

# Prinzipien der Gestaltung computergestützter Lernhilfen für Schüler mit geistiger Behinderung

Friedbert Jochum

FSQ-Bericht Nr. 08-05-15

Forschungsschwerpunkt Software-Qualität  
Fachhochschule Köln  
Steinmüllerallee 1  
51643 Gummersbach  
friedbert.jochum@fh-koeln.de  
Mai 2008

**Abstract:** Der Beitrag diskutiert auf der Grundlage sonderpädagogischer und informatischer Forschungsergebnisse sowie praktischer Erfahrungen im Rahmen von Entwicklungsprojekten drei grundlegende Fragen: (a) welche Qualitätsansprüche werden an computergestützte Lernhilfen für Schüler mit geistiger Behinderung aus der Nutzungsperspektive gestellt, (b) welche Prinzipien der konstruktiven Qualitätssicherung und (c) welche methodischen Schritte erfordert die Entwicklung solcher Systeme, damit sie diesen Qualitätsansprüchen genügen und darüber hinaus leicht gewartet, gepflegt und portiert werden können?

## 1 Einleitung

Die Entwicklung von Software für Menschen mit geistiger Behinderung ist eine anspruchsvolle Herausforderung, bei der weitgehend unbekanntes Terrain betreten wird. Insbesondere erfährt man auf sehr elementare Weise, dass der Mensch im Mittelpunkt stehen muss, wenn man gute Resultate erzielen will und nicht der Computer. Handelt es sich doch bei Software um ein rein *sprachlich/symbolisches Artefakt*, dessen Qualität aufs Engste mit geistigen Dimensionen seiner Benutzer wie Sprache, Kommunikation, Wahrnehmung, Kognition und Motivation korrespondiert. Subtil abgestimmte Computerprogramme bieten gerade deshalb, aber auch wegen ihrer dynamischen und interaktiven, auf mehrere Sinne ausgerichteten Gestalt, sozusagen als „geistige Prothesen“ mehr als alle anderen Medien differenzierte Möglichkeiten, Menschen mit geistiger Behinderung in ihrer Entwicklung zu fördern und bei der Alltagsbewältigung zu unterstützen.

Aufgrund der relativ kleinen Zielgruppe sind allerdings nur wenige geeignete Produkte allgemein verfügbar. Zum Teil fehlt es aber auch an fein abgestimmten Methoden und Instrumenten für die Erschließung der Anforderungen und die Entwicklung neuer Systemansätze, die dieser sehr speziellen Nutzergruppe mit oft individuell variierenden Mehrfachbehinderungen gerecht werden.

Seit April 2000 besteht zwischen dem Forschungsschwerpunkt Software-Qualität der Fachhochschule Köln und der Städtischen Förderschule Geistige Entwicklung (vormals Städtische Schule für Geistigbehinderte) in Köln-Vogelsang eine lose Kooperation mit dem Ziel, Möglichkeiten des Computereinsatzes im Unterricht dieser Schule auszuloten. Neben der Realisierung von Prototypen für den praktischen Einsatz im Schulunterricht wurden auch Methoden für die Nutzer- und Anforderungsanalyse [Vo08] sowie die Systementwicklung [Jo08a] auf ihre Eignung hin erprobt, verfeinert und weiterentwickelt. Folgende praktische Ergebnisse konnten bisher in Form von Prototypen vorgelegt werden: ein multimediales Edutainmentspiel für den Umgang mit Standardsoftware [Mü01], ein 3D-Simulationsspiel zur Verkehrserziehung [GJ03], ein 3D-Simulationsspiel zum Einkauf im Supermarkt [LM04] und ein 2D/3D-Simulationsspiel zum selbstständigen Wohnen [EO07].

Der eigentliche Sinn solcher Computerlernspiele ist der Transfer der in den virtuellen Computerwelten erzielten Lernerfolge auf Situationen des praktischen Lebensvollzugs der Spieler, etwa beim Überqueren einer innerstädtischen Straße ohne Fußgängerampel, beim Kochen von Spaghetti mit Soße in einem Behindertenwohnheim oder beim Einkauf von Eiern, Milch und Käse im Supermarkt. Inwieweit eine solche Transferleistung im konkreten Fall gelingt, ist eine sonderpädagogische Fragestellung und kann von der Informatik nicht beantwortet werden. Die Informatik-Forschung kann jedoch die in der Sonderpädagogik erhobenen Forderungen hinsichtlich der Gestaltung von Lernsoftware aufgreifen und dazu geeignete Methodenansätze entwickeln.

Hierbei stellen sich drei grundlegende Fragen, denen im vorliegenden Beitrag, auch vor dem Hintergrund der oben erwähnten eigenen Erfahrungen, nachgegangen wird: (a) welche Faktoren sind für den Lernerfolg und die Transferleistung in die Lebenswelt der Schüler aus sonderpädagogischer Sicht relevant, und welche Qualitätsansprüche resultieren daraus an die Lernsoftware aus der Nutzungsperspektive, (b) welche Prinzipien der konstruktiven Qualitätssicherung und (c) welche methodischen Schritte erfordert die Entwicklung solcher Systeme damit diese den Qualitätsansprüchen genügen und darüber hinaus leicht gewartet, gepflegt und portiert werden können? Frage (a) wird in Kapitel 2 diskutiert, Frage (b) in Kapitel 3 und Frage (c) in Kapitel 4. Kapitel 5 fasst abschließend die Ergebnisse zusammen und geht kurz auf offene Fragen ein.

Auf Standards des Usability Engineering, des Requirements Engineering und des Software Engineering, die hier zum Teil einfließen, zum Teil aber auch methodisch nachgeordnet sind, wird nicht näher eingegangen. Dazu sei auf die einschlägige Fachliteratur verwiesen. Außerdem wird im vorliegenden Beitrag weitgehend auf die Darstellung einzelner Details zugunsten einer Gesamtsicht verzichtet. In [Jo08] werden die Anforderungen an Lernsysteme aus sonderpädagogischer Sicht und in [Jo08a] die methodischen Hintergründe der Systementwicklung ausführlicher diskutiert.

## 2 Qualitätskriterien

### 2.1 Sonderpädagogische Zielsetzungen

Die sonderpädagogischen Forderungen [B199, SL99, Sc00, Br06, Pf07] können dahingehend zusammenfasst werden, dass sich Lernsoftware für Schüler mit geistiger Behinderung funktional, strategisch und ergonomisch an den inhaltlichen und didaktischen Konzepten des Unterrichts sowie an den individuellen Motiven, Interessen, sprachlichen, kommunikativen, kognitiven wie motorischen Kompetenzen der Schüler orientieren muss. Außerdem sollten die Systeme möglichst leicht an die individuellen Fähigkeiten der Schüler hinsichtlich des Schwierigkeitsgrads, der Strukturiertheit und des Detailliertheitsgrads der Lerneinheiten, der sachlichen und bedientechnischen Hilfen sowie der Interaktionsformen adaptierbar sein. Um eine hohe Transferleistung zu erzielen, sollten die Lerninhalte und deren Darstellung in den Spielwelten handlungsleitend sein und der Lebenswirklichkeit der Schüler möglichst nahe kommen. Darüber hinaus wird die Forderung erhoben, auch partnerschaftliches Lernen, Arbeiten und Spielen durch die Lernsoftware zu ermöglichen. Schließlich sollte zur Unterstützung der Lehrpersonen ein geeignetes diagnostisches Instrumentarium zur Lernzielkontrolle sowie zur Dokumentation des Lernverhaltens und Lernfortschritts einzelner Schüler und Schülergruppen zur Verfügung stehen.

Aus diesen sonderpädagogischen Zielsetzungen lassen sich spezifische Kriterien für die Nutzungsqualität von Lernsystemen ableiten, die den drei Kategorien Unterrichtstauglichkeit, Schülertauglichkeit und Lehrertauglichkeit zugeordnet werden können. Daraus ergeben sich aber auch Anforderungen an die Systemarchitektur.

### 2.2 Unterrichtstauglichkeit

Die zentrale Forderung bezüglich der Unterrichtstauglichkeit ist, dass sich Lernprogramme in die Ziele und Inhalte des regulären Unterrichts integrieren lassen müssen und inhaltlich für die Schüler bedeutsam hinsichtlich ihrer Lebenswirklichkeit sein sollten. Dies betrifft in erster Linie die *pragmatisch-semantische Ausgestaltung der Systemfunktionalität* und zwar hinsichtlich der methodisch-didaktischen Einbettung in den Unterricht, der Operationalisierung der Lernziele sowie der inhaltlichen Bezugnahme auf den Unterrichtsgegenstand bzw. die Lebenswelt der Schüler. Ein weiterer wichtiger Aspekt ist, dass die Lektionen des Lernprogramms in der zugehörigen Unterrichtseinheit vor- und nachbereitet werden können. Dies erfordert eine sinnvolle Aufteilung in Lernschritte, welche mit und welche ohne Computerhilfe auszuführen sind sowie interaktive Elemente zur inhaltlichen und methodischen Einbindung in die entsprechende Unterrichtseinheit.

Jede Lernfunktion muss darüber hinaus den *Bedeutungs- und Geltungsansprüchen* des Unterrichts bzw. der Lebenswirklichkeit der Schüler genügen, damit die Lern- und Transferziele erreicht werden können. Das heißt, die Daten, auf denen die Funktionen operieren, müssen die relevanten Unterrichtsgegenstände bzw. Alltagsgegenstände inhaltlich adäquat repräsentieren, und die Funktionen müssen die Lernziele des Unterrichts so operationalisieren, dass sie bei ihrer korrekten Anwendung gültige Resultate liefern – und bei inkorrekt Anwendung ungültige Resultate.<sup>1</sup>

Hinsichtlich der Geltung sind außerdem *zweckbedingte Reihenfolgen* von Teilschritten zu beachten, insbesondere beim handlungsleitenden Lernen. Andererseits sollte das Lernprogramm keine Reihenfolgen von den Schülern fordern, die sachlich nicht notwendig sind. Dies würde zum Einüben nicht sachgemäßer (zumindest aber nicht zweckrationaler) Handlungsketten führen und sowohl die Komplexität des Lernens als auch die der Systembenutzung unnötig erhöhen (es sei denn, es sollen „motivationale Umwege“ eingeführt werden). Teilschritte ohne sachlich zwingende Reihenfolgen müssen daher als *nebenläufige Prozesse* modelliert werden, so dass beliebige Ausführungsordnungen (auch das gleichzeitige Ausführen mehrerer Teilhandlungen beim partnerschaftlichen Lernen) als gültig bewertet werden.

### 2.3 Schülertauglichkeit

Die zuvor diskutierte Ausrichtung von Lernsoftware an den Zielen und Inhalten des Unterrichts erfordert aufgrund heterogener Schülerprofile eine Anpassung an individuelle Förderpläne im Rahmen des regulären Unterrichts, d.h. inhaltliche und didaktische Differenzierungen der Lektionen sowie deren individuelle Ergänzung um motivationale Elemente, Hilfesysteme und Funktionen für partnerschaftliches Lernen. Ebenso muss die Mensch-Computer-Interaktion individuell auf die kognitiven und motorischen Stärken und Schwächen der Schüler abgestimmt werden. Die kognitive und feinmotorische Weiterentwicklung der Schüler und Fortschritte beim Lernen verlangen darüber hinaus eine sich über die Zeit dynamisch verändernde Adaptionmöglichkeit für die genannten Bereiche.

**Lektionen:** Lernschwache Schüler bedürfen mehr Unterstützung durch stärkere Strukturierung und kleinere Schritte und mehr Erklärungen als lernstarke Schüler, um einerseits Langeweile zu verhindern, wenn die Aufgaben zu leicht sind und andererseits Frustration, wenn die Aufgaben zu schwer sind. Somit hat das methodisch-didaktische Konzept Einfluss auf den Aufbau der Lektionen. Dies bedeutet für den Systementwurf, dass eine Lektion für ein und dieselbe Aufgabenstellung je nach Lernniveau und kognitivem Vermögen des Lernenden inhaltlich wie strukturell unterschiedlich fein gegliedert werden muss. Dies wirkt sich auch unmittelbar auf die Hilfesysteme aus.

---

<sup>1</sup> Die Möglichkeit einer inkorrekten Funktionsanwendung wird allerdings von Lehrern wie auch in der sonderpädagogischen Forschung oft kritisch gesehen, da dies Fehlverhalten antrainieren könnte.

**Hilfesysteme:** Zwei Arten von Hilfesystemen sind zu unterscheiden. Aufgabenabhängige, sachliche Hilfsfunktionen zum Zwecke der Erklärung ganzer Lektionen, einzelner Teilschritte zur Lösung von Aufgaben oder zur sachlichen Information über den Lerngegenstand sowie technische Bedienhilfen zur Erklärung von Benutzungsoberflächenelementen und Navigationsmöglichkeiten im System. Die sachlichen Hilfen müssen wie die Lernfunktionen den Geltungs- und Bedeutungsansprüchen des Unterrichts genügen und ebenso individuell auf die Schüler abstimbar sein.

**Motivation:** Für Lernfunktionen sind Geltung und Bedeutung die wichtigsten Gütekriterien. Für motivationsfördernde Funktionen gelten hingegen eher Gütekriterien wie Spaß, Aufmerksamkeit, Spannung oder Interesse. Hier können Geltung und Bedeutung sogar hinderlich sein. Umso wichtiger ist es „Lernfunktionen“ von „Motivationsfunktionen“ deutlich wahrnehmbar zu trennen, damit keine falschen Inhalte gelernt werden können. Dramaturgische Elemente [z.B. Mi05, Ha04] bei der Spielgestaltung sowie eine entsprechende Anbahnung und Nachbereitung im einbettenden Unterricht unter Beachtung individueller Vorlieben der Schüler können wesentlich zu einer Motivationssteigerung beitragen.

**Partnerschaftliches Lernen:** Partnerschaftliches Lernen zur Erhöhung der sozialen und kommunikativen Kompetenz erfordert neben einer geeigneten Unterrichtsthematik die soft- und hardwaretechnische Systemvernetzung sowie zusätzliche Funktionen zur Koordination und Kommunikation zwischen den Schülern. Dies beeinflusst auch die Gestaltung der Benutzungsoberfläche und der Interaktionsprozesse. Partnerschaftliches Lernen zwischen Schülern mit unterschiedlichem Lernniveau und unterschiedlichen kognitiven wie motorischen Kompetenzen stellt zusätzliche Anforderungen an das System, da sowohl die Interaktions- als auch die Lernprozesse und die Benutzungsoberfläche bei jedem Schüler unterschiedlich ausgestaltet sein müssen.

**Mensch-Computer-Interaktion:** Jede Funktion muss nicht nur den Geltungs- und Bedeutungsansprüchen der Lernaufgabe genügen, damit sie effektiv benutzt werden *kann*, sondern muss auch zusammen mit ihren Daten, auf denen sie operiert, eine wahrnehmbare, verstehbare, selbsterklärende, auffordernde und zugleich auch motivierende äußere Form auf der Benutzungsoberfläche aufweisen, damit sie auch tatsächlich effektiv von den Schülern benutzt *wird*. Es ist leicht nachvollziehbar, dass dies von individuellen psychischen Faktoren der Schüler wie kognitive und kommunikative Kompetenzen sowie von deren Motivierbarkeit und Interessen abhängt. Darüber hinaus sind oft auch körperliche Beeinträchtigungen zu berücksichtigen wie z.B. feinmotorische Einschränkungen, Seh- und Hörschwächen, eingeschränktes Farben- und Formensehen oder Probleme mit der Auge-Hand-Koordination. Diese können auch in Kombination zueinander auftreten. Das Zusammenspiel von Multimedialität, Interaktivität, Dynamik und 2D/3D-Virtualität bietet vielfältige Möglichkeiten, Schüler mit geistiger Behinderung zu fördern und zu motivieren. Dennoch sollte damit behutsam umgegangen werden. Denn nicht alle Themen eignen sich für eine 3D-Darstellung, und nicht alle Schüler können sich im virtuellen dreidimensionalen Raum sicher orientieren. Zu viele Details und eine zu hohe Dynamik könnten außerdem viele Schüler überfordern. Individuelle Abstimmungen entscheiden hier über Erfolg oder Misserfolg.

## 2.4 Lehrertauglichkeit

Aus Sicht der Lehrenden wäre sicher ein weiteres Bündel an Hilfen zur Unterstützung ihrer Arbeit beim Einsatz von Lernsoftware nützlich. Dazu zählen etwa Funktionen zur Auswahl von Lernzielen, Lektionen oder Lektionsvarianten im Lernprogramm, zur Erstellung und Pflege von Schülerprofilen, zur Systemkonfiguration und -nivellierung hinsichtlich individueller Schülerprofile, zur Lernzielkontrolle, zur Diagnose und Dokumentation des Lernverhaltens der Schüler. Diese Arbeitsmittel müssen auf die Bedürfnisse der Lehrenden und deren Arbeitsbedingungen zugeschnitten sein. Aber auch hinsichtlich der Auswahl des Lernprogrammtyps [Br06] sollten die Arbeitsbedingungen Berücksichtigung finden. Kann eine Lehrperson aufgrund der Klassengröße, Klassenzusammensetzung, des Entwicklungsstands der Schüler und anderweitiger Verpflichtungen die für einen anvisierten Lernprogrammtyp nötige Hilfestellung sowie die nötige Vor- und Nachbereitung im Unterricht überhaupt leisten?

## 2.5 Systemarchitektur

Aufgrund der geforderten Adaptierbarkeit der Lektionen, Hilfen und Interaktionsformen an die individuellen Bedürfnisse der Schüler ist eine komponentenbasierte Systemarchitektur unabdingbar, wobei die fachspezifischen Aspekte (Lektionen, Hilfen, Schülerprofile, Schülerverhalten) von den technologischen Aspekten getrennt werden sollten. Dies erleichtert auch die Realisierung, Wartung, Pflege und Portierung des Systems. Zur Unterstützung partnerschaftlichen Lernens bietet sich eine Client-Server-Architektur an. Dies ermöglicht darüber hinaus, die Hauptrechenleistung auf dem Server zu konzentrieren und die Sonderschulen mit ihrer in der Regel schwachen Rechenausstattung durch schmale Clients zu entlasten.

## 3 Begründungsprinzipien

Unterrichts-, Schüler- und Lehrertauglichkeit von Lernsoftware können nicht aus einer formalen oder technologischen Warte aus in ein System hineinprogrammiert werden, sondern bedürfen einer methodischen Grundlegung [Mi97, Ja01] in der Nutzungspraxis, so dass das resultierende System den Geltungs- und Bedeutungsansprüchen des regulären Unterrichts und der individuellen Förderpläne genügt und für die verschiedenen Zielgruppen adäquat benutzbar ist. Ausgangsbasis und Ziel der Entwicklung computergestützter Lernsysteme sollte also die konkrete (intendierte) Unterrichtspraxis sein, mit der dort vorherrschenden (Fach-) Sprache und Kommunikationskultur, den zu unterstützenden Unterrichtseinheiten, Lernzielen und didaktischen Konzepten sowie den Stärken, Schwächen und Vorlieben der unmittelbar beteiligten Schüler (Abbildung 1).

Da aber Softwareentwickler, Lehrer und Schüler in der Regel unterschiedliche Sprachen, Perspektiven, Interessen, Kompetenzen und fachliche Hintergründe haben, muss die Erschließung der Unterrichtspraxis *einvernehmlich* zwischen allen Beteiligten erfolgen (**dialogisches Begründungsprinzip**).

Angesichts der Behinderungen der Schüler kann dies aber nur gelingen, wenn der inhaltliche Diskurs durch ein *exploratives* und *experimentelles* Prototyping [F184] unterstützt wird (**pragmatisches Begründungsprinzip**). So haben Usability-Studien [Ei07, Mü01] gezeigt, dass sich die Systemanforderungen im Detail, insbesondere hinsichtlich der Mensch-Computer-Interaktion sowie der didaktischen und motivationalen Ausgestaltung der Lektionen, letztlich nur empirisch oder experimentell mit den betroffenen Schülerinnen und Schülern selbst ermitteln lassen. Praktische Erfahrungen in diversen Projekten haben aber auch gezeigt, dass ein reines Prototyping ohne konzeptionellen Hindergrund die Lernzielkontrolle sehr erschwert und zu unstrukturierten, schwer wartbaren und änderbaren Systemen führt, was auch eine verminderte Adaptierbarkeit der Funktionalität und Interaktionsformen zur Folge hat.

Die experimentelle Erschließung der Schüler-Anforderungen bedarf also zusätzlich einer *sprachkritischen Rekonstruktion* [KL96] der Unterrichtspraxis. D.h. die Benennung, Intension und Extension [Ca68, Fr75] aller relevanten Begriffe und deren Beziehungen untereinander werden kritisch hinterfragt und einvernehmlich, eindeutig, lückenlos und zirkelfrei explizit nachvollziehbar dokumentiert, zum Beispiel mit Hilfe der Unified Modeling Language UML (**semantisches Begründungsprinzip**). Ziel ist die Herausarbeitung der Funktions-, Daten- und Interaktions-Semantik des zu entwickelnden Systems als konzeptionelle Grundlage für die formale statische und dynamische Ausgestaltung des Systems nach Innen wie nach Außen: Die zu erlernenden Handlungen werden zusammen mit den Vor- und Nachbedingungen (und den Invarianten) in solche Teilabläufe zerlegt, die das System simulieren soll (Systemfunktionen) und in solche, die vom Benutzer auszuführen sind. Die Vorbedingungen beschreiben die Zustände der involvierten Dinge, die vor der Funktionsanwendung oder Benutzerhandlung erfüllt sein müssen und die Nachbedingungen die Zustände, die durch die Funktionsanwendung oder Benutzerhandlung hergestellt werden. Damit ist die *Funktionssemantik* spezifiziert. Die Menge aller Vor- und Nachbedingen (und Invarianten) spezifiziert die notwendigen und hinreichenden Wertebereiche auf denen die Funktionen operieren (*Datensemantik*). Die *Interaktionssemantik* ist durch den Kontrollfluss der wechselseitigen Benutzerhandlungen und Funktionsanwendungen sowie dem damit einhergehenden Datenfluss spezifiziert.

Wichtig ist also, computergestützte Lernsysteme für Schüler mit geistiger Behinderung pragmatisch, semantisch und dialogisch zu begründen, um sicherzustellen, dass alle Anforderungen und Bedürfnisse in diesem hochsensiblen Bereich auch tatsächlich erfüllt werden und in die Systemarchitektur einfließen. Hierbei empfiehlt es sich, die Unterrichtseinheiten, in die das Lernspiel eingebunden werden soll, als Ausgangsbasis zu nehmen (Systempragmatik). Daraus können die relevanten Inhalte hinsichtlich der Funktionen, Daten und Interaktionen des Systems (Systemsemantik) hergeleitet und auf der Benutzungsoberfläche des Systems in adäquater Form zum Ausdruck gebracht bzw. in Komponenten und Schnittstellen<sup>2</sup> zusammengefasst werden (Systemsyntax).

---

<sup>2</sup> Eine Komponente kapselt alle Funktionen, die auf dem gleichen inhaltlich zusammenhängenden Datenbereich operieren, zusammen mit diesen Daten in einer Einheit und veröffentlicht alle exportierten und importierten Dienste über Schnittstellen.

Ein Vorgehen nach dieser methodischen Ordnung nennen wir **konstruktiv**. Systeme werden danach nicht aus einer formalen oder technischen Perspektive *programmiert*, sondern aus der Nutzungspraxis heraus *konstruiert*, d.h. zielgerichtet, lückenlos, explizit nachvollziehbar und unter Wahrung von Geltung und Bedeutung schrittweise aufgebaut. Hierbei kommen die elementaren (sprachlogischen) Konstruktionshandlungen Prädikation, Abstraktion und Komposition [Sc97, WO05] zur Anwendung. Die insgesamt erforderlichen Prozesse bei der Systementwicklung werden im folgenden Kapitel diskutiert.

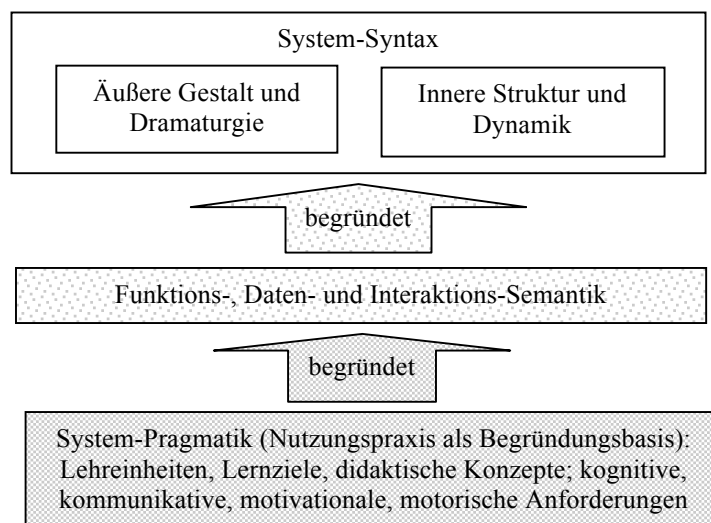


Abbildung 1: Pragmatisch-semantische Systembegründung

#### 4 Konstruktions- und Gestaltungsprozesse

Die hohen und schwer zugänglichen Anforderungen hinsichtlich der Schülertauglichkeit einerseits und die Notwendigkeit einer modularen, adaptierbaren Systemarchitektur andererseits erfordern zwei unterschiedliche Entwicklungslinien: Eine nach außen gerichtete Gestaltung mit dem Ziel, das System interaktiv in den Nutzungskontext einzubinden, und eine nach innen gerichtete Konstruktion mit dem Ziel einer komponentenbasierten Systemarchitektur. Beide Linien sind für die gegebene Domäne und Zielgruppe gleichermaßen wichtig, traditionell aber in unterschiedlichen Disziplinen mit jeweils eigenen Denkansätzen verwurzelt. Die Interaktionsgestaltung – verwurzelt im Usability Engineering – kümmert sich zwar intensiv um die Einbindung des Systems in den Nutzungskontext, vernachlässigt aber den inneren Systemaufbau. Umgekehrt stellt die Entwicklung komponentenbasierter Systeme – verwurzelt im Software Engineering – den Aufbau änderungsfreundlicher und wiederverwendbarer Systeme in den Mittelpunkt, vernachlässigt aber die Mensch-Computer-Interaktion.



Eine Verbindung der beiden Disziplinen auf einer aggregierten Ebene der Vorgehensmodelle [z.B. Co03] oder Lebenszyklen [z.B. Py03] erscheint für den vorliegenden Fall eher ungeeignet. Stattdessen nehmen wir eine *architekturzentrierte Sicht* ein und fragen nach dem *kleinsten sinnvollen Architekturelement*, mit dem beide Bereiche gemeinsam operieren. Als ein solches Architekturelement hat sich die Systemfunktion (Lern-, Hilfe-, Diagnosefunktion, etc.) herauskristallisiert. *Bedeutung* und *Geltung* sind hier die grundlegenden Qualitätskriterien, denen darstellende, formal/strukturelle und technische Aspekte methodisch nachgeordnet sind. Sie verankern die Systemfunktionalität im Unterricht hinsichtlich der Lerngegenstände, didaktischen Konzepte und Lernziele und bilden einerseits die konzeptionelle Basis für eine inhaltlich sinnvolle Strukturierung des Systems bei zusätzlicher Beachtung softwaretechnischer Anforderungen. Andererseits bilden sie auch die konzeptionelle Grundlage für eine adäquate Ausgestaltung der Mensch-Computer-Interaktion, bei zusätzlicher Beachtung psychischer und feinmotorischer Stärken und Schwächen der Benutzer. Gültige semantische Modelle der Lektionen sind aber auch eine notwendige Voraussetzung für die Lernzielkontrolle und die Diagnose des Lernverhaltens der Schüler.

Drei miteinander korrespondierende Prozesse mit jeweils spezifischen Zielsetzungen, Qualitätskriterien und Stakeholdern sind also zu unterscheiden (Abbildung 2):

- (a) Die *sprachkritische Rekonstruktion* der einbettenden Unterrichtseinheiten und didaktischen Konzepte gemeinsam mit den Lehrpersonen (Unterrichts- und Lehrertauglichkeit) in mehreren Begründungs-Widerlegungs-Revisions-Zyklen. Sie liefert mit der Funktions-, Daten- und Interaktions-Semantik die konzeptionelle Grundlage für die beiden anderen (nebenläufigen) Prozesse (b) und (c). Dies bedeutet aber nicht, dass sämtliche Systemfunktionen spezifiziert sein müssten, bevor die anderen Prozesse starten können.
- (b) Die *experimentelle Gestaltung* der Mensch-Computer-Interaktion mit häufigen Design-Evaluations-Redesign-Zyklen unter expliziter Einbeziehung der unmittelbar betroffenen Schüler sowie der Lehrpersonen. Dies liefert konkrete Anhaltspunkte für den audiovisuellen und dramaturgischen Aufbau der Spielwelten und die rückwirkende semantische Verfeinerung der in Prozess (a) spezifizierten Lern- und Hilfefunktionen hinsichtlich Dynamik, Schwierigkeitsgrade, Strukturiertheit, Detailliertheit und Motivation (Schülertauglichkeit).
- (c) Die *sprachlogische Konstruktion* fachlicher Systemkomponenten und Schnittstellen sowie deren Kopplung mit plattformspezifischen Komponenten<sup>3</sup> (Anpassbarkeit, Wartbarkeit, Änderbarkeit, Portierbarkeit), ggf. mit Rückwirkung auf Prozess (a). Aufgrund der Komponenten-Architektur bleiben ggf. nötige Änderungen durch Ergebnisse aus Prozess (b) lokal, ohne Auswirkungen auf das Restsystem. Eventuell neu erforderliche Komponenten können im Rahmen *evolutionärer* Prototyping-Zyklen [F184] in das System integriert werden.

---

<sup>3</sup> In [Fi07] sind die einzelnen Schritte bei der sprachlogischen Konstruktion eines komponentenbasierten Simulationsspiels für selbstständiges Wohnen beschrieben.

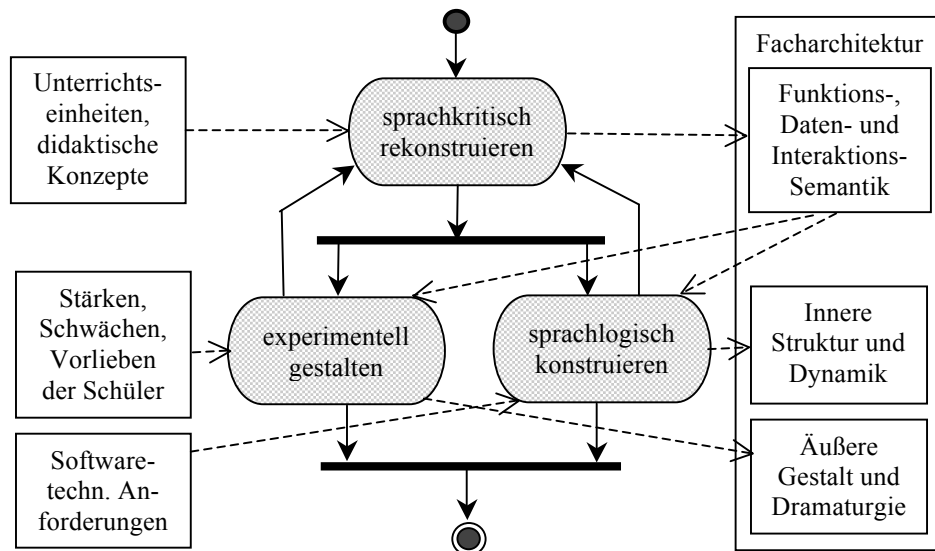


Abbildung 2: Konstruktives Gestalten einer schüler-, lehrer- und unterrichtstauglichen Systemarchitektur (UML-Aktivitätsdiagramm)

## 5 Zusammenfassung

Computergestütztes Lernen für Schüler mit geistiger Behinderung erfordert ein ausdifferenziertes multimediales Instrumentarium, abgestimmt auf die Erfordernisse des Unterrichts, der Schüler und der Lehrer. Herausragendes Qualitätsmerkmal ist dessen feine Justierbarkeit hinsichtlich der Stärken, Schwächen und Vorlieben einzelner Schüler und Schülergruppen. Dazu bedarf es einer in der Nutzungspraxis pragmatisch, semantisch und dialogisch begründeten, komponentenbasierten Systemarchitektur sowie geeigneter Methoden zu deren Entwicklung.

Der hier in Grundzügen skizzierte methodische Ansatz kreuzt Engineering- und Design-Perspektiven auf der Ebene von Architekturelementen und beschreibt den Aufbau von Lernsystemen konsequent von außen nach innen. Damit wird eher die Sichtweise von *Softwarearchitekten* eingenommen als die von Softwareentwicklern oder Interaktionsdesignern. Softwareentwickler neigen all zu oft dazu, system- und programmiertechnische Aspekte in den Vordergrund zu stellen. Dabei besteht die Gefahr, dass die resultierende Software zwar algorithmisch und technisch effizient und robust, fachlich jedoch naiv, ineffektiv und für die Benutzer wenig transparent ist. Dies gilt aus nahe liegenden Gründen in verstärktem Maße für die Entwicklung behindertengerechter Systeme. Andererseits führt eine Fokussierung auf die Mensch-Computer-Interaktion bei Vernachlässigung softwarearchitektonischer Erfordernisse zu Systemen, die zwar optimal in den Nutzungskontext eingebunden, aber nur sehr schwer wartbar, änderbar und adaptierbar sind.

Wird Software hingegen aus der Nutzungspraxis heraus *konstruktiv gestaltet*, wie hier vorgeschlagen, werden solche Probleme aus den beiden Lagern weitgehend vermieden. Dieser Ansatz lässt sich natürlich auch auf andere anspruchsvolle Domänen und Zielgruppen übertragen.

Hinsichtlich einer Transfer fördernden Ausgestaltung von Lernsystemen sind noch viele Fragen offen. So etwa die nach der Übertragbarkeit von Gestaltungsprinzipien „realer Räume“ aus Architektur und Design auf die Gestaltung „virtueller Räume“, um intuitiv handlungsleitende Spielwelten zu schaffen, die der Lebenswirklichkeit der Schüler möglichst nahe kommen und trotzdem überschaubar sind. Viel versprechend erscheinen hier insbesondere Untersuchungen, inwieweit sich phänomenologische [z.B. Me06], umweltsychologische [z.B. Me87] oder dramaturgische [z.B. Mi05] Ansätze eignen, Empathie, Motivation und Konzentration und damit auch die Transferleistung bei der anvisierten Zielgruppe zu steigern.

Bei allem Bemühen um maßgeschneiderte, behindertengerechte Softwaresysteme sollte jedoch nicht die in der Sonderpädagogik erhobene Forderung aus den Augen verloren werden, computergestütztes Lernen in Sonderschulen angemessen in das soziale Milieu des Unterrichts einzubetten und durch Lehrpersonen zu begleiten.

## Literaturverzeichnis

- [Bl99] Blesch, G.: Transfer computerunterstützten Lernens. In: [La99], S. 137-156.
- [Br06] Bröner, M.: Arbeiten, Lernen, Spielen – Der Einsatz des Computers bei Schülern mit geistiger Behinderung. Verlag modernes leben, Dortmund, 2006.
- [Ca68] Carnap, R.: Logische Syntax der Sprache. Wien, 1968.
- [Co03] Constantine, L. et al.: Usage-Centred Design and Software Engineering: Models for Integration. International Conference on Software Engineering, pp 54-61, 2003.
- [Ei07] Eitner, A.: Evaluation der Usability eines Lernspiels für geistig behinderte Kinder und Jugendliche. Diplomarbeit, Fakultät für Informatik und Ingenieurwissenschaften, FH Köln, April 2007.
- [EO07] Eitner, A.; Otrzonek, B.: Anforderungsermittlung für ein Lernspiel zur Erlangung von Wohnkompetenz. FSQ-Bericht Nr. 07-07-01, Forschungsschwerpunkt Software-Qualität, FH Köln, Juli 2007.
- [Fi07] Fischer, S.: Entwicklung eines Pilotsystems einer Lernsoftware für geistig behinderte Kinder und Jugendliche. Diplomarbeit, Fakultät für Informatik und Ingenieurwissenschaften, FH Köln, April 2007.
- [Fl84] Floyd, C.: A Systematic Look at Prototyping. In: Budde, R.; Kuhlkamp, K.; Methiassen, L.; Zyllighofen, H. (eds.): Approaches to Prototyping, Springer-Verlag, Berlin u.a., 1984.
- [Fr75] Frege, G.: Funktion, Begriff, Bedeutung. In: Patzig, G.(Hrsg.): Fünf logische Studien, Vandenhoeck&Ruprecht, Göttingen, 1975.
- [GJ03] Geßler, B.; Jochum, F.: KidS – Ein computergestütztes Verkehrslernspiel für geistig behinderte Kinder. In: medien + erziehung, Heft 3, 47. Jahrgang, Juni 2003.
- [Ha04] Hagebölling, H. (Ed.): Interactive Dramaturgies – New approaches in Multimedia Content and Design. Springer-Verlag, Berlin, Heidelberg, New York, 2004.
- [Ja01] Janisch, P.: Logisch-pragmatische Propädeutik – Ein Grundkurs im philosophischen Reflektieren. Velbrück Wissenschaft, Weilerswist, 2001.

- [Jo08] Jochum, F.: Computergestützte Lernhilfen für Schüler mit geistiger Behinderung – Teil1: Qualitätsanforderungen aus der Nutzungsperspektive. FSQ-Bericht, Forschungsschwerpunkt Software-Qualität, FH Köln, 2008. (in Arbeit)
- [Jo08a] Jochum, F.: Computergestützte Lernhilfen für Schüler mit geistiger Behinderung – Teil2: Methodische Grundlagen der Systementwicklung. FSQ-Bericht, Forschungsschwerpunkt Software-Qualität, FH Köln, 2008. (in Arbeit)
- [KL96] Kamlah, W.; Lorenzen, P.: Logische Propädeutik – Vorschule des vernünftigen Redens. Metzler-Verlag, Stuttgart, Weimar, 1996.
- [La99] Lamers, W. (Hrsg.): Computer- und Informationstechnologie – Geistigbehindertenpädagogische Perspektiven. Selbstbestimmtes Leben, Düsseldorf, 1999.
- [LM04] Lukoschek, A.; Manka, S.: Prototyping bei der Entwicklung von Lernsoftware für Kinder. Diplomarbeit, Fakultät für Informatik und Ingenieurwissenschaften, FH Köln, Mai 2004.
- [Me87] Mehrabian, A.: Räume des Alltags – Wie die Umwelt unser Verhalten bestimmt. Reihe Campus, Frankfurt/M, New York, 1987.
- [Me06] Meisenheimer, W.: Das Denken des Leibes und der architektonische Raum. Köln, 2006.
- [Mi05] Mikunda, C.: Der verbotene Ort oder die inszenierte Verführung. Econ-Verlag, Düsseldorf, 2005.
- [Mi97] Mittelstraß, J.: Konstruktive Begründungstheorie. In: Der Flug der Eule – Von der Vernunft der Wissenschaft und der Aufgabe der Philosophie (Kap. IV). Suhrkamp-Verlag, Frankfurt/M, 1997.
- [Mü01] Müller, K.: Prototypische Entwicklung einer individuellen computerbasierten Lernhilfe für einen geistig behinderten Jungen. Diplomarbeit, Fakultät für Informatik und Ingenieurwissenschaften, FH Köln, Januar 2001.
- [Pf07] Pfeffer-Hoffmann, C.: E-Learning für Benachteiligte – eine ökonomische und mediendidaktische Analyse. Mensch und Buch Verlag, Berlin, 2007.
- [Py03] Pyla, P. et al.: Towards a Model-Based Framework for Integrating Usability and Software Engineering Life Cycles. International Conference on Software Engineering, pp 54-61, 2003.
- [Sc97] Schienmann, B.: Objektorientierter Fachentwurf – ein terminologiebasierter Ansatz für die Konstruktion von Anwendungssystemen. Teubner-Verlag, Leipzig 1997.
- [Sc00] Schönfelder, J.: Softwareergonomische Richtlinien für Computerprogramme im Bereich der Geistigbehindertenpädagogik. In: Zeitschrift für Heilpädagogik 51, Heft 9, 2000, S. 368-377.
- [SL99] Studer, F.; Luder, R.: Methodisch-didaktische Überlegungen zum Computereinsatz in der Einzelförderung von geistig behinderten Menschen. In: [La99], S. 189-203.
- [Vo08] Voss, F.: Computerunterstützung bei der Entwicklung von Alltagskompetenzen für Menschen mit geistiger Behinderung – Eine methodenkritische Nutzer- und Anforderungsanalyse. Masterarbeit, Fakultät für Informatik und Ingenieurwissenschaften, FH Köln, Juli 2008. (in Arbeit)
- [WO05] Wedekind, H.; Ortner, E.; Inhetveen, R.: Informatik als Grundbildung (6 Aufsätze). In: Informatik\_Spektrum, Springer-Verlag, Berlin, Heidelberg, Heft 2/2004 – Heft 1/2005.

## Danksagung

Der GEW-Stiftung Köln danke ich für die zweijährige finanzielle Unterstützung (2005-2007) des Projekts *Computergestützte Lernhilfen für Kinder und Jugendliche mit geistiger Behinderung – Gestaltung nutzergerechter Mensch-Computer-Systeme* im Forschungsschwerpunkt Software-Qualität an der FH Köln, auf dessen Ergebnissen der vorliegende Beitrag teilweise aufbaut.