briefen, die auch unerwünscht im Briefkasten landen können. Die Analogie gilt auch für die einfachen Anwendungen des WorldWideWeb, bei denen zusätzlich nicht-textuelle Nachrichten ausgetauscht werden können. Für unsere Zwecke genügt es, folgende Aktivitäten zu unterscheiden:

- à Lesen und Schreiben von Textnachrichten, die als »Aushänge« an elektronischen Anschlagbrettern veröffentlicht werden;
- à Lesen und Gestalten einfacher Hypermedia-Dokumente (in der Beschreibungssprache HTML), die im WWW veröffentlicht werden.

In beiden Fällen kann der Zugang auf geschlossene Gruppen regional beschränkt werden.



- Funktion »*Virtuelle Büchereien*« (WWW, Datenbanken)  
Zugriff auf Multimediadateien, die auf einem WWW-Server oder in einer Datenbank veröffentlicht worden sind. Die Dateien können auf Anforderung des Benutzers zu einem anderen Computer im Netz transferiert werden, so daß der Empfänger die Texte, Grafiken oder Filme betrachten oder sich die Audiodateien anhören kann. Zusätzlich können diese Dateien Verweise auf beliebige Stellen in anderen lesbaren Dateien haben, so daß sie ein System von hypermedialen Dokumenten bilden.  
Die großen Bibliotheksdatenbanken und das WWW bieten derartige Funktionen in einer bisher nicht erlebten Perfektion für riesige Informationsmengen an. Für die Schule sind allerdings hauptsächlich nur die Teilbereiche anwendbar, die den herkömmlichen Nachschlagewerken und Enzyklopädien entsprechen. Da die Lehrer und Schüler diese Dienste nur gelegentlich benötigen, sind sie auf benutzerfreundliche Gestaltung der Navigationsdialoge und semantisch angemessene hypermediale Aufbereitung der Dokumente angewiesen – eine Anforderung, die heute erst selten erfüllt wird.

Es ist nicht mein Ziel, hier die technischen Details der Transferprozesse, der Erstellung von eigenen WWW-Dokumenten, oder das Downloading von Dokumenten auf den eigenen WWW-Browser zu beschreiben. Stattdessen werde ich die oben skizzierten Funktionen für die Möglichkeiten der asynchronen Kommunikation in Lehr/Lernsituationen näher untersuchen. Kapitel 4 enthält eine Auswahl an Szenarien.

Die einfachste Form der Unterrichtsorganisation ist die der geschlossenen Schülergruppe auf der Basis einer homogenen technischen Plattform (z.B. ein homogenes lokales Netzwerk mit PCs), doch sobald räumliche Verteilung ins Spiel kommt und die Anzahl der Teilnehmer sich erhöht, kann eine solche Homogenität kaum aufrecht erhalten werden. Wenn z.B. einer der Kommunikationspartner in einem anderen Netz als dem Internet arbeitet (z.B. in einem der kommerziellen Systeme wie CompuServe oder AOL oder in einem der privaten nicht-kommerziellen Systeme wie FidoNet oder ComLink), werden einige der genannten Funktionen möglicherweise nicht verfügbar sein. Die Kommunikationspartner werden sie durch andere technischen Formen – meist unter Reduktion des Benutzungskomforts – ersetzen müssen.

Als ein Beispiel für die Darstellung eines Ersetzungsprozesses nehmen wir die technische Umsetzung von Gruppendiskussionen: Diese Art der Kommunikation kann in geschlossenen Benutzergruppen am einfachsten auf der Basis von Newsgroups realisiert werden: wenn eine Person einen Text zu einer Gruppendiskussion beitragen will, wählt sie ein Brett mit einem passenden Themenbereich und gibt die Nachricht dort ab. Interessierte Benutzer können dann auf das Brett zugreifen (genauer: die Kopfzeilen aller (neuen) Nachrichten lesen, um dann die interessierenden auf ihren Rechner zu holen). Die Initiative der Informationsbeschaffung liegt voll in der Hand des Benutzers (der, nebenbei bemerkt, so auch seine Kommunikationskosten steuern kann).

Eine Newsgroup kann für alle offen sein oder auf eine bestimmte Benutzergruppe beschränkt werden; das Anbringen einer Nachricht kann durch einen Moderator gesteuert sein: er kann Nachrichten ablehnen, kommentieren oder kürzen. Gegenwärtig existieren zehntausende von Newsgroups im Internet, von denen allerdings viele nur regionale Bedeutung haben. Für die Schule relevante Newsgroups sind z.B. mit den Domains `k12.*` (z.B. `k12.lang.francais`), `school.*` und `schule.*` gekennzeichnet.

Wenn die Computersysteme einer Gruppe im wesentlichen homogen sind, kann sog. »Groupware« genutzt werden. Normalerweise wird dies in einheitlich organisierten Unternehmen so gehandhabt. Es gibt jedoch auch einige Groupware-Systeme, die in inhomogenen Computernetzen installiert werden können. Die gegenwärtigen Bemühungen, einen einheitlichen Standard für die Funktionalität solcher Software festzulegen (ISO93), soll auch hier die Interoperationalität sicherstellen. Für Schulen ist solche spezielle Software für Gruppenkommunikation allerdings noch nicht verfügbar.

#### **4. Szenarien für computergestützte Kommunikation im Unterricht**

In diesem Kapitel werden einige ausgewählte Szenarien präsentiert, die sich auf Berichte von Schulen stützen, in denen die Telematik in Lerneinheiten integriert wurde. Da dieser Artikel das didaktische Design der Software behandelt, werde ich für die Szenarien nur die Punkte herausgreifen, die die speziellen Anforderungen an die Software als Kommunikationsmittel im Unterricht und als Lernmaterial gestellt werden.

### **Szenario 1: »Virtuelles« Klassenzimmer, »virtuelles« Seminar**

*Kommunikationsattribute:*

- Partner: ein Lehrer, mehrere Schüler;
- Initiative: Lehrer;
- Hauptziel: Wissenstransfer;
- Typischer Anwendungsbereich: Teleteaching.

*Softwarekategorien:*

- Standardkommunikations- und -präsentations-Software, spezielle Groupware.

Der normale Unterricht an einer Schule ist so organisiert, daß in einer Klasse eine Gruppe von Schülern mit einem Lehrer an einem Unterrichtsgegenstand arbeitet. Diese Situation wird im »virtuellen Klassenzimmer« reproduziert, wobei die Notwendigkeit entfällt, die Teilnehmer an einem bestimmten Ort oder zu einer bestimmten Zeit zusammenzuführen. Daraus resultiert eine veränderte Lehrerrolle:

- der Wissenstransfer vom Lehrer zu den Schülern ist weiterhin möglich, aber durch die Asynchronität der Kommunikation verlangsamt, so daß anderes computergestütztes Unterrichtsmaterial eingesetzt werden muß, um den Wissenserwerb zu verbessern;
- die Rolle des Lehrers als Berater, Mentor und Tutor wird verstärkt; die Schüler können ihn einfacher direkt befragen, ohne die allgemeine Gruppendiskussion zu unterbrechen; der Lehrer kann eine solche Frage entweder an die ganze Gruppe weitergeben oder nur dem fragenden Schüler antworten.

### **Szenario 2: Kooperation zwischen »virtuellen« Klassenzimmern**

*Kommunikationsattribute:*

- Partner: verschiedene Lehrer, verschiedene Lerngruppen;
- Initiative: Lehrer;
- Hauptziel: soziale Kontakte; Meinungen zu verschiedenen Kulturen entwickeln;
- Typischer Anwendungsbereich: Klassenpartnerschaften.

*Softwarekategorien:*

- Standardkommunikations-Software und spezielle Groupware.

Wenn das erste Szenario (virtuelles Klassenzimmer, virtuelles Seminar) über eine Schule hinaus eingesetzt wird, erreicht es eine neue Dimension: von den Lehrern angeleitet, arbeiten Schüler in verschiedenen geographisch auseinander liegenden Schulen an einem speziellen Thema. Der Wissenstransfer steht auch hier im Vordergrund; es kann erwartet werden, daß die Vorkenntnisse der Schüler und die didaktische Methodik der beteiligten Lehrer sehr unterschiedlich sind, so daß mit zusätzlichen Hindernissen zu rechnen ist.

Beispiele für dieses Szenario können in den Fächern Geographie, Ökologie, Mathematik und Biologie gefunden werden. Sehr oft wird auch der Fremdsprachenunterricht mit integriert, wenn die Schulen in verschiedenen Kulturkreisen liegen (Gorny/Sarnow, 1993).

**Szenario 3: »Virtuelle« Projektgruppen**

*Kommunikationsattribute:*

- Partner: ein Moderator oder mehrere, viele Teilnehmer;
- Initiative: alle;
- Hauptziel: kooperatives Lösen eines Problems, Entwicklung von kooperativen Fertigkeiten;
- Typischer Anwendungsbereich: Projektunterricht;

*Softwarekategorien:*

- Spezial- (oder Standard-) Groupware.

Der Schritt von den beiden ersten Szenarien hin zu projektorientiertem Lernen ist nur sehr klein. Telematik-basierte Projektgruppen können auf der Basis von Newsgroups oder mit spezieller Groupware organisiert werden. Der Frontalunterricht des traditionellen Klassenzimmers wird durch die zusätzlichen Kommunikationsmöglichkeiten zwischen Lehrer und Schülern, Lehrer und einzelndem Schüler und den Schülern untereinander aufgebrochen. Die Rolle des Lehrers verschiebt sich völlig hin zum Moderator und Berater.

Peter Gorny

Mit einer Themenstellung ausgestattet, können die Schüler an großen Projekten weitgehend selbstorganisiert arbeiten. Diese Form wird vor allem im Sekundarbereich und an Hochschulen eingesetzt.

Ein besonders erwähnenswertes Projekt ist AQUADATA, an dem sich europaweit Hunderte von Schulen beteiligen, um die Wassereigenschaften von Flüssen und Bächen zu untersuchen (beteiligte Fächer: Biologie, Chemie, Physik, Geografie, Informatik und Mathematik). Dieses und andere Beispiele computerunterstützter Projekte werden von Gorny/Sarnow (1993) beschrieben.

#### **Szenario 4: Aktive Informationsbeschaffung (mit Hilfe von Fragebögen etc.)**

*Kommunikationsattribute:*

- Partner: ein Lehrer, mehrere Schüler;
- Initiative: Lehrer;
- Hauptziel: Meinungsumfragen, Wissenserwerb von Experten;
- Typische Anwendungsbereiche: Staatsbürgerkunde, Geschichte;

*Softwarekategorien:*

- Standardkommunikationssoftware

Diese sehr populäre und einfach umzusetzende Szenario kann in beinahe allen Fächern eingesetzt werden. Die Schüler stellen zu einem bestimmten Thema einen Fragebogen zusammen und senden ihn an bestimmte bekannte Einzelpersonen oder veröffentlichen ihn in einer Newsgroup. Die eingehenden Antworten werden gesammelt und ausgewertet und dienen als Basis für weitere Umfragen zum Thema.

Häufig entwerfen und veröffentlichen Schüler, z.B. eine ganze Gymnasialklasse, die Ergebnisse ihrer Untersuchung im WWW (Beispiele bei Gorny/Sarnow (1993)).

#### **Szenario 5: Informationssuche und -aufbereitung**

*Kommunikationsattribute:*

- Partner: zwei oder mehr Lehrer, zwei oder mehr Klassen;
- Initiative: Lehrer;
- Hauptziel: Informationsbeschaffung, statistische Auswertung;

- Typische Anwendungsbereiche: Naturwissenschaften, Mathematik, Staatsbürgerkunde.

*Softwarekategorien:*

- Standardkommunikationssoftware, spezielle Software für den Datenbankzugriff.

Neben dem Beispiel von AQUADATA, das schon im dritten Szenario (virtuelle Projektgruppen) angesprochen wurde, kann ein Projekt zwischen einer deutschen und einer spanischen Schule in Mathematik genannt werden, bei dem die Inhalte Statistik, Programmierung und Soziologie waren (Projektsprache: Englisch). Beide Klassen entwarfen gemeinsam einen von der jeweils anderen Klasse zu beantwortenden Fragebogen über die Familien der Schüler (Alter der Schüler, Geschlecht, Geschwister, Gewicht und Größe, Fernsehgewohnheiten, Gewicht und Größe der Eltern etc.). Nach der Phase der Datensammlung folgte eine statistische Evaluation mithilfe von Tabellenkalkulationsprogrammen. Die Ergebnisse wurden in Tabellen und Grafiken aufbereitet und jeweils auch der anderen Schule geschickt (Gorny/Sarnow, 1993).

### **Szenario 6: Selbstorganisiertes offenes Lernen**

*Kommunikationsattribute:*

- Partner: ein oder mehrere Schüler;
- Initiative: Schüler;
- Hauptziel: Beschaffung von Information, die von Experten erarbeitet wurde, sowie Entwicklung selbständiger Problemlösungen;
- Typische Anwendungsbereiche: individuelle Aufgaben mit starkem Gewicht auf Literaturrecherche;

*Softwarekategorien:*

- spezialisierte Lernsoftware, allgemeine Kommunikationssoftware, spezielle Groupware.

Dieses Szenario umfaßt die Aktivitäten von Schülern, die Informationen zu einem bestimmten Thema sammeln und dieses gesammelte Material strukturieren und bewerten – ein Prozeß, der nur bei gleichzeitigem Lernen (Wissenserwerb) möglich ist. Das erarbeitete

Material wird für selbständige Problemlösungen verwendet, die wieder im Internet veröffentlicht werden können. Die Schüler werden ermutigt, auf akademischen oder journalistischen Information-Servern zu suchen. Die Ergebnisse werden strukturiert und in einer lokalen Datenbank oder im WWW gespeichert, so daß andere sie abrufen können. Die notwendige Technik basiert entweder auf dem Internet oder auf Spezialsoftware zum Zugriff auf die Information-Server.

Besonders leicht handhabbar ist die Technik des World Wide Web, die es Autoren erlaubt, Dokumente verschiedenster Modi miteinander zu verknüpfen, indem interaktive Verweise auf Stellen in anderen Dokumenten gesetzt und so neue hypermediale Verknüpfungen erstellt werden.

Die didaktischen Ziele dieser Art von Unterrichtsaufgabe bestehen darin, Techniken zur Informationsbeschaffung zu entwickeln, die gefundenen Dokumente beurteilen zu lernen und relevantes Material aus einer Vielzahl von Dokumenten herauszusuchen und als neue interaktive HTML-Dokumente zu präsentieren. Weiterführende Aktivitäten schließen die Nutzung von Simulations- und Modellierungssoftware ein bis hin zur Programmierung eigener Modelle als Java-Applets.

Schüler werden so in die Lage versetzt, ihre Arbeitsergebnisse im WWW zu präsentieren, eine Herausforderung, die erheblich höher motiviert als das Abliefern eines Textes bloß für den Lehrer. Aus diesem Grunde wird dieses Verfahren an vielen Universitäten in den USA und Europa bereits praktiziert. In Deutschland soll auf diese Weise durch die »Initiative Schulen ans Netz« (SAN, 1996) ebenfalls eine Steigerung der Medienkompetenz bei Schülern erreicht werden.

## **5. Softwareergonomie – State of the Art der Entwicklung von Benutzungsoberflächen für Büroanwendungen**

### **Aspekte der Softwareergonomie**

Da das didaktische Design im Zentrum dieses Aufsatzes steht, kann die Bedeutung der Software-Ergonomie nur auf dieses Ziel hin skizziert werden. Diese Wissenschaft befaßt sich hauptsächlich mit dem Entwurf und der Evaluation von Benutzungsoberflächen in Bezug auf die Handhabung von Softwaresystemen (sowie mit den dahinterstehenden

theoretischen und praktischen Problembereichen). Die Kriterien für die Designentscheidungen und für die Evaluation sind aus der Psychologie und der Arbeitswissenschaft übernommen, und setzen voraus, daß *Benutzern* in einer *Organisation* (z.B. in einem Unternehmen) *Aufgaben* zugewiesen wurden, die sie zur Erreichung von Zielen der Organisation erledigen sollen, und daß diese Aufgaben mit einem Computer als *Werkzeug* bearbeitet werden müssen. Es sei hier angemerkt, daß die Forschungen der Software-Ergonomie längst den so gesetzten engen Rahmen der arbeitsbezogenen Software-Werkzeuge gesprengt haben und das ganze Gebiet der Mensch-Computer Interaktion (Human-Computer Interaction, HCI) und der benachbarten Felder der kognitiven Psychologie, Informatik und des Grafik- und Industriedesigns einbeziehen, so daß viele interdisziplinäre Projekte entstanden sind.

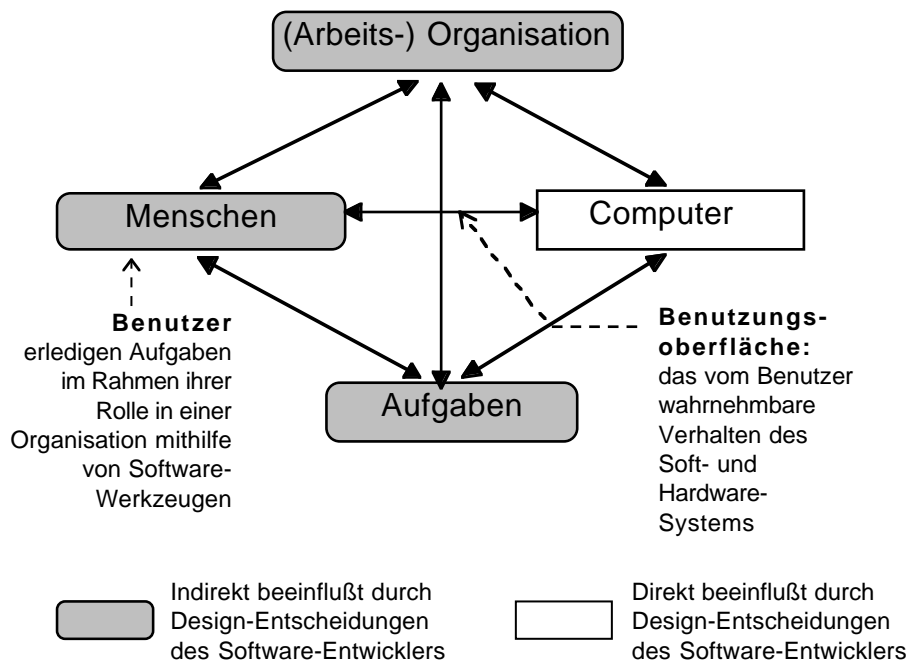


Abbildung 3: Einflussfaktoren beim Gebrauchstauglichkeits-Design von Arbeitsplatzsoftware

Im engeren Sinne befaßt sich die Software-Ergonomie also nicht mit dem Entwurf von Computerspielen, Werbe-, Freizeit-, oder Lernsoft-



ware, »Edutainment« und »Infotainment«, sondern mit arbeitsorientierter Software, aber eine vorsichtige Übertragung der Ergebnisse in den Bereich Lernsoftware ist trotzdem möglich.

### **Anforderungen und Vorgaben für Benutzungsoberflächen**

Softwareentwickler sehen sich sehr unterschiedlichen rechtlichen Bestimmungen und informellen Empfehlungen gegenüber. Neben den normalen gesetzlichen Bestimmungen (wie Haftung, Urheberrecht etc.) wurde eine neue formale Anforderung entwickelt: die »Mindestvorschriften bezüglich der Sicherheit und den Gesundheitsschutz bei der Arbeit an Bildschirmgeräten«, die die Europäische Union 1990 herausgegeben hat (EU-Verordnung 90/270/EEC) und die in den meisten Mitgliedsstaaten der EU in der nationalen Gesetzgebung verankert worden ist. In Verbindung mit weiteren Rechtsvorschriften nationaler Organisationen müssen bei der Entwicklung von Software und Hardware für Büroanwendungen detaillierte Qualitätsanforderungen bei der Auswahl und/oder des Entwurfs der Hardware, Einrichtung und Beleuchtung sowie des Raumklimas und Strahlenschutzes am Arbeitsplatz berücksichtigt werden. Für die Software werden ebenfalls Mindestbedingungen über die Aufgabenangemessenheit, Steuerbarkeit und Individualisierbarkeit sowie der Vermeidung von Überwachungsmechanismen festgelegt.

Wesentlichen Einfluß auf die Formulierung der EU-Richtlinie und der entsprechenden deutschen Bildschirmarbeitsverordnung hatte die Arbeit der International Standardization Organization, die in der Norm ISO 9241, Teil 10, die Begrifflichkeit der Software-Ergonomie definiert und mit Beispielen veranschaulicht hat. (ISO 9241 ist wortgleich mit der englischen Fassung der deutschen Norm DIN 29241 und der Europäischen Norm EN 9241.)

Von der 17-teiligen Norm ist für uns der Teil 10 »Dialogprinzipien« besonders wichtig: Software sollte folgende Kriterien erfüllen:

- *Aufgabenangemessenheit,*
- *Selbstbeschreibungsfähigkeit,*
- *Steuerbarkeit,*
- *Erwartungskonformität,*
- *Fehlerrobustheit,*
- *Individualisierbarkeit,*
- *Erlernbarkeit.*

Weiterhin ist es in unserem Zusammenhang wichtig, daß die Norm ISO 9241 in Teil 11 die »Gebrauchstauglichkeit« (Usability) definiert als  
*Gebrauchstauglichkeit =*  
*Effektivität + Effizienz + Benutzerzufriedenheit*

Neben den o.g. rechtlichen Vorschriften und Normen müssen sich Softwareentwickler zusätzlich mit den Vorgaben in den Style Guides der großen Softwarefirmen wie Apple, IBM, Microsoft oder anderer Institutionen wie der OSF oder eines spezifischen Kunden auseinandersetzen. Diese Style Guides sollen sicherstellen, daß die jeweiligen Entwickler ein gleichbleibendes Aussehen (»Look and Feel«) der Benutzungsoberflächen der verschiedenen Software-Systeme einhalten. Wie wir später zeigen werden, reicht es allerdings nicht aus, diesen Style Guides zu folgen, um EU-konforme und den ISO-Kriterien genügende Software zu entwickeln.

### **Eine exemplarische Methode zur Entwicklung von Benutzungsoberflächen**

Da wir hier nicht die verschiedenen Methoden für das Design von Benutzungsoberflächen diskutieren können, werde ich nur die von unserer Arbeitsgruppe entwickelte Methode exemplarisch vorstellen: »MUSE II – Method for User Interface Engineering« (Gorny, 1997). Sie basiert auf

- der Activity Theory (Leontiev, 1978),
- der Handlungsregulationstheorie, die z.B. für das Verfahren KABA (Kontrastive Aufgabenanalyse) angewendet wurde (Dunkel, 1993), und
- der Werkzeug/Material-Metapher, die besonders anschaulich die objektorientierten Entwicklungsmethoden für Benutzungsoberflächen unterstützt (Daldrup, 1996)

und bildet die theoretische Basis des Software-Ergonomie-Beratungswerkzeugs EXPOSE (Gorny, 1995a), das von unserer Arbeitsgruppe entwickelt wird.

Unsere Methode geht von der – in der Informatik breit akzeptierten – Annahme aus, daß eine systematische (also methodengeleitete) Entwicklung von Software zu einer höheren Produktqualität führen muß als das häufig noch anzutreffende »Basteln«. Dagegen stehen etwa

Auffassungen, die dadurch die Design-Kreativität gerade bei der Gestaltung der Benutzungsoberflächen eingengt sehen und deshalb Methoden des »Rapid Prototyping« z.B. beim Grafik-Design bevorzugen. Wir fordern dagegen mit unserer Methode für das Gesamt-Design von Benutzungsoberflächen das »Slow and Principled Prototyping«.

Zusammenfassend gesagt ist die Methode MUSE II *modellgeleitet* (Aufgabenmodell, Benutzermodell (Rolle), Benutzungsmodell), *kontextbezogen* (durch Einbeziehung aller Einflüsse der Arbeitsorganisation auf die Weise, in der ein Mensch seine Aufgabe erledigt), *hochgradig iterativ* (durch wiederholte Betrachtung des Entwurfsgegenstandes aus verschiedenen Sichten) und *exemplarisch* (durch Auswahl einiger hinreichend typischer Einsatzfälle der Software, statt einer formalen Modellierung des Gesamtsystems).

MUSE II strukturiert den Entwurfprozeß in drei Sichten, die jeweils wiederholt eingenommen werden können:

- die zweckgebundene (oder konzeptuelle) Sicht,
- die interaktionsbezogene Sicht,
- die präsentationsbezogene Sicht.

Sie erlaubt die Festlegung der Anforderungen an die Benutzungsoberfläche in allen drei Sichten auf der Basis von Regeln, die aus Forschungsergebnissen der Arbeitspsychologie, der Kognitionspsychologie und der Arbeitswissenschaft sowie aus der Norm ISO 9241 und der EU-Richtlinie 90/270/EWG stammen. Diese Regeln verbinden Analyseparameter mit Entwurfsentscheidungsparametern.

Der Methode MUSE II folgend gliedert sich der Entwurf von Benutzungsoberflächen in folgende Schritte für arbeitsbezogene Software:

- Festlegung der Aufgaben des Benutzers,
- Festlegung der Objekte, die während der Aufgabenbearbeitung benutzt werden (»virtuelle Objekte«, vergleichbar den realen Gegenständen, die der Benutzer ohne Computerunterstützung bearbeiten würde), und der Benutzeraktionen, mit denen die virtuellen Objekte verändert werden,
- Entwurf der Softwarefunktionen,
  - à die die Benutzeraktionen mit dem realen Objekt ersetzen sollen (Anwendungsfunktionen),und weiterer Softwarefunktionen, die benutzt werden,
  - à um die Software einem spezifischen Bedarf des Benutzers anpassen zu können (Adaptierfunktionen),

- à um das Computersystem zu steuern (Steuerfunktionen)
- à um die Benutzung der Software durch Hilfeinformationen, Fehlermeldungen etc. zu unterstützen (Metafunktionen),
- Entwurf der Interaktion zwischen Benutzer und System,
- Entwurf der Präsentation der virtuellen Objekte und der sonstigen Interaktionsobjekte.

## 6. Verknüpfung didaktischer und software-ergonomischer Methoden für das Softwaredesign

### MUSE II als Leitschnur für das didaktische Design

Unterrichtssoftware entwickeln heißt, nach der Festlegung der allgemeinen Bedingungen des Unterrichts (Ziele, Inhalte, Methoden und Organisation) die Benutzungsoberfläche so zu entwerfen, daß sie die höchstmögliche Gebrauchstauglichkeit bietet (siehe Abbildung 4).

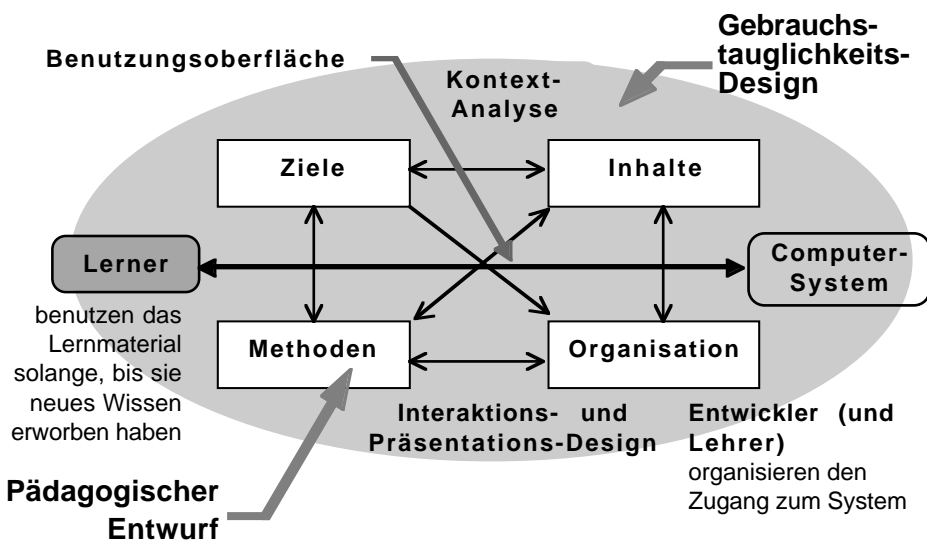


Abbildung 4: Vereinigung des Pädagogischen Entwurfs (Abbildung 1) mit dem Gebrauchstauglichkeits-Design (analog Abbildung 3) führt zum Didaktischen Design

Bei der Anwendung von MUSE II auf Lernsoftware lassen sich die Entwurfsentscheidungen mit den folgenden Schritten – in Analogie zu den oben beschriebenen für Arbeitssoftware – kennzeichnen:

- Festlegung der Ziele, Inhalte, Methoden und Organisation des Lernens,
- Festlegung der virtuellen Objekte, mit denen der Lerner im Lernprozeß hantiert,
- Entwurf der Softwarefunktionen,
  - à die die Handlungen des Schülers mit realen Objekten (beim Arbeiten ohne Computer) ersetzen sollen (Anwendungsfunktionen)
- und weitere Softwarefunktionen, die benötigt werden,
  - à um dem Lehrer zu ermöglichen, die Software für bestimmte Lernergruppen anzupassen, methodische und organisatorische Vorgaben einzustellen und die Software individuellen Lernanforderungen anzupassen (Adaptierfunktionen),
  - à um das Computersystem zu steuern (Steuerfunktionen),
  - à um die Benutzung der Software durch Hilfeinformationen, Fehlermeldungen etc. zu unterstützen (Metafunktionen),
- Entwurf der Interaktion zwischen Lerner (bzw. Lehrer) und System, wobei der Kenntnisstand und die Fähigkeiten des Lerners in Betracht gezogen werden muß,
- Entwurf der Präsentation der virtuellen Objekte und der sonstigen Interaktionsobjekte, wobei die *Affordanz* und die Anforderungen der *Semiotik* an die Präsentation wegen des Status von Lehrern und Lernern als »gelegentliche unregelmäßige Benutzer« besonders in berücksichtigt werden müssen.

In der Psychologie bedeutet »Affordanz« – grob gesagt – die vom Benutzer wahrgenommenen Benutzungsmöglichkeiten von Gegenständen. Die Darstellung der Interaktionsobjekte muß der intendierten Benutzung entsprechen, zum Beispiel:

- Button -> Start einer Aktion oder Funktion
- Menüpunkt -> Start einer Aktion oder Funktion
- Menü -> Auswahl einer von mehreren Aktionen oder Funktionen (es sollte nicht für Parameteränderungen verwendet werden)
- Checkbox -> Auswahl eines oder mehrerer Parameter
- Radio-Button -> Auswahl genau eines Parameters
- Piktogramm -> Auswahl eines Items, das ein Datenobjekt oder

ein Programm in Form eines Piktogramms (Icon) repräsentiert

- Tool-Bar -> Auswahl einer Aktion oder Funktion aus einer Folge von Piktogrammen (sie sollte nicht für Parameteränderungen verwendet werden)
- Text field -> Eingabe von Text
- (Scroll) List -> Auswahl einer Textzeile aus einer Liste von Textzeilen
- Datenfeld -> Eingabe von numerischen Daten
- Slider -> Eingabe von kontinuierlichen numerischen Daten, insbesondere physikalischer Größen mit Meßgenauigkeiten (z.B. für Zeit, Länge, Temperatur etc.)

(Daß die Affordanz nicht selbstverständlich aus der Gestalt des Gegenstandes hervorgeht, sei am Beispiel normaler Türen erläutert: jeder von uns hat schon einmal an dem Griff einer Tür gedrückt oder gezogen, die sich nun gerade nur in die andere Richtung öffnen ließ.)

Die »Semiotik« stellt den Zusammenhang her zwischen Inhalt und Semantik der Daten einerseits und Präsentation der Daten auf der Benutzungsoberfläche andererseits. Es ist zum Beispiel nicht zu empfehlen, eine Kurve über eine Folge von diskreten Werten zu ziehen (z.B. die verkaufte Stückzahl einer Ware pro Jahr über eine Reihe von Jahren); stattdessen ist hier eine Präsentation der jährlichen Verkaufszahlen mit einem Balkendiagramm angebracht. Um aber eine zeitliche Abfolge von Temperaturmessungen darzustellen, wäre eine Kurvendarstellung die semiotisch richtige Wahl, weil dadurch u.a. dem Betrachter auch suggeriert wird, daß er zwischen zwei Werten interpolieren kann.

### **Didaktisches Design WWW-basierter Lernsoftware**

Wenn man den kognitiven Konstruktivismus als Meßlatte an herkömmliche tutorielle und Trainingssoftware legt, wird offensichtlich, daß diese Software die höheren kognitiven Prozesse des Lehrens und Lernens nicht effektiv unterstützt. Die Möglichkeiten eines Schülers, Hilfestellungen zu erhalten, während er versucht, Informationen zu beschaffen und sich dadurch neues Wissen anzueignen, sind bei

Simulations- und Modellierungssoftware sowie beim Rollenspiel mit erweiterter Unterstützung durch telematik-gestützte Kommunikation und Informationsbeschaffung am größten (vgl. Thissen, 1997).

MUSE folgend müssen wir bei der Entwicklung derartiger Software folgende Entscheidungen treffen:

- Festlegung der Ziele, Inhalte, Methoden und der Organisation des Lernens:

Design-Entscheidung: selbstorganisiertes offenes Lernen – bevorzugt mit den folgenden Arten von Lernsoftware:

- à Software zum Problemlösen – Modellierung von Mikrowelten,
- à Kommunikationswerkzeuge und -medien,
- à Informationsbeschaffung,
- à Simulationen,
- à Rollenspiel.

- Festlegung der virtuellen Objekte, mit denen der Lerner beim Lernprozeß hantiert (abhängig vom Lern-Inhalt),
- Entwurf der Softwarefunktionen (Details vgl. oben),
- Entwurf der Interaktion zwischen Lerner und System, speziell mit den Java-Applets,
- Entwurf der Präsentation der virtuellen Objekte mit HTML, bzw. mit Java, und in multimedialen Dokumenten (Details wie oben).

Es ist in diesem Zusammenhang zu bemerken, daß die gegenwärtigen Versionen von HTML-3 bzw. 4 und Java noch etliche Wünsche in den Bereichen Navigation, Interaktionstechnik und Layout offen lassen, so daß gegebenenfalls hier Einschränkungen in Kauf genommen werden müssen.

### **Beispiel**

In diesem Text ist es unmöglich, die interaktiven Eigenschaften des Beispiels darzustellen. Deshalb werde ich die Lernsoftware mit ihren didaktischen Merkmalen nur grob skizzieren:

- Inhalt: Interaktive Manipulation von grafisch dargestellten mathematischen Funktionen (Funktionsgraphen),
- Typ: Software als Begleitmaterial für Vorlesungen,
- Anwendungsgebiet: Einführung in Generative Computer Graphics.

Diese Software wird in solche Vorlesungen eingebunden, die folgende didaktischen Merkmale hat:

- Ziel: Erwerb von wissenschaftlicher und praktischer Kompetenz in der Entwicklung von Computer Graphics-Software,
- Inhalt: nach der facheigenen wissenschaftlichen Systematik geordnet,
- Methode: Dozentenvortrag, ergänzt durch Tafelbilder und Folien,
- Organisation: Frontalunterricht.

Umsetzung: interaktive Simulationssoftware (z.B. Java-Applets, eingebettet in tutoriellem Material (Abbildung 5):

- Ziel: Vertiefung des Verständnisses der Interpolation und Approximation bei grafischen Darstellungen,
- Inhalt: Interpolation und Approximation in 2D und 3D mit Hermite-, Bezier- und B-Spline-Funktionen,
- Methode: grafische Präsentation einer Ausgangssituation, interaktive Manipulation der Kurven und Oberflächen, tutorielle Erklärung der Funktionen,
- Organisation: selbstorganisiertes individuelles Lernen (orts- und zeitunabhängig von der Vorlesung, aber auf sie ausgerichtet).

Das Ergebnis (Abbildung 5) zeigt nur eines der verschiedenen Java-Applets, das von einer studentischen Projektgruppe zu diesem Thema entwickelt wurde. In geeigneten Fällen können statt derartiger Applets Video- und Computeranimationen angewendet werden. Der umgebende Text ist – wie oben gesagt – nicht als unabhängiges Tutorial zu verstehen, sondern bezieht sich auf einen bestimmten Teil einer Vorlesung.

Das Beispiel wurde im Rahmen des Projekts *MuSIK* entwickelt (*Medien-unterstütztes Studium der Informatik*), das die Verbesserung des Informatikstudiums an der Universität Oldenburg zum Ziel hat. Gegenwärtig sind folgende Module fertiggestellt oder in Arbeit:

- Einführung in die Informatik (Datenstrukturen – Animation und Modellierung von Algorithmen, Visualisierung von Strukturen),
- Theoretische Informatik (Modellierung von Automaten mit verschiedenen visuellen Sprachen, z.B. Zustandsgraphen, Petri-Netze und das Testen der Modelle durch Simulation),
- Softwareentwicklung (Modellierung und Simulation von Projektentwicklungsprozessen, Visualisierung von Datenstrukturen, hierarchischen Klassen, etc.),
- Betriebssysteme, Netzwerke und Informationssysteme (Modellierung, Simulation der Betriebslast),



Netscape: Lernsoftware zu "Computer Graphics"

Back Forward Reload Home Search Guide Images Print Security Stop

Location: <http://www-cg-hci.informatik.uni-oldenburg.de/~pgse96/>




Abbildung 1




Abbildung 2

**Einleitung**

Die folgenden Hypertexte sollen Euch mit verschiedenen Methoden der mathematischen Darstellung von Kurven vertraut machen, die dann algorithmisch mit einem Computer angewandt werden können.

Die Fähigkeit der Darstellung von Abbildern reeller Gegenstände (siehe Abbildung 1, 2) wie auch die Erschaffung völlig neuer "Dinge" im Computer erfordern, daß dieser mehr als nur einfache waagerechte und senkrechte Linien darstellen kann, wie es die Beschaffenheit der darstellenden Hardware nahelegt. Für immer realistischer wirkende Abbildungen (die Ansprüche auf dem Gebiet der Computergraphik sind in den letzten Jahren in unvorstellbarem Maße gestiegen) wird zum Beispiel versucht, dreidimensionale Ebenen so auf das zweidimensionale Ausgabegerät zu projizieren, daß ein echt wirkendes Objekt sichtbar wird. Grundvoraussetzung für die Visualisierung dieser Ebenen ist die Darstellung von Kurven, die auf dem i.A. gerasterten Ausgabegerät (z.B. Bildschirm) natürlich nur angenähert werden können. Dazu werden einzelne Bildpunkte (Pixel) dann gesetzt, wenn sie dem mathematischen Kurvenverlauf am nächsten kommen.

Wie aber kennt das Programm oder dessen Autor den Kurvenverlauf?  
Um einen Algorithmus zur Kurvenberechnung erstellen zu können, wird erst ein mathematisches Modell benötigt. Als effektiv, flexibel und sinnvoll implementierbar hat sich das Modell der parametrisierten kubischen Kurven erwiesen: insbesondere bietet es Vorteile bei der Erstellung komplexerer Kurvenverläufe durch Aneinanderreihung einzelner Kurvensegmente.

Verschiedenen Kurventypen, die auf dem Prinzip der parametrisierten kubischen Kurven beruhen, sollen im folgenden Abschnitt erklärt werden.

---

**Die Hermite Kurve**

**Einleitung**

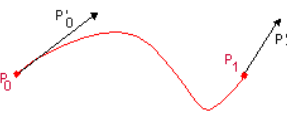
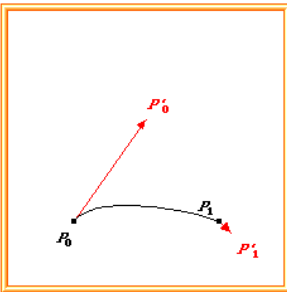


Abbildung 1



In diesem Abschnitt werden die mathematischen Grundlagen der Hermite-Splines erklärt. Als Leitlinie dient die Fragestellung:

**Wie zeichnen wir eine Kurve?**

Zum Beispiel kann ein Kurvensegment folgende Eigenschaften haben (siehe Abbildung 1):

1. Anfangs- und Endpunkt sowie
2. Steigung in Anfangs- und Endpunkt.

Aus diesen Einzelsegmenten lassen sich kompliziertere Kurven zusammensetzen.

Um selber diese Eigenschaften ausprobieren zu können, haben wir ein Applet zur Interaktion an einer Hermite Kurve beigefügt. Mit der linken Maustaste einen der vier Kontrollpunkte markieren, die Maustaste dabei gedrückt halten und in die gewünschte Richtung bewegen.

Abbildung 5: Ausschnitte aus einem Tutorial zur Vorlesung »Generative Grafische Datenverarbeitung« mit einem Java-Applet (Kasten unten links) zur interaktiven Manipulation eines Kurvenstücks.

- Anwendungsorientierte Informatik, z.B.:
  - à Computer Graphics (Visualisierung von Interpolationsalgorithmen, Interaktion mit Bildelementen, Manipulation von Farbmodellen, Modellierung von 3-D-Szenen einschließlich virtuelle Realität),
  - à Softwareergonomie und Mensch-Computer Interaktion (Modellierung von Benutzungsoberflächen und Interaktionsobjekten, Farbwahrnehmung, Simulation von Systemverhalten, etc.),
  - à Bildbearbeitung und Mustererkennung (Visualisierung der Algorithmen und interaktive Manipulation der Bilder),
  - à Intelligente Tutorielle Systeme und CBT (Benutzermodellierung, Lernstrategien, Tutorverhalten etc. und Visualisierung der Effekte),
  - à Anwendungen der Informationstechnik in Management, Wirtschaft und Verwaltung, in Produktionsprozessen, Ökologie etc. (Modellierung, Simulation und Visualisierung der Strukturen).

URL des Projekts MuSIK: <http://www-cg-hci.informatik.uni-oldenburg.de/~musik>

URL der o.g. studentischen Projektgruppe: <http://www-cg-hci.informatik.uni-oldenburg.de/~pgse96>

## **7. Zusammenfassung**

In diesem Aufsatz habe ich den Versuch unternommen, die Vorteile aufzuzeigen, die die Zusammenführung informatischer und pädagogischer Methoden bei der Gestaltung von Lernsoftware bringt, insbesondere auch solcher, die hoch-interaktiv ist und auf den multimedialen Möglichkeiten des WWW basiert. Diese Zusammenführung von Pädagogischem Entwurf von Unterricht und Lernmaterial mit dem Gebrauchstauglichkeits-Design für Arbeitssoftware definiert das Didaktische Design von Lernsoftware.

Es war ferner mein Ziel, verständlich zu machen, daß es unabdingbar ist, auch Lernsoftware nach den strengen Methoden des Software-Engineering und der Software-Ergonomie zu entwickeln, um auch hier

*Peter Gorny*

eine Produktqualität zu erreichen, die den enormen Entwicklungsaufwand rechtfertigt.

Weiter wurde gezeigt, daß ins Internet eingebundene Lernsoftware besondere Bedeutung für die zwischenmenschliche Kommunikation hat, und daß deshalb zusätzliche didaktische Anforderungen definiert werden müssen. Gerade diese Art von Software ist es, die heute schon die Verschiebung der Bedeutung der vorhandenen Inforamtionstechnologie repräsentiert: der Wandel des »Rechners« zum »Kommunikator«.

## **Danksagung**

Die hier vorgestellten Ergebnisse verdanke ich der Unterstützung vieler Kollegen in Schulen und Hochschulen, die bereitwillig in den Projekten PLUTO, EXPOSE, MUSE II und MuSIK mitgewirkt haben. Besondere Vorarbeiten hat mein Mitarbeiter Herr Dipl.-Math. Hilko Donker geleistet.

Die Projekte wurden aus folgenden Quellen gefördert: EU-Programme COMETT II und TEMPUS (PLUTO), BMBF-Programm Arbeit und Technik (EXPOSE), Nieders. Ministerium für Wissenschaft und Kultur (MUSE II), Hochschulsonderprogramm III (MuSIK).

Herrn Robert Schüttler danke für seine Unterstützung bei der Übertragung meines ursprünglich als englischer Vortrag geschriebenen Textes ins Deutsche.

## **Literatur**

- Daldrup, U.: (Un)Ordnung im Gestaltungsprozeß menschengerechter Software. Frankfurt u. a. (Peter Lang) 1996.
- DBS – Deutscher Bildungsserver. Datenstand Nov. 1997. URL: <http://dbs.schule.de/>
- Dunckel, H.: Kontrastive Aufgabenanalyse. Stuttgart (Teubner) 1993.
- Gorny, P.: PLUTO in Siebenmeilentiefeln durchs globale Dorf. In: LOG IN, 13. Jg. (1993), H. 3, S. 15-19.
- Gorny, P.; Sarnow, K.: Computergestützte Kommunikation im Unterricht. In: Troitzsch, K. (Hrsg.): Informatik und Schule '93. Fachtagung der Gesellschaft für Informatik, 11.-13. Oktober in Koblenz. Heidelberg u. a. (Springer-Verlag) 1993.
- Gorny, P. (1995a): EXPOSE – HCI-Counseling for user interface design. IN: Nordby, K. et al (eds.): Human-Computer Interaction – Interact '95. London etc. (Chapman & Hall) 1995.

*Didaktisches Design telematik-gestützter Lernsoftware*

- Gorny, P. (1995b): Telematyka w Szkolach. Komputer w Edukacji, nr. 2/1995 (Auf Polnisch). Engl. Fassung: Telematics in Schools, Eingelad. Vortrag. Fachtagung Informatics in Schools, Kielce (Polen) 1995. Manuskript.
- Gorny, P.: MUSE II – Kontextbezogener Entwurf von Benutzungsoberflächen. Tutorial zur Fachtagung Software-Ergonomie 1997. Dresden, 3.-6. März 1997. Manuskript.  
URL: <http://www-cg-hci.informatik.uni-oldenburg.de/Ergonomie97/Kontextbezog.BO-Design.pdf>.
- Gumin, H., Meier, H. (Hg.), Einführung in den Konstruktivismus / mit Beitr. von Heinz von Foerster u.a., 2. Aufl., München ; Zürich (Piper) 1995.
- ISO (1993): International Standards Organisation, Joint Technical Committee 1/SC 18, WG 1: Draft for »Group Communication«, »Preliminary User Requirements for Group Communication« and »User Requirements for Asynchronous Computer Referencing«, 1993.
- Kehoe, B.: Zen and the Art of the Internet – A Beginner's Guide to Internet. Chester, PA, 1992. Autorenadresse: brendan@cs.widener.edu.
- Leontjev, A. N.: Activity, conciousness, and personality. Englewood Cliffs NJ: Prentice Hall, 1978.
- Meyer, H.: Didaktische Modelle. Frankfurt (Scriptor) 1991.
- Rohe-Krebeck, K.: Probleme, Chancen und Möglichkeiten von Lernprogrammen im Mathematikunterricht der Sekundarstufe I. Staatsexamensarbeit. Fachbereich Mathematik, Carl v. Ossietzky Universität Oldenburg, 1993.
- SAN – Initiative Schulen ans Netz. URL: <http://www.san-ev.de/>
- Sarnow, K.: Unterrichtsvorschläge aus dem europäischen Schulprojekt ESP In: LOG IN, 13 (1993), H. 3, S. 49-54.
- Thissen, F.: Das Lernen neu erfinden – konstruktivistische Grundlagen einer Multimedia-Didaktik. In: Beck, U.; Sommer, W. (Hrsg.): LEARNTEC 97 – Europäischer Kongreß für Bildungstechnologie und betriebliche Bildung. Karlsruhe: 1997, S. 69-79. URL: <http://machno.hbi-stuttgart.de/~thissen>