

SOFTWARE-ERGONOMIE: STAND UND ENTWICKLUNGSTENDENZEN

H.-J. Bullinger, K.-P. Fährnrich, J. Ziegler, Stuttgart

Historische Entwicklung der Software-Ergonomie

Software-Ergonomie als ein interdisziplinäres Gebiet beschäftigt sich mit der Analyse, Gestaltung und Evaluation interaktiver Rechnersysteme. Durch die enge Kopplung der Software-Ergonomie (SWE) an den Bereich der technischen Entwicklung von Rechnersystemen und Softwareprodukten interagiert sie auf zwei Weisen mit der technischen Gestaltung:

- Zum einen greift sie neue technische Entwicklungen auf, um damit neue Lösungsmöglichkeiten für die Fragen der Rechnerbenutzung zu erarbeiten,
- zum anderen kann sie von einem benutzerorientierten Standpunkt aus Anstöße und Richtlinien für Entwicklungen liefern.

Eine lange Phase in der Verwendung von Rechnersystemen war sicherlich dadurch gekennzeichnet, daß die treibende Kraft in der Technikentwicklung lag und Benutzbarkeitsaspekte nur implizit mitentwickelt wurden. Selbst in den frühesten Zeiten der Computerentwicklung gab es allerdings schon vereinzelt Stimmen, die darauf hinwiesen, daß die Verwendung von Computersystemen nicht ausschließlich als technisches Problem zu sehen sei, wie das folgende Zitat belegt:

- "Any machine coding system should be judged quite largely from the point of view of how easy it is for the operator to obtain results" (Mauchley, 1947, zit. nach Gaines, 1986)

Solche Aussagen, die allerdings seltene Einzelfälle darstellten, mußten im Computerbenutzer den professionellen Spezialisten, den Computeroperator, sehen. Es bedurfte der Entwicklung interaktiver Time-Sharing-Systeme in den frühen 60er Jahren (z.B. das MIT MAC System von 1963, das RAND/JOSS System oder das BASIC System des Dartmouth College, siehe Gaines, 1986), um langsam den Aufmerksamkeit auch auf andere Benutzergruppen als den spezialisierten Programmierer zu richten. Schon damals wurde vereinzelt die zukünftige Bedeutung weiterer Benutzergruppen vorausgesehen, wie es sich in dem folgenden Zitat ausdrückt:

- "The future is rapidly approaching when professional programmers will be among the least numerous and least significant system users." (Mills, 1967, zit. nach Gaines, 1986)

Auf der wissenschaftlichen Seite markiert wohl die Arbeit von Shackel (1959) zur Gestaltung von Rechnerkonsolen den Beginn der Beschäftigung mit er-

gonomischen Fragen des Rechneinsatzes. Die erste multidisziplinäre Konferenz in diesem Bereich dürfte das IBM Scientific Computing Symposium on Man-Machine-Communication in Yorktown Heights im Jahre 1965 gewesen sein (vgl. Gaines, 1986). Schon die Zusammensetzung dieser Konferenz deutete auf die starke Interdisziplinarität des Fachgebietes hin. 1969 erschien die erste Fachzeitschrift in diesem Bereich, das International Journal of Man-Machine-Systems. Anfang der 70er Jahre erschienen die ersten wichtigen Buchveröffentlichungen, die sich mit software-ergonomischen Fragen auseinandersetzten (Martin: Design of Man-Computer-Dialogues, 1973; Weinberg: Psychology of Computer Programming, 1971; Sackman: Man - Computer Problem Solving, 1970). Seit 1981 wird in den USA die CHI (Computer-Human Interaction - offizielle Konferenz der ACM) abgehalten. In den letzten beiden Jahren hat sich massiv das internationale und nationale Normungswesen dieser Fragestellungen angenommen (z.B. ISO TC 159/SC4/WG5).

Für den europäischen Bereich zeichnet sich erstmalig mit dem HUFIT-Vorhaben (HUFIT: Human Factors Laboratories in Information Technology) (vgl. Fähnrich & Ziegler, 1987) im Rahmen des ESPRIT Programmes eine breite Zusammenarbeit auch insbesondere mit der Industrie ab.

In Deutschland sind, abgesehen von einigen Vorläuferarbeiten (oder Arbeiten in verwandten Gebieten wie der Anthropotechnik), seit Beginn der 80er Jahre ein verstärktes Interesse und zunehmende Aktivitäten im Bereich der Software-Ergonomie zu beobachten. Seit 1980 findet regelmäßig jährlich die Tagung Mensch-Maschine-Kommunikation statt, in der sich ein Kreis von interessierten Wissenschaftlern und Praktikern zusammengefunden hat. Für eine breitere Öffentlichkeit stellte sich das Fachgebiet erstmals mit der ersten ACM-Tagung zur Software-Ergonomie 1983 in Nürnberg (Balzert, 1983) dar. Die Fortsetzung dieser Tagungsreihe 1985 in Stuttgart (Bullinger, 1985), sowie jetzt 1987 in Berlin reflektiert ein steigendes Bewußtsein sowie eine konsolidiertere wissenschaftliche Auseinandersetzung mit dem Problemkreis ergonomischer Softwaregestaltung. Diese Tagungen haben weit über den deutschen Raum hinaus Beachtung gefunden. Eine vermehrte internationale Beachtung der Arbeiten im deutschsprachigen Raum wird sicherlich auch durch das Abhalten der zweiten internationalen IFIP-Konferenz in Stuttgart im September dieses Jahres gefördert werden.

Für eine weitere vertiefte Diskussion der historischen Entwicklung des Gebietes Software-Ergonomie vergleiche auch (Gaines, 1986), sowie Shackel (1985).

Arbeitsgebiete der Software-Ergonomie

Forschung im Bereich der Software-Ergonomie ist angewandte Forschung. Es könne viele Parallelen zu klassischen ergonomischen Themenstellungen gezogen werden: Die drei Hauptforschungsgebiete beziehen sich auf die Analyse, die

Gestaltung und die Evaluation von interaktiven Rechnersystemen im weitesten Sinne und den darauf implementierten Softwareprodukten. Shackel formuliert dazu:

- ...the study of, and the application of human factors knowledge to all aspects of the relation between the human, the machine and the environment which directly influence the safe, efficient, acceptable and satisfying usage of IT systems."

Im Bezug auf die in der ISO anstehenden Normierungsarbeiten wird momentan ein Referenzmodell für software-ergonomische Aktivitäten diskutiert, das den Rahmen für den Bereich der Software-Ergonomie absteckt:

- Software-ergonomische Gestaltungsarbeiten sollten immer von einer umfassenden Betrachtung und Analyse des zugrundeliegenden Arbeitssystems ausgehen. Dabei sind zum einen die im Arbeitssystem zu verrichtenden Aufgaben (Aufgabenanalysen, Repräsentation von Arbeitsaufgaben, Aufgaben- und Tätigkeitsinhalte, spezifische Anwendungsbereiche etc.) von Interesse. Zum anderen werden verschiedene Benutzer bzw. Benutzergruppen (anatomische, physiologische und psychologische Fähigkeiten und Fertigkeiten, kommunikatives Verhalten, Benutzermodelle) betrachtet. Die Fortentwicklung der Technologie (Systemleistungen bzw. -limitationen wie Kapazitäten, Geschwindigkeiten, Antwortzeiten, Verlässlichkeit, neue qualitative Entwicklungen) sind zu beachten.
- Im gestalterischen Bereich erstrecken sich die Arbeiten über die Spezifikation der im Arbeitssystem zu verwendenden Werkzeuge, der Funktionalität, der Auslegung bzw. Realisierung unterschiedlicher Dialogformen und Dialogtechniken sowie letztendlich der Gestaltung der Schnittstelle des Systems mit dem Benutzer an der Oberfläche (Informationsdarstellung, Gruppierung und Codierung etc.). Peripher zu beachten sind hierbei Fragen der Gestaltung der entsprechenden Arbeitsmittel (Tastaturen, Bildschirm etc.) sowie der verwendeten Anpaßmittel (Stuhl, Tisch etc.). Durchaus Einfluß auf die direkte Mensch-Maschine-Schnittstelle hat außerdem die Arbeitsumgebung (Licht, Lärm, Klima etc.).
- Auf einer Metaebene wird im Bereich der Methoden- und Werkzeugentwicklung für die Implementation von Benutzerschnittstellen gearbeitet. Stichworte in diesem Bereich sind Rapid-Prototyping, User-Interface-Management-Systeme, Daten- und Methodenbanken für den Entwickler. In diesem Kontext wird auch verstärkt der tatsächliche und mögliche Einfluß der Software-Ergonomie auf den Entwicklungsprozeß (Design-Cycle-Model) von informationstheoretischen Produkten untersucht. Besonders stark diskutiert werden dabei momentan sogenannte benutzerpartizipative Vorgehensweisen.

Gegenwärtiger Stand

Die enorm gestiegene Leistungsfähigkeit von Informations- und Kommunikationstechnologien bei gleichzeitig stark gesunkenen Preisen hat sowohl zu neuen Lösungsmöglichkeiten wie auch zu neuen Problemstellungen durch die zunehmende Durchdringung der Arbeitswelt, des öffentlichen und privaten Bereiches mit Rechnerprodukten geführt. Neue und zahlenmäßig starke Benutzergruppen sind erstmalig in Kontakt mit Hard- und Softwaresystemen gekommen. Dementsprechend muß sich die Software-Ergonomie mit vielfältigen Benutzermerkmalen, -voraussetzungen und -einstellungen auseinandersetzen, wenn die zunehmenden Probleme einer benutzergerechten Gestaltung gelöst werden sollen.

Im kommerziellen Bereich stellen massendaten-orientierte Anwendungen mit ihren für den Benutzer relativ inflexiblen und stark vorstrukturierten Interaktionsmöglichkeiten nach wie vor den Schwerpunkt dar. Hier liegt das zahlenmäßig größte Potential für software-ergonomische Verbesserungen. Software-ergonomische Lösungen sind hier nicht ohne eine Änderung der Aufgabenstruktur selbst möglich.

Zum anderen gewinnen multifunktionale Anwendungen eine immer stärkere Bedeutung. Multifunktionale Arbeitsstationen unterstützen ein breites Spektrum an Aufgaben in einem integrierten System. Für die Software-Ergonomie erwachsen hieraus mehr Anforderungen bezüglich der Konsistenz von Benutzerschnittstellen, der durchsichtigen Strukturierung der vielfältigen Informationen, die auf einem solchen System verfügbar sind, sowie eine verbesserte Unterstützung beim Auffinden und Nutzen der für eine Aufgabe erforderlichen Funktionalität.

Die Entwicklung von leistungsfähigen, dezentralen Arbeitsplatzrechnern mit entsprechenden Hardwarevoraussetzungen wie grafikfähige Bildschirme, Maus etc. hat auch die Entwicklung neuer Benutzerschnittstellen in einem entscheidenden Maße gefördert. Beispielgebend sind hier die langjährigen, benutzerorientierten Forschungsarbeiten am XEROX PARC Forschungszentrum, die zu grafikorientierten Benutzerschnittstellen nach dem Prinzip der direkten Manipulation geführt haben.

Die Entwicklung des Xerox Star Systems hatte weitreichende Auswirkungen für die Entwicklung nachfolgender Bürosystemprodukte und die generelle Sichtweise auf Benutzerschnittstellen. Kennzeichnend ist dabei, daß dem Benutzer auf graphische Weise vertraute Objekte der Büroumgebung am Bildschirm präsentiert werden, die auf einfache und konsistente Weise unter Zuhilfenahme eines Zeigeeinstrumentes manipuliert werden können (siehe Shneiderman, 1982). Entscheidend ist die objektorientierte Vorgehensweise: Der Benutzer hat in direkter Weise Zugriff auf Informationsobjekte, die in seiner Aufgabe bedeutungsvoll sind

und muß nicht erst zwischen rechnerspezifischen Objekten und seinen eigentlichen Arbeitsgegenständen übersetzen. Die Sprachebene, auf der die Kommunikation zwischen Mensch und System stattfindet, ist näher an die Ebene der eigentlichen Aufgaben herangerückt (Hutchins u.a., 1986). Dieses Prinzip findet sich z.B. auch bei Tabellenkalkulationsprogrammen.

Die bisherigen Anwendungen direkter Manipulation beziehen sich im wesentlichen auf universelle horizontale Rechnerwerkzeuge wie z.B. Textverarbeitung, Graphikerstellung, Kalkulation u.a. In diesen Bereichen scheinen die weiteren Entwicklungsmöglichkeiten beschränkt zu sein. Ein weitaus größeres Potential ist hier in vertikalen, problemspezifischen Anwendungen zu sehen, für die bisher nur wenige Arbeiten bezüglich des potentiellen Einsatzes direkt manipulativer Techniken durchgeführt wurden. Es wird ganz entscheidend darauf ankommen, geeignete konzeptuelle Modelle zu entwickeln, die für den Benutzer sichtbar und zugänglich gemacht werden können und manipulatives Operieren auf den Aufgabenobjekten erlauben.

Techniken der Mensch-Rechner-Interaktion

Bereits 1973 stellte Martin eine Auflistung 23 unterschiedlicher Interaktionstechniken zusammen und teilte diese in rechner- und benutzerinitiierte Techniken ein. Einsatzfelder und Benutzungseigenschaften wurden dabei lediglich auf informelle Weise diskutiert (kennzeichnend ist das Fehlen von Literaturhinweisen auf empirische Studien in Martin's Buch). Seitdem ist eine Vielzahl von Studien veröffentlicht worden, die sich auf die Gestaltung von Interaktionstechniken und empirische Ergebnisse zu ihrer Benutzbarkeit beziehen (ein guter Überblick findet sich in Shneiderman, 1987). Eine systematische Einteilung verfügbarer Interaktionstechniken mit einer Zuordnung ihrer jeweiligen Benutzbarkeitscharakteristika (usability criteria) fehlt allerdings bislang, obwohl diese Frage als ein zentraler Gegenstand des Software-Ergonomie angesehen werden kann (vgl. Ziegler, 1987).

Von vielen Seiten wurde das Prinzip der direkten Manipulation als entscheidender Fortschritt in der Mensch-Rechner-Interaktion aufgenommen und dieser Form in häufig unkritischer Weise positive Benutzungseigenschaften zugeschrieben. Einige Vorteile dieser Form dürften unbestritten sein, aber auch hier fehlen bisher klare Aussagen über Aufgabentypen und Benutzergruppen, für die diese Interaktionsform besonders geeignet ist. Es kann vermutet werden, daß Interaktionstechniken aus der gesamten Spannweite zwischen formalen Techniken (z.B. Kommando- und Programmiersprachen), direkter Manipulation bis hin zu natürlichsprachlichen Schnittstellen jeweils Vor- und Nachteile in spezifischen Aufgaben- und Benutzergruppenkontexten haben, die unter Umständen sogar

komplementär oder einander entgegengerichtet sein können (Trade-off Problematik).

These: Direkte Manipulation ist in vielen Diskussionen als die Lösungsmöglichkeit zur Benutzerschnittstellengestaltung bezeichnet worden. Zu einer Bewertung von Interaktionsformen ist jedoch zunächst eine geeignete Klassifikation und systematische Zuordnung von Benutzungseigenschaften in Abhängigkeit von Aufgabe und Benutzer dringend erforderlich. Dies ist ein zentrales Thema für die Software-Ergonomie.

Integrierte multimodale Mensch-Rechner-Kommunikation

Die vergangenen 10 Jahre haben wesentliche Fortschritte bei einigen Interaktionsformen gebracht. Besonders zu nennen sind hier natürlich-sprachliche Systeme wie direkt graphisch manipulative Systeme. Es wird z. Zt. diskutiert unter welchen Randbedingungen welche Interaktionsformen zu bevorzugen ist. Bezeichnet man einmal an formale Sprachen angelehnte Interaktionsformen (z.B. Kommandosprachen) natürlich-sprachliche Systeme sowie direkte graphische Manipulation als drei basale (generische) Interaktionsformen (vgl. Bullinger & Fähnrich, 1984), so ist primäres Ziel für zukünftige Generationen von Benutzer-Interfaces, aufgaben- und benutzeradäquate Integrationen dieser basalen Interaktionsformen zu finden. Dieser Gedanke findet sich auch an anderer Stelle in der Literatur wieder (vgl. Hayes, 1986). Auf diesem momentan noch eher spekulativen Gebiet wird weltweit momentan vielleicht von 10 Forschungsgruppen ernsthaft gearbeitet. Erste Prototypen befinden sich in der Entwicklung (Wetzel u.a., 1987). Diese Systeme beschäftigen sich momentan vorwiegend mit der Integration von direkt manipulativen Benutzerschnittstellen und natürlich-sprachlichen Systemen. Ein anderer Ansatz ist die Integration von Gestik-Input in direkt manipulative Systeme (Koller u.a., 1986). Diese neuen multimodalen integrierten Interaktionsformen werden an der Benutzerschnittstelle Redundanz anbieten. Sie werden damit neue Wahlmöglichkeiten eröffnen. Damit diese neuen Wahlmöglichkeiten den Benutzer nicht unnötig belasten, wird gleichzeitig an wissensbasierten Komponenten gearbeitet, die eine Dialogsteuerung oder Unterstützung bei der Dialogsteuerung offerieren. An dieser Stelle sind jedoch verstärkte Forschungs- und Entwicklungsanstrengungen notwendig.

These: Benutzerschnittstellen der Zukunft werden die bisher gängigen basalen Interaktionsformen in integrierter Form an der Benutzerschnittstelle anbieten (integrierte multimodale Benutzerschnittstellen). Dazu werden adaptierende (wissensbasierte) Ablaufsteuerungskomponenten benötigt werden. Diese werden auch unter dem Begriff Benutzerassistent oder Benutzeragent diskutiert.

Flexibilisierung von Rechnerwerkzeugen

Viele existierende Anwendungen, insbesondere im Bereich der Groß-EDV, sind durch weitgehend vordefinierte und starre Dialogabläufe gekennzeichnet. Solche Lösungen entsprechen den verständlichen Anforderungen an Datenqualität und Datensicherheit sowie der starken ablauforganisatorischen Standardisierung der Vorgangsbearbeitung in den Unternehmen. Bei dem zu beobachtenden organisatorischen Trend hin zu einem erweiterten Tätigkeitsspektrum z.B. in der Sachbearbeitung und den bei einer kundenorientierten Bearbeitung zunehmenden Anforderungen an die Behandlung von Sonderfällen und die Erarbeitung von individuellen Lösungen (Bullinger, 1984) können solche Systeme allerdings auf Dauer keine ausreichende Unterstützung bieten. Notwendig sind hier Softwarewerkzeuge, die sowohl flexiblere Aufgabenerledigung als auch flexiblere Möglichkeiten der Dialogführung erlauben. Beispiele für solche flexiblen Werkzeuge sind z.B. in den bereits erwähnten Spreadsheets zu sehen. Allerdings dürfen dabei die Anforderungen an die Verfügbarkeit von aktuellen Datenbeständen sowie die Einbeziehung standardisierter Teile der Aufgabenerledigung nicht unberücksichtigt bleiben. Bislang mangelt es hier noch an überzeugenden Lösungen etwa in Verbindung von Groß-EDV und dezentralen Arbeitsplatzrechnern, die eine entsprechende Flexibilität für den Benutzer vor Ort zur Verfügung stellen könnten. Abhilfe erwartet man von einer Klasse von Systemen, die als "verteilte Systeme" bezeichnet werden können (Ness & Reim, 1987).

Benutzerseitig bedeutet die Entwicklung flexiblerer Systemlösungen einen erhöhten kognitiven Planungsaufwand bei der Aufgabenerledigung, der aus ergonomischer Sicht aber durchaus als förderlich und wünschenswert im Sinne einer Aufgabenerweiterung angesehen werden kann, sofern er nicht zu Überforderungen führt. Gleichzeitig müssen in flexiblen Systemen besondere Anforderungen an die Benutzbarkeit der unterschiedlichen Funktionen hinsichtlich der Konsistenz und der einfachen Handhabbarkeit im syntaktischen und Ein-/Ausgabebereich gestellt werden. Eine erhöhte Komplexität bei der Planung der Aufgabe und Bestimmung der erforderlichen Systemfunktionen, die dabei zu nutzen ist, und eine hohe Komplexität bei der Durchführung der Aufgabe, d.h. bei der Nutzung der Funktionalität, kann den Benutzer insgesamt leicht überfordern und führt zu unangemessen hohem Lernaufwand und Fehlermöglichkeiten.

These: Im arbeitsorganisatorischen Bereich existiert ein starker Trend weg von tayloristischen (stark arbeitsteiligen) Arbeitssystemen und hin zu mehr ganzheitlichen Tätigkeiten mit entsprechender Funktionsintegration zu beobachten. Dies bedeutet auch stärker funktionsintegrierte technische Unterstützungssysteme. An der Benutzerschnittstelle wird dies zwangsläufig zu einer

Flexibilisierung der Anwendung entsprechender Software-Werkzeuge führen. Im Sinne der Arbeitspsychologie und Ergonomie ist diese Entwicklung hin zu einer Aufgabenerweiterung durchaus positiv zu betrachten, sofern sie belastungs- und beanspruchungsseitig im wünschenswerten Rahmen bleibt. Daher sind bei diesen Systemen besonders hohe Anforderungen an die software-ergonomische Gestaltungsqualität zu stellen.

Wissensbasierte Unterstützung der Mensch-Rechner-Interaktion

Auch bei einer zunehmenden Verbesserung von Benutzerschnittstellen mit Hinblick auf die Benutzbarkeit der angebotenen Software-Funktionalität ist es absehbar, daß traditionelle Ansätze zur Gestaltung von Benutzerschnittstellen nur ein beschränktes Potential für eine erweiterte Nützlichkeit und Benutzbarkeit von interaktiven Systemen bieten. Das traditionelle Vorgehen ist dabei gekennzeichnet durch die Gestaltung des expliziten Kommunikationskanals, wie z.B. der Informationsdarstellung, der Art des Funktionsaufrufs etc. Der implizite Kommunikationskanal, d.h. die Verwendung von gemeinsamen Wissen als Kommunikationsbasis wird dabei nicht ausgenutzt (Fischer & Gunzenhäuser, 1986).

Das im System implementierte Wissen kann sich auf unterschiedliche Aspekte und Ebenen der Mensch-Rechner-Interaktion beziehen. Zum einen kann es Wissen über die Interaktionsmöglichkeiten selbst sein und z.B. für adaptive Systeme oder aktive, intelligente Hilfefunktionen herangezogen werden. Zum zweiten kann Wissen über das Anwendungsgebiet und die Aufgaben des Benutzers dazu verwendet werden, um z.B. innerhalb der Anwendung sinnvolle Vorschläge zu entwickeln, Erklärungen zu liefern oder Pläne für die Aufgabendurchführung aufzubauen. Der Bereich des aufgabenbezogenen Wissens bietet ein erhebliches Potential für einer verbesserte Unterstützung des Benutzers (vgl. Ball & Hayes, 1980). Zukünftige Systeme werden deshalb in weitaus stärkerem Maße als heute nicht nur das "Wie" einer Aufgabenerledigung, sondern auch das "Was" unterstützen müssen.

Hierzu müssen im System Informationen auf dem entsprechenden Abstraktionsniveau zur Verfügung stehen. Dies erfordert bei der Realisierung zumeist die Verwendung wissensbasierter Techniken bei der Systemimplementation.

Der Bedarf nach einer wissensbasierten Unterstützung ist um so stärker einzuschätzen, je mehr die Anwendung auf eine problemspezifische Fragestellung ausgerichtet ist oder mehrere Anwendungsbereiche in einem multifunktionalen System zusammenkommen. Bei horizontalen Werkzeugen ohne einen spezifischen Anwendungsbezug (z.B. Textverarbeitung) sowie für reine Hilfestellungen bezüglich der syntaktischen Benutzung von Funktionen, wird dagegen der Einfluß wissensbasierter Techniken vermutlich relativ gering bleiben.

Die vieldiskutierten Benutzermodelle werden deshalb weniger Modelle unterschiedlicher Benutzer oder Benutzergruppen in Bezug auf geeignete Interaktionsmöglichkeiten sein, sondern eher Modelle des Benutzerwissens im Bereich der jeweiligen Anwendung bzw. Modelle zu Zielen und Handlungsstrategien der jeweiligen Aufgabe des Benutzers.

These: Die Gestaltung der Interaktionsmöglichkeiten allein führt nur zu einer beschränkten Verbesserung der Nützlichkeit und Benutzbarkeit eines Systems. Ein erhebliches Potential liegt deshalb in der Bereitstellung von Anwendungs- und Aufgabenwissen unter Verwendung wissensbasierter Techniken.

Theoretische Ansätze zur Untersuchung kognitiver Aufgaben

Es kann nicht erwartet werden, daß für das Gebiet der Software-Ergonomie in absehbarer Zeit eine geschlossene theoretische Basis zur Verfügung stehen wird. Dies liegt zum einen darin begründet, daß die Software-Ergonomie ein junges Gebiet ist. Zum zweiten ist sie ein interdisziplinäres Gebiet und muß deshalb Rückgriff auf den theoretischen Stand in den beteiligten Fachdisziplinen nehmen, der - wie z.B. in der kognitiven Psychologie - ebensowenig als abgeschlossen betrachtet werden kann.

Dennoch sind in den vergangenen Jahren eine Reihe von Entwicklungen deutlich geworden, die zumindest für Teilbereiche des komplexen Gebietes der Software-Ergonomie eine bessere theoretische Fundierung versprechen. Ausgehend von der Benutzerseite sind dies zum einen verschiedene Kommunikations- und Handlungsmodelle, zum anderen konkrete, formale und quantifizierbare Ansätze zur Modellierung der kognitiven Aufgaben des Benutzers. Das Ziel der letztgenannten Ansätze ist eine Prädiktion quantitativer Größen des Benutzerverhaltens, wie Performanz, Lernzeiten oder Fehler des Benutzers.

Als Ausgangspunkt solcher Methoden muß sicherlich das GOMS-Modell von Card, Moran und Newell (1983) betrachtet werden, in dem die Aufgabe des Benutzers durch kognitive Ziele, Operatoren, Methoden als größere Handlungseinheiten und entsprechende Kontrollmechanismen in einer hierarchischen Weise modelliert wird. Experimentelle Untersuchungen haben befriedigende Übereinstimmung zwischen theoretischen Vorhersagen und den beobachteten Leistungsgrößen geliefert. Aufbauend auf diesem Ansatz sind die Arbeiten von Polson und Kieras (1985) zu sehen, in denen das für eine Aufgabenausführung erforderliche Benutzerwissen durch die Regeln eines Produktionssystems ausgedrückt wird. Durch die starke Formalisierung eines solchen Modelles kann es implementiert und für Simulationszwecke herangezogen werden. Aus dem modellierten Wissen wird eine kognitive Komplexität der Aufgabe abgeleitet, die den Zeitbedarf für das Erlernen einer Aufgabe in linearer Weise bestimmt. Großangelegte experimentelle

Untersuchungen von Aufgaben aus dem Bereich des Editierens von Text und Graphik sowie einigen anderen Bereichen haben die theoretischen Vorhersagen der jeweiligen Modellierungen in außergewöhnlich guter Weise bestätigt. (Polson u.a. 1986, Ziegler u.a. 1986).

Ein zweiter Ansatz geht davon aus, die vom Benutzer durchgeführten Eingabeaktionen zur Steuerung des Systems als Sprache zu betrachten und diese grammatikalisch zu beschreiben. Der am weitesten entwickelte Vertreter dieser Richtung ist die sogenannte "Task Action Grammar" von Green und Payne (Green u.a. im Druck). Auch diese können anhand der Zahl grammatikalischer Regeln Vorhersagen über Lernen und Fehler machen. Da sie das erforderliche Benutzerwissen auf maximal verdichtete Weise darstellen, sind sie besonders geeignet, um die Konsistenz einer Benutzerschnittstelle zu überprüfen.

Ein entscheidender Vorteil solch theoriegeleiteter formaler Methoden besteht darin, daß sie eine Erklärung empirischer Ergebnisse liefern und damit eine stärkere Generalisierbarkeit von Aussagen ermöglichen. Ein Mangel vieler herkömmlicher Studien zur Benutzbarkeit von Systemen ist sicherlich darin zu sehen, daß die erhobenen Befunde nur in einem engen Aufgaben- und Systemkontext, in dem sie erhoben wurden, als gültig betrachtet werden können.

These: Verstärkte Aktivitäten im Bereich theoretischer und methodischer Entwicklungen sind notwendig, um die Software-Ergonomie weg vom Einzelfall-Empirismus hin zu verallgemeinerbaren Richtlinien und Gestaltungsmethoden zu führen. Vergleichende Untersuchungen zur Benutzbarkeit von Rechnersystemen werden allerdings auch in Zukunft weiterhin ihren Stellenwert behalten.

Evaluation von Benutzerschnittstellen

Die Bewertung der Gestaltung einer Benutzerschnittstelle ist in traditioneller Sicht ein Schritt, der sich an die Gestaltungsphase anschließt. Dieses Vorgehen ist in verschiedener Hinsicht unbefriedigend, wenn die Software-Ergonomie nicht einer rein korrektiven Funktion haben soll. Wünschenswert ist eine möglichst frühe Einschaltung software-ergonomischer Bewertungen im Designprozeß (Shackel, 1985). Erste Evaluationsmöglichkeiten sollten schon im Stadium des konzeptionellen Entwurfs und der Spezifikation möglich sein. Prädiktive Methoden, wie im vorausgegangenen Kapitel dargestellt, können hierbei hilfreich sein. Die Vorgehensweise des "rapid prototyping" erlaubt eine frühzeitige empirische Bewertung in einem iterativen Designprozeß (Good u.a., 1984).

Eine grundlegende Kontroverse im Bereich der Evaluation sowie in der Software-Ergonomie allgemein bezieht sich auf die Verwendung "harter" oder "weicher" Untersuchungsmethoden (vgl. hierzu Newell & Card, 1985, die für die

Notwendigkeit harter wissenschaftlicher Methodik in diesem Bereich plädieren, sowie die Gegenposition von Carroll & Campbell, 1986). Für die Systemevaluation bedeutet die Verwendung harter Methoden die Erhebung quantitativer, statistisch abgesicherter Daten, wie z.B. Ausführungs- und Lernzeiten, Fehlerraten u.a. unter genau kontrollierten Bedingungen. Um eine Bewertung der erhobenen Daten zu ermöglichen, ist hier meist ein Vergleich mit anderen Systemen oder Gestaltungsalternativen erforderlich. Eine bessere Vergleichbarkeit könnte durch die Entwicklung und Verwendung standardisierter Benchmark-Aufgaben für bestimmte Aufgabenbereiche erreicht werden, die Referenzpunkte für die Systembewertung liefern können.

Auf der Seite der "weichen" Methoden stehen explorative und subjektive Vorgehensweisen, wie z.B. Beobachtung des Benutzers, Benutzerbefragungen oder auch die Anwendung einer Checkliste durch den Experten. Weiche Methoden können ganz besonders wertvoll für das Entdecken und lokalisieren von Systemmängeln sein. In vielen Fällen führen sie mit einem weitaus geringerem Aufwand zu wesentlichen Aussagen für die Systembewertung.

Das grundlegende Kriterium bei der Methodenwahl für die Evaluation von Benutzerschnittstellen ist die Frage nach der Zielsetzung: harte Methoden erscheinen geeigneter für die Frage "Wie gut ist ein System?", weiche Methoden für die Frage "Was ist schlecht an einem System und warum?" (die Unterscheidung "how good" und "why bad" stammt von C. Lewis, pers. Mitt.).

Vor der Auswahl einer geeigneten Methodik steht die Frage nach den zu verwendenden Kriterien für eine Systembewertung. Hier ist eine klare Trennung zu machen zwischen Eigenschaften, die in der Systemgestaltung direkt beobachtbar sind, und Benutzungseigenschaften, die nur indirekt über den Umgang des Benutzers mit dem System zu erfassen sind. Checklisten z.B. sollten möglichst objektiv Systemeigenschaften abprüfen, wobei dem Merkmal eine entsprechende Bewertung bezüglich der Benutzbarkeit zuzuordnen ist.

Die Kriterien des DIN-Entwurfs 66234, Teil 8 zur Dialoggestaltung scheinen hier eine problematische Zwischenstellung einzunehmen, da zu ihrer Operationalisierung auf System- und Benutzungseigenschaften zurückgegriffen werden muß. Es gibt aber durchaus hilfreiche Ansätze, um diese Kriterien in einer Bewertungsmethodik umzusetzen (Paetau & Oppermann in diesem Band).

Shackel (1985) gibt vier Kriterienklassen an, durch die Benutzbarkeit (usability) definiert wird und die hier in verkürzter Form dargestellt werden.

- Effektivität (Effectiveness): Der erforderliche Bereich von Aufgaben muß mit einer Leistung erfüllbar sein, die über einer bestimmten Schwelle liegt (von einem bestimmten Anteil der jeweiligen Benutzergruppen in einem bestimmten Bereich von Nutzungsmöglichkeiten),

- Erlernbarkeit (Learnability): innerhalb eines bestimmten Zeitraums und mit einem bestimmten Trainings- und Unterstützungsaufwand,
- Flexibilität (Flexibility): Möglichkeiten der Anpassung des Systems an eine bestimmte Variabilität der Aufgaben und der Umgebung in Bezug auf die ursprünglich spezifizierten Aufgaben.
- Benutzereinstellungen (Attitude): innerhalb annehmbarer Bereiche von Beanspruchung und bei anhaltenden Arbeitszufriedenheit.

Obwohl auch diese Kriterien sicherlich noch nicht vollständig operationalisiert werden können, scheinen sie den Bereich der Benutzbarkeit gut abzudecken und sind prinzipiell geeignet, um die Anforderungen an die Benutzbarkeit eines Systems sowohl spezifizierbar als auch überprüfbar zu machen.

- These: Bei der Evaluation von Benutzerschnittstellen müssen sich "harte" und "weiche" Methoden ergänzen. Die dabei verwendeten Kriterien müssen sich eindeutig auf die Benutzbarkeit beziehen und sollten nicht überlappend sein.

Literaturverzeichnis

- Ball, E. & Hayes, P. (1980): Representation of Task-Specific Knowledge in a Gracefully Interacting User Interface Technical Report, Carnegie Mellon University, CMU-CS-80-123.
- Balzert, H. (Hrsg.) (1983): Software-Ergonomie. Tagung 1/1983 des German Chapter of the ACM am 28.-29.4.1983 in Nürnberg, Stuttgart: Teubner.
- Bullinger, H.-J. (Hrsg.) (1985): Software-Ergonomie '85. Mensch-Computer-Interaktion. Tagung 3/1985 des German Chapter of the ACM am 24. und 25.9.1985 in Stuttgart; Stuttgart: Teubner.
- Bullinger, H.-J. (1984): Die Durchdringung der Unternehmen mit integrierten Bürosystemen - Die Phasen der Büro-rationalisierung; In: Warnecke, H.J. & Bullinger, H.-J. (Hrsg.), Integrierte Bürosysteme - Zukunftsichere Strategien und erfolgreiche Anwendungen, 3. IAO Arbeitstagung, Berlin u.a.: Springer, 1984.
- Bullinger, H.-J. & Fähnrich, K.-P. (1984): Symbiotic Man-Computer Interfaces and the User Assistant Concept. In: Salvendy, G. (Ed.): Human-Computer Interaction, Amsterdam: Elsevier.
- Card, S. u.a. (1983): The Psychology of Human-Computer Interaction, Lawrence Erlbaum, Hillsdale N.J.
- Carroll, J.M. & Campbell, R.L. (1986): Softening up hard science: Reply to Newell and Card. User Interface Inst., IBM Watson Research Center, Yorktown Heights, New York.
- Fischer, G. & Gunzenhäuser, R. (Hrsg.) (1986): Methoden und Werkzeuge zur Gestaltung benutzergerechter Computersysteme. Berlin - New York: de Gruyter.

- Gaines, B.R. (1986): From Timesharing to the sixth Generation: The Development of Human-Computer Interaction. Part I, Int. Journal Man-Machine-Studies, 24, 1-27.
- Good, M.D. u.a. (1984): Building a user-derived interface. Communications of the ACM, 27, 1984, pp. 1032-1043.
- Green, T.R.G. u.a. (im Druck): Formalizable Models of User Knowledge in Human-Computer Interaction. In: Green, Hoc, Murray & Veer (eds.): Theory and Outcomes in Human-Computer Interaction. London: Academic Press.
- Hayes, P. J. (1986): Steps toward Integrating Natural Language and Graphical Interaction for Knowledge-based Systems. In: Proc. of the 7th European Conference on Artificial Intelligence 21-25 July 1986, Brighton, pp. 456-465.
- Hutchins, E.L. u.a. (1986): Direct Manipulation Interfaces. In: Norman, D. & Draper, S. (Hrsg.): User Centered System Design, 1986.
- Koller, F. u.a. (1986): Integration verschiedener Eingabetechniken am Beispiel eines Graphikeditors. WISDOM Report FB-IAO-86-30, Stuttgart.
- Martin, J. (1973): Design of Man-Computer Dialogues. Prentice-Hall, Englewood Cliffs N.J., 1986.
- Newell, A. & Card, S.K.: The Prospects for Psychological Science in Human-Computer Interaction. Human-Computer Interaction, 1, pp. 209-242.
- Ness, A.J. & Reim, F. (1987): Gestaltung und Betrieb von verteilten Systemen im Büro: Dimensionen der Integration; In: Fähnrich, K.-P. (Hrsg.), Fortschritte der neuen Informations- und Kommunikationstechniken, Online'87, 10. Europäische Kongreßmesse für Technische Kommunikation, Kongreß 7, Hamburg, 1987.
- Paetau & Oppermann (1987): In diesem Band.
- Polson, P. & Kieras, D. (1985): An approach to the formal analysis of user complexity. Int. Journal Man-Machine Studies, Vol. 22, pp. 365-394.
- Polson, P. u.a. (1986): A Test of a Common Elements Theory of Transfer. In: Proc. CHI'86, Boston, April 13-17.
- Sackman, H. (1970); Man-Computer Problem Solving; Princeton, Auerbach.
- Shackel, B. (1959): Ergonomics for a Computer, Design, 120, pp.36-39.
- Shackel, B. (1985): Human Factors and Usability - Whence and Whither? In: Bullinger (1985).
- Shneiderman, B. (1982): The Future of Interactive Systems and the Emergence of Direct Manipulation. In: Behaviour and Information Technology, Vol. 1, 3, pp. 237-256.
- Shneiderman, B. (1987): Designing the User Interface - Strategies for Effective Human-Computer Interaction. Addison Wesley, Reading (Mass.).
- Weinberg, G.M. (1971); The Psychology of Computer Programming; van Nostrand Reinhold, New York.
- Wetzel, R.P. u.a. (1987): A Logic Oriented Approach to Knowledge and Databases

Supporting Natural User Interaction. Deictic Interaction System - Query Environment; LOKI Report KR-GR 5.3/KR-NL 5, Stuttgart.

Ziegler, J. u.a. (1986): On Using Production Systems for Cognitive Task Analysis and Prediction of Transfer of Cognitive Skill. 3rd European Conf. On Cognitive Ergonomics, Paris.

Prof. Dr.-Ing. habil. H.-J. Bullinger
Fraunhofer Gesellschaft
Institut für Arbeitswirtschaft
und Organisation
Silberburgstraße 119 a
7000 Stuttgart 1