

1.6 Informatikgrundlagen und Mensch-Computer-Kommunikation

Ronald Hartwig¹, Michael Herczeg²

¹human interface.design Hamburg, ²Universität zu Lübeck

1 Einleitung

Ein zentraler Aspekt im Anwendungsbereich des Computer Supported Cooperative Learning (CSCL) ist die Entwicklung der dazu notwendigen Medien und Programme. Im Kapitel 4 wird beschrieben, wie die Umsetzung der für CSCL benötigten Programme und Medien erfolgen kann. An dieser Stelle sollen hingegen einige dazu notwendige grundlegende Konzepte der Informatik, die bei der Realisierung von Software für CSCL sowie bei der Entwicklung von Inhalten für CSCL, die auf Software beruhen, kurz vorgestellt werden. Es kann hier jedoch kein Wert auf Vollständigkeit hinsichtlich allgemeiner informatischer Grundlagen gelegt werden.

2 Überblick

Die Realisierung von Software im Zusammenhang mit CSCL findet sich vor allem in folgenden Formen:

1. *Autorensysteme, Content-Management-Systeme oder Entwicklungsumgebungen* für die Erstellung und Wartung der Inhalte (im Folgenden Module genannt);
2. *Lernraumsysteme (auch LMS „Learning Management System“ genannt)*, die den Lernprozess unterstützen, z.B. durch curriculare und terminliche Übersichten, oder Möglichkeiten zum Anfügen von Annotationen sowie dem Zwischenspeichern und Weiterverarbeiten von Inhalten. Dazu gehören auch *Kommunikations- und Kooperationsysteme*, die die Kommunikation und Zusammenarbeit zwischen den Lernenden untereinander und den Lernenden und Ihren Betreuern unterstützen, also Chatsysteme, Nachrichtensysteme, kollaborative Arbeitsumgebungen, gemeinsame Dokumente und virtuelle Räume;
3. *Inhalte der Module*, sofern sie über reine semantische Auszeichnung (HTML, XML) hinausgehen, wie zum Beispiel interaktive Präsentationen, Selbsttestaufgaben, Animationen, Simulationen oder Spiele.

Für alle diese Systeme können in der Arbeitswissenschaft (Ulich 2001), Psychologie (Zimbardo et al. 2003), Pädagogik (Kerres 2001; Schulmeister 2001) und Informatik entwickelte Theorien und Konzepte aufgegriffen und weiter entwickelt werden, um die effektive, effiziente und zufrieden stellende Nutzung zu gewährleisten. Dies wird unter dem Begriff der

Gebrauchstauglichkeit (engl.: *Usability*) zusammengefasst (ISO 9241). Diese befasst sich mit dem Zusammenspiel von Benutzern und interaktiven Software-Systemen, der so genannten *Mensch-Computer-Interaktion* oder *Mensch-Computer-Kommunikation* (engl.: „Human Computer Interaction“ kurz „HCI“) (Herczeg 1994; 2004). Daneben sind natürlich auch softwaretechnische Qualitätsaspekte, wie die Verfügbarkeit, Korrektheit und Robustheit, sowie die Wartbarkeit und Wiederverwendbarkeit für den Einsatz erheblich. Diese werden im Kapitel 5 genauer betrachtet.

Jede Art von Software ist prinzipiell nach dem EVA-Prinzip konzipiert: Eingabe-Verarbeitung-Ausgabe, d.h. eine Eingabe (das können Daten oder Ereignisse sein) wird aufgrund einer im System festgelegten Programmlogik deterministisch zu einer Ausgabe verarbeitet. Eingabeereignisse und Daten werden vom Benutzer an der so genannten *Benutzungsschnittstelle* (engl.: *User Interface* kurz *UI*) eingegeben (z.B. über die Tastatur) oder ausgelöst (z.B. durch Anklicken eines „Start“-Knopfes). Die Eingabedaten bzw. Eingabeereignisse können aber auch von anderen Systemen, z.B. Ereignisse von anderen Programmen oder Daten aus einem Netzwerk, stammen. Ebenso kann die Ausgabe an den Benutzer, zum Beispiel als Bilder und Zeichen über einen Bildschirm oder als Töne über einen Lautsprecher wieder zurück ausgegeben werden. In der Programmlogik ist festgelegt, wie das System auf die Eingabe reagieren soll, wie es die Eingabeinformationen weiterverarbeiten soll und wie es eine geeignete Ausgabe erzeugen soll. Das System selbst ist dabei in einem Zustand, der durch die Eingaben des Benutzers und nach den Vorgaben der Programmlogik verändert wird.

Eine Voraussetzung für die praktisch verwendeten Programmierkonzepte ist dabei, dass gleiche Eingaben bei gleichem Ausgangszustand des Systems und gleicher Programmlogik auch zum gleichen Ausgabeergebnis führen (*Determinismus*). In der Praxis ebenfalls wichtig ist die Eigenschaft, dass nach einer Eingabe, einem definierten Zustand und einer definierten Programmlogik nach definierter Zeit eine Ausgabe erfolgt (*Determiniertheit*).

3 Verteilte Systeme

Computersysteme werden heutzutage meist in *Computernetzen* (Tanenbaum 2000) zusammengeschaltet. Dabei können die verschiedenen beteiligten Computersysteme unterschiedliche Aufgaben wahrnehmen. Man spricht in diesem Zusammenhang auch von *verteilten Systemen*. Zunächst soll einmal betrachtet werden, welche Arten von derartigen Systemen im Bereich des CSCL Verwendung finden:

Als *Stand-Alone-Applikation* oder kurz *Applikation* wird Software bezeichnet, die auf dem Rechner des Benutzers selbst installiert ist und nur ihm zur Verfügung steht. Ein typisches Beispiel einer solchen Applikation ist ein Textverarbeitungsprogramm. Es verwendet die Eingaben des Nutzers oder andere vorgegebene Informationen und verarbeitet sie nach vorgegebenen Regeln zu einer neuen Ausgabe. Es sind auch Applikationen, zum Beispiel eine Uhr, denkbar, die nur durch ihren Aufruf einmal gestartet werden, und dann keine weiteren Eingaben erwarten und kontinuierlich eine Ausgabe liefern.

Im Gegensatz dazu stehen *verteilte Systeme*. Dabei kommunizieren über festgelegte Kanäle zwei oder mehr Systeme über ein Netzwerk miteinander. Dabei wird zwischen zwei Arten der Kooperation unterschieden: Client-Server-Systeme und Peer-to-Peer-Systeme.

Client-Server-Systeme haben eine vorgegebene Hierarchie und Aufgabenteilung. Das bekannteste Beispiel, gerade auch im Bereich des CSCL, ist sicher der *Web-Client* (Internet-Browser) auf der einen Seite und der *Content-Server* (z.B. ein LMS) auf der anderen Seite. Der Client dient dabei als Schnittstelle zum Benutzer und nimmt auf dessen Rechner zunächst Eingaben entgegen (zum Beispiel den Klick auf einen Link auf einer Webseite), übersetzt sie in eine Sprache für die Kommunikation zum Server, auf dem die Daten gespeichert sind, und schickt diese Daten an den Server.

Der Server wertet die vom Client des Benutzers übermittelte Anfrage aus, sucht oder berechnet die Ausgabedaten und gibt sie, zunächst ebenfalls wieder in einer maschinenlesbaren Form, wieder an den Client zurück. Der Client wiederum übersetzt diese Daten dann für die lokale Verwendung, zum Beispiel indem er die angeforderte Webseite anzeigt.

Wenn dem Client-Programm nur noch wenige, einfache Verarbeitungsschritte verbleiben und der Großteil der Informationsverarbeitung auf dem zentralen Server stattfindet, so spricht man von einem *Thin-Client*. Im umgekehrten Fall spricht man hingegen von einem *Rich-Client*, wobei beide Bezeichnungen die Vorteile der jeweiligen Lösung betonen sollen. Thin-Clients erheben geringere Anforderungen an die Leistungsfähigkeit des Rechners bzw. seiner Software beim Benutzer. Solche Systeme waren in lokalen Netzen als Terminal-Mainframe-Modelle („Großrechner“) lange Zeit verbreitet und wurden erst mit dem Aufkommen kostengünstiger, leistungsfähiger *Desktop-Systeme* auf der Basis von Personal-Computern (PC) teilweise verdrängt. PCs können viele Aufgaben sofort vor Ort erledigen und müssen nicht erst auf die vergleichsweise langsame und kostenintensive Datenübermittlung warten. Mit dem Aufkommen immer leistungsfähigerer und kostengünstigerer Vernetzungen zwischen den Rechnern wird das Client-Server-Prinzip z.B. in Form der Internet-Clients und -Server oder des so genannten *Application-Sharing* (Verteilung von Funktionalität auf verschiedene Computersysteme im Netz) aber zunehmend auch für das Internet attraktiver. Rich-Clients, die erhebliche Teile der Informationsverarbeitung selbst vornehmen und nur komprimierte Rohdaten über das Netz übertragen, empfehlen sich, um komplexe, auch stärker multimediale Interaktionsmöglichkeiten ohne Verzögerungen durch den zwischenzeitlichen Datentransfer und das Warten auf einen zentralen Server zu erlauben.

Ein wichtiger Vorteil von Client-Server-Systemen gegenüber Stand-Alone-Applikationen ist die zentrale Bereitstellung von Funktionalitäten und die daraus folgende einfachere Wartbarkeit. Schließlich muss eine Software so nur auf einem zentralen Rechner und nicht in den oftmals sehr heterogenen (Hardware-/Software) Umgebungen der Nutzer installiert und gepflegt werden. Ein weiteres Beispiel sind virtuelle Labore, bei denen mittels kleiner lokaler Client-Programme ein zentrales Labor gesteuert werden kann. Hier ist der Vorteil des vernetzten Systems, dass aufwändige und teure Systeme, die von den Lernenden oder einzelnen Lehrenden nur selten oder kurzzeitig verwendet werden, nur einmal zentral angeschafft und betrieben werden müssen und dann vielen, räumlich und zeitlich verteilten Lernenden zur Verfügung stehen. Auch umfangreiche Datenbestände können von Web-Servern oder Datenbanken häufig besser zentral gehalten werden. Den Lernenden werden auf deren Anfragen

hin dann nur die Teile übermittelt, die tatsächlich benötigt werden. Wenn allerdings gleich bleibende Inhalte, zum Beispiel statische Teile eines Lernmoduls oder größere Videodateien, immer wieder übertragen werden sollen, weicht man vom strengen Client-Server-Paradigma ab und sendet unveränderliche Inhalte nur einmal und lässt sie zur weiteren Nutzung auf dem Rechner des Lernenden verbleiben. Die heutigen Internet-Browser sind zu diesem Zweck nicht nur als reine Netz-Clients konzipiert, sondern auch in der Lage, als Stand-Alone-Applikation ohne Server, Daten, die lokal auf der Festplatte oder einer CD-ROM/DVD vorliegen, zu verarbeiten und dem Nutzer verfügbar zu machen. Sie können auch beide Betriebsarten mischen.

Im Bereich des CSCL sind Client-Server-Systeme mit Blick auf die Kooperation unentbehrlich. Kommunikation erfordert in der Regel einen gemeinsamen Kommunikationsraum oder zumindest eine zentrale Anlaufstelle, an der man Kooperations- und Kommunikationspartner auffinden kann. Typische Kommunikationsserver, die häufig Teil des LMS sind, sorgen dafür, dass Gruppen von Lernern und Lehrenden verwaltet werden und ihre Anwesenheit bei Bedarf an andere potenzielle Kommunikationspartner gemeldet wird (*Awareness*).

Wichtig für Client-Server-Systeme ist die ausreichende Leistungsfähigkeit und Zuverlässigkeit sowohl der Server als auch der hinführenden Netzwerke, so dass Warte- oder gar Ausfallzeiten für die Lernenden vermieden werden können. Weiterhin ist von technischer Seite zu beachten, dass öffentlich zugängliche Server immer dem Risiko von Angriffen ausgesetzt sind.

Peer-to-Peer-Systeme unterscheiden sich von Client-Server-Systemen dadurch, dass es keine festgelegte Rangfolge und Aufgabenteilung gibt, sondern dass ähnliche oder gleichartige Systeme miteinander kommunizieren. Sie finden sich häufig im Bereich der (synchronen) Kommunikation. Dadurch wird die häufig sehr datenintensive Kommunikation zwischen den Partnern nicht durch den potenziellen Engpass eines zentralen Servers gefährdet, sondern die Partner sprechen direkt miteinander.

Daneben existiert noch eine Vielzahl weiterer Netzformen, die aber hier nicht weiter im Detail vorgestellt werden (zum Beispiel *Lastverteilungssysteme*). Zukünftig interessant könnte das *Distributed-Computing* (verteiltes Rechnen) werden, um Aufgaben, die für ein einzelnes System zu groß sind, auf geeignete Teile und Teilsysteme in einem Netzwerk aufzuteilen.

4 Intelligente Systeme

Eine Besonderheit unter den verteilten Systemen stellen die so genannten *Agentensysteme* dar. Im Vergleich zu den bisher beschriebenen Systemen sind diese autonomer, d.h. sie arbeiten nachdem sie aktiviert wurden auch ohne direkte Benutzeraufforderung im Hintergrund weiter oder nehmen stellvertretend Teilaufgaben wahr. So können Agentensysteme zum Beispiel die Suchgewohnheiten eines Benutzers analysieren und darauf basierend im Hintergrund bereits selbstständig weitere Suchanfragen durchführen, um diese dann bei Bedarf schneller bereitstellen zu können (Teuteberg, 2001). Die Unterscheidung zu anderen Systemen ist allerdings durch die Vielfalt von Agentensystemen nicht sehr trennscharf, denn letz-

endlich sind auch Agentensysteme nur Anwendungen, die früher oder später als Teil eines Client-Server oder eines Peer-to-Peer-Netzwerkes wieder mit dem Nutzer in Kontakt treten. Das Ziel ist, dass so genannte *Intelligente Agenten* dem Benutzer Routinetätigkeiten abnehmen könnten.

Ähnlich den intelligenten Agenten verspricht man sich von *Adaptiven Systemen*, dass es gelingen wird, dass sich das System dem Nutzer so intelligent anpassen kann, dass dieser bei seiner Arbeit bzw. in seinem Lernprozess unterstützt wird. Das System wertet dazu die ihm zur Verfügung stehenden Nutzungsinformationen aus, sei es aus Abfragen, Datenbanken oder aus statistischer Auswertung des Nutzungsverhaltens und interpretiert diese. Daraus werden dann Schlussfolgerungen über die für den Benutzer potenziell hilfreiche Unterstützung gewonnen werden.

Dieser Schlussfolgerungsprozess ist aber für komplexe Aufgaben- und Handlungskomplexe nicht einfach. Es besteht die Gefahr, dass zu stark vereinfachte Benutzermodelle Benutzerhandlungen falsch interpretieren. Es ist zum Beispiel durchaus fragwürdig, ob die Häufigkeit eines Aufrufs tatsächlich etwas über die Qualität des Inhaltes aussagt oder ob nicht ungewollt selbst verstärkende Effekte einen Zirkelschluss (häufige Anwahl → besonders hervorheben → noch häufigere Anwahl) erzeugen. Die nötige Intelligenz und auch Erfahrung, die zur Interpretation einer Benutzerhandlung eines Menschen notwendig ist, darf dabei nicht unterschätzt werden. Sonst besteht die Gefahr, dass der Lerner am Ende zunächst einmal ein mentales Modell darüber bilden muss, was das System von ihm glaubt verstanden zu haben, oder aber dass der Eindruck entsteht, man habe keine Kontrolle darüber, was in welcher Form dargeboten wird.

Ein weiteres Problem der (automatischen) adaptiven Systeme als auch der vom Benutzer selbst anzupassenden *adaptierbaren Systeme* ist, dass bei der Kommunikation über das System mit mehreren Personen kein gemeinsames Systemmodell mehr besteht, d.h. Benutzer A und Benutzer B haben beim Blick auf den gemeinsamen Informationsraum unterschiedliche Sichten und bekommen unter Umständen Probleme dabei, über das Gleiche zu sprechen, da es sich für beide unterschiedlich darstellt.

Ein weiteres Konzept intelligenter Unterstützungssysteme wurde bereits lange Zeit diskutiert und in ersten Prototypen entwickelt. Es handelt sich dabei um *Intelligente Tutoren*, die Benutzer insbesondere in ihrem Lernprozess beobachten und führen sollen (Sleeman & Brown 1982). Diese Systeme waren als *Wissensbasierte Systeme* oder auch *Expertensysteme* realisiert und enthielten eine formale Repräsentation des zu lernenden Wissens. Bei der Realisierung dieser Systeme hat es sich gezeigt, wie schwierig es ist, menschliches Wissen in Computersystemen zu repräsentieren. Die besten Ergebnisse wurden mit fachlich sehr eng gefassten Systemen erreicht, die sehr genau definierbares und begrenztes Wissen enthielten und dieses dann zur Steuerung oder Beurteilung der Lernprozesse nutzten. Eine weitere Schwierigkeit war die geeignete Nutzung des repräsentierten Wissens. Wann und wie sollten die Lerner informiert oder angeleitet werden? Für die Zukunft sind in gut definierten Bereichen intelligente Tutoren vorstellbar, die mit den oben vorgestellten Ansätzen intelligenter Agenten und adaptierbarer Systeme zusammenwirken.

5 Mensch-Computer-Interaktion

Die beschriebenen Unterscheidungen zwischen den verschiedenen Systemkonzepten und Vernetzungsmodellen sind für den Benutzer häufig unsichtbar. Sie sind eher aus technisch-ökonomischer Sicht interessant, wenn es darum geht, sich für eine bestimmte Technologie zu entscheiden. Für den Lernenden hingegen ist von zentralem Interesse, ob ihn die ihm zur Verfügung gestellten Medien und Hilfsmittel tatsächlich beim Lernen unterstützen. Entscheidend sind dafür zunächst natürlich die pädagogische und didaktische Konzeption des Angebotes und die sinnvolle Einbettung in ein möglicherweise kollaboratives Gesamtkonzept. Dies wird in den anderen Abschnitten dieses Kapitels ausführlich beschrieben. Daneben ist aber auch wichtig, dass das Medium selbst auch die Leistungsfähigkeit des Menschen berücksichtigt. Die *Ergonomie* beschäftigt sich, aus der Arbeitswissenschaft kommend, allgemein damit, Werkzeuge so an den Menschen anzupassen, dass dieser möglichst schädigungsfrei, effektiv, effizient und zufrieden stellend mit ihnen arbeiten kann. Die *Software-Ergonomie* beschäftigt sich entsprechend mit Software-Werkzeugen.

5.1 Modelle der Mensch-Computer-Kommunikation

Die Kommunikation zwischen einem Menschen und einem Computersystem läuft, wie übrigens auch die Kommunikation zwischen Menschen, in Form eines *Dialoges* ab (Herczeg 1994; 2004). Dabei senden sich die beiden Kommunikationspartner gegenseitig Nachrichten zu. Diese Nachrichten enthalten Informationen, die das Wissen und daraufhin das Verhalten der beiden Kommunikationspartner verändern können. Sie erzeugen also eine gegenseitige Wirkung, eine Wechselwirkung, auch *Interaktion* genannt.

Die Nachrichten, die sich die beiden Kommunikationspartner zusenden, müssen in irgendeiner Weise präsentiert werden. Dies kann zum Beispiel in Form von Text und Graphik oder auch in Form von Tönen (Einzeltöne, Klänge, Musik) oder bewegten Bildern (Animation, Film) erfolgen. Diese vielfältigen Formen Informationen in Nachrichten zu verpacken nennt man *Multimedia*, also die Verwendung verschiedener Medien zur Kommunikation.

Bei der Auseinandersetzung eines Menschen mit einer Anwendungswelt, zum Beispiel bei der Arbeit, dem Lernen oder beim Spiel, kann der Computer unterschiedlichste Rollen spielen:

- Computer als *Arbeitsmittel* (Ressource);
der Mensch setzt den Computer ein
- Computer als *Werkzeug* (Funktion, Automat);
der Mensch bedient den Computer
- Computer als *Partner* (Assistent, Agent, Partner, Überwacher);
der Mensch kommuniziert mit dem Computer
- Computer als *Medium* (Mittler zur Anwendungswelt);
der Mensch interagiert über den Computer
- Computer als *Virtuelle Realität* (der Computer wird unsichtbar);
der Mensch bewegt sich in einer künstlichen Welt

Derartige Rollen eines Computers in der Mensch-Computer-Kommunikation nennen wir auch *Kommunikationsparadigmen*. Sie spielen als Basiskonzepte eine wichtige Rolle bei der Konzeption und der Gestaltung interaktiver Computersysteme.

Im Bereich des CSCL sind insbesondere die Kommunikationsparadigmen Partner und Medium von besonderer Bedeutung. Das Partnermodell (*Assistent*) haben wir schon im Zusammenhang mit Agentensystemen kennen gelernt. Das Modell des Mediums tritt immer dann auf, wenn der Computer zugunsten der Wahrnehmung eines Lerninhaltes in den Hintergrund tritt. Man spricht dann auch von *Lernmedien*. CSCL kann in besonderen Fällen auch in naturalistisch visualisierten dreidimensionalen *Lernräumen* stattfinden, in denen sich die Lerner virtuell treffen. In einem solchen Fall wird das Modell der virtuellen Realität benutzt.

5.2 Software-Ergonomie

Die Realisierung von CSCL-Systemen erfordert die anwendungs- und benutzergerechte Gestaltung dieser Systeme (Hartwig et al. 2002a; 2002c; Herczeg 1994; 2004), da der Kommunikations- und Lernprozess sonst wesentlich durch die Auseinandersetzung mit dem System behindert würde.

Dies bedeutet, dass – eine geeignete didaktische Konzeption vorausgesetzt – die Bedienung des Lernmediums und der darin eventuell enthaltenen interaktiven Elemente die physiologischen und kognitiven Leistungsgrenzen des Menschen berücksichtigt. So ist beispielsweise die visuelle Wahrnehmung durch die Fähigkeiten der Augen aber auch der Leistungsfähigkeit der weiteren Verarbeitungsprozesse im Gehirn begrenzt. Schriften, die zu klein oder zu kontrastarm dargestellt werden, können nicht richtig wahrgenommen werden, wohingegen zu viele, nur sehr kurz dargestellte Informationen die Merkfähigkeit überfordern können. Ähnliches gilt für die Wahrnehmung von Tönen oder für andere Sinnesmodalitäten. Neben der reinen Informationsaufnahme ist auch die Informationsverarbeitung innerhalb der kognitiven Prozesse des Lernalters begrenzt.

Ähnliches wie für die Wahrnehmungsprozesse gilt auch für die Handlungsprozesse, bei denen ausgehend von einer Zielsetzung und einer Handlungsplanung einzelne Aktionen mit Hilfe des Computersystems durchgeführt und eventuell korrigiert werden müssen.

5.3 Gebrauchstauglichkeit

Um die Software-Ergonomie praktisch umzusetzen und ursprünglich in Arbeitskontexten operationalisierbar bewerten zu können, wurde das Konstrukt der Gebrauchstauglichkeit (engl.: Usability) entworfen. In der ISO 9241 (Teil 11) wird diese als das Produkt aus *Effektivität*, *Effizienz* und *Zufriedenheit* bezogen auf eine bestimmte Nutzergruppe, einen bestimmten Nutzungskontext und eine bestimmte Aufgabe definiert. Besonders wichtig ist dabei die Einschränkung auf *bestimmte* Nutzergruppen, Nutzungskontexte und Aufgaben, d.h. eine universale Gebrauchstauglichkeit (auch umgangssprachlich als „Benutzerfreundlichkeit“ bezeichnet) kann es grundsätzlich so nicht geben.

Aus der Definition ergibt sich aber auch bereits, wie die Gebrauchstauglichkeit festzustellen ist. Zunächst muss geprüft werden, ob das angebotene Lernmittel oder der Lernraum dem

Lernenden überhaupt ermöglicht, seine Teilaufgabe innerhalb seines (selbstdefinierten) Lernprozesses durchzuführen. Möchte der Lernende beispielsweise eine Nachricht an einen Partner versenden, so kann relativ leicht geprüft werden, ob ihm dies gelingt. Der Grad der Zielerreichung definiert somit die Effektivität. Dann kann im Sinne der Effizienz geprüft werden, wie viel Aufwand dazu notwendig war. Dies kann als Vergleichsmaß verwendet werden, gegenüber alternativen Arbeitsformen. Beispielsweise kann man vergleichen, wie viel aufwändiger ein Versand einer Nachricht per Fax gegenüber einem Versand per E-Mail wäre. Als Drittes wird dann geprüft, wie zufrieden der Benutzer mit dem Ablauf dieses Vorganges im Allgemeinen und mit dem Werkzeug im Besonderen war.

Diese Art der Bewertung ist im Abschnitt Qualitätsmanagement (Abschnitt 4.5) noch einmal thematisiert. An dieser Stelle sei ausdrücklich davor gewarnt, Zufriedenheit als *hinreichendes* Kriterium für Gebrauchstauglichkeit anzusehen. Faktoren wie die „resignative Zufriedenheit“ (Reduktion der Ansprüche) und andere psychologische Abwehrmechanismen zwingen zu großer Sorgfalt bei der Planung der empirischen Erhebung der Zufriedenheit und dazu, auch die Effektivität und Effizienz analytisch ausreichend zu berücksichtigen.

Wichtig ist, dass die ergonomische Gestaltung eines interaktiven Systems bereits bei dessen Planung einsetzt und nicht erst bei der Gestaltung der Benutzungsschnittstelle. Die Kenntnis und Berücksichtigung von Nutzungskontext, spezifischen Eigenschaften und Anforderungen der Benutzer sowie deren Ziele und Aufgaben sind essentiell für den Erfolg der Entwicklung und des späteren Einsatzes eines interaktiven Systems.

Im Prozess der Erstellung von Systemen aus dem Bereich CSCL ordnen sich die Softwaretechnik und Software-Ergonomie gewissermaßen „unterhalb“ der Mediendidaktik ein und helfen anhand von Nutzungsszenarien die Anwendungsabläufe proaktiv zu gestalten (siehe z.B. Hartwig et al. 2002b), anstatt reaktiv nur noch Mängel festzustellen. Einige wenige, aber aussagekräftige ergonomische Qualitätssicherungsmaßnahmen, wie zum Beispiel Benutzer-tests, stellen dann sicher, dass am Ende nicht die Bedienbarkeit der Umsetzung sondern das didaktische Konzept im Mittelpunkt der Lernaufmerksamkeit stehen kann.

5.4 Dialogprinzipien

Aus dem Bereich der Software-Ergonomie sind Kriterien und Prinzipien entstanden, die einige der fundamentalen Anforderungen an ein interaktives System mit Blick auf dessen Gebrauchstauglichkeit beschreiben (ISO 9241). Sie sollen helfen, die Medien und Werkzeuge, die dem Lerner für die Bewältigung seiner Teilaufgaben innerhalb des CSCL Lernprozesses angeboten werden, zu planen und zu bewerten:

5.4.1 Aufgabenangemessenheit

Die wichtigste Frage ist: „*Bietet das System alle benötigten Funktionen?*“ und dies in einer Form, die dem Nutzer auch wirklich dienlich ist. Beispielsweise ist ein E-Mail-Programm zum Versenden von Textnachrichten vielleicht aufgabenangemessen, aber weniger zur gemeinsamen Arbeit an einem zentralen Dokument geeignet, denn es entstehen leicht Inkonsistenzen. Dieses Prinzip vereint bereits die beiden Hauptfragen: „*Leistet es was es soll?*“ und „*Ist der Aufwand der Aufgabe angemessen?*“.

Zur Bewertung des Aufwandes stehen im Kontext Lernen keine absoluten Zahlen zur Verfügung, aber ein Vergleich zu einer papiergestützten Arbeitsform gibt erste Anhaltspunkte. Wenn das dauerhafte Markieren eines Abschnittes in einem Papierdokument mittels eines Textmarkers in Sekunden geschehen kann, so kann dies in einem Lernraum schwierig (z.B. Herauskopieren in eine andere Anwendung und dann dort markieren = ineffizient) oder gar unmöglich (ineffektiv) sein. Dabei sind solche Punkte dann im Gesamtzusammenhang zu sehen und es ist insbesondere zu prüfen, ob die gewählte Form dann zumindest per saldo noch einen Nutzen gegenüber einer anderen Alternative bietet.

Häufig setzen Formen des so genannten „*Blended Learning*“ hier an und mischen die Vorteile traditioneller Lehr- und Lernformen, die sich in ihrer Handhabung und/oder Nützlichkeit als (noch) überlegen erweisen mit computergestützten Möglichkeiten. Aus Sicht der Gebrauchstauglichkeit ist dies zu begrüßen und bietet aber auch die Perspektive, an welchen Stellen noch ungenutzte Potenziale einer computerbasierten Lösung liegen.

5.4.2 Steuerbarkeit

Ein wesentliches Merkmal ergonomischer interaktiver Systeme ist die Steuerbarkeit des Ablaufs durch den Benutzer. Sowohl Reihenfolge als auch Geschwindigkeit können vom Benutzer, soweit von der Aufgabe her sinnvoll, selbst bestimmt werden. Dieses kommt den Ideen moderner konstruktivistischer Lerntheorien nahe (Niegemann 2001), die beim Lernen ebenfalls Selbstständigkeit im Lernprozess als entscheidendes Qualitätsmerkmal ansehen. Doch didaktische Gründe können im Einzelfall auch gegen bestimmte Freiheitsgrade sprechen, d.h. eine weniger steuerbare Alternative kann sich als aufgabenangemessener erweisen. Hier zeigt sich, dass die Prinzipien weder voneinander unabhängig noch widerspruchsfrei sind. Es ist im Einzelfall zu prüfen und abzuwägen, ob und wie der Nutzen einer freieren Alternative den einer mehr geführten Alternative überwiegt. Prinzipiell gilt aber gerade im Bereich kleinerer Teilaufgaben, z.B. Ansehen eines Videos, Kommunizieren in einem Chat-system oder Ausprobieren einer Simulation, dass die Entscheidung, wann welche Vorgehensweise den Benutzer optimal unterstützt, am besten diesem selbst überlassen bleibt.

5.4.3 Individualisierbarkeit

Der Lernende sollte nicht nur die vorhandenen Möglichkeiten selbstgesteuert anwenden können, sondern, in gewissen Grenzen, das System auch seinen Gegebenheiten und Vorlieben anpassen können (adaptierbare Systeme). So verhindern beispielsweise feste Seitengrößen häufig das sinnvolle parallele Arbeiten mit mehreren Inhalten oder kooperativen Werkzeugen oder die Anpassung an leistungsfähigere Monitore.

Wie bereits bei der Adaptivität erwähnt, ist auch bei Individualisierbarkeit grundsätzlich der potenzielle Nutzen einer individuellen Anpassung gegen die möglichen Nachteile abzuwägen. So können ungünstige Einstellungen für den Benutzer unbemerkt z.B. wichtige Informationen verdecken oder die Lesbarkeit erheblich verschlechtern. Versuche haben gezeigt, dass Benutzer mit selbst angepassten Systemen teilweise deutlich schlechter arbeiten konnten, als mit vorgegebenen Lösungen, da sie mangels entsprechender Kenntnisse im Sinne der Aufgabe ungünstige Einstellungen gewählt hatten.

5.4.4 Erwartungskonformität

Es ist bei der Bedienung eines Systems meist hinderlich, wenn sich interaktive Elemente anders verhalten, als man es anhand ihres Aussehens oder ihrer Positionierung erwarten würde. Werden beispielsweise Inhalte in Form von Web-Seiten dargeboten, dann ist es für die Lerner belastend, wenn sich diese dann nicht wie solche verhalten. Klassisches Beispiel ist dabei die Kodierung von Links. Wenn die Links in einem Dokument blau und unterstrichen und im nächsten rot und fett gedruckt sind, so muss der Lerner immer wieder nicht nur den Lernstoff erlernen, sondern auch das Bedienkonzept der Web-Seite.

Bei der Entwicklung der Systeme wird leider häufig übersehen, dass diese neben vielen anderen im täglichen Ablauf verwendet werden und der Lerner sich wesentlich weniger um das Konzept einer Web-Site oder eines Systems kümmern kann, als der Entwickler, der sich über einen längeren Zeitraum ununterbrochen damit beschäftigt. Auch die Nutzung von Handbüchern oder Online-Hilfen wird in der Regel überschätzt. Schließlich ist der Lerner gerade dabei, sich ein mentales Modell einer häufig komplexen Wissensdomäne anzueignen und jeder Aufwand, der für das Erlernen des vorgesehenen Systems verwendet wird, verringert die Effizienz im eigentlichen Lernprozess.

Natürlich kann es auch hier didaktisch begründete Ausnahmen geben, um gewollt Überraschungen oder Unsicherheiten zu provozieren, aber überall dort, wo dies nicht ausdrücklich vorgesehen ist, sollte darauf geachtet werden, dass alles sich so leicht wie möglich bedienen lässt und keine Überraschungen auftauchen. Ziel muss eine Gestaltung sein, die konsistent mit den bei den Lernenden zu erwartenden Kenntnissen über die Systembedienung ist.

5.4.5 Selbstbeschreibungsfähigkeit

Ähnlich verhält es sich mit der Selbstbeschreibungsfähigkeit. Soweit irgend möglich sollte auch bei bisher unbekannten Bedienelementen bereits aus deren Darstellung ersichtlich werden, was sie beinhalten bzw. auslösen. So sollten beispielsweise Bedienelemente einer interaktiven Simulation erwartungskonform sein und damit ähnlich den Schaltflächen, die der Nutzer aus dem Alltag der Rechnerbedienung kennt.

Damit Lerner nicht durch aufwändige Hilfesysteme die Nutzung der Lernumgebung und der Zugriff auf die Inhalte erst erlernen müssen, muss das Ziel sein, unmittelbar verständliche Systeme zu schaffen, die keinen nennenswerten Lernaufwand erfordern. Dies kann durch Wiederverwendung bekannter Dialogkonzepte erfolgen oder durch metaphorische Ansätze, bei denen Erfahrungen aus der meist materiellen Welt auf die Bedienung des Systems übertragen werden können (Dutke 1994).

5.4.6 Fehlertoleranz

Benutzer machen Fehler. Im Zusammenhang mit der Ergonomie eines Systems im Kontext CSCL ist zunächst einmal von Fehlern bei der Bedienung des Systems die Rede und nicht etwa von inhaltlichen Fehlern, die eventuell ein geplanter Teil des Lernprozesses sind.

Die Mindestanforderung an ein System muss deshalb sein, dass die Fehler nicht sofort irreversible Folgen haben. D.h. bevor ein Benutzer versehentlich die Arbeit einer ganzen Gruppe mit einem Klick vernichtet, muss er entweder noch einmal gewarnt werden und so die Mög-

lichkeit bekommen, seinen Fehler zu bemerken und zu vermeiden und/oder es muss danach eine Möglichkeit geben, den Fehler wieder rückgängig zu machen (Undo-Funktion). Sind solche Mechanismen nicht vorhanden und dem Benutzer ist das Risiko bewusst, so wird die gesamte Bedienung mit Angst vor einem solchen Fehler belastet sein und so die Zufriedenheit aber auch die Effizienz negativ beeinflusst.

Darüber hinaus ist es aus ergonomischer Sicht wünschenswert, wenn bei durch das System erkannten Fehlern, der Benutzer nicht nur einen höflichen Hinweis bekommt, sondern auch konstruktive Hinweise zur Korrektur. Automatische Fehlerkorrekturen sind hingegen mit Vorsicht zu betrachten. Sie können dazu führen, dass bestimmte Fehlhandlungen sich „einschleifen“, da sie ohne negative Folge bleiben. Dies kann den Wechsel zu einem weniger fehlertoleranten System erschweren und unerwünschte Lerneffekte nach sich ziehen.

Völlig selbstverständlich ist die Forderung nach Fehlerrobustheit, d.h. dass ein System durch eine Fehlhandlung des Benutzers nicht in einen undefinierten Zustand geraten darf (keine „Systemabstürze“).

5.4.7 Lernförderlichkeit

Im Kontext des CSCL missverständlich ist die Forderung nach der Lernförderlichkeit der Gestaltung eines Systems. Gemeint ist in diesem Fall nicht, wie weit das System das Lernen der im Lernszenario vorgesehenen Inhalte und Kompetenzen unterstützt (das fiele in den Bereich der Aufgabenangemessenheit), sondern wie das Erlernen der Bedienung des Systems unterstützt wird. Dies ist eng verwandt zu den Forderungen nach Selbstbeschreibungsfähigkeit und Erwartungskonformität aber auch Fehlertoleranz. Wenn etwas Neues im System angeboten wird, so soll der Benutzer bekannte Modelle soweit möglich darauf abbilden (Erwartungskonformität), neue Inhalte und Konzepte aus der Gestaltung selbst ableiten (Selbstbeschreibungsfähigkeit) und dann gefahrlos ausprobieren (Fehlertoleranz) können.

7 Ausblick

Angesichts der dargestellten informatischen und nachrichtentechnischen Möglichkeiten können computergestützte Lernplattformen entstehen, die völlig neue Nutzungskonzepte des Lehrens und Lernens ermöglichen. Verteilte, intelligente und mobile Systeme stellen neue Präsentations-, Interaktions- und Kommunikationsräume für das Lernen bereit. Es ist Aufgabe der Planer und Entwickler dieser Systeme das Wissen um die Möglichkeiten aber auch Grenzen der Mensch-Computer-Interaktion frühzeitig im Entwicklungsprozess zu berücksichtigen. Auf diese Weise kann der beabsichtigte Lernprozess in den Mittelpunkt der Entwicklung gestellt werden und die Mediendidaktik genauso Berücksichtigung finden, wie software-technische und software-ergonomische Aspekte. Neue Nutzungskonzepte müssen jedoch zeigen, dass sie im Sinne einer Gebrauchstauglichkeit effektiv, effizient und zufrieden stellend die Aufgaben der Lehrer und Lerner im Lernprozess unterstützen. Neue Medien und Werkzeuge müssen ihre Nützlichkeit beweisen, um eine sinnvolle Ergänzung oder aber sogar ein Ersatz für herkömmliche bewährte Formen des Lehrens und des Lernens darzustellen.

Literatur siehe Gesamtliteraturverzeichnis im Buch.