

Hans Dieter Hellige (Hg.)

Mensch-Computer-Interface: Zur Geschichte und Zukunft der Computerbedienung

2008

<https://doi.org/10.25969/mediarep/3664>

Veröffentlichungsversion / published version

Buch / book

Empfohlene Zitierung / Suggested Citation:

Hellige, Hans Dieter (Hg.): *Mensch-Computer-Interface: Zur Geschichte und Zukunft der Computerbedienung*. Bielefeld: transcript 2008. DOI: <https://doi.org/10.25969/mediarep/3664>.

Erstmalig hier erschienen / Initial publication here:

<https://doi.org/10.14361/9783839405642>

Nutzungsbedingungen:

Dieser Text wird unter einer Creative Commons - Namensnennung - Nicht kommerziell - Keine Bearbeitungen 3.0 Lizenz zur Verfügung gestellt. Nähere Auskünfte zu dieser Lizenz finden Sie hier:

<https://creativecommons.org/licenses/by-nc-nd/3.0>

Terms of use:

This document is made available under a creative commons - Attribution - Non Commercial - No Derivatives 3.0 License. For more information see:

<https://creativecommons.org/licenses/by-nc-nd/3.0>



HANS DIETER HELIGE (Hg.)

Mensch-Computer-Interface

Zur Geschichte und Zukunft

der Computerbedienung

[transcript] Kultur- und Medientheorie

Hans Dieter Hellige (Hg.)
Mensch-Computer-Interface

HANS DIETER HELDIGE (Hg.)
Mensch-Computer-Interface.
Zur Geschichte und Zukunft der Computerbedienung

[transcript]

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation
in der Deutschen Nationalbibliografie; detaillierte
bibliografische Daten sind im Internet über
<http://dnb.d-nb.de> abrufbar.

© 2008 transcript Verlag, Bielefeld



**This work is licensed under a Creative Commons
Attribution-NonCommercial-NoDerivatives 3.0 License.**

Umschlaggestaltung: Kordula Röckenhaus, Bielefeld

Umschlagabbildung: Heinz Obein (das ist Michael Sowa),
»Dompteur«; © Michael Sowa. Für die Genehmigung zum Wieder-
abdruck ist der Autor Michael Sowa und der Edition Inkognito, Ber-
lin sehr dankbar.

Satz: Dieter Hellige, Bremen

Druck: Majuskel Medienproduktion GmbH, Wetzlar

ISBN 978-3-89942-564-2

Gedruckt auf alterungsbeständigem Papier mit chlorfrei
gebleichtem Zellstoff.

Besuchen Sie uns im Internet:

<http://www.transcript-verlag.de>

Bitte fordern Sie unser Gesamtverzeichnis
und andere Broschüren an unter:

info@transcript-verlag.de

INHALT

Vorwort
HANS DIETER HELDIGE

7

Paradigmenwandel der Computer-Bedienung aus technikhistorischer Perspektive

Krisen- und Innovationsphasen in der
Mensch-Computer-Interaktion
HANS DIETER HELDIGE

11

Interfaces der Mainframe-Ära: Vom Automatischen zum Interaktiven Computing

Die ergonomischen Erfindungen der Zuse-Maschinen
im internationalen Kontext
HORST ZUSE

95

Zeigen, Zeichnen und Zeichen.
Der verschwundene Lichtgriffel
FRIEDER NAKE

121

Gestaltungsprobleme und Designkonzepte in der Mensch-Computer-Interaktion

Benutzergerechte MCI in einer dynamischen Welt –
Eine Gestaltungsaufgabe
HORST OBERQUELLE

157

Vom Persönlichen Computer zum Sozialen Medium.
Paradigmenwechsel der Mensch-Computer-Interaktion
MATTHIAS MÜLLER-PROVE

173

**Alternativen zum Desktop-Computing:
Der Körper und Gegenstände als Interfaces**

Wege und Irrwege der Mensch-Maschine-Kommunikation
beim Wearable Computing

INGRID RÜGGE

199

Die Rückkehr des Sensorischen: Tangible Interfaces
und Tangible Interaction

EVA HORNECKER

235

**Die »MCI der Zukunft«: Multisensorik, Everywhere Interfaces
und proaktive Umgebungs-Intelligenz?**

Ubiquitous Computing: Ein neues Konzept der
Mensch-Computer-Interaktion und seine Folgen

MICHAEL FRIEDEWALD

259

Die Interaktion des Menschen mit seiner intelligenten Umgebung
The Human-Environment-Interaction (HEI)

JOSÉ L. ENCARNAÇÃO, GINO BRUNETTI, MARION JÄHNE

281

**Paradigmenwandel der Computer-Bedienung
aus kultur- und geistesgeschichtlicher Perspektive**

Auf dem Weg zum »Finalen Interface«.

Ein medienhistorischer Essay

WOLFGANG COY

309

Interaktion im Kontext

JÖRG PFLÜGER

323

AutorInnenverzeichnis

391

VORWORT

Computer bedürfen wie jedes Kommunikations- und Informationsmedium eines »Interface«, über das die Funktionen angewählt, abgewickelt und kontrolliert werden. Die Gestaltfindung für Bedienschnittstellen und die nutzergerechte Anpassung sind seit jeher besonders schwierig und langwierig. Die Annahme neuer Interfaces durch die Nutzer gelingt vielfach gar nicht oder braucht oft Jahrzehnte, wie es Tastatur, Maus und Datenhandschuh belegen. Denn bei den Mensch-Maschine- und Mensch-Computer-Schnittstellen müssen neben technisch-funktionalen Aspekten eine große Palette ergonomischer und psychologischer Anforderungen sowie soziale Kontexte berücksichtigt werden. Als Brücke zwischen dem technischen System und den Nutzern sind die Interfaces zudem stark von Vorerfahrungen und Vorverständnissen der Entwickler und Nutzer abhängig. In kaum einem Bereich der Informatik spielen daher *hermeneutische* Phänomene wie mentale Modelle, Benutzermodelle, Metaphern und Leitbilder sowie Konstruktionsstile und Technikkulturen eine so eminente Rolle wie hier.

Obwohl Interfaces vielfach über Erfolg oder Misserfolg entscheiden, gilt die Bedienschnittstelle seit jeher mehr als ein Annex und unscharfer Randbereich der Informatik und Informationstechnik und nicht als eine zentrale Gestaltungsaufgabe. Auch die Technik- und Wissenschaftsgeschichte haben dem Thema Mensch-Computer-Interaktion bisher nur eine relativ geringe Aufmerksamkeit geschenkt. Für die »Fachgruppe Informatik- und Computergeschichte« in der »Gesellschaft für Informatik« war diese Forschungslücke Anlass zu einem Workshop auf der GI-Jahrestagung im Herbst 2005 und ein sich an diese erste Bestandsaufnahme anschließendes längerfristiges Buchprojekt.

Als Ergebnis legen Informatiker und Informatikhistoriker mit diesem Band eine erste Langzeitbilanz der Mensch-Computer-Interaktion vor. Sie bieten sowohl Gesamtüberblicke der Entwicklung aus technik- und geistesgeschichtlicher Sicht als auch spezielle Studien zur Bedienproblematik einzelner Epochen und Interfacetechniken. Dadurch entsteht eine große Linie von den Bedienschnittstellen der frühen Mainframe-Welt über die interaktiven PC-Interfaces bis zu den neuesten Entwicklungen des Wearable, und Tangible Computing sowie zur proaktiven Ambient Intelli-

gence. Die Einzelanalysen münden jeweils in theoretische Betrachtungen und kritische Rückblicke auf die MCI-Forschung sowie Ausblicke auf die Zukunft.

Dabei ist es das Ziel des Bandes, einen Bogen zu spannen von historischen Krisenphasen und Innovationssprüngen der Mensch-Computer-Interaktion zur aktuellen Problematik. Diese ist geprägt von einer immer stärker expandierenden multimedialen bzw. multimodalen Funktionalität bei immer kleineren Bedienschnittstellen. Die Autoren diskutieren die verschiedenen Lösungswege für die derzeitige Interface-Krise, insbesondere die Ansätze für »Natural Interfaces«, Avatare und natürlichen Dialog mit dem Rechner sowie die Konzepte für eine Ablösung von geräteartigen Medien durch unsichtbare »intelligente Umgebungen«. Der historisch-genetische Zugang zur Mensch-Computer-Interaktion kann zwar aktuelle Kontroversen über den »richtigen« Weg zur Überwindung der Krise nicht entscheiden. Doch in der Langzeitperspektive treten durchgängige Probleme und typische Fehler- und Engpass-Situationen an der Mensch-Computer-Schnittstelle viel deutlicher in Erscheinung als in der gewohnten relativ kurzfristigen Problemlösungsperspektive der Entwickler.

Bremen, April 2008

Hans Dieter Hellige

**PARADIGMENWANDEL DER COMPUTER-BEDIENUNG
AUS TECHNIKHISTORISCHER PERSPEKTIVE**

KRISEN- UND INNOVATIONSPHASEN IN DER MENSCH-COMPUTER-INTERAKTION

HANS DIETER HELIGE

Das Thema Mensch-Maschine-Kommunikation (MMK) bzw. Mensch-Computer-Interaktion (MCI) hat bisher in der Computer- und Informatikgeschichte eine relativ geringe Aufmerksamkeit gefunden. Es gibt nur wenige Spezialmonographien und auch in den »Annals of the History of Computing« muss man lange suchen, bis man auf Beiträge oder Ausführungen zu dieser Thematik stößt. Die Geschichtsschreibung spiegelt dabei aber nur den Sachverhalt wider, dass die Bedienschnittstelle seit jeher mehr als ein Annex und unscharfer Randbereich der Informatik angesehen wird und nicht als eine zentrale Gestaltungsaufgabe, die über Erfolg oder Misserfolg entscheiden kann. Nur sehr langsam setzt sich die Einsicht durch: »The interface between the user and the computer may be the last frontier in computer design.« (Foley 1987, S. 83) Welchen Beitrag die Informatikgeschichte zur Theoriedebatte in der MCI leisten kann, soll zunächst am Beispiel der Begriffsgeschichte, der Disziplin-Genese und der kontroversen Entwicklungsmodelle erörtert werden. Vor diesem Hintergrund wird dann im Hauptteil ein Überblick über die Langzeitentwicklung der MCI gegeben.

1 Die MCI als Gegenstand historischer Langzeitbetrachtung

Die traditionell randständige Positionierung der MCI kommt bereits in gängigen Bezeichnungen der Fachterminologie zum Ausdruck: Langezeit betrachtete man die Mensch-Maschine-Interfaces als Bestandteil der *Computer-Peripherie* und reduzierte sie damit auf bloße Hardware-Anhängsel. Diese Sichtweise entstand bereits 1945 mit der Übernahme des Begriffes »input-output« aus der Elektrotechnik, Elektronik und Automatisierungs-

technik in die frühe Computer Science durch John von Neumann.¹ Im Unterschied zu den frühen Computerpionieren Stibitz, Atanasoff, Eckert, Mauchly, Aiken, die noch keinen eigenständigen Begriff für das Bediensystem kannten, verstand v. Neumann die von ihm »input-output-organs« genannten Bedienschnittstellen als einen abgesonderten Bereich der Computer-Architektur. Doch gegenüber den »active organs« mit logischen Funktionen sah er im »Input-Output« lediglich untergeordnete »organs serving ›memory‹ functions«. Die verschiedenen Bedienprozesse reduzierten sich für ihn auf die bloße Zu- und Abfuhr von Daten- und Programmträgern bzw. die Abwicklung der Aufgabenpakete, und er setzte sie mit dem dem Gehirn dienenden neuronalen Sensorik- und Motorikapparat gleich (Neumann 1945, S. 35; 1958, S. 29). Auch für Alan Turing waren »Input/Output« »external organs«, die er auf der Ebene mechanischer Arbeiten der »servants« und der »girls« verortete, während er die geistigen Tätigkeiten des Planens und Durchdenkens der Programme den »masters« vorbehalten wollte (Turing 1947, S. 118 ff.). Die im frühen Computing entstandene hierarchische Arbeitsteilung bei der Organisation der Ein- und Ausgabe blieb bestimmend für die ganze Mainframe-Ära, sie bewirkte auch nachhaltig eine Geringschätzung der randständigen »terminal facilities« bzw. des »terminal equipment« (Everett 1951, S. 74) und des unmittelbaren Umgangs mit der Anlage, vor allem im kommerziellen Rechnereinsatz.

Dass gerade diese Marginalisierung der »input/output devices« sich als ein entscheidendes Hemmnis für die gesamte weitere Computer-Entwicklung erweisen würde, wurde von wenigen so klar durchschaut wie von Frederick P. Brooks. Er forderte in seinem berühmten IFIP-Vortrag von 1965, mit dem er den Begriff *Computer Architecture* in der Community bekannt machte, daß die Computer Architects die Schaffung neuer »input-output devices« zu einer Kernaufgabe machen sollten, denn keine Engineering-Aktivität könne so sehr neue Computeranwendungen stimulieren. Human-Factors-Aspekte im Computing sollten, so forderte er, künftig die gleiche Aufmerksamkeit genießen wie die Cockpits von Militärflugzeugen. Mit Blick auf eine komplexe Designsicht schlug er für diesen Bereich die Bezeichnung »architecture of input-output systems« vor (Brooks 1965, S. 89). Diese war für Brooks ein Teilbereich der Computer-Architektur, die er als das Gesamtinterface zwischen dem Benutzer und dem Computer verstand: »The whole discipline of system architecture has its central concern in the definition of the interface between a com-

1 Nach »The Oxford English Dictionary« (2.Aufl. 1989) wurde »Input-Output« seit dem 18. Jh. für Kapital-Einsatz und -Ertrag, seit dem Ende des 19. Jh. für Stoff- und Energie-Einsatz/-Ertrag verwendet und war seit den 1920er Jahren besonders in der Elektrotechnik und Elektronik verbreitet.

puter system and its users«. Der Interface-Architekt solle sich nicht primär als »scientist« sehen, sondern als »toolsmith«, der die Berührungszonen zwischen System und User menschengerecht gestaltet und dabei nach und nach alle Sinne einbezieht: »If we recognize our artifacts as tools, we test them by their usefulness and their costs, as is proper.« (Brooks 1977 S. 626; zum Architektur-Begriff siehe Hellige 2004b, S. 436-448) An Brooks knüpften zwar designorientierte Ansätze an (Carroll 1983; Bennett 1985), doch der Schwerpunkt von Architekturkonzepten im MCI-Kontext verschob sich schnell auf Funktionsebenen-Modelle und Standardisierungsrahmen für eine strukturierte Entwicklung von User Interfaces (z.B. Farber 1989). Letzlich hatte Brooks mit der Propagierung des Architekturbegriffs in dieser Sphäre keinen Erfolg.

Stattdessen bürgerten sich seit den 60er Jahren die weniger designbezogenen Bezeichnungen *man-machine interface* und *man-computer interface* ein, die in Deutschland »Bedien-« oder »Benutzerschnittstelle« genannt wurden. Der Interface-Begriff bezeichnete schon seit Ende des 19. Jh. Grenzschichten zwischen zwei Flächen oder Komponenten, so zuerst ab 1880 in der Chemie zwischen zwei Flüssigkeiten, und später dann in der Elektrotechnik und Elektronik zwischen physikalischen Teilsystemen (Verbindungselemente wie Stecker, Buchsen und Übergänge zwischen Geräten, Netzteilen).² In der Computertechnik taucht *interface* zuerst im Laufe der 50er Jahre auf. Die anfängliche physikalische Akzentsetzung prägte offenbar auch noch den kurz vor 1960 entstandenen Begriff »man-machine interface«: »Where a complex machine represents the principal artifact with which a human being cooperates, the term ›man-machine interface‹ has been used for some years to represent the boundary across which energy is exchanged between the two domains.« (Engelbart 1962) Doch bald überwog die system- und kommunikationstechnische Perspektive, das »man-computer interface« wurde nun als Grenz- bzw. Übergabestelle zwischen System und Umgebung oder zwischen Systemkomponenten gesehen. Allerdings relativierte sich damit auch seine Bedeutung, es bildete nur noch eines unter vielen »internal and external interfaces«. Der deutsche Begriff »Schnittstelle« postuliert ausdrücklich gemäß DIN 44300, dass es sich hierbei um einen »Übergang an der Grenze zwischen zwei gleichartigen Einheiten handelt.«

Genau diese Annahme einer Grenz- und Trennungsfläche zwischen Mensch und Maschine kritisierten die Pioniere und Promotoren der interaktiven MCI Douglas Engelbart und Joseph Licklider bereits 1962/63 als

2 Face steht hierbei für »surface«; die anthropomorphe Deutung von »interface« im Sinne »von Angesicht zu Angesicht mit dem Computer« entspricht nicht der Wortentstehung, vgl. »The Oxford English Dictionary« (2.Aufl. 1989).

irreführende Vorstellung. Für Engelbart handelt es sich vielmehr um eine Kopplung von menschlichen und technischen Prozessen am »man-artifact interface«, bei der die physikalischen Aktionen mit dem komplexen »matching process between the deeper human processes« verwoben sind: »Quite often these coupled processes are designed for just this exchange purpose, to provide a functional match between other explicit-human and explicit-artifact processes buried within their respective domains that do the more significant things.« (Engelbart 1962; siehe Friedewald 1999, S. 154 f.)

Auch nach Licklider widerspricht die schematische Aufteilung der Bedienfunktionen auf den Nutzer und die Maschine dem synergetischen bzw. symbiotischen Charakter der MCI: »The crucial regions for research and development seem to lie on both sides of the literal interface.« Deshalb hielt er »man-computer intermedium« für den adäquateren Begriff, da das »physical intermedium« sowohl »the user's station«, die Konsole, als auch »the user's entire workspace« umfasse (Licklider 1965a, S. 92 f.). Sie stelle so eine enge mediale Verbindung zwischen dem Computer- und dem Anwendungssystem her. Da Licklider diese grundsätzlichen Betrachtungen und sein Plädoyer für die Bezeichnung »intermedium« anstelle von »interface« nicht in seinen berühmten Aufsätzen brachte, sondern in dem weniger rezipierten Buch »The Library of the Future« von 1965, fand sein Begriff keinen Eingang in die Terminologie der Computer Science. Erst in neuerer Zeit taucht »Intermedium« in der medien- und kulturwissenschaftlichen Debatte wieder auf, allerdings meist mit Multimediabezug. Da sich jedoch neuerdings wieder die Einsicht durchsetzt, dass es sich beim Interface um ein »Medium im Medium« handelt (Grassmuck 1995), bestünde durchaus Bedarf für einen breiter gefassten Intermedium-Begriff.

Als in den 1970/80er Jahren mit den Terminals und ersten Mikrocomputern der »naive Benutzer« in Erscheinung trat und die Computerbedienung Gegenstand systematischer wissenschaftlicher Forschung und bewusster Gestaltung wurde, sprach man zunehmend von »user interface«. Man meinte damit die »I/O devices and processes« und die sie kontrollierende Software, also eigentlich »the computer interface to the user« und gerade nicht »the user interface to the computer« (Grudin 1993, S. 114). »User Interface« wurde bald zum Kernbegriff der »Human-Computer Interaction«. Obwohl sich der Begriff später immer mehr vom »engineering model« auf das »user task model« und die Software-kontrollierte Dialoggestaltung verlagerte und sich die Forschungsperspektive in Richtung eines umfassenden MCI-Forschungsansatzes ausweitete, wurde dennoch an der Interface-Bezeichnung festgehalten. Damit blieb nicht nur die eminent mediale Bedeutung ausgeblendet, sondern es wurde auch

die anfangs dominierende Ingenieurperspektive der Computerbedienung im Begriff festgeschrieben (Grudin 1993, S. 115). Mit der Informatikgeschichte läßt sich also herausarbeiten – was hier nur sehr grob geschehen ist – wie die Fachterminologie ältere fehlorientierende Sichtweisen bis in die Gegenwart bewahren kann.

Die untypische Disziplinentwicklung der MCI

Das Gebiet der physikalischen Intermedien und der MCI bietet noch weit mehr Ansatzpunkte für lohnende historische Langzeitbetrachtungen. So kann eine Rekonstruktion der Disziplinwerdung besonders nützlich sein für die Einschätzung des erreichten Wissenschaftsstatus und des Wissenschaftstypus der MCI. Diese hat sich nach ersten Anläufen wissenschaftlicher Beschäftigung um 1960 und jahrzehntelangem Experimentieren erst Anfang der 80er Jahre zu einer eigenständigen Disziplin entwickelt. Doch während die Professionalisierung, Literalisierung und Institutionalisierung mit der Einrichtung von Lehrstühlen und der Gründung von Fachzeitschriften, Konferenzen und Fachgesellschaften ab 1982/84 recht erfolgreich verlief, kam die systematische Theorie- und Methodenbildung nur sehr langsam voran (nach dem Ansatz von D. v. Engelhardt, vgl. Hellige 2004a, S. 2 ff.). Die Erforschung der Hardware- und Software-Interfaces hat sich zwar von rein deskriptiven Bestandsaufnahmen über Toolkits und vergleichende Effizienz-Messungen zu Klassifikationen des realisierten Interface-Spektrums und schließlich zu kompletten Design-Space-Systematiken vorgearbeitet. So strukturieren die vorliegenden recht unterschiedlichen Taxonomien und Design-Space-Matrizen den Lösungsraum auf der Basis von Raumdimensionen, Aktionsflächen-Bedarf, physikalischen Eigenschaften, Bandbreite der Interaktion, (In)Direktheitsgrad der Operationsweisen sowie von logischen Funktionen und Dialogstilen³. Ebenso hat man versucht, User-Interface-Systeme in Ebenen zu untergliedern, die von der untersten, der physikalischen Ebene, über die lexikalische, syntaktische, semantische und konzeptionelle bis zur höchsten Ebene, dem organisatorischen Kontext, reichen (Jacob 1996, S. 177).

Doch scheiterte bisher der Versuch, Designmerkmale- und –prinzipien theoretisch zu begründen und durch Systematisierung des Interface-Designraums die Basis für einen rationellen Entwurf aus dem Baukasten zu schaffen. Die Gründe dafür erkannte Donald A. Norman schon 1983 bei seinen Versuchen, wissenschaftlich begründete Designprinzipien für die MCI aufzustellen, in deren ausgeprägtem Zielkonflikt-Charakter:

3 Vgl. u.a. Buxton 1983; Hutchins/Hollan/Norman 1986; Foley/van Dam u.a. 1990; Card/Mackinlay/Robertson 1990; Frohlich 1992; Bos 1993; MacKenzie 1995.

»A central theme of our work is that, in design, there are no correct answers, only tradeoffs. Each application of a design principle has its strengths and weaknesses; each principle must be interpreted in a context.« (Norman, 1983, S. 3) Aufstellungen von Designprinzipien, Taxonomien, Designraum-Systematiken und Ebenenmodelle geben zwar analytische Einblicke in das Zusammenspiel der Designdimensionen, doch blenden sie alle akteursbezogenen Kontexte und Trade-offs sowie alle gestalthaften und hermeneutischen Aspekte des Interface-Entwurfs aus. Obwohl bei Software-Interfaces erste Ansätze einer Ordnung der Entwurfsmethoden und einer Strukturierung des Designprozesses erkennbar sind, ist eine wissenschaftlich fundierte Designmethodik noch in weiter Ferne (siehe den Überblick bei Baecker/Grudin/Buxton/Greenberg 1995, Kap. 2). Denn durch die permanente Ausweitung des Nutzerkreises vom Computerspezialisten und professionellen Anwender zum Computerlaien, durch die Verschiebung Ortes des »user interface« – vom Rechenzentrum zum Abteilungs- und Arbeitsplatzrechner und weiter in die Alltags- und Lebenswelt - und schließlich durch den Wandel des Computers vom Rechenautomaten zum Werkzeug und »instrumentellem Medium« (Schelhowe 1997) änderten sich auch immer wieder die Designanforderungen und -methoden grundlegend. Man gelangte daher bisher kaum über empirienahe Designleitfäden hinaus zu einer »allgemeinen und eindeutigen Theorie in der HCI«: »Es ist unmöglich zur jetzigen Zeit eine solche abzuleiten; wahrscheinlich wird es nie möglich sein.« (Dix/Finlay/Abowd/ Beale 1995, S. 20) Die MCI bleibt somit noch immer wesentlich eine *Design*-Disziplin, eine »black art«, in der »engineering design« und »creative design« mit wissenschaftlichen Methoden zusammenkommen müssen (Wolf/Rode/Sussmann/Kellog 2006). Sie weicht damit noch stärker als die übergeordnete Disziplin Informatik vom klassischen szientistischen Entwicklungsmuster ab (zur Rolle des Designs in der MCI siehe unten den Beitrag von Oberquelle).

Theoretisierung und Methodisierung scheiterten vor allem auch an dem stark inter- bzw. multidisziplinären Charakter des Gebietes. Denn bei den Mensch-Maschine- und Mensch-Computer-Schnittstellen müssen neben technisch-funktionalen Aspekten eine große Palette ergonomischer und psychologischer Anforderungen sowie soziale Kontexte berücksichtigt werden. Als Brücke zwischen dem technischen System und den Nutzern sind die Interfaces zudem stark von deren Vorerfahrungen und Vorerständnissen der Entwickler und Nutzer abhängig. Die Tastatur, der Joystick und die Maus sind nicht einfach aus mechanischen Konstruktionselementen zusammengesetzt worden, sondern Resultate komplizierter metaphorischer Prozesse und technikkultureller Wechselbeziehungen. In kaum einem Bereich der Informatik spielen daher *hermeneutische* Phäno-

mene wie mentale Modelle, Benutzermodelle, Metaphern und Leitbilder eine so eminente Rolle wie hier. Da diese über Gestaltvorbilder, Technikulturen und Konstruktionsstile in hohem Maße *kulturell* und *historisch* verankert sind, stoßen systematisierende und theoretische Ansätze hier schnell an Grenzen. Historisch angelegte MCI-Forschungsberichte, technikgeschichtliche und technikgenetische Studien bieten demgegenüber einen reichen historischen Erfahrungsschatz über Designstile, über erfolgreiche und gescheiterte Interface-Metaphern, über Leitbild- und Metaphern-Wanderungen zwischen den Techniken sowie über die periodische Wiederkehr von Übertragungen.⁴

Die gegenläufigen Entwicklungslogiken der MCI

Eine historische Langzeitbetrachtung der Intermedien der Informations- und Computertechnik ermöglicht darüber hinaus auch konkretere Einblicke in die spezifische Entwicklungsdynamik von Bedienphilosophien und Bauweisen der Interfaces als sie die bisherigen, relativ schematischen Entwicklungsphasenmodelle bieten. Denn so wie die Wissenschaftsentwicklung der wesentlich gesellschaftlich und kulturell bedingten Gestaltungsdisziplin MCI nicht einfach einem szientistischen Stufenkonzept folgt, so wenig unterliegt ihr zentraler Gegenstandsbereich einer klar definierbaren technischen, ökonomischen, sozialen oder kulturellen Eigenlogik. Für alle vermeintlich eindeutigen Entwicklungsrichtungen lassen sich vielmehr Gegenteilstrends erkennen. So hat man die MCI-Entwicklung als einen unablässig fortschreitenden Prozess der *Miniaturisierung* gesehen: Durch den Übergang von mechanischen, elektromechanischen zu elektronischen Bauelementen mit immer höherer Integrationsdichte gehe die Entwicklung nach Bell's Law zu immer kleineren, breiter verteilten Gerätefamilien, die von entsprechend miniaturisierten Interfaces und Displays bedient werden. (Bell/Chen/Rege 1972; Weiser 1991). Als letzte Konsequenz des Moore's Law würden Computer bzw. Mikroprozessoren samt Interfaces in Alltagsgegenstände eingebettet oder in Gestalt von Sensoren am oder von Nano-Robotern gar im Körper plaziert (Kurzweil 2002). Dem Trend zum ständig verkleinerten und letztlich »unsichtbaren Computer« widerspricht, dass die Miniaturisierung mit der Bedienungsfreundlichkeit kollidiert und dass daher eine Gegenteilstendenz zur Größensteigerung zu beobachten ist, etwa bei Notebooks, bei Großbildschirmen, Walldisplays und Rundumprojektionen in CAVES.

4 Siehe hierzu die frühen informatischen Überblicke von Carroll/Mack/Kellog 1988; Streit 1988, die MCI-Forschungsberichte von Grudin sowie die technikgenetischen Studien von Mambrey/Paetau/Tepper 1995; Konrad 2004, Kap. 2.; vgl. auch Hellige 1996a, S. 15-35.

In anderen Entwicklungsmodellen steht der Prozess der zunehmenden *Abstraktion* bzw. *Entmaterialisierung* im Zentrum: Von physikalischen Bedienteilen wie Hebel, Schalter, Kurbel, Knöpfen, Schalttafeln und Tastaturen gehe die Entwicklung über Textmenüs, flächenartigen Bedienschnittstellen und virtuellen Schaltflächen zur virtuellen Realität als immateriellem Interaktionsraum (Walker 1990). Doch diesem Trend zum quasi immateriellen Intermedium steht entgegen, dass bei informationstechnischen Interfaces noch immer bestimmte Grundfunktionen mit mechanischen Stellgliedern ausgeführt werden und dass sich Joysticks großer Beliebtheit erfreuen. Zudem wird gegen eine allgemeine Virtualisierung und den Verlust haptisch erfahrbarer Interfaces ein »physical turn« mit »graspable« bzw. »tangible objects« propagiert, also eine Gegenbewegung mit dem Ziel der (*Wieder*)*Vergegenständlichung* (vgl. Fishkin/Moran/Harrison 1998 und unten den Beitrag von Hornecker).

Damit korrespondiert eine Sichtweise, die die MCI-Entwicklung als einen Prozess der *Verräumlichung* der Bedienschnittstellen interpretiert, der von punktuellen, eindimensional-linearen zu 2D und schließlich zu 3D-Interfaces führt (z.B. Meadow 1970, S. 31-34; Johnson 1999, S. 28 ff. und Robben 2006). Das Interface entwickle sich danach zunehmend zu einem »Interplace« (Erickson, 1993). Dieser These vom auf den »graphical turn« folgenden »spatial turn« widerspricht jedoch die schleppe Ausbreitung virtueller Räume und 3D-Interfaces sowie das weitgehende Scheitern immersiver Interaktionsformen. Als weitere Entwicklungstendenz hat man die *Zunahme der Modalität* bzw. eine ständige Ausweitung der Kommunikationskanäle gesehen (Balzert 1988). Danach gehe die Entwicklung von monomodalen textuellen und visuellen Interfaces zu multimodalen Interfaces, die zusätzlich über akkustische, olfaktorische, haptische Schnittstellen alle Sinne ansprechen (vgl. unten den Beitrag von Encarnação/Brunetti/Jähne). Das Ziel sei ein Zustand, in dem der Benutzer nicht mehr in der Lage sei, »to distinguish between actual presence, telepresence, and virtual presence.« (Sheridan 1992) Doch über sehr bescheidene Ansätze in Unterhaltungsmedien hinaus ist der angebliche Trend zum »multisensorischen Computer« bisher noch nicht hinausgelangt, da die technischen Realisierungsprobleme und Kosten immens hoch sind und sich bislang viele Erwartungen in Kombinationen von »natural input modes« als Mythen erwiesen haben (Oviatt 1999). Zudem widerspräche der sensorische Informationsüberschuss perfekter Telepräsenz dem stark situativen und selektiven Charakter alltäglicher Medienutzung (vgl. Dertouzos 1999, S. 113).

Andere Ansätze wiederum sehen in einer Aufhebung der maschinellen Form der Interfaces die vorherrschende Tendenz der MCI, also in einer fortschreitenden *Humanisierung* der Interaktionsformen. Die Entwicklung

gehe von technikgeprägten zu immer natürlicheren Dialogformen mit dem Ziel einer quasi-menschlichen Kommunikation mit dem Computer auf der Basis von Sprache, Gestik und Körpersprache (siehe die Belege unten in Kap. 8). Für einige ist der Endpunkt der Entwicklung ein humanoider Computer in Avatar- oder Robotergestalt, der dem Menschen als Kommunikationspartner gegenübertritt. Der Vermenschlichungs-Tendenz widersprechen andererseits Auffassungen einer zunehmenden *Einbettung* der Interfaces in Alltagsgegenstände, ein Entwicklungstrend, der letztlich in einer Welt von »smart objects« endet, die »sehen« und »denken« können und den Menschen mit Programmintelligenz dienen (siehe u.a. Norman 1998, Denning 2002).

Beide Entwicklungsmodelle überschneiden sich mit einer weiteren Richtung, die in der *Steigerung des Intelligenzniveaus* der Mensch-Computer-Kommunikation den zentralen Entwicklungsfortschritt sieht. Bereits in den 50er und frühen 60er Jahren führten erste Erfolge der KI zu Prognosen der Realisierung des intelligenten Mensch-Computer-Dialogs um 2000 (Turing 1950, Mooers 1959; Ramo 1961) In den 80er Jahren erlebten diese Erwartungen durch Expertensysteme und Rechner der 5. Generation eine Renaissance (Shackel 1985), ebenso beflügelte die Entwicklung von Software-Agenten erneut die Erwartungen in intelligente Bedienschnittstellen (Kay 1984, 1990; Laurel 1990; Negroponte 1997; Maybury/Wahlster 1998). Doch abgesehen von den gewaltigen Umsetzungsproblemen sprechen die mit diesen Ansätzen verbundenen massiven Eingriffe in die Arbeits- und Lebenswelt dagegen, sie als nicht hinterfragbare Entwicklungsrichtung zu betrachten. Aus technikhistorischer und -soziologischer Perspektive werden diese Entwicklungsmodelle vielmehr als konkurrierende Leitbilder und Diskursangebote der MCI-Community gewertet, aber nicht als Leitfaden für die historische Rekonstruktion.

Die folgende Darstellung unterstellt daher keine der beschriebenen Entwicklungslogiken, sondern gliedert die MCI-Entwicklung idealtypisch nach grundlegenden Bedienparadigmen in vier große Abschnitte: in die manuell bedienbare Rechen- und Informationstechnik, in das vorweg arrangierte, automatisch abgewickelte Computing sowie das Interaktive und Proaktive Computing (in Kurzform: *Manual, Automatic, Interactive* und *Proactive Computing*). Diese Typen treten zwar zeitlich nacheinander in Erscheinung, sie werden aber nicht als normative Entwicklungsstufen verstanden, sondern als Wahlmöglichkeiten für bestimmte Benutzergruppen und Einsatzzwecke.

Der eigentliche historische Ablauf wird dagegen aus innovationstheoretischer und kulturalistischer Perspektive betrachtet, also unter dem Aspekt von Technologie-Lebenszyklen und sich wandelnden Technikstilen und Benutzerkulturen. Dabei zeigt sich schnell, dass Intermedien und In-

teraktionsformen anderen Entwicklungsmustern folgen als die übrige Hardware. Denn Bedienphilosophien und Bauweisen der Interfaces wälzen sich im Unterschied zu der vom Moore's Law-getriebenen Regelmäßigkeit der Bauelemente-Entwicklung höchst unregelmäßig und auch nur in langen Zeiträumen um. Es gibt hier einen Wechsel von kürzeren Krisen-, Such- und Innovationsphasen, die von einer divergenten Entwicklung unterschiedlicher Interface-Techniken bestimmt ist, und langfristigen Stabilisierungs- und Reifephasen mit ausgeprägt konvergenter Entwicklungstendenz. Im Folgenden möchte ich nun entscheidende Krisen- und Innovationsphasen der Intermedien herausgreifen und dabei die innovations- und diffusionstheoretische Betrachtung mit hermeneutischen Fragestellungen kombinieren.

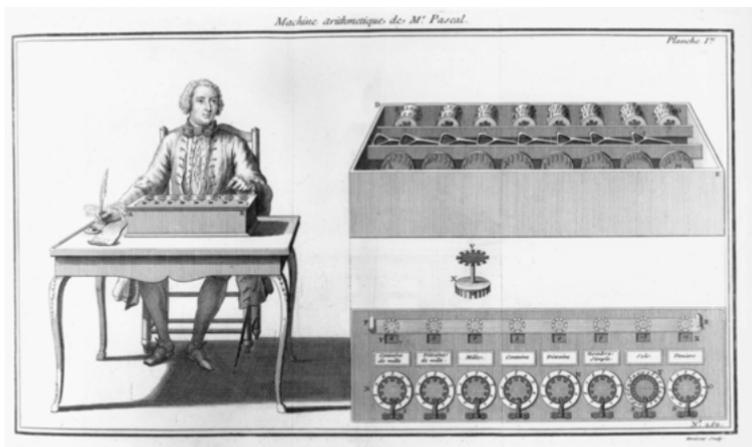
2 Interfaces der manuellen Rechen- und Informationstechnik

Obwohl die Computertechnik nicht aus der Entwicklung der Rechen-, Schreib- und Kommunikationsmaschinen heraus entstanden ist, hat sie stark von deren Interface-Angebot profitiert. Mechanische und elektromechanische Rechenmaschinen können lediglich Einzel- oder Teilberechnungen maschinell durchführen. Ihre funktionale Begrenzung erzwingt deshalb eine interaktive Bedienung im Rahmen eines umfassenden nutzergeführten Rechenplanes. Den Rechengang steuerte der Benutzer der frühen mechanischen Additions- und Subtraktions-Rechenmaschinen über Kurbeln, Schieber, Stellräder und andere direkt mit den Rechengetriebenen gekoppelte Stellelemente. Die Ablesung der Resultate erfolgte über meist aus der Uhrentechnik entlehnte Zahlendisplays. Diese enge Anbindung der Bedienschnittstellen an den Wirkmechanismus entspricht dem genuinen Ingenieurverständnis (»engineering model«), sie ist kennzeichnend für die Frühphase vieler Techniken. Auch bei den Rechenmaschinen bildeten sich erst im Laufe der Entwicklung Bauweisen heraus, die sich stärker am Nutzerverständnis der Bedienung (»user task model«) orientierten und die enge Bindung von Ein- und Ausgabe und Wirkmechanismus lösten (vgl. Gentner/Grudin 1996).

Zuerst wurden bei den Vierspezies-Maschinen besser bedienbare Bauweisen entwickelt, die das Einstellwerk sowie Umdrehungs- und Resultatwerk entkoppelten und so die Zahleneingabe von dem Rechengang mit der Kurbel trennten. Zu vereinzelt Rechenmaschinen-Entwürfen mit Tastatur (Parmelee, Hill) und Klaviatur (Schilt) kam es erst um 1850, der eigentliche Durchbruch erfolgte aber erst mit der speziell für Büro-zwecke entwickelten Volltastatur-Addiermaschine von Dorr Eugène Felt

Ende des 19. Jahrhunderts und der Entwicklung der 10er-Block-Tastatur durch Sundstrand nach 1900.⁵ Die Tasteneinstellung rationalisierte besonders bei Büro-Additionsmaschinen wegen der Vorzüge kurzer Stellwege und genormter Einstellwerke die Nutzung erheblich. Schließlich wurden mit dem Übergang zum elektromechanischen Antrieb seit den 1930er Jahren auch die Kurbeln durch Summen- oder Resultattasten abgelöst. Die Kopplung von Interface und Rechengetrieben war damit endgültig gelöst, die Rechenmaschine erhielt den Charakter einer Blackbox, mit der die Nutzer nur noch über das Drücken von Tasten und das Ablesen von Resultatfenstern in Verbindung traten. Die Rechenmaschine hatte sich so der Bedienphilosophie der Tastatur-basierten Büromaschinen angepasst, wo der Trend zur Separierung der Bedienebene und zur Schnittstellen-Konvergenz schon Jahrzehnte früher eingesetzt hatte.

Abb. 1: Die Pascal-Rechenmaschine im Gebrauch, überdimensionierte Darstellung in einem Kupferstich im »Recueil des machines et inventions approuvés par l'Academie Royale des Sciences«. Clermont- Ferrand 1735



Das um 1850 in Anlehnung an die Tastenform der Blasinstrumente entstandene Keyboard hatte sich um 1870 gleichzeitig in der Börsenticker-, Fernschreiber- und Schreibmaschinenteknik gegenüber der jahrzehntelang dominierenden Metaphern- und Konzeptvielfalt durchsetzen können. Diese war geprägt durch Rückgriffe auf traditionale Ein-/Ausgabetechniken wie Klaviaturen (Schreibklaviere, Klaviatur-Telegraphen), Kompass- bzw. Uhren-Formen (Nadel-, Zeigertelegraphen) und verschie-

5 Vgl. bes. Martin 1925, S. 66 f.; 91ff.; Lange 1986, S. 87ff.; Marguin 1994, S. 123 ff.; de Beauclair 2005, S. 18-32.

denartige Pen-Devices (Handschrift-Faksimileübertragung und Teleautographen). Im Marktsegment für Laien überlebte die anfängliche Vielfalt der Bauweisen und Metaphern noch lange. Hier war nämlich durch die rein professionelle Morsetelegraphie, die mit einem minimalistischen Eintastenmedium dem Benutzer die gesamte Code-Umwandlung überließ, eine fatale Interface-Lücke in der lokalen und innerbetrieblichen Textkommunikation entstanden. Der Mangel an einer laiengerechten elektromechanischen Textkommunikation bewog auch Erfinder wie Elisha Gray und Thomas Alva Edison zur Vereinfachung der Typendrucker-Technik zu von jedermann bedienbaren Börsentickern, Geschäfts- und Stadttelegraphen. Sie ersetzten dabei endgültig die Klaviatur durch die Typentastatur und gaben damit auch das Vorbild für die erste in größeren Serien hergestellte Remington-Schreibmaschine von 1874.⁶

In der professionellen Informations- und Kommunikationstechnik kam es mit der Etablierung der alphanumerischen Tastatur zu einer paradigmatischen Schließung, die bis heute fortwirkt. Denn alle Versuche, die zeitbedingten Designentscheidungen bei der Tastaturbelegung (QUERTY- bzw. QWERTZ-Tastensatz) oder bei der Tastenanordnung (ungeteilte 4-Reihenordnung der Zahlen und Buchstaben) durch alternative Keyboard-Designs (Dvorak-Tastatur, Split-Keyboards, Health Comfort Keyboard) zu revidieren, schlugen fehl (Lewis/Potosnak/ Magyar 1997). Als folgenreich für die verspäteten Ansätze zu einer laiengerechten elektromechanischen Textkommunikation erwies sich auch die seinerzeitige Einführungskonstellation. Denn Markteinführung und -ausbreitung des »Teletypewriter« wurden durch das noch leichter bedienbare »natural speech interface« Telefon verzögert. Durch dieses historische Zusammentreffen wurde die technisch bereits Ende des 19. Jahrhunderts mögliche Kommunikation zwischen Fernschreibmaschinen um Jahrzehnte hinausgeschoben. Die mit Electronic Mail und »Bürofernschreiben« (Teletex) in den 1970er Jahren einsetzende neue Welle der elektronischen Textkommunikation wurde dann aufgrund ihrer anfänglich massiven ergonomischen Defizite von der bewusst laiengerecht gestalteten Telefax-Technik noch einmal um über ein Jahrzehnt zurückgeworfen (Hellige 1995). Dies zeigt, wie folgenreich sich sowohl frühe Festlegungen wie auch strategische Defizite der MMK und MCI auf die Entwicklung von informationstechnischen Märkten auswirken können.

6 Dies beruht auf meinem noch unveröffentlichten Text: »Metaphern und Konstruktionsstile bei der Gestaltfindung der Mensch-Maschine-Schnittstelle in der Telegrafie« (Ms. 1994, siehe unter: <http://www.artec.uni-bremen.de/team/hellige/veroeff>), der demnächst stark erweitert erscheinen soll.

3 Die Bedientechnik im automatisch abgewickelten Computing

Wie die manuelle Rechen- und Informationstechnik imitierte und variierte auch die Computertechnik in der Pionierphase Bedienschnittstellen etablierter Techniken. Aufgrund ihrer völlig anderen Bedienphilosophie, dem vorrangierten programmgesteuerten Prozessablauf, griff sie aber nicht auf die Interfaces der vom Benutzer manuell interaktiv bedienten Rechenmaschinen zurück, sondern auf die der Webautomaten bzw. Musikautomaten. Denn das »automatic computing« ergab sich nicht einfach aus Entwicklungstrends traditioneller Rechengерäte und –maschinen oder mathematischer Verfahren, sondern entstand aus dem Projekt von Charles Babbage, die manufaktuelle Tabellenberechnung von Gaspard Riche de Prony mit Methoden industrieller Arbeitsteilung und Mechanisierung weiterzuführen. Babbage organisierte die maschinelle Ausführung von Rechenplänen (»plans and systems of computing«) mit Hilfe von im Prinzip frei kombinierbaren mechanischen Rechen- und Speicherwerken und realisierte diese ab 1833 in Ansätzen in seiner »Analytical Engine« (Babbage 1832, S. 138; Babbage 1864, S. 110). Für die Programmabwicklung seiner »really automatic calculating machine« griff er auf das Vorbild verketteter Lochkartensätze bzw. Lochstreifen von Jacquard-Webstühlen zurück. Die »communication« von »orders to the machine« erfolgte über »arrangements« verschiedener »sets of cards«, die jeweils Daten, Variablen, Operationen und Arbeitsschritte repräsentierten (Babbage 1837, S. 45 f.). Für die Planung der Rechenoperationen und die Verkettung der Kartensätze entwickelte Babbage bereits eine Tabellen-Darstellung, die Ada Lovelace (1843, S. 73 ff.) noch weiter ausbaute. Hierdurch wurde die Programmierung vereinfacht und von der weniger anspruchsvollen Maschinenbedienung abgesondert. Die Resultate der Berechnung sollten zur Vermeidung von Ablese- und Übertragungsfehlern gleich in Gestalt fertiger Tabellen ausgedruckt werden. Das von Babbage entwickelte Prinzip maschineller Rechenprozesse mit Lochkarten- bzw. Lochstreifen-Eingabe und Tabellen-Ausgabe bestimmte, obwohl von ihm selber nicht realisiert, über ein Jahrhundert die Pläne, Projekte und entwickelten Systeme programmgesteuerter Rechenautomaten und Lochkarten-Maschinen.

Da die als Automatenbeschickung angelegte »Computernutzung« sehr schwerfällig war, suchten Erfinder bereits um 1900 nach flexibleren Formen der Programmeingabe und fanden sie zunächst in der Stecktafel. Nach dem Vorbild der Telefon-Schalttafel erfand Otto Schöffler 1895 den »logischen Vermittlungsschrank«, der ab 1905 auch Bestandteil von Lochkartenmaschinen wurde (Zemanek 1983, S. 100 ff.). Die Einführung des »removable«, »portable« bzw. »automatic plugboard« in der IBM

601 von 1934 und der DEHOMAG D11 erlaubte sogar die Speicherung, Auswechslung und Weitergabe der Steckprogramme (Kistermann 2000, S. 224 f.). Ab 1930 wurden Stecktafeln auch bei Buchungsmaschinen und seit den 40er Jahren schließlich bei Analog- und Digitalrechnern als Programmier- und Bedienschnittstelle eingeführt. Beim ENIAC erstreckte sich die Plugboard-Metapher sogar über die Programmeingabe hinaus auf die aufgabenspezifische Verdrahtung der Rechereinheiten. Die Programmierung, Anlagensteuerung und Maschinenkonfiguration waren bei diesem Typ der Verknüpfungsprogrammierung noch nicht von einander getrennt. Obwohl die manuelle Programmieretechnik recht umständlich und fehlerträchtig war, hielt sie sich noch bis in die 60er Jahre. Denn sie erlaubte vielen Firmen die Weiternutzung alter Lochkarten-Datensätze und bot außerdem die Möglichkeit, die Programmierung statt durch teures DV-Personal durch die mit den Sachproblemen vertrauten Sachbearbeiter vornehmen zu lassen (Trimble 1986, S. 24 ff.; Hellige 1998, 191 ff.). Doch für Laiennutzer war eine derart dicht am technischen Prozess angesiedelte Bedientechnik nicht geeignet.

Auf Dauer aussichtsreicher war ein anderer Ansatz für die Ein- und Ausgabe von Programmen und Daten, nämlich der tastenbasierte Fernschreiber. Der wurde bereits 1908 bzw. 1915 von Percy E. Ludgate und Torres y Quevedo als Ergänzung zur Lochstreifen-Steuerung vorgesehen, doch erst Konrad Zuse, George R. Stibitz und Howard H. Aiken verwendeten ihn in ihren Pioniercomputern als ein reguläres Ausgabe- und gelegentlich auch als Eingabemedium. Samuel Alexander (1948, S. 248) bezeichnete 1947 in dem ersten Gesamtüberblick über »Input and Output Devices« den bereits mit Binärcode operierenden Teletypewriter als *das* geeignete Medium für die Mensch-Computer-Kommunikation: »Thus it was quite natural to borrow from this highly developed communication art for the initial development of input and output devices for conveying information into and out of electronic digital computers.« Durch Vorrichtungen für die automatische Lochstreifen-Fertigung und den Lochstreifen-gesteuerten Papiausdruck entstanden in dieser Zeit die rechnerangepassten »Flexowriter«, die bis in die frühen 50er Jahre zu den »principle I/O units« avancierten (van Dam 1966, S. 240). Die Computertechnik hatte damit Anschluss an die Standard-Interfaces der traditionellen elektromechanischen Büromaschinenteknik gefunden. Die hierdurch erreichbare Bandbreite der Mensch-Computer-Kommunikation war zwar nur sehr begrenzt, sie reichte aber für die rein alphanumerische Eingabe der Programmbeefehle und die Ausgabe numerischer Daten bei dem die frühe Entwicklung bestimmenden »scientific computing« meist aus. Denn noch immer wurden hier Computer überwiegend zur Lösung mathematischer

Berechnungen in der Technik, den Naturwissenschaften und der Mathematik genutzt (Aiken 1956, S. 31).

Abb.2: Bedienkonsole, Teletypewriter, Bandeinheiten und Tafel mit Ablaufplan der Rechenjobs beim ersten UNIVAC am US Census Bureau 1951 (The History of Computing. An Encyclopedia of the people and Machines; www.computermuseum.li/.../ UNIVAC-1-Census.htm)



Der Bewältigung größerer Datenmengen, vor allem im »business computing«, dienten nach wie vor Lochstreifengeräte, Lochkarten-Einheiten sowie schnelle Drucker für die alphanumerische Ausgabe. Mitte der 50er Jahre kamen auch erste Plotter und Kathodenstrahlröhren als Kurvenzeichner hinzu (vgl. den Überblick bei Chapin 1962, Kap. 5 u. 6, bes. S. 71). Für die Ein- und Ausgabe sowie Speicherung von Massendaten führten John Presper Eckert und John W. Mauchly 1950/51 beim UNIVAC Magnetbandgeräte ein, die wegen ihrer wesentlich größeren Geschwindigkeit und Leistungsfähigkeit bald zur regulären Ausstattung von Mainframes gehörten. Dennoch hielt sich das seit langem etablierte Hollerith-Input-Output-System einschließlich Stecktafeln als Rechner-Peripherie noch lange, da es zuverlässiger und flexibler war und von den Sachbearbeitern unmittelbar gelesen und manipuliert werden konnte (Davies 1954, S. 317 f.). Insgesamt verlief die Entwicklung der Hardware-Interfaces in dieser Zeit in relativ traditionellen Bahnen, größere Fortschritte gab es nur bei den Bedienschnittstellen für Programmierer. Hier wurde die anfangs durchgängige Programmerstellung und -eingabe in Maschinen-code durch die Entwicklung von Assemblern, Compilern und schließlich

durch die Schaffung elaborierter Programmiersprachen deutlich vereinfacht. Die damit erzielte Annäherung an die natürliche Sprache verminderte die bisherige hohe Anpassungsleistung des professionellen Benutzers an die Maschine und nährte schon um 1950 die Illusion einer baldigen Erreichbarkeit der *Konversation* zwischen Mensch und Computer. Doch die tatsächliche Entwicklung der MCI ging in die Gegenrichtung.

Ein folgenreicher Wandel der Organisation der Rechnernutzung ergab sich nämlich aus dem Problem der zeitlichen Abstimmung von Ein-/Ausgabe-Prozeduren und Rechengeschwindigkeiten. Die Input-Outputgeräte waren zwar für immer höhere Leistungen ausgelegt worden, doch sie konnten letztlich nicht mit den ständig steigenden Prozessorgeschwindigkeiten Schritt halten. Man koppelte sie daher vom Rechenbetrieb ab und ließ über Pufferspeicher und Interrupt-Mechanismen viele Ein-/AusgabEinheiten auf den Rechner zugreifen (Sutherland 1967, S. 52). Durch die generell eingeführte Zwischenspeicherung der Programme und Daten auf Magnetbändern, Lochkarten oder Lochstreifen wurde die vom Rechenbetrieb unabhängige Vorbereitung und Stapelung von Jobs wie Ausdruck der Resultate möglich, und im Hinblick auf die Auslastung der extrem teuren Computer auch notwendig. Daraus entwickelte sich im Laufe der 50er Jahre der fließbandartige Stapelbetrieb (batch processing), der zu einer wesentlich besseren Ausnutzung der Rechenkapazitäten führte.

Die Kehrseite der optimierten Mainframe-Ökonomie war freilich eine stark arbeitsteilige Organisation aller Arbeitsgänge. Während die Erbauer der Pioniermaschinen und die frühen Computernutzer noch Programmierer, Maschinen-Benutzer und -Bediener in einer Person waren, führte der Rechenzentrumsbetrieb nun eine strikte Arbeitsteilung ein: Er zerteilte das Computing in die hoch qualifizierte Programmerstellung, die monotone Kartenlochung bzw. Lochstreifen-Fertigung (gemäß der üblichen Arbeitsteilung meist Frauenarbeit), die Programm- und Massendateneingabe sowie die Ausgabe und die Operator-Aufgaben bei der Rechanlage. Auch die Optimierung der Computerbedienung erfolgte nun getrennt nach diesen Teilprozessen, was zwar die Rationalisierung, nicht aber die innovative Ausweitung der Computernutzung beförderte. Im Gegenteil, denn dem Benutzer wurde durch die fabrikmäßige Organisation des Stapelbetriebes der Zugang zum Computer weitgehend versperrt. Die »user« lieferten nur noch Lochkarten und Programmanweisungen ab und empfangen, je nach Auslastung der Anlage nach Stunden oder Tagen, die Ausdrucke der Ergebnisse zurück. Dies führte nicht nur oft zu akuten Engpässen, sondern auch zu langen Wartezeiten und Wiederholtschleifen bei fehlerhaften Programmen (vgl. bes. Nake 1995, S. 32 ff.).

Nur eine Minderheit von Experten oder Programmierern im »scientific computing« hatte noch direkten Zugang zum Rechner. Als Zugriffs-

medium diente diesen neben den Flexowritern vor allem die *Konsole* (*console, desk*). Die von den Rechen- und Steuereinheiten separierte Computerkonsole war bereits von Zuse in der Z3 (1938-41) und Z4 (1941-45) als zentrales Bedienpult zur Maschinensteuerung und -überwachung, zur Programmabwicklung sowie als Ein-/Ausgabe-Einheit geschaffen worden. Da er seine Rechenautomaten aus Nutzersicht konzipiert hatte, sah er von Beginn an eine integrierte Bedieneinheit vor. Die ersten amerikanischen Pioniermaschinen folgten dagegen mehr dem »engineering model« und verteilten die verschiedenen Input-/Output-Einrichtungen und Bedieneinheiten wie beim ENIAC über die Anlage (siehe unten den Beitrag von Horst Zuse). Die Bedienebene hatte sich hier noch nicht vom Wirksystem des Rechenautomaten gelöst. Das Vorbild für Zuses Konsolen-Design wie auch der späteren Pioniermaschinen bildeten die Steuerpulte von Maschinen bzw. die Leitstände der Anlagentechnik, die sich ihrerseits an die Form der Orgelkonsole (»organ console«) angelehnt hatten. Unabhängig von Zuse entstanden das Control Panel beim britischen Colossus (1943) sowie die Konsolen beim Model V des Bell-Computers von Stibitz (1946/47) und beim Harvard Mark II und III von Howard Aiken (1947 bzw. 1949). Der Begriff Konsole bürgerte sich auch nur langsam seit 1948 ein und erst seit dem UNIVAC und LEO (1951) und der IBM 701 und 702 (1952) gehörten »operator's consoles« zur Standard-Ausstattung von Mainframes.

Im Zuge der Entwicklung des Rechenzentrums und des Stapelbetriebes wurde das Input-/Output-System von der Operator-Konsole abgetrennt und beides nun in klimatisierten Räumen von den unmittelbaren Nutzern abgeschottet. Nur wer in den Genuss des direkten Zugangs zur Rechenanlage kam, konnte seine Programme ausprobieren und interaktiv »debuggen«: »Dieser ›User‹ war voll beschäftigt, die Maschine nach seinem Wunsche laufen zu lassen. [...] Dieser ›User‹ hatte ein Programm, das er fast persönlich durch den Computer zog.« (Alberts 2005, S. 208). Als mit dem Größenwachstum der Mainframes der Kreis der Privilegierten des »Open-Shop-Betriebes« immer kleiner wurde und die Unzufriedenheit der »Kunden« des »Closed-Shop« zunahm, wurde die gravierende Interface-Lücke des maschinenzentrierten Computerbetriebes offenbar. Zum fehlenden Direktzugang kam die noch immer höchst restriktive, von der Maschine diktierte monologische Form der »Kommunikation« mit dem Rechner über lange maschinenlesbare Befehlssequenzen hinzu: »Most machines are still a complex interconnection of logical elements provided with relatively primitive input-output facilities, usable by human beings only through a very *detailed* set of instructions from them that are almost micrological in nature.« (Carr 1962, S. 158) Es setzte daher in den 50er Jahren eine Suchphase nach dialogischen, »symbiotischen« Interfaces ein,

die in der Lage wären, dem Benutzer wieder einen direkten Zugang zum Rechner und interaktive Eingriffsmöglichkeiten in Rechen- und Verarbeitungsprozesse zu bieten und so die weitgehende Arbeitsteilung bei der Rechnernutzung wieder zurückzuschrauben (zum Wechsel vom Monolog zum Dialog siehe Meadow 1970, S. 3-15).

4 Die Anfänge des Interaktiven Computing

Als Entgegnung auf die Rechenfabrik entstanden um 1950 die anthropomorphen Leitideen eines »man-computer dialogue« bzw. einer »man-computer communication« nach dem Vorbild der »human conversation«: »The computer system should optimize around the characteristic variabilities of real time human norms for effective system performance rather than try to fit the human into an alien pace that may ostensibly be more convenient from program and equipment considerations.« (Sackman 1967, S. 435 und Kap. 9) Die Bemühungen um eine technische Realisierung des »conversational principle« waren dabei sehr breit angelegt, das Suchfeld zur Schließung der Interface-Lücke erstreckte sich von weitreichenden KI-Visionen eines bald erreichbaren natürlichsprachigen Dialogs mit dem Rechner über die rein textuelle Konversation bis zu einer ganzen Reihe neuartiger grafischer Interaktionsformen (zum Leitbild des »conversational computing« siehe Pflüger 2004, S. 370-379).

Frühe Visionen einer natürlichen Konversation mit dem Computer reichen allerdings bereits in die Anfänge des modernen Computing zurück. Schon 1933-45 entwarf der große Pionier der analogen Computertechnik Vannevar Bush mit seinem berühmten hypothetischen Desktop »Memex« das Szenario eines persönlichen Text-, Bild- und Sprachverarbeitungs- und Austauschsystems (Bush 1945, siehe auch unten den Beitrag von Müller-Prove). Er legte seiner medienkombinatorischen Fiktion allerdings noch traditionelle Informationstechniken wie Trockenfotografie, Lochkarten-, Mikrofilm- und Faxtechnik zugrunde, er dachte aber auch schon an Erweiterungen mit mobilen Miniaturkameras und Sprachverarbeitung. Doch Leitbildfunktion erhielt diese Vision eines interaktiven Hypermediums erst um 1960, als mit der Time-Sharing-Technologie die computertechnischen Voraussetzungen gegeben waren (Hellige 1996b, S. 207; Oinas-Kukkonen 2007). Die Idee einer Konversation mit dem Computer ist implizit auch schon bei John v. Neumann und Alan Turing erkennbar. Sie betrachteten die Kommunikation mit dem Computer als Überwindung von Sprachdifferenzen durch eine Kette von logischen und physikalischen »translations«: »[...] problems can be coded, i.e. prepared in the language the machine can understand [...] a unit which can un-

derstand these instructions and order their execution.« (Burks/Goldstine/v. Neumann 1946, S. 34 f.) Wegen unterschiedlicher Intelligenzanforderungen der verschiedenen Übersetzungsvorgänge hielt Turing deren arbeitsteilige Organisation vorerst für notwendig. Doch für die Zukunft erwartete er schon 1947 eine Zunahme des Sprachvermögens der Computer, so dass in wenigen Jahrzehnten eine quasi natürliche Konversation zwischen Mensch und Maschine möglich würde (Turing 1947, S. 122).

Abb. 3: Der Mark III-Computer von Aiken in einer Karrikatur von Boris Artzybasheff im Time-Magazine (Bd. 55, 4, 23.1.1950); sie symbolisiert die Erwartungen einer Lösung der Interface-Krise durch KI und Kybernetik.



Die Erwartungen in die möglichen Dialog- und Übersetzungsleistungen von Computern schaukelten sich dann im Laufe der 50er Jahre immer weiter hoch. So beabsichtigte Harry Huskey 1949/50 mit seinem Computer »ZEPHIR« den Bau »eines Dolmetscher-Gehirns, das jeden beliebigen Text aus dem Englischen automatisch in drei Fremdsprachen übersetzen kann« (Strehl 1952, S. 33 ff.). Der IBM-Forschungsmanager Rex Rice sagte in dem Szenario »Computers of the Future« von 1959 voraus, dass als Folge der »Microminiaturization« schon in 10-15 Jahren »Micro-Miniature Computers« Programme in »self-organizing-systems« erstellen und auf diese Weise die direkte Übersetzung der menschlichen Sprache in Maschinensprache leisten würden. Der führende Pionier für Information Retrieval Systems Calvin N. Mooers glaubte 1959 sogar, dass in ca. zwan-

zig Jahren der Informationssucher in natürlicher Sprache den Computer im »Information Center« um Auskunft über ein Fachproblem bitten und dieser mit einem maschinell formulierten State-of-the-Art-Report in dem zuvor gewünschten Umfang mündlich oder schriftlich antworten wird.

In den 50er Jahren kam auch die Idee eines »phonetic typewriters« auf, der jeden gesprochenen Satz verstehen und als generelles »natural communication link between men and machines« fungieren sollte. Zwar wollte man damit auch die »serious bottlenecks« in den großen militärischen Computersystemen beseitigen, doch das Hauptinteresse galt der weiteren Automatisierung der Büroarbeit: »The ultimate goal of workers in the area of automatic speech recognition is the construction of a device which, in effect, replaces a secretary taking dictation [...]« (Marill 1961, S. 34). Bereits die ersten zwischen 1952 und 1960 von Harry F. Olson (1975, S. 16-18) bei RCA entwickelten Prototypen eines »phonetic typewriter« verfolgten das Ziel, die Arbeit der Sekretärin auf die Korrektur der Ausdrucke zu reduzieren. Doch so sehr auch die Vision des Sprechschreibers die Forschung über Sprachverarbeitung beflügelte, gelangte man über eine noch recht unzuverlässige, mit viel Trainingsaufwand verbundene Erkennung einzelner Lautgruppen und festdefinierter Sprachbefehle nicht hinaus (Rabiner/Juang 1993, S. 6-10).

So kam es im Laufe der 60er Jahre zu einer großen Ernüchterung, denn »despite many illuminating discoveries, the physical realization of automata that will recognize natural speech seems still far away.« (Lindgren 1965, S. 114) Man musste erkennen, dass die schnellen Erfolge bei der Frequenzerlegung der Sprache und der Einzellauterkennung in den 50er Jahren nur ein erster Schritt zur Spracherkennung waren und dass die wirklich komplizierten Probleme der Sprachverarbeitung, die Phonem-, Wort- und Satzerkennung und gar das Sprachverstehen noch völlig ungeklärt waren. Daniel Bobrow (1967, S. 52, 55), einer der führenden Forscher auf dem Gebiet der »natural language communication«, warnte seine Kollegen vor einer Unterschätzung der Komplexität natürlicher Sprache, die sich gerade durch variierende Aussprache, fehlende Präzision, vage Anspielungen und Spontaneität auszeichne. Er stellte auch die Grundannahme der Forschung in Frage, dass »natural language« immer »the most natural medium for communications« sei, eine Erkenntnis, die in der Speech-Interface-Community jedoch bald wieder in Vergessenheit geriet.

Eine ähnliche Enttäuschung erlebte man bei dem Versuch, den Rechner zur Schrifterkennung zu nutzen und so per Mustererkennung mit diesem zu kommunizieren. Turing ging schon 1950 davon aus, dass Computer demnächst »written language« verarbeiten könnten. Auch Andrew D. Booth (1956, S. 18 f.), ebenfalls englischer Computerpionier, sah in der Erkennung gedruckter und handschriftlicher Eingaben nach Loch-

streifen, Lochkarten und Magnetbandeinheiten »the next form of input to a computing machine, in order of sophistication«. Nach ersten Erfolgen bei der Buchstabenerkennung rechneten viele schon für die späten 50er Jahre mit der breiten Einführung von Lesemaschinen. Ebenso erwarteten Licklider und Pioniere des interaktiven Time-Sharing wie Herbert Teager um 1960, dass die User sehr bald an einer Konsole handschriftliche Eingaben machen könnten, die der Computer in Realzeit verarbeitete. Doch trotz erster Systeme der »Intelligent Machine Research Corporation« für das Lesen genormter Buchstaben ab 1953 gelangen Versuche mit einer noch sehr groben Erkennung handschriftlicher Eingaben auf einem Bildschirm oder einem Digitalisiertablett erst Mitte der 60er Jahre. Der wirkliche Durchbruch der *Optical Character Recognition* (OCR) kam sogar erst in den 70er Jahren, wobei es sich aber um eine sehr spezielle Zusatzeinrichtung zum Rechner für Maschinen- und Druckschrift-Erkennung handelte und nicht um das ersehnte handschriftliche »computer graphics user/machine interface« (Rubenstein 1957; Hornbuckle 1967, Mori/Suen/Yamamoto 1992).

Da diese ersten Bemühungen, in einem Gewaltstreich die ›Sprachlücke‹ zwischen Mensch und Computer über dessen Anpassung an die natürlichsprachige oder schriftliche Kommunikationsweise zu schließen, scheiterten, suchte man nach anderen Wegen, das Leitbild des »conversational computing« zu realisieren. Eine Richtung setzte auf die entgegengesetzte Strategie: die Anpassung der menschlichen Sprache an die Maschine. So hielt es Douglas Engelbart seinerzeit für denkbar, dass die Computernutzer sich den Maschinencode so weitgehend aneigneten, dass die teuren CRT Displays und Buchstabengeneratoren vermieden werden könnten (Engelbart 1963, S. 20). Noch weiter ging der Chef des US-Computerherstellers TRW Simon Ramo (1961, S. 49 f.) mit seiner Forderung, durch die Schaffung einer neuen computergerechten Symbolsprache den die Entwicklung hemmenden Rückstand der »input output devices« auszugleichen: »A new type of English will come into common use, based on quantitative or logically qualitative expressions, with a controlled vocabulary to back it up, with rigid syntax and grammar, with phonetic spelling. [...] As man-machine communication becomes common, this language will become the ›natural‹ one to use.« Die binäre »universal language of the future«, die auch keine Dezimalzahlen mehr kennt, würde unterstützt durch Bedeutung tragende Farbcodes auf dreidimensionalen »visual displays« und »standard rythmical acoustic patterns.« Am Ende sollte der Computer alle banalen Alltagsaufgaben automatisch abwickeln und so den Zwang, mit der Maschine zu kommunizieren, auf komplexe Probleme, Normabweichungen und übergeordnete statistische Analysen reduzieren (ebda., S. 51). Ähnlich sah Andrew D. Booth (1960, S. 360) die Ab-

lösung eines Großteils der Ein-/Ausgabe-Arbeit voraus: »Thus computers of the future will communicate directly with each other and human beings will only be called on to make those judgements in which aesthetic considerations are involved.«

Realisten dagegen, die weder an eine baldige Vermenschlichung der Computersprache noch an eine Maschinenanpassung der natürlichen Sprache glaubten, setzten auf bewährte alphanumerische Ein-/Ausgabetechniken der traditionellen Büromaschinenteknik. Sie wollte mithilfe eines telegrammartigen Schreibmaschinen-Dialogs mit dem Rechner kommunizieren. Die Tastaturen wurden durch zusätzliche Funktionstasten für Standard-Operationen und Overlay-Folientasten für Spezialfunktionen noch stärker auf den Computer zugeschnitten. Neu an diesem an sich konservativen Lösungsansatz war das Abgehen vom vorab erstellten vollständigen Programm zugunsten eines kleinschrittigen Problemlösens im Wechsel mit dem Computer, wie es beim Debugging an der Konsole schon länger praktiziert worden war. Im Entwicklerteam des Whirlwind am MIT entstand daraus 1951/52 die »keyboard interaction«. Ab 1952 wurde am ORDVAC bereits ein »1024-spot display« mit Licht- und Tonsignalen für das interaktive Erstellen und Testen von Programmen erprobt (Davis 1965, S. 26). Seit 1953 entwickelte sich aus diesen Ansätzen im MIT-Projekt »Memory Test Computer« das Konzept des »interactive computing« und des Direktzugriffs der User auf den Rechner (»hands-on approach«, vgl. Friedewald 1999, S. 81, 110; Waldrop 2001, S. 144 ff.).

Computer wurden in diesem Kreis von Gegnern des Lochkartenbasierten Closed-Shop-Betriebes um Wesley Clark vorrangig als Werkzeuge gesehen, die der Benutzer von der Konsole aus ganz selbständig im »interactive use« bedienen sollte. Die »keyboard interaction« an der Konsole oder am Terminal wurde sehr bald die vorherrschende Form des »conversational computing«, 1967 schätzte Licklider, dass mehr als 95% aller »consoles« Teletypewriter waren (Licklider 1968, S. 211). In Time-Sharing-Systemen wurde sogar die Kommunikation von Keyboard zu Keyboard möglich, so dass Mooers 1963 von »reactive typewriters« sprach: »a device as cheap to have and as easy to use as the phone.« (zit. nach Van Dam 1966, S. 279) Doch so bahnbrechend dieser neue interaktive Bedienmodus auch war, so sehr zeigten sich vor allem im militärischen Bereich die Grenzen der weitgehend textuellen Interaktion. So kam es hier zu den ersten Ansätzen für grafische Interfaces: Mit klar erkennbarer militärischer Metaphorik entstanden um 1950 im Rahmen des computergestützten Realzeit-Informationssystems SAGE zur Radarüberwachung der CRT-Monitor sowie die ersten Tracking- und Pointing-Devices für eine zweidimensionale Mensch-Computer-Kommunikation.

5 Die militärische Frühphase der »graphical communication devices«

Die Aufgabe, eine Vielzahl von weiträumigen Flugbewegungen für das Überwachungspersonal überschaubar zu machen, erzwang eine Umwälzung der Bedienschnittstelle zwischen Mensch und Computer. Die bislang verwendeten Flexowriter und »hard copy output devices« sowie die Ein-/Ausgabegeräte auf Lochkartenbasis waren nach der Feststellung des Whirlwind-Chefdesigners Jay Forrester bei militärischen Realzeitrechnern als »terminal equipment [...] relatively inadequate«. Eine Lösung sah er allein in der Weiterentwicklung der Kathodenstrahlröhre zu einem Video Display. Der aus diesen Bemühungen hervorgegangene Monitor (»refresh graphic screen«), der in ersten Vorläufern an der Wende 1948/49 im Whirlwind I entstanden war und seit 1951 als reguläres Ausgabemedium diente, ähnelte anfangs noch stark einem Radarbildschirm. Seit 1952/53 wurde er im Cape Cod-Sector, dem Vorläufer des SAGE-Systems, regulär zur interaktiven Spurverfolgung von Flugbahnen eingesetzt.

Abb. 4: Die interaktive Bildschirmeingabe mit »light gun« im Radarüberwachungssystem SAGE (IBM-Archiv; http://www03.ibm.com/ibm/history/exhibits/vintage/vintage_4506VV2216.html)



Als erstes Tracking-Device wurde Ende der 40er Jahre im Rahmen von SAGE der Joystick zum Computer-Eingabemedium weiterentwickelt. Er imitierte in Gestalt und Funktionsweise den bereits Anfang des 20. Jahrhunderts entstandenen »control stick« von Autos und Flugzeugen. Unmittelbares Vorbild des Joysticks war der bereits 1942 in Deutschland als Lenkwaffen-Steuerungssystem für Flugzeug-Cockpits entstandene »Steuerknüppel« (Henschel Hs 293). Die aus dem Slang der Piloten stammende, 1917 zuerst belegte sexuell-anzügliche Bezeichnung »Joystick« erschien spätestens 1955 auch in amerikanischen Patentschriften im Zusammenhang mit Positioniergeräten (Robert H. Peterson, RCA). Wie später auch bei der »Maus« und beim »Puck« setzte sich die bildhafte Namensgebung gegen alle technischen Bezeichnungen durch, ja die zusätzliche Metapher aus der Lebenswelt förderte nicht unwesentlich die Akzeptanz des Eingabemediums bei den Nutzern.

Doch als Positionsgeber reüssierte im SAGE-System nicht der wenig zielgenaue Joystick, sondern die an der Jahreswende 1948/49 entstandene »light gun«. Deren Erfindung wird aufgrund der späteren Selbstzuschreibung meist dem »overall system designer« von SAGE Robert Everett zugesprochen, ohne dass der je ein Patent dafür beantragt hat. Nach Norman Taylor, einem der führenden SAGE-Entwickler, sei das Problem der Ansteuerung eines nicht ansprechbaren Bildschirmpunktes der Anlass für Everetts Idee gewesen.⁷ In seinem 1957 angemeldeten und 1962 erteilten Patent für ein interaktives »Data Processing System« taucht eine »light pen« zwar als Eingabegerät auf, allerdings mit dem Verweis auf das 1956 angemeldete Patent der »Light Gun Assembly« von Ralph G. Mork von IBM.⁸ Die Benennung der »light gun« ging entweder auf die »electron gun« zurück, d.h. die Fokussierungsvorrichtung in der Kathodenstrahlröhre, oder sie war, was wahrscheinlicher ist, eine Anspielung auf die Funktionalität des Eingabemediums. Denn die bestand in einem symbolischen Beschießen verdächtiger Flugobjekte, um dem Rechner zu signalisieren, welche Flugbahnen er kontinuierlich verfolgen und auf dem Bildschirm als Graphen abbilden sollte. Als Gestaltvorbild ist auch relativ ein-

7 Norman Taylor (1989) schildert die Erfindung so: »So Bob Everett, our technical director, said ›we can do that easily. All we need is a light gun [sic] to put over the spot that stops and well get a readout as to which one it is.‹ So he invented the light gun that afternoon and the next day we achieved man machine interactive control of the display — I believe for the first time.«

8 Siehe die US Patente Nr. 3037192 und 2915643 sowie »A Perspective on SAGE«, S. 391. Carl Machover (1978, S 38) schreibt die Erfindung Ivan Sutherlands Bruder Bert zu. Die Idee, mit einem »Electronic Tracing System« auf einem CRT-Bildschirm zu schreiben, war übrigens schon 1946 von Lloyd Espenschied von den Bell Labs patentiert worden (US Patent 674395).

deutig eine Handwaffe erkennbar: »The Whirlwind light gun was shaped like a backward pistol.« (Ross 1988, S. 64)

Die Prägung durch den militärischen Entstehungskontext konnten diese Ein- und Ausgabemedien jedoch schon bald zu einem großen Teil abstreifen. Der Monitor näherte sich dem rechteckigen Erscheinungsbild etablierter informationstechnischer Bildschirmsysteme, nämlich dem Fernseher und dem Library-Screen von Mikrofilm-Lesegeräten. Aus der unhandlichen »gun« entwickelten Benjamin Gurley und Charles E. Woodward (1959) am MIT-Lincoln Laboratory 1957/58 mithilfe der Halbleiter-Bauweise eine leichtere und besser bedienbare Miniaturversion, die sie Light »pen« nannten. Parallel dazu schufen Skramstad, Ernst und Nigro (1957) ein Erprobungssystem für die zivile und militärische Objektverfolgung an Bildschirmen einen »Light Pencil« in sehr ähnlicher Technik. Die für das SAGE-System charakteristische starre Einbindung des Bedienungspersonals in die halbautomatische Ablaufkette von »detection, decision and response« wurde von dem Studenten Dom Combelec im Rahmen einer Master-Thesis zu einem offenen interaktiven Designprozess umgewandelt. Dieses wohl allererste CAD-Programm ließ schon 1950 erkennen, »that interactive displays were the real tool to link people with computers« (Taylor 1989, S. 20). Douglas T. Ross, der 1954 sein erstes Programm für die manuelle Grafikeingabe per Lightgun geschrieben hatte, entwickelte ab 1959 zusammen mit Steven A. Coons am MIT das zivile »pen-tracking« und die interaktive graphische Objekt-Erzeugung am Bildschirm zu den ersten CAD-Systemen weiter (Ross 1988, S. 68 f.).

Die Kombination von Monitor-Display und Lightpen-Eingabe galt sehr bald als die ideale »workstation« für die interaktive Computergrafik (Gilmore 1989, S. 46). Doch die Lightpen blieb wie ihr Vorläufer ein reiner Pulsdetektor und somit kein »natürliches« Schreib oder Zeichengerät, wie es der Name suggerierte. Ihre noch immer bestehende Unhandlichkeit und vor allem ihre vertikale Nutzungsweise sah Licklider noch 1976 (S. 94) als eine Erbe der Lightgun an: »The result was to tire the user's arms, to fail to take advantage of their years of experience with pens and pencils on horizontal surfaces, and to doom the light pen to an early grave.« Gleichwohl blieb die Lightpen noch bis weit in die 80er Jahre ein besonders im CAD und der grafischen Datenverarbeitung verwendetes Eingabemedium, ja in Automatisierungssystemen, Operationssälen und Wahlmaschinen wird sie bis heute als Alternative zur Maus verwendet.

Beim Joystick blieb die militärische Metaphorik zwar in der Folgezeit durch den häufigen tatsächlichen Einsatz bei Waffensystemen oder die virtuelle militärische Verwendung in Computerspielen immer erhalten. Dennoch wurden Joysticks auch in vielen nicht-militärischen Umgebungen zur Maschinen-, Geräte- und Fahrzeugsteuerung eingesetzt. So fan-

den sie besonders in Automatisierungssystemen zur Bahnsteuerung, in Kränen und Aufzügen und in Flugzeugen (Fly-by-wire-Systemen) Verwendung. Später wurden sie auch aus Platzgründen als Mausersatz in Keyboards integriert. Demnach kann bei Mensch-Computer-Schnittstellen, die vorwiegend auf einer *Gestaltmetapher* beruhen, der Zusammenhang zum Ursprungsmilieu gelöst und mit völlig abweichenden Funktionalitäten verbunden werden.

Anders verhielt es sich dagegen mit dem zentralen Kontrollraum des SAGE-Systems, dem Urbild des computergestützten Kontrollzentrums. Es bestand aus der Großbildanzeige, die die einzelnen Überwachungszonen der Monitore zu einem kontinentalen Lageplan zusammenrechnet und mit der sogenannten »light cannon« auf die Wand projizierte.⁹ In ihm kamen alle wichtigen Informationen zusammen, es war daher auch das militärische Entscheidungszentrum, so wie es die traditionellen »map tables und counters« noch im Zweiten Weltkrieg gewesen waren. Das »control center« mit dem »wall-display« war im Unterschied zu den *Gestaltmetaphern* Lightgun und Joystick vorwiegend *Funktions- und Systemmetapher*. Der Übertragungsvorgang betraf hier in hohem Maße Aspekte der Arbeitsteilung und -organisation im Gesamtsystem, der Hierarchie und Entscheidungsstrukturen. Das Ursprungsmilieu wurde damit über die Metaphernverwendung teilweise mitübertragen. Die Lösung aus dem militärischen Entstehungskontext erfolgte deshalb beim Kontrollzentrum erst sehr viel später. Nach gescheiterten Bemühungen von SAGE-Entwicklern wie Jay Forrester und Angehörigen des militärisch-industriellen Komplexes, in den späten 50er und frühen 60er Jahren, die »War Rooms« als Lösungsansatz für die zentrale Überwachung und Steuerung aller großen Firmen und Organisationen durchzusetzen, gelang es erst in den 80er Jahren mit den Group Decision Support-Systems, die »War-Rooms« in »Konferenzräume« umzuwandeln. In ihnen repräsentierten die Wall Displays nun keine Lagepläne mehr, sondern Wandtafeln und Whiteboards für die Gruppenkooperation. Es erscheint mir ein lohnendes Aufgabenfeld für die Technikgeschichte und die HCI-Forschung zu sein, die Prägnungen von Metaphern durch den Entstehungskontext und die jeweils mögliche Dekontextualisierung vergleichend zu analysieren.

9 Das SAGE-Kontrollzentrum hatte bereits einen bisher nicht beachteten Vorläufer, das von der RCA 1945/46 entwickelte TELERAN »Traffic Control Center«, das die Radar-Flugraumüberwachung verschiedener Bereiche zu Lagebildern integrierte und für dichte Verkehrszonen bereits den Computereinsatz vorsah (Ewing/Smith 1946/47, besonders die Abb. der Computer-Version auf S. 611).

6 Die Innovationsphase grafischer Interfaces und der mehrdimensionalen Interaktionswelt

Die vom militärischen Computing ausgelösten Nachfrageimpulse hatten die Suche nach interaktiven Mensch-Computer-Interfaces angestoßen. Der Whirlwind II Computer und das SAGE-Project waren, wie Licklider es formulierte, »the fountainheads of oscilloscopes with light pens (guns) and man-computer interaction« (Licklider 1965b, S. 477). Doch für den erforderlichen grundlegenden Wandel der Hardware- und Software-Intermedien reichten sie allein nicht aus. Es bedurfte eines komplexen Zusammenspiels personeller und institutioneller Innovationsbedingungen, damit der Paradigmenwechsel in der Mensch-Computer-Beziehung gelingen konnte. Dazu gehörte eine mit dem »Closed-Shop-Betrieb« unzufriedene »programmer opposition« als Rekrutierungsfeld für ein innovatives Entwicklerpersonal. Diese universitäre Subkultur von Programmierern, die nach Alternativen zu den erstarrten Bedienphilosophien der kommerziellen Computerhersteller suchte, erfuhr in den USA eine im MCI-Bereich bisher beispiellose staatliche Förderung (zu deren Bedeutung siehe Myers 1998).

Die konzeptionelle und strategische Initialzündung erfolgte dabei durch Schlüsselfiguren wie Licklider und Robert Taylor, die im entscheidenden Augenblick für die Forschungsfinanzierung im Computerbereich zuständig waren. Bei ihnen handelte es sich um multidisziplinäre Gatekeeper, die als eigentlich Fachfremde zwischen den Entwicklern und Nutzern der Technik standen und die früh die zentrale Bedeutung der Mensch-Computer-Interaktion für die Ausweitung der Computernutzung in Denk- und Arbeitsprozessen erkannten (Friedewald 1999, S. 121). Es ging beiden gerade nicht um eine bloß *technische* Optimierung von Interfaces, sondern um neue *mediale* Nutzungsweisen von Computern, bei denen nicht mehr programmgesteuerte Berechnungen, sondern die Informationsbeschaffung und -verarbeitung, die dynamische Repräsentation von Ideen sowie die Kooperationsunterstützung im Mittelpunkt standen. Alles zusammen sollte menschliches Denken und Problemlösen auf eine neue Stufe stellen: »Through such interaction, the heuristic capabilities of men and the algorithmic capabilities of computers - the highest capabilities of the cooperating partners - can be melded together to produce what may turn out to be a significant augmentation of intellectual power.« (Licklider 1967, S. 40)

Die medienkombinatorische Erneuerung der MCI

Das MCI-Erneuerungsprogramm von *Joseph C. R. Licklider* beruhte auf praktischen Erfahrungen mit Computern und Time-Sharing-Versuchen, auf einer Auswertung des SAGE-Projektes sowie auf einer ganzen Reihe von eigenen Aufsätzen und State-of-the-Art-Reports über Interfaces und Computer-Bedienung. All dies verdichtete er in theoretischen Studien und Grundsatzartikeln, von denen einige wie »Man-Computer Symbiosis« von 1960 Leitbild-Funktion für das Interactive Computing erhielten. Er entwickelte in diesen Schlüsseltexten die Methode einer Kombination neuer Interfaces und Medien aus bekannten Mustern, denn für ihn stand fest: »A new concept is achieved, not by creating a new schema *ab initio*, on a custom basis, but by adapting an old schema or, if necessary, arranging several refurbished schemata into a new, complex structure.« (Licklider 1965a, S. 3) Um sich dabei aber von bekannten Techniken zu lösen, formulierte er die »schemata« abstrakt, denn dann war es möglich »to break them down and recombine them into new configurations, always on the lookout for new elements, if one is to progress.« (Ebda., S. 92 f.)

Auf der Basis ergonomischer Bewertungen vorhandener Interfaces und der Projektion zu erwartender technischer Verbesserungen spielte er verschiedene Arrangements von »intermedia« durch. So untergliederte er den MCI-Designraum in ein grafisches Interaktionsschema (»The Oscilloscope-and-Light-Pen Schema« einschließlich diverser Pointing Devices), in eine textuelle Interaktionsform (»The Typewriter-Hardcopy-Display Schema«), in ein der klassischen Tafel entsprechendes »Group-Computer-Interaction Schema« sowie das am gewohnten »writing desk« orientierte »Consoles and Workspaces Schema«, das bereits in Richtung »Desktop-Computer« wies (Licklider 1960, S. 9 ff.; 1965a, S. 93-104). In ihm sah er für die Zukunft auch das größte Potential, da es der Leitidee der »man-computer symbiosis« schon sehr nahe komme: »Devising an electronic input-output surface on which both the operator and the computer can display, and through which they can communicate, correlated symbolic and pictorial information.« (Licklider/Clark 1962, S. 121) Die »general-purpose console« künftiger Zeiten beschrieb Licklider (1965c, S. 508) als ein multimediales Universal-Interface, das neben Bildschirm und Tastatur auch Light-Pen, Digitalisier-Tablett, Kamera, Projektor, Mikrophon, Lautsprecher, Telefon, Uhr sowie für die 3D-Ansteuerung einen »bowling ball« aufweisen sollte. Die von ihm anfangs stark favorisierte kombinierte Sprach- und Handeingabe, wie sie Herbert Teager nach dem Vorbild des »Memex« von Vannevar Bush plante, betrachtete er Mitte der 60er Jahre dagegen nur noch als ein *Fernziel*, ebenso mobile Interfaces in der Art von Electronic Paper »for fast, natural, nonconstraining, effective interaction«:

»If I had three wishes, one would be for a thin sheet of plastic - sensitive to a stylus, capable of bright display in colour with selective erasure, controllable by computer, and inexpensive.« (Licklider 1968, S. 210)

Neben den Rekombinationen neuartiger Intermedien aus vorhandenen Komponenten erwog Licklider Mitte der 60er Jahre auch bereits grafische Dialogtechniken, etwa visuelle Symbole und menügesteuerte Dialoge. So wollte er dynamisch im Programmablauf sich ändernde Schaltfelder (»light buttons«) auf den Bildschirm bringen und durch Lightpen oder Joystick ansteuern. Ebenso forderte er unterschiedliche Interaktionssprachen für Experten und Anfänger, wobei er schon den Grundgedanken der »Direkten Manipulation« vorwegnahm: »The computer should continually tell the neophyte what options are open to him and, when an instruction is partly defined, what additional arguments are required.« (Licklider 1967, S. 63, 67) Derartige Überlegungen flossen unmittelbar in die ARPA-Förderungsprogramme für neue Programmierkonzepte und graphische Interaktionstechniken ein, so wie er auch mit Forschungsgeldern die Entwicklung und Erprobung neuer Interfaces gezielt vorantrieb.

Die künftigen Nutzungskonzepte für das »dynamic medium computer« entwarf Licklider aber nicht aus den kombinatorischen Arrangements von Hard- und Software, sondern aus grundlegenden Anwendungsszenarien. Er unterschied dabei die lokale Kooperation (»project meeting as a model«), die individuelle Kommunikation über ein Computernetz (»face to face through a computer«), die On-line-Bibliothek und den universalen Zugang zum Wissen (»library of the future«, »distributed intellectual resources«) und schließlich die sich selber organisierenden Nutzergemeinschaften (»on-line interactive communities«) (Licklider/Taylor 1968, S. 23 ff.). Damit wurde wie bereits in Bushs Memex-Aufsatz die Medienkombinatorik auf der Anwendungsebene durch die Generierung praxisnaher Szenarien ergänzt. Hierdurch wie durch sein stark Grafik-orientiertes Interaktions-Konzept hatte sich Licklider endgültig von seinen früheren KI-basierten anthropomorphen Visionen einer direkten Konversation mit dem Rechner freigemacht und war auf die pragmatischen Ansätze eingeschwenkt, wie sie besonders Engelbart und die Pioniere der graphischen Datenverarbeitung vertraten. Ihre Entwicklungsaktivitäten wurden dann auch neben Time-Sharing-Systemen und Computernetzen der dritte Hauptschwerpunkt des ARPA-Förderungs-Programms.

Auch *Douglas C. Engelbart* legte als Quereinsteiger in die Computing Community das Hauptgewicht nicht auf Berechnungen, sondern auf die Informationsverarbeitung, auf das »Individual as a User, Generator, and Retriever of Information« (Engelbart 1961) Er hatte sich schon ab 1958 im Rahmen von Überlegungen über ein »technical information center« näher mit dem Memex-Aufsatz beschäftigt und in den Folgejahren

gründlich rezipiert (Bourne/Engelbart 1958 und Oinas-Kukkonen 2007, der aber die frühen Aufsätze nicht heranzieht). Er übernahm von Bush nicht nur die Idee eines »Microdocumentation System« und einer persönlichen Arbeitsstation, sondern auch dessen Methode, »to combine lower-level technologies into a simple machine with a higher level function.« (Engelbart 1961; Nyce/Kahn 1991, S. 44). Ganz im Gegensatz zum damaligen KI-gläubigen Mainstream der Computer-Community bildete die Entwicklung neuer Interfaces und die Schaffung einfacher Werkzeug-Kombinationen für die Informationsverarbeitung den Ausgangspunkt seiner MCI-Forschung, denn hiervon erwartete er eine bedeutende Erweiterung intellektueller Tätigkeiten. Dabei ging er bei der Auslotung des physikalischen Design-Space der Interaktionstechniken im Gegensatz zu den in der Interface-Entwicklung üblichen Ad-hoc-Entwurfsmethoden sehr systematisch vor.¹⁰

Nach frühen Designstudien für Feedback-Lernmaschinen Ende der 50er Jahre spielte Engelbart 1962/63 alle menschlichen Dateneingabemöglichkeiten durch. Von den bekannten »man-machine interfaces« hielt er allein die Standard-Tastatur für die Interaktion mit dem Computer geeignet, während Morsetaste und das »keyboard-shorthand system« den Benutzer überforderten. Die so viel propagierten »most natural« Interface-Techniken, die Handschriftenerkennung und die »voice communication« schieden ebenfalls für ihn aus, denn »we have no way of knowing how long it will be before a system-applicable speech recognition machine will be able to decode full English transmission.« (Engelbart 1963, S. 14-16) Die Alternativen für die alphanumerische Eingabe sah er in einer 5-Tasten-Einhand-Tastatur (»chording device«) bzw. in einem daraus abgeleiteten Datenhandschuh (»glove«) für die Eingabe von »five-key codes«. Für die Ansteuerung eines von ihm »indicator mark« benannten Cursors erprobte er vergleichend die bekannten Positionsgeber Lightpen, Joystick und Trackball und konzipierte selber mit »fiber-optic light pipes« « bestückte Datenhandschuhe. Neben den manuellen »screen-select devices« entwickelte sein Team »feet control devices« (später kamen noch »nose-pointing control« und »knee control« hinzu) und erwog sogar schon »eye position tracking schemes« (Engelbart 1963, S. 23-25; English/Engelbart/Berman 1967, Engelbart 1988, S. 194 ff.).

Engelbart ordnete für komparative Konzeptstudien und Vergleichstests alle »input devices« in einer mehrdimensionalen Matrix an: »Much as the periodic table of the elements has characteristics which define groups

10 Siehe hierzu wie zum Folgenden vor allem Friedewald 1999, S. 168-185; ihm verdanke ich auch eine Kopie des Briefes bzw. Memorandums von Engelbart an Robert Taylor vom 5.4.1963 aus der Douglas C. Engelbart Collection, Stanford University Library.

along rows and columns, we laid out a grid of existing devices. And just as the periodic table's rules have led to the discovery of certain previously unknown elements, this grid ultimately defined the desirable characteristics of a device that didn't exist. That device was the mouse.« (Maisel/Engelbart 1996; ähnlich schon im 3. Engelbart-Interview, März 1987) Aber beim Antrieb der Maus knüpften er und William English direkt an ein konkretes historisches Vorbild an, das Planimeter mit zwei orthogonalen Laufrädern, ein Designfehler, der erst 1973 durch die Ballmouse von Ronald Rider behoben wurde (Rider 1974, Aford 1990, S. 395). Doch insgesamt erfolgte die Umsetzung seines von Licklider und Taylor geförderten Forschungsprogramms für neue Interaktionsmedien und die graphische Computerkommunikation bereits auf der Basis einer ersten Systematik des Input-/Output-Equipments und einer frühen Interaktions-Design-Methodik.

Auf der medialen Eben kombinierte Engelbart die Interfaces Bildschirm, Maus, Keyboard und Chording Device zu einer Arbeitsstation für professionelle Informationsverarbeitung. Durch Software-Tools für Textverarbeitung, Grafiken, hypertextartige Dokumenten-Organisation entstand daraus ein viele PC-Errungenschaften vorwegnehmendes dynamisches Medium (»dynamically cooperative man-machine intelligent team«, Engelbart 1960) Die kombinatorische Synthese erfolgte dabei wie bei Bushs »Memex« nicht aus dem Gerätepark, sondern von praktischen Nutzungsszenarien her und auf der Basis einer »hypothetical description« des Systems der »Intelligenzverstärkung«. Das Ergebnis seiner medienkombinatorischen Entwicklungsmethode präsentierte er in der berühmten Demo von 1968, bei der er die ganze Spannweite seiner medialen Neuschöpfungen vorführte (Textverarbeitung, E-Mail, Hypertext, Tele-/Videoconferencing). Die Bilanz dieser Vorgehensweise war eine außergewöhnliche Verbreiterung des Such-, Experimentier- und Entwicklungsfeldes für innovative Interfaces und die Entdeckung und Erprobung des Computers als eines dynamischen Mediums während der 60er und 70er Jahre.

Die Interface-Innovationen der 60er Jahre

Aufgrund dieser Anstöße und der optimalen Innovationsbedingungen kam es zu dem erstaunlichen Sachverhalt, dass im Laufe nur *eines* Jahrzehnts ein Großteil der physikalischen Intermedien und Software-Interfaces entstanden, die noch heute die Grundlage des interaktiven Computing ausmachen. Dazu gehörten neben verbesserten Lightpen- und Joystick-Bauweisen die grafischen Koordinatengeber *Maus*, *Puck* und *Trackball*. Wie die Maus beruhten auch Puck und Trackball auf Gestalt- und Bedienmetaphern aus dem Bereich der Zeichengeräte und mathemati-

schen Instrumente. Der Maus-ähnliche Puck, auch Sensor Probe genannt, der per Lupe und Fadenkreuz eine weit genauere Ansteuerung von Punkten auf Bildschirmen, Grafiktablets oder Menutablets ermöglichte, lehnte sich eng an bestehende Präzisionszeichengeräte an, er wurde demgemäß auch überwiegend als professionelles Interface beim CAD eingesetzt. Vorläufer des Trackballs (Tracker Ball, Roller Ball, Bowling Ball) waren bereits in der Mitte des 19. Jahrhunderts analoge Rechengeräte für die »harmonische Analyse« wie die Kugelkalotte bzw. das Kugel-Rollplanimeter. Eine neue Welle von »ball type computing devices« entstanden im Zweiten Weltkrieg für Feuerleitsysteme, Scheinwerfer-Steuerungen und Flugbahnberechnungen. Obwohl »ball computers« bereits Ende der 40er Jahre auch für Werkzeugmaschinen-Steuerungen verfügbar waren, wurden die ersten »control balls« für das Bildschirm-Tracking und zur Cursor-Lenkung erst 1961 bzw. 1966 patentiert (Alexander 1961; Laman 1966).

Im gleichen Zeitraum führten die erheblichen Mängel der Lightpen-Bedienung zu dem Digitalisiertablett mit Stifteingabe, dem »input drafting table for handwritten and drawn programs« von 1961, nach ihrem Erfinder auch Teager-Table genannt. Durchgesetzt hat sich aber erst das von Malcolm R. Davis und Thomas O. Ellis (1964) in der Rand Corporation geschaffene Rand-Tablet, aus dem sich bis heute unzählige Varianten von Pen-Tablets entwickelten. Die Spannweite reicht von kleinen Sketchpads bis zu Digitizern im Zeichenbrett-Format. Die Koordinationsmängel zwischen Ein- und Ausgabe wurden einerseits Anlass für eine Direktverkopplung von Monitor und Tablet (»GRAIL-Console« bei Rand) und andererseits zu einer Verlegung des elektrischen Koordinaten-Gitters auf den Bildschirm. Damit wurde 1962 durch Christian Paul Charles Lesage von der »Compagnie des Machines Bull« auch die Touchpanel- und Touchscreen-Entwicklung angestoßen. Sein »manually controlled coding device« (U.S. Patent 3.220.00, 23.11.1965) sollte untrainierten Laien die Bedienung von Computern ohne die Zwischenschaltung komplizierter Eingabegeräte erlauben. Der Touchscreen gilt seither aufgrund seiner Bedienung durch einfache Zeigegesten als der einfachste und natürlichste aller Positionsgeber, denn »there is no spatial mapping between input device and the screen.« (Douglas/Mithal 1997, S. 72) Ben Shneiderman brachte gar die Entstehung dieses elementarsten aller Eingabemedien mit Michelangelos Darstellung der Erschaffung des Menschen in Verbindung: »Inventors of the touchscreen in the 1960s may have been inspired by this image in their cultural unconscious. Touchscreens have an unrivaled immediacy, a rewarding sense of control, and the engaging experience of direct manipulation.« (Shneiderman 1991, S. 93)

Auch dem Pen-basierten Grafik-Tablett sagte man damals eine große Zukunft voraus, da es dem natürlichen Schreiben und Zeichnen am

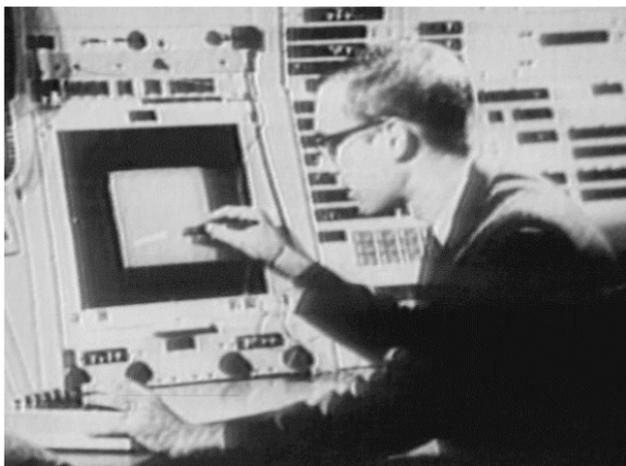
nächsten käme, denn »the nearer the operator action approaches the natural act of writing or drawing the more successful is the man/machine relationship.« (Evans 1969, S. 11). Doch die Trennung von Ein- und Ausgabemedium ließ die Tablets mit Graphic Pencil bzw. Stylus bis heute nicht über eine professionelle Nischentechnik hinauskommen. Auch der »Berührbildschirm« blieb wegen seiner starren Dialogführung lange Zeit auf Public Access Systeme beschränkt. Sein Einsatz im industriellen Umfeld wurde zudem bis vor kurzem durch die mangelnde Genauigkeit der Ansteuerung kleinerer Ziele mit dem Finger behindert. Hier wie in der professionellen grafischen Datenverarbeitung blieb es lange bei Lightpen- oder Stift-basierten Eingabesystemen. Für den Allgemeinbedarf fügte man um 1965 in die Tastatur »cursor control keys« (Schreibmarken-Tasten, Pfeiltasten) ein, die bei Buchungsmaschinen schon seit den 50er Jahren üblich waren (Lange 1986, S. 107 ff.). Später kamen noch der drehknopfartige »Pro Pointer« und »spin-wheels« als in Keyboards integrierbare Positionsgeber hinzu. Mit diesen nachgeschobenen fortschrittlich-konservativen Lösungen wurde auch das weit verbreitete Standard-Interface mit begrenzten, aber preiswerten 2D-Interaktions-Möglichkeiten ausgestattet.

Die 2D und 3D-Grafik-Revolution in der MCI

Neben den diversen neuen Hardware-Interfaces entstand im Laufe der 60er Jahre auch der Grundstock an graphischer Interaktions-Software. Hier war es vor allem Ivan Sutherlands Pioniersystem »Sketchpad« von 1962/63, das den Umgang mit dem Computer auf eine neue Stufe stellte: »Heretofore, most interaction between man and computers has been slowed down by the need to reduce all communication to written statements that can be typed. In the past, we have been writing letters to rather than conferring with our computers.« (Sutherland 1963, S. 329 und unten Nake) Mit den interaktiven Grafik-Programmen von Sketchpad wurde es hingegen möglich, dass der Benutzer direkt mit dem Computer konferierte »through the *medium* of line drawings.« Der Übergang von den Programmzeilen zur grafischen 2D-Interaktion wurde von Sutherland nicht nur als eine bedeutende Ausweitung der Bandbreite in der Mensch-Computer-Interaktion gesehen, sondern als ein grundlegender medialer Wechsel: »The Sketchpad system, by eliminating typed statements [...] in favor of line drawings opens up a new area of man-machine communication.« Durch die verwendete »picture language perfectly natural to the human« könne der Nutzer komplizierte Zusammenhänge viel leichter überschauen und verstehen. Es entstehe ein »novel communication medium«, dessen Einsatzspektrum von der mathematischen Modellierung über Konstrukti-

onszeichnungen und Schaltkreisentwürfen bis zu »artistic drawings«, Karikaturen und animierten Grafiken reiche (Sutherland 1962, S. 9, 18; 1967, S. 61).

Abb. 5: Ivan Sutherlands Grafiksystem Sketchpad von 1963, das die interaktive grafische Kommunikation mit dem Rechner revolutionierte (Machover 1999)



Wie Sutherland und andere Pioniere der grafischen Datenverarbeitung erwartete auch Licklider vom Übergang vom »one-dimensional stream of alphanumeric characters« zur Kommunikation mit zwei- und dreidimensionalen grafischen Skizzen und Bildern eine Erschließung breiter Nutzerschichten: »[...] I think that it will probably be realistic graphics that dominates the mass introduction of computing into the lives of the people.« (Licklider 1967, S. 41; 1976, S. 95) Licklider war derart überzeugt von der grafischen Wende der MCI, dass er als Organisator der Forschungsförderung im Computerbereich die Grafik zu einem Kern seines Forschungsprogramms machte und sogar zur Absicherung dieses Schwerpunktes Sutherland zu seinem Nachfolger berief (Norberg/O'Neill 1996, S. 125 ff.).¹¹ Als Leiter der ARPA-Computer-Abteilung formulierte dieser nun seinerseits strategische Überlegungen für den MCI-Bereich und Zukunftsszenarien für Online-Systeme und die grafische Interaktion. Ein Schwerpunkt waren darin »neuartige Eingabegeräte«, denn diese »können dem

11 Auch dessen Nachfolger Robert W. Taylor war sehr an der Computergrafik und der MCI interessiert, vgl. Taylor 1968; Norberg/O'Neill 1996, S. 131 ff.

Computer eine größere Vielfalt an Informationen zuführen«. Auch von besseren Computer Displays sowie neuen Visualisierungs- und Simulationstechniken erwartete er eine Erweiterung des menschlichen Wahrnehmungshorizontes, (Sutherland 1967, S.; 1970, S. 57). So wie er durch die Verknüpfung von Darstellung, Bearbeitung, Speicherung, Simulation, Animation und Übertragung grafischer Informationen ständig neue mediale Möglichkeiten erkundet hatte, so wollte jetzt über die Verbesserung und Kombination von Interfaces und Interaktionstechniken dem Computer ein breiteres Anwendungsfeld in Wissenschaft, Technik und Schulwesen eröffnen.

Nach der erfolgreichen Erschließung der 2D-Welt gab Sutherland noch während der 60er Jahre auch den Anstoß für eine erste Inangriffnahme des dreidimensionalen Interaktionsraumes. Bereits 1963/64 nutzten John E. Ward und Douglas Ross im interaktiven Grafik-Projekt »Kludge« am MIT das Ball-Computing-Prinzip in einem 3D »rate-control joystick«, den sie »chrystal ball« bzw. »globe« nannten. Sie manipulierten mit ihm dreidimensionale CAD-Grafiken auf einem Bildschirm, »so that one can get a feeling of the figures from all sides.« (Lindgren 1966, S. 65 f.; Wildes/Lindgren 1985, S. 350 f.) Lawrence G. Roberts, der erste Chefentwickler des von Licklider und Taylor angeregten *Arpanet*, baute die Lightpen durch Ultraschall-Sensoren zu einem 3D-Tracking-Device aus und erweiterte zusammen mit Timothy E. Johnson die Sketchpad-Software in den 3D-Bereich (Roberts 1989). Von dem Übergang zur Dreidimensionalität versprach sich Roberts (1964) eine weitere Erleichterung der Mensch-Computer-Kommunikation, denn: »A graphical language is tremendously powerful because it is a natural form of human representation and it derives richness and economy from its multi-dimensional character.« Sutherland selbst ging Mitte der 60er Jahre dann den nächsten Schritt von *statischen* 2D- und 3D-Objekten zu einer *dynamischen* Modellierung dreidimensionaler Räume.

Er schuf im Anschluss an frühe Helm-Display-Konstruktionen von Flugsimulatoren, wie sie Comeau und Bryan (1961) für die U.S. Airforce entwickelt hatten, und an multisensorische 3D-Kino-Systeme von Morton Heilig (Halbach 1994; Schröter 2004, S. 180 ff.) in den Jahren 1965-68 das erste Computer-gesteuerte »Head-Mounted Device« (HMD) mit zwei Miniatur-CRTs. Er experimentierte auch bereits 1966/67 mit durch Kopfhaltung gesteuertem Navigieren in virtuellen 3D-Räumen, die durch Drahtgitter angedeutet waren. Trotz der noch primitiven Visualisierung begriff er dieses Display bereits als Einstieg in eine völlig neue Interface-Welt, in der der Benutzer sich schwerkraftlos in mathematischen Räumen bewegen kann. Die Bedienung sollte nicht mehr durch Tastaturen, Knöpfe, Joysticks, Grafiktablets oder sonstige »manual input devices« erfolgen,

sondern durch »natürliche« Muskel- und Körperbewegungen, die über Sensoren erfasst würden. Selbst die Eingabe über die Augenstellung (»measure rotation of the eyeball«, Sutherland 1968, S. 757) wurde erwo-gen, aber wegen technischer Schwierigkeiten zunächst aufgegeben.

Ähnlich wie bei Engelbart und Licklider kulminierte Sutherlands kombinatorische Erkundung des MCI-Designraums in einem fiktiven Intermedium, das alle neuen »methods of controlling machines« integrierte. In dem Essay »Ultimate Display« von 1965 beschrieb er ein »kinesthetic display«, das mit einem Joystick mit »force feed back capability«, mit Hand-, Arm- bzw. Ganzkörperbewegungen oder Sprache gesteuert wird und daher auch über eine exquisite Audioausgabe verfügt. Dieses neuartige Computer-Display sollte nach Möglichkeit das gesamte natürliche Sinnespektrum ansprechen und dadurch zugleich die physikalische Welt wie auch fiktive Modellwelten abbilden können. So wie das Mikroskop die mikroskopische Welt und das Fernrohr die makroskopische Welt erschlossen habe, so würden Computer-Displays den Menschen befähigen, die Strukturen der vom Menschen erzeugten mathematischen Welt in einem elektronischen Mechanismus zu simulieren und zu erkunden, wobei es gleichgültig sei, ob die simulierten und manipulierten Objekte den Naturgesetzen folgten oder nicht:

»If the task of this display is to serve as a looking-glass into the mathematical wonderland constructed in the computer memory, it should serve as many senses as possible. [...] The ultimate display would, of course, be a room within which the computer can control the existence of matter [...] With appropriate programming such a display could literally be the Wonderland into which Alice walked.« (Sutherland 1965a, 507 f.)

Viele Hardware- und Software-Komponenten wie auch Nutzungskonzepte der Virtual Reality waren somit bei Sutherland schon in der zweiten Hälfte der 60er Jahre vorweggenommen, es fehlte nur noch der Begriff. Hier blieb er noch langezeit auf die literarischen Metaphern des »looking glass« und des »mathematical wonderland« fixiert. Doch auch ohne einen konkreten Leitbegriff lenkte Sutherland die MCI-Community in die Richtung multimodaler Interaktionsformen. Er erweiterte die von Engelbart erschlossene zweidimensionale Matrix der Ein-/Ausgabemöglichkeiten, die noch stark an Texten, statischen Grafiken, hierarchischen Begriffsnetzen und klassischen Kommunikationsmedien orientiert war, zu einem nahezu unerschöpflichen mehrdimensionalen multisensorischen Designraum und gab damit wichtige Anstöße für die spätere Natural Interface-Forschung. In der Praxis galt allerdings wegen der hohen Kosten und der unzureichenden Speicher- und Prozessorkapazitäten bis in die 80er und 90er Jah-

re hinein die Hauptaufmerksamkeit nicht den umfassenden Visionen von dreidimensionalen Datenräumen, Computer-simulierter multisensorischer Telepräsenz und ›Virtueller Realität‹, sondern den »conventional interfaces, working in flatlands« (Blundell/Schwarz 2005, S. 10).

Am Ende trieb Sutherland seine kombinatorischen Überlegungen über die Interfaces und Medien hinaus auf die Ebene der On-Line-Systeme. Denn von einem Zusammengehen der sich bisher getrennt entwickelnden »Process Control«, »Inquiry Systems«, »Problem-Solving«- und »Instrumentation-Systems« versprach er sich ungeahnte neue Möglichkeiten der Arbeitserleichterung, Arbeitsteilung und Wissensbeschaffung. Resultate wären »automated libraries«, die die Informationswünsche der User kennen, »on-line programming systems« einschließlich »program sharing« und räumlich verteilte »on-line design systems«: »We have yet to combine the functions of the design system and the inquiry system [...] The time is ripe to collect the techniques of pattern recognition, process control, and heuristic programming together to gain a new capability« (Sutherland 1965b, S. 11 f.). Durch fortschreitende Integration von »process control on-line studies« und »automatic problem-solving work« könnten in Zukunft »automata« die menschliche Interaktionsarbeit immer mehr ersetzen. Die Computer würden Wissen über ihre jeweilige Umgebung erlangen, den Benutzer von sich aus fragen und selber aktiv werden. Im Unterschied zu Licklider und vor allem zu Engelbart rechnete Sutherland nämlich mit einer relativ kurzen Übergangsperiode der ›symbiotischen‹ Online-Interaktionswelt. Daher beschäftigte er sich bereits 1965 mit dem Gedanken eines Menschen und Dinge verkoppelnden proaktiven Computing als Grundlage einer zukünftigen »leisure society«: »What I am predicting is that today's interest in systems in which a man and a machine get together on-line will be replaced in the distant future by interest in systems in which a computer gets directly on-line with the real world, sensing and interacting with it directly through transducers. The ›real world‹ with which such systems interact will include human beings, of course.« (Ebda., S. 13) Er gelangte damit noch Mitte der 60er Jahre wohl als einer der Ersten zu der Vision einer sich selber regelnden Welt von ›intelligenten‹ Dingen und Alltagsprozessen.

Die medienkombinatorische Synthese der PC-GUI-Welt

Mit der durch die neuen Positionsgeber und Zeiginstrumente ermöglichten grafischen Manipulation von 2D- und 3D-Objekten auf einem Rasterbildschirm wurden nicht nur die Grenzen der bisherigen zeilengebundenen textuellen Interaktion gesprengt, es erschloß sich nun auch ein nahezu unendlicher Raum von Software-Interfaces. Denn durch bloßes

Zeigen auf einen Punkt, eine Linie, eine Fläche, ein Wort, ein Feld mit Schaltfunktion oder ein Symbol konnten die unterschiedlichsten Software-Objekte und -Prozesse angewählt und Aktionen ausgelöst werden.¹² Das Zeigeelement entwickelte sich so zu einem Navigator zwischen virtuellen Schaltflächen wie Windows, Icons, Menus und Softkeys, mit denen sich verschiedene Bedeutungsebenen ansteuern, Zeichenprozesse aktivieren und mediale Funktionen realisieren ließen (siehe dazu unten den Beitrag von Nake). Dadurch entstanden völlig neuartige, Hardware-unabhängige symbolische Interaktionsmöglichkeiten. Sie erst verliehen dem Computer den Charakter eines dynamischen Mediums, das neben die klassischen Informations- und Kommunikationsmedien trat. Zu den bisherigen *Geräte-* und *Hardware-*Interface-Metaphern trat so eine Fülle von *Software-*Metaphern, die vor allem durch Sutherlands Schüler Alan Kay in der Ende der 60er Jahre konzipierten »Desktop-Metapher« gebündelt und im »graphical user interface« zu einem multifunktionalen »metamedium« integriert wurden (siehe Friedewald 1999, S. 249 ff; Barnes 2007).

Kay setzte sowohl die graphische wie auch die multimodale Entwicklungsrichtung seines Lehrers fort, doch ging es ihm nicht um synthetische Räume und ein mathematisches Wunderland, sondern um ein universelles Mikrocomputer-Lernmedium, das im Gegensatz zur passivisierenden TV-Berieselung und behaviouristischen Lernautomaten Wissens- und Phantasieräume aktiv und spielerisch erschließt und so die kreative Entfaltung fördert. Unter dem starken Einfluß der Reformpädagogik der 60er Jahre und in bewusster Anknüpfung an die Medienfiktionen von Bush, Licklider, Engelbart und Sutherland entwickelte er seine Vision eines »personal, portable information manipulator« ebenfalls in Form eines Zukunftsszenarios und einer »fantasy machine« (Kay 1972). Diese sollte den medialen Spielraum ausloten und zugleich als »Holy Grail version« des zukünftigen »personal computer« die Designer in ihren alltäglichen Entwicklungsarbeiten immer wieder an die ursprünglichen Zielsetzungen erinnern (Kay 1975, S. 4). Bei der Konkretisierung der Medienvision griff er weder auf das »Ultimate Display« Sutherlands noch auf die Electronic Library- und »Public Information Utility«-Konzepte der 60er Jahre zurück, sondern orientierte sich in Anlehnung an McLuhans Medientheorie an der Buchmetapher. So nannte er seine von einem grafischen Terminal ausgehende Medienprojektion ausdrücklich »*Dynabook*«. Als ein »carry anywhere device« sollte dieses aktive, dynamische Medium langfristig viele alte Medien in sich aufnehmen und völlig neue, noch nicht erfundene ermöglichen:

12 Um 1975 prägte David Canfield Smith im Rahmen seiner von Kay betreuten Dissertation für das generalisierte Konzept symbolischer Schaltfelder den Begriff »icon« (Perry/Voelcker 1989, S. 50).

»Imagine having your own self-contained knowledge manipulator in a portable package the size and shape of an ordinary notebook. How would you use it if it had enough power to outrace your senses of sight and hearing, enough capacity to store for later retrieval thousands of page-equivalents of reference materials, poems, letters, recipes, drawings, animations, musical stores, waveforms, dynamic simulations, and anything else you would like to create, remember, and change?« (Kay 1975, S. 2)

Obwohl Kay seine »Fantasy amplifier«-Vision und das »Personal Computing«-Szenario durchaus auch als »science fiction« verstanden wissen wollte, war er doch bei deren technischer Umsetzung darum bemüht, sich ganz realistisch an verfügbare Interfacetchniken zu halten (Kay 1972). Denn er setzte, auch hier den medienkombinatorischen Vorbildern folgend, nicht auf Technologiesprünge in der Zukunft, sondern auf die Neukombination erprobter Techniken. Auch er glaubte nicht an schnelle Erfolge der KI, denn »nobody knows whether artificial intelligence is a 10-year problem or a 100-year problem.« (Kay/Postman 1999) Er wollte die Interface- und Interaktions-Techniken nicht intelligenter machen, sondern ähnlich wie Bush und Engelbart in Fortführung erkennbarer Trends der vorhandenen Techniken mit Computern völlig neue Repräsentationsmöglichkeiten für Ideen schaffen: »We haven't gotten any smarter, we've just changed our representation system. We think better generally by inventing better representations; that's something that we as computer scientists recognize as one of the main things that we try to do.« (Kay 1989, S. 6) Über das Manipulieren von Repräsentationen sollten sich die Dynabook-Nutzer eigene Denk- und Phantasieräume aufbauen und mit anderen kommunizieren: »External mediums serve to capture internal thoughts for communications and, through feedback processes, to form the paths that thinking follows.« (Kay 1977, S. 231). Seinen eigenen Anteil an der Entstehung dieser »Personal Dynamic Media« (Kay/Goldberg 1977) erblickte Kay daher auch nicht so sehr in einer radikalen Neuentwicklung als vielmehr in der Integration von bereits Bestehendem, im Aufgreifen von Nutzerwünschen der Computerlaien sowie im Zusammenfügen von objektorientierten Softwaretools, symbolischen Interaktionsansätzen und neueren Interfaces zu einem universellen Spiel-, Wissens-, Design-, Planungs- und Kommunikations-Medium: »All the ingredients were already around. We were ready to notice what the theoretical frameworks from other fields of Bruner, Gallway, and others were trying to tell us. What is surprising to me is just how long it took to put it all together.« (Kay 1990, S. 196, siehe zu Kay unten den Beitrag von Pflüger)

Mit der Integration der graphischen Interaktionstechniken und der neuen grafikorientierten Interfaces bzw. Intermedien zur Desktop-GUI-

Welt fand die medienkombinatorische MCI-Revolution der 60er Jahre ihren Abschluss. Während *vor* Kay alle Neuerungen im Umgang mit Computern noch fast ausschließlich professionellen Anwendern galten, ermöglichte nun ein Angebot leicht zu bedienender Werkzeuge, mit Analogiebildungen, die die Computerbedienung viel enger an die Büro- und Alltagswelt heran rückten, selbst Kindern, die Funktionen und Inhalte des Metamediums selbst zu bestimmen und auszufüllen. Kays Designstrategie lief ja darauf hinaus, die Nutzer zu befähigen, sich ihre »personal tools« aus Bausteinen und Objekten selber zusammensetzen: »The burden of system design and specification is transferred to the user. This approach will only work if we do a very careful and comprehensive job of providing a general medium of communication which will allow ordinary users to casually and easily describe their desires for a specific tool.« (Kay/Goldberg 1977, S. 40 f.) Die neue Welt von Software-Interfaces befreite auf diese Weise die Computernutzung von den bisher dominierenden Zwängen vordefinierter algorithmischer Prozesse und Programmieranforderungen, sie lastete ihm damit aber auch die Mühe eines kleinschrittigen Vorgehens auf, ein Problem das Kay schon bald veranlasste, wie seine Lehrer Sutherland und Minsky nach intelligenter Entlastung des Users durch Automaten zu suchen.

7 Die Selektions- und Stabilisierungsphase des Interactive Computing

Alle wesentlichen Neuerungen bei Hard- und Software-Interfaces der 60er Jahre erfolgten noch in der Welt der Time-Sharing-Systeme. Erst im Laufe der 70er Jahre trat neben das an einen Mainframe oder Minicomputer angeschlossene Terminal der »Personal Computer« als geschlossene Hardware-Software-Konfiguration. Die Idee von »personal computers«, »personal consoles« bzw. »personal terminals« entstand zwar bereits kurz nach 1960 in der Time-Sharing-Community, doch handelte sich dabei nur um »remote input-output consoles which have all characteristics of a user's own personal computer with respect to access.« (Teager 1961 und weitere Belege bei Hellige 1996b, S. 218 f.) Erst Ende der 60er Jahre vollzog Kay den Bruch mit der Time-Sharing-Technologie und setzte nun voll auf Mikrocomputer und die LSI-Technik als Basis seines »personal computing medium«. Prototypisch wurde dieses Konzept erstmals 1971-76 in dem »Alto« am Xerox PARC realisiert, doch erst in der zweiten Hälfte der 70er Jahre begann die Markteinführung von Workstations und PCs. Hierdurch änderte sich der Umgang mit Computern grundlegend, denn: »data processing was liberated from centralized, fortress-like clean-rooms attended

by men in white coats, to appear on nearly every desktop and in millions of homes.« (Netravali 1999, S. 202)

Die durch die Fortschritte der Mikroelektronik forcierte PC-Ausbreitung profitierte in technischer Hinsicht entscheidend vom MCI-Innovations Schub der 60er Jahre, sie selber gab der Evolution der physikalischen Intermedien aber zunächst keine wesentlich neuen Anstöße. Vor allem aus Kostengründen dominierte in der PC-Entwicklung noch bis zum Anfang der 80er Jahre die *textuelle* Interaktion mit Keyboard und Drucker als wichtigsten Ein-/Ausgabemedien. Bildschirme waren anfangs die große Ausnahme, auch Maus, Joystick und Trackball blieben kostenbedingt auf professionelle Anwendungen beschränkt, die eher auf Mainframes und Workstations liefen. So beklagte Sutherland 1971 das Ausbleiben der Graphik-Revolution: »But today, after nearly ten years, we don't find acres of draftsmen using computer output terminals; we don't see widespread application of computer graphics in architecture, mechanical design, or even in numerical parts of programming. Why does it take so long for these very good ideas to catch on?«¹³ So kam es, dass die Massenausbreitung der graphischen Interface-Innovationen der 60er Jahre erst 15-20 Jahre später einsetzte. Ende der 80er Jahre hatte der Normal-PC-Besitzer, wie Roberts (1989, S. 65) voller Erstaunen feststellte, nahezu dieselbe grafische Hardware- und Software-Ausstattung, über die die Pioniere bereits in den 60igern verfügten: »So today, one has more or less the same capability even though the computers are a lot faster. And so you see a lot of the same programs, in terms of drawing programs and the like as Sketchpad originated.« Dies zeigt einmal mehr, mit wie langen Zeiträumen man bei der tatsächlichen Einführung und breiten Durchsetzung neuer Intermedien und Interaktionstechniken zu rechnen hat.

Die nun einsetzende *Diffusion* veränderte die Entwicklungsbedingungen und -richtungen der MCI grundlegend, es kam zu einer paradigmatischen Schließung der Interface-Entwicklung. Auf der Hardware-Seite führte das Massenprodukt PC zu einer starken Einengung der anfänglichen Vielfalt der Hardware-Schnittstellen auf die Standardkonfiguration »Bildschirm-Keyboards-Maus«. Während der noch nicht als marktfähiges Produkt geschaffene »Alto« die Wahl zwischen »mouse«, »joystick«, »stylus« und »tablet« offen ließ, sah das kommerzielle Folgeprojekt, der zwi-

13 Alan Kay ergänzte sein berühmtes »Predicting the Future«-Zitat von 1989 mit einer ähnlichen Feststellung: »Another way to predict the future is to realize that it takes a very long time – about 10 to 20 years – to get a technology out of the research lab and into everyday life. It's very difficult to get brand new ideas out in less than a decade; in the case of the transistor, it took almost 25 years. No matter what you do, it may take several companies, several different groups of people, several different areas of venture capital funding and more before you get something back.« (Kay 1989)

schen 1977 und 1981 als »office automation system« entwickelte »Xerox Star« nur noch die Maus als Eingabeinstrument vor (Johnson u.a. 1989). Die Maus hatte damit aufgrund ihrer relativ günstigeren ergonomischen Eigenschaften gegenüber den anderen Positionsgebern gesiegt, sie wurde zum universalen Ansteuerungsinstrument. Die Maus erlebte zwar eine Reihe von Verbesserungsinnovationen, doch trotz der Entwicklung von der Räder- über die Rollmaus zur drahtlosen Optomaus, trotz der Variation bei ihren Funktionstasten und der gelegentlichen Hinzufügung eines Rollrades oder eines Trackballs blieb die grundlegende Funktionalität als Instrument der Cursor-Ansteuerung bis heute nahezu unverändert. Das Papier blieb, obwohl es ursprünglich durch das »Electronic Paper« auf dem CRT-Display abgelöst werden sollte, neben dem Bildschirm weiterhin das wichtigste Ausgabemedium und ein wesentlicher, meist verdrängter Bestandteil der sich etablierenden intermedialen Konstellation. Diese blieb über mehr als 20 Jahre nahezu unverändert, obwohl sich Prozessorleistung, Arbeitsspeicher- und Festplattenkapazität eines PC in diesem Zeitraum weit mehr als vertausendfachten (Baudouin-Lafon 2004, S. 15).

Abb.6: Xerox Star 1810, der 1981 als erster kommerzieller PC mit grafischer Benutzungsoberfläche herausgebracht wurde. (Quelle Xerox-PARC-Archiv; img.zdnet.com/techDirectory/_STAR1.GIF)



Während die Hardware-Innovationen stagnierten, verlagerte sich die Entwicklungsdynamik in diesem Zeitraum zu den Software-Interfaces hin. Es entstand hier eine immer vielfältigere Welt von virtuellen Objekten und Operationen im zwei- und bald auch dreidimensionalen Datenraum, die

der Benutzer per Mausklick ansteuern bzw. auslösen konnte (ausführlich dazu unten Müller-Prove). Durch die zunehmende Visualisierung verschob sich der Anteil der Eingabe-/Ausgabeoperationen am Gesamtvolumen der Computerbefehle beträchtlich. Lag der Input-/Output-Anteil in den 70er Jahren noch bei ca. 10%, stieg er bis zur Mitte der 90er Jahre auf rund 85%: »Die Rechnersysteme scheinen uns auf merkwürdige Weise zu imitieren, indem sie mehr und mehr ihrer Leistungsfähigkeit der Ein- und Ausgabe widmen, denn drei Viertel unserer Hirnrinde dienen dem Sehen, also unserem wichtigsten Sinnesorgan.« (Dertouzos 1999, S. 107) Doch auch die technische Evolution der Software-Interfaces beruhte während der Selektions- und Stabilisierungsphase noch immer weitgehend auf dem Innovationsschub der späten 60er und frühen 70er Jahre: »Most of the code in modern desktop productivity applications resides in the user interface. But despite its centrality, the user interface field is currently in a rut: the WIMP (Windows, Icons, Menus, Point-and-Click) GUI based on keyboard and mouse has evolved little since it was pioneered by Xerox PARC in the early '70s.« (Van Dam 1999) Selbst im Jahre 2006 ergab eine Bestandsaufnahme der HCI-Entwicklung nur geringe grundlegende Fortschritte seit dem noch immer vorbildlichen »Xerox-Star«-Design von 1981: »HCI hasn't produced major innovations in the last 20 years; the WIMP interface today is almost identical to what it was in the 1980s.« (Canny 2006, S. 26)

Die Monopolstellung der Maus unter den Positioniergeräten blieb bis in die 90er Jahre unangetastet. Erst durch den Aufstieg der Portables in den späten 80er Jahren und der mobilen Computer seit den frühen 90ern erlebten Joystick und Trackball, aber auch der Touchscreen eine größere Verbreitung. Waren diese bis dahin weitgehend auf Nischenanwendungen wie den Spielektor, professionelle Spezialanwendungen oder öffentliche Terminals beschränkt, so ersetzen sie nun als »built-in pointing devices« in Laptop- und Notebook-Keyboards die Maus (Reinhardt 1991). Die ergonomischen Defizite der reinen Formen führten bald zu einer vielgestaltigen Variation und Rekombination verschiedener Zeigeinstrumente und miniaturisierter Grafiktablets zu neuen Mischformen.

Den Anfang machten noch in den 80er Jahren die »Wobbleplate«, die bei Xerox (US-Patent 431113) als Verbesserung der getrennten Pfeiltasten-Bedienung entwickelt wurde, sowie Touch Tablet und Touchpad, die Hewlett-Packard und Epson aus der Kombination von Softkey-Prinzip und miniaturisiertem Touchscreen schufen. Der von Ted Selker für die IBM-Notebooks entwickelte »Trackpoint« verkleinerte den schon seit den 60er Jahren verwendeten drucksensiblen isometrischen Joystick und plazierte ihn inmitten der Tastatur. Doch trotz einer nahezu zehnjährigen aufwendigen Entwicklung und mannigfacher Imitationen bei anderen

Computerherstellern gelang dem Trackpoint weder die erhoffte Totalverdrängung der Maus noch eine führende Stellung bei den »keyboard pointing devices« (Thompson 1996). In den 90er Jahren folgten noch weitere Kombinationen taktile Zeigeinstrumente, die wie Twintouch, Touchmouse, Isopoint, Keymouse, Unmouse und Touch-Trackball als Mausersatz bei Notebooks und Handhelds dienen sollten oder die wie Puck Pointer, Ballpoint-Pen und Trackmouse nur ein sehr begrenztes Navigierfeld benötigten (Hinckley/Sinclair 1999). Aus der Interface-Lücke, die durch den Aufstieg miniaturisierter mobiler Computer entstanden war, ging so ein ganzes Feld von Interface-Varianten hervor, die mit ihren mannigfachen Kreuzungen, Rekombinationen und Mimikry-Effekten ein Musterbeispiel evolutionärer Technikgenese darstellen, wie sie Basalla (1988) theoretisch entwickelt hat.

Für eine noch weitergehende Schrumpfung der Computergröße reichten aber auch diese Größenanpassungen und evolutionären Weiterentwicklungen bestehender Ein-/Ausgabemedien nicht mehr aus. Eine Unterschreitung der durch das Bedienfeld als kritischem Engpassfaktor definierten Gerätegröße war nur möglich durch Lösungsansätze, die den Bereich grafikorientierter Hardware-Interaktionstechniken hinter sich ließen. Neben der Miniaturisierungs-Problematik bei der Bedienfläche gab es seit der Mitte der 80er Jahre aber noch weitere Anlässe für eine Suche nach Alternativen zum sich gerade etablierenden Desktop- und WIMP-Paradigma. Da war die zunehmende Ausbreitung von Public Access Systemen wie Informations-Kioske, Online-Kataloge und Selbstbedienungsterminals, bei denen weder die Maus noch komplizierte GUI-Dialoge geeignet sind. Doch auch beim Desktop-Computing selber bewirkte das kleinschrittige Vorgehen der »Direkten Manipulation« mit zunehmender Komplexität der Anwendungs-Software immer mehr Verdruss. Infolge der aus Marktstrategien resultierenden Hardware-Software-Spirale bei PCs und der skalenökonomisch bedingten geringen Spezialisierung wuchs die Funktionalität der Softwarepakete derart, dass die Errungenschaften des Graphical User Interface, Visualisierung und sequentielle Eingaben mit ständigem Feedback, immer dysfunktionaler wurden (Norman 1998, S. 72 ff.). Schließlich nahm infolge der vermehrten multimedialen Ausstattung von Workstations und PCs, des Einsatzes von Wall-Displays und 3D-Visualisierungen, insbesondere von Virtual Reality-Systemen, die Asymmetrie zwischen Eingabe- und Ausgabetechniken ständig zu: »Computers today are deaf, dumb, and blind. They communicate with their environment principally through an slow serial line to a keyboard and mouse.« (Pentland 1995, S. 71) All dies löste noch in den 80er Jahren ein Interesse an breitbandigeren Eingabe-Systemen und »natürlicheren« Formen des »Mensch-Computer-Dialogs« aus, der sich in verstärkten Ent-

wicklungsaktivitäten auf den Gebieten der Sprach-, Schrift-, Blick- und Gestenerkennung niederschlug. Hieraus entwickelten sich nach 1990 als neue MCI-Leitbilder das »Non-WIMP-User-Interface«, die »Post-Mouse-Era« und, positiv gewendet, die »Natural User Interfaces«.

8 Natural User Interfaces als Alternative zum Desktop-Paradigma

Die Hoffnung auf eine baldige Realisierung einer quasi natürlichen Kommunikation mit Computern wurde trotz permanenten Scheiterns nie aufgegeben. So plädierte Raymond S. Nickerson 1969 für eine Umkehrung der HCI-Forschung: »the need for the future is not so much for computer-oriented people as for people-oriented computers« (zit. nach Newell 1985, S. 231). Fünf Jahre später verkündete Morton I. Bernstein (1974, S. 194) das neue Entwicklungsziel: »With the tremendous developments in computing which we have witnessed in the past two decades, it is now time that computers should be ›humanized‹ and that many men and woman be liberated from distance between men and machines.« Schon 1980 sollten Ein- und Ausgaben in der natürlichsten Weise erfolgen: »One will speak those things normally spoken and draw pictures and write equations for those things normally drawn and written. The computer's response will be the spoken word, perhaps a song where appropriate, pictures, graphs, and charts, and even the printed word.« (Ebda., S. 183). Wenige Jahre später formulierte Peter F. Jones als neues MCI-Design-Prinzip, »that the human interface with the computer is best when it resembles human communication as much as possible.« (zit. nach Newell 1985, S. 231) Im Jahre 1987 verkündete dann der IBM-Forschungschef den natürlichen Umgang mit Computern als ein wesentliches Charakteristikum der »Next Computer Revolution«: »A considerable fraction of the computer's power will go to making the machine easier to use by accommodating a natural human-machine interaction based for example on voice and handwriting.« (Peled 1987, S. 40) In der ersten Hälfte der 90er Jahre wurden die anfangs noch separaten Bestrebungen zur Realisierung von »natural language« bzw. »natural gesture interfaces« unter den Begriffen »Natural I/O« und »Natural User Interfaces« zu einem neuen Leitbild mit hohen Erwartungen erhoben: »Natural I/O is the ultimate user-friendly interface. It places the burden of communication squarely on the computer rather than on the human: Instead of the user having to learn the computer's preferred interface, the computer must deal with the user's preferences.« (Caudill 1992, S. 135)

Sprach- und Gesten-Interfaces als Hoffnungsträger

Im Mittelpunkt der Bemühungen um eine natürlichere MCI stand noch immer die Sprachverarbeitung. Da selbst nach der ersten großen Ernüchterung in den 60er Jahren die Sprache weiterhin als die natürlichste Kommunikationsform auch zwischen Mensch und Rechner galt, wurden die Forschungen zu »speech interfaces« auch in den folgenden Jahrzehnten kontinuierlich weitergetrieben. Daraus gingen in den 70er Jahren erste noch sprecherabhängige Spracherkennungssysteme mit kleinem Vokabular und Einzelworteingabe hervor (Norberg/O'Neill 1996, S. 224-239). Die aus jedem Fortschritt folgenden Versprechungen der Entwickler und Anbieter vom endgültigen Durchbruch endeten jedoch regelmäßig in einer Ernüchterung: »Over the past four decades it has often been felt that the solution to the problem of ›machine recognition of speech‹ is just around the corner.« (Hogan 1983, S. 178)

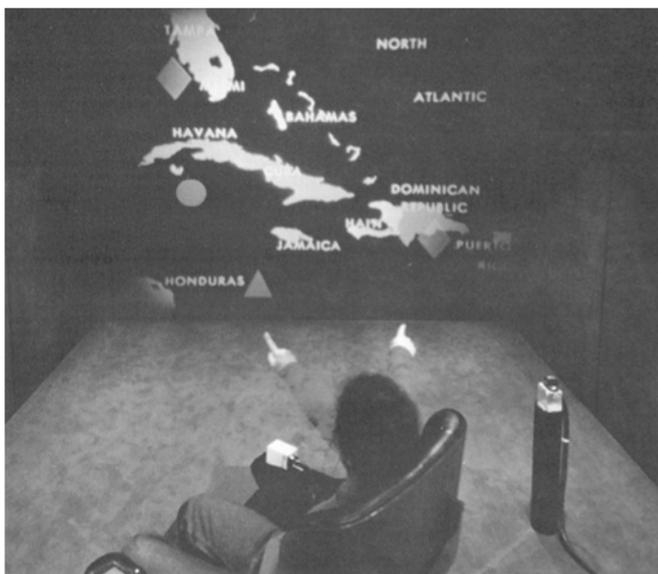
Höhere Speicher- und Prozessorleistungen ermöglichten seit Mitte der 80er Jahre die Verwendung größerer Wörterbücher und erste Kontextabgleiche auf der Basis statistischer Überprüfungen von Worthäufigkeiten und Wortkombinationen, dies alles lief freilich noch immer auf Großrechnern. Erst seit den 90er Jahren kamen erste Produkte für PCs und Workstations auf den Markt, die allerdings erst Ende des Jahrzehnts für den Allgemeingebrauch geeignet waren (vgl. u.a. Rabiner 1995). Nach 2000 begann die Einführung von Sprachverarbeitungssystemen für mobile Geräte, diese haben aber mit ähnlichen Ressourcenproblemen zu kämpfen wie vordem die stationären PC-Systeme. Doch trotz ständig leistungsfähigerer Prozessoren und erfolgreichem Einsatz von statistischen Methoden (Markov-Ketten) bietet die Verarbeitung sprecherunabhängiger fließender Sprache noch immer für den Alltagsgebrauch von Diktier- und Dialogsystemen unbefriedigende Erkennungsraten. Nur die Sprachsteuerungs- und -dialogsysteme mit begrenztem Befehlssatz arbeiten schon relativ zuverlässig. Die gegenüber den GUI-Interfaces langsamere Sprachein-/ausgabe, fehlendes oder unzureichendes Feedback sowie umständliche Korrekturprozeduren lassen »speech interfaces« für den Benutzer noch immer nicht als die natürliche Alternative zur WIMP-Welt erscheinen, als die sie seit langem propagiert wurden. Rabiners Erwartung (1993, S. 495), dass in den Jahren nach 2000 die natürlichsprachige Interaktion mit »totally unrestricted vocabulary, syntax, and semantics, including the capability of translating telephony« erreicht würde, hat sich so wie die früheren Prognosen für die Sprachverarbeitung als zu optimistisch erwiesen. »Human-human relationships«, so das ernüchterte Resultat von Ben Shneiderman (2000, S. 63), »are rarely a good model for designing effective user interfaces. Spoken language is effective for human-human interaction but often

has severe limitations when applied to human computer interaction.« Auch die Bestandsaufnahmen von Kato (1995, S. 10060) und Ogden/Bernick (1998, S. 150 ff.) stellten fest, dass die jahrzehntelange Erfahrung gezeigt habe, dass Voice Interfaces sehr spezieller Natur seien und daher nicht als generelle Lösung des »man-machine interface« taugten, zumal den »Voice-Processing Technologies« der für die zwischenmenschliche Kommunikation charakteristische Kontextbezug fehle. Trotz dieser Einsichten werden Speech-Systems in den meisten MCI-Zukunftsszenarien nach wie vor als *die* Patentlösung für die Mensch-Computer-Kommunikation gehandelt.

Als die nach der Sprachverarbeitung »natürlichste« Interaktionstechnik galt und gilt die Gesteneingabe. Da in der zwischenmenschlichen Verständigung mit sehr einfachen, aber aussagekräftigen Hand- und Körpergesten sehr komplexe Sachverhalte kommuniziert werden können, erhofft sich die MCI-Forschung vom gezielten Gesteneinsatz eine vereinfachte Handhabung der zunehmend komplexeren Anwendungsprogramme. Als Gesten dienen dabei sowohl vereinbarte Hand- und Fingerstellungen, Kopf- oder Körperbewegungen als auch »drawing gestures« nach dem Muster von Korrekturzeichen oder stenografischen Kürzeln (Huang/Pavlovic 1995). Die Entwicklung von Gesteneingabetechniken setzte bereits in den 60er Jahren ein, zunächst mit dem Schwerpunkt »pen-based gestures« im Anschluß an die »Light-pen gestures« in Sutherlands »Sketchpad« und die Gestenerkennung auf dem Rand-Tablet. Um 1970 kam dann auch die maschinelle Interpretation von Handgesten ins Spiel, hier führte das Problem der Interaktion mit Walldisplays, die speziell beim US-Militär eingesetzt wurden, zu ersten Entwicklungen von Gesture-Systemen.

Im militärischen Kontext entstand ab 1976 auch das bahnbrechende »Put That There-System« der »Architecture Machine Group« am MIT um Nicholas Negroponte und Richard Bolt. Hier wurde das Muster eines »Media Room« geschaffen, in dem der User von einem Sessel aus über kombinierte Sprachbefehle und Zeigegesten »command events« auf einem visualisierten Einsatzfeld auslöste (Bolt 1980, siehe die Abb. 7). Das als »image of an office of the future« geplante System wurde zwar durch ein »spatial data management« zu einem persönlichen Dokumentenbearbeitungs- und Lesemedium ausgebaut (»Dataland« und »Books without pages«), aber mit seinem zentralen Steuerstand im abgedunkelten Raum und der Interaktion mit dem Walldisplay per Zeige-, Touch- und Sprachkommandos ähnelte es eher noch den Command-and Control-Interfaces der SAGE-Ära als den immersiven Virtual Environments der 80/90er Jahre (Bolt 1979; Brand 1990, S. 170 ff.; Wagner 2006).

Abb. 7: Das »Put-that-there-System« der »Architecture Machine-Group« am MIT kombinierte erstmals Sprach- und Gesten-Erkennung (Bolt 1980)



In den 80er Jahren wurde das Anwendungsspektrum der Gesteneingabe stark ausgeweitet. Die Kombination der Zeigefunktion mit Menüs ermöglichte bald die direkte Manipulation von beliebigen Text-, Grafik- und Multimedia-Dokumenten. Doch erst die Einführung von Datenhandschuhen und Videokameras für das Tracking und Erkennen von Gesten erschloss über die bloßen Zeigegesten hinaus den Bereich emblematischer bzw. semiotischer Gesten, d.h. symbolischer Handbewegungen mit kulturell tradiert oder vorab definierter Bedeutung. Der Datenhandschuh, den Engelbart bereits als Eingabemöglichkeit erwogen hatte, entstand zuerst 1976/77 an der School of Art & Design der University of Illinois in Chicago. Die Computerkünstler und Interface-Entwickler Daniel J. Sandin und Thomas A. DeFanti entwickelten den »Sayre Glove«, benannt nach dem Anreger der Idee Richard Sayre, um damit Sound- und Lichteffekte bei elektronischen Musik- und Multimedia-Performances zu »dirigieren« (Sturman/Zeltzer 1994).

Auch die Gestenerkennung per Videokamera ging aus der Nutzung des Computers zu künstlerischen Zwecken hervor. Myron W. Krueger setzte 1969/70 als erster Videokameras zum Tracking von Hand- und Körpergesten ein, da er Sutherlands HMD und die 3D-Zeigegeräte in seinen experimentellen »responsive environments« als zu störend empfand.

Er schuf durch die Projektion von Gesteneingaben auf eine Wand zwischen 1969/70 und 1987 eine ganze Reihe von interaktiven Rauminstallationen, die als Vorformen der Virtuellen Realität angesehen werden können. In der hierdurch erreichten Kombination von »gesture input« mit Großdisplays sah er die Basis für ein »new aesthetic medium«, für das er Mitte der 70er Jahre den Begriff »Artificial Reality« prägte (Krueger 1991, S. XIII). Bereits 1970 hatten er und Sandin sogar schon an eine Rundumprojektionsfläche für 3D-Visualisierungen gedacht. Unter Kruegers Einfluss entwickelten Sandin, Cruz-Neira und DeFanti dann 1991 tatsächlich das »CAVE Automatic Environment« (Turner/Krueger 2002). Damit waren Computerkünstler und Mediendesigner als eine neue Gruppe von Innovatoren auf dem Gebiet der Bedienschnittstellen und Interaktionsmedien hervorgetreten, die vor allem der Simulation virtueller Räume, der Ganzkörpergestik und der multimodalen Ausweitung der Interaktionsformen neue Impulse gaben.

Die Virtual Reality als Kombinationskonzept für die »Natural Interfaces«

Durch die Mediendesigner erweiterte sich das bis dahin stark auf HMDs und Cockpit-Simulationen fixierte 3D-Interface-Forschungsprogramm der US Air Force unter der Leitung von Thomas A. Furness in Richtung auf weniger kostspielige, auch zivil verwendbare »virtual spaces« bzw. »virtual worlds«. Doch letztlich gelang die Öffnung der Entwicklung erst nach 1989 im Rahmen der zivilen Ausgründung des »Human Interface Technology Lab« (Furness 1986 und 1988; Rheingold 1991, Kap. 9). Erfolgreicher bei der Zusammenführung der 3D-Simulationsentwicklungsrichtungen wurde so ein weiterer Akteur, die NASA. Sie war besonders an Flugsimulatoren für das Astronauten-Training auf der Erde sowie an Teleaktorik- und Telepräsenz-Medien für Raummissionen interessiert und wollte deshalb Interfacetechniken wie Datenhandschuhe, Videokameras, Head- und Helmet-Mounted Devices, die Methoden der Gestensteuerung Computer-generierter Objekte und der 3D-Visualisierung zu einem funktionsfähigen Gesamtsystem integrieren (Ellis 1995, S. 21 ff.). Sie initiierte ab 1981 die Entwicklung verbesserter Head- bzw. Helmet-Mounted Devices auf LCD-Basis (»Virtual Visual Environment Displays«), die die ergonomischen Mängel der in der Air Force verwendeten HMDs überwand, und unterstützte besonders die Weiterentwicklung von Thomas G. Zimmermans »optical flex sensor« (1981/82, U.S. Patent 4.542.291, 17.9.1985) zu einem marktfähigen Produkt.

Den zwischen 1985-87 geschaffenen »DataGlove« verstanden seine Entwickler Thomas Zimmerman und Jaron Lanier (1987, S. 192), nun

nicht mehr nur als ein spezielles Gesteninterface, sondern als eine »natürliche« Alternative zu Maus und Joystick in der Manipulation von 2D- und 3D-Objekten und als Zugang zur neuen Welt der Computersimulation: »Just as speech is our natural means of communication, the human hand is our natural means of manipulating the physical world. As computer systems begin to simulate the physical world, the technologies presented in this paper suggest a broad spectrum of possibilities to a wide variety of users. It is increasingly important that we shape the simulated world of our computer in ways which reflect our human universe, rather than allow ourselves to be shaped by our machines.« Das Erscheinen des DataGlove auf dem Markt wird in historischen Abrissen meist als das Schlüsselereignis für die Genese der »Virtual Reality« gewertet und in unmittelbarem Zusammenhang mit der von Jaron Lanier beanspruchten Prägung des Begriffs gebracht.¹⁴ Doch entgegen Laniers Selbststilisierung als VR-Guru war es vor allem der Leiter des Virtual Environment Workstation Project (VIEW) am NASA-Ames Research Center Scott S. Fisher, der die entscheidenden Weichen für die Integration der einzelnen Entwicklungsstränge stellte.

Der »artist-engineer« Fisher verknüpfte ab 1981 die Stereoskopie- und 3D-Simulations-Konzepte der Architecture Machine Group, die vorwiegend aus der militärischen Sphäre stammenden HMD-Entwicklungen und die künstlerische Richtung kollaborativer Telepräsenz-Experimente Myron Kruegers (Rheingold 1991, Kap. 6). Durch stufenweise Integration der einzelnen Eingabe-/Ausgabe-Techniken schuf er ein »three-dimensional visual environment« (1982) bzw. ein »virtual environment display system controlled by operator position, voice and gesture« (Fisher 1986, S. 1) Wie Morton Heilig und Sutherland wollte er am Ende alle Sinne in die »Telepresence«-Simulation einbeziehen, also auch »auditory feedback«, »tactile interaction«, »eye-tracking«, »full arm and body interac-

14 Das erste Auftreten des Fachbegriffs ist nach wie vor ungeklärt. Lanier behauptete 1999, er habe den Begriff »in the early days of our company« (1985 gepr.) zur Abgrenzung von Sutherlands »virtual worlds« geprägt, um den Vernetzungsaspekt zu betonen. 2001 datierte er vager auf »in the mid-to-late 1980s«. In dem bekannten Vortrag auf der CHI '87 (Zimmerman/Lanier 1987) ist nur von »virtual objects« bzw. »environments« die Rede, erst 1989 taucht »virtual reality« in Laniers Texten auf, als schon andere Autoren den Begriff verwendeten (Lanier 1989). Die im Oxford English Dictionary nachgewiesene erste Verwendung durch Yaak Garb (1987) knüpft an den Begriff der »virtual machine« an und bezieht sich ohne jeden Hinweis auf die 3D-Interaktion nur allgemein auf die zunehmend abstrakter werdende Computertechnik. Die Menschen operieren am Ende nur noch fern der »physical reality« in »virtual realities«, d.h. im »mind space« und in »selfcontained symbolic worlds« (Vgl. allgemein hierzu Schroeter 2004, S. 206 ff.; 216 ff.).

tion« (Fisher 1990, S. 429 ff.). Seine Ziele waren dabei eine »operator interface configuration that features human matched displays and controls for transparent, natural system interaction« und »collaborative virtual environments.« (Fisher 1986, S. 1) Die »virtuellen Räume« wurden so zu einem Integrationskonzept für alle Einzeltechniken der »natural interfaces« und zu einem Schmelztigel vieler Medien, ja am MIT Media Lab sah man darin den Inbegriff eines User Interface: »Interactive virtual environments provide perhaps the greatest generalization of the notion of a user interface, as they seek to »immerse« a user into a machine-made world. A user should be able to interact with objects or agents in these environments in as natural a manner as possible.« (Darrell/Pentland 1995, S. 135)

Die in Prototypen erreichte Konvergenz vieler Interface- und Sensortechnologien ließ sehr bald die Erwartung aufkommen, die Verwirklichung des »ultimate display« bzw. des »final interface« stünde unmittelbar bevor. Die »Virtuelle Umgebung« erschien als »eine neue Epoche der Mensch-Maschine-Kommunikation«, die das Potential besitze, Industrie, Gesellschaft und den Alltag der Menschen innerhalb kurzer Zeit völlig umzuwälzen (Astheimer/Böhm u.a, 1994, S. 282; Walker 1990; Jacobson 1992). Auf dem Höhepunkt des 1988/89 abrupt einsetzenden Hypes entwickelten sich die VR-Leitbilder zu einer regelrechten Heilsbotschaft:

»Does this mean that VR represents the early stages of the arrival of the ultimate medium? Or, as in the past, are we just projecting this longstanding desire for the essential copy and for physical transcendence - for a dream machine - onto another promising but inadequate technology? [...] It is fitting that VR appears at the end of the millennium. One suspects that the ultimate display wears a messianic crown of thorns; it is a technoGodot, long awaited but yet to arrive. It may even be an expression of our desire to assume a Godlike control of reality.« (Biocca/Levy 1995, S. 13)

Doch die bereits ab 1991/92 erkennbare »interface crisis« bei Virtual Environments machte die Grenzen der VR deutlich, und der Höhenflug war schon nach wenigen Jahren zu Ende (Bowman/Kruijff 2005, S. 13). Man mußte einsehen, dass trotz erreichter Fortschritte »the major technical problems that existed in the late 1960s remain with us today« (Holloway 1992, S. 180). Denn HMDs, Datenhandschuhe und Bodysuits waren für die Alltagsnutzung zu invasiv, und die Immersion isolierte den Nutzer völlig von seiner Umgebung. Die extrem hohen Einstiegskosten schränkten das Anwendungsspektrum stark ein und ließen die VR wie ein Jahrzehnt zuvor die Computergrafik zu einer »solution in search of a problem« werden (Machover/Tice 1994, S. 15). Der »Cyberspace« als Gegenwelt zum bestehenden Interface-Repertoire und als finales Medium wurden so

entmystifiziert zu »Spatial Input« und »3D User Interfaces«. Diese entwickelten sich zu einer zunehmend erfolgreichen Spezialtechnik, die *neben* die bestehenden Intermedien trat, diese aber nicht verdrängte. Der integrale Ansatz der multisensuellen und multimodalen Virtual Environments wurde wieder zurückgestellt zugunsten einer intensiveren Erforschung der einzelnen Natural Interface-Techniken.

Die zweite NUI-Welle: Augen, Gesicht und Hirn als Eingabemedien

Die Gesture-Eingabetechniken, die sehr vom VR-Hype profitiert hatten, litten besonders stark unter dessen Ende: Statt der favorisierten Datenhandschuhe und HMDs wurden 3D-Erweiterungen der klassischen Positioniergeräte Maus, Trackball, Joystick und Grafiktablett, also Space mouse, Spaceball, 3D-Joystick und 3-Draw-System zu Standard-Eingabegeräten der nun das Gebiet beherrschenden »Desktop-VR«. ¹⁵ Auch die Stilisierung des mit skriptuellen Gesten gesteuerten Pencomputers zum universalen intelligenten Informations- und Kommunikationsmedium, das den Desktop-PC endgültig ablösen sollte, scheiterte schon nach wenigen Jahren kläglich an den Mängeln der Handschriftenerkennung und des Gestendialoges. Erst neuerdings erleben einzelne, als *Zusatztechnik* verwendete Zeigegesten bei mobilen Geräten größere Verbreitung. Von dem nach der Markteinführung des »DataSuit« 1987 ausgelösten Body-suit-Mythos blieb nur ein bescheidenes Randgebiet des Arm- und Bodytracking übrig. Der Schwerpunkt von Forschung und Entwicklung verlagerte sich stattdessen zu Input-Techniken mit kürzeren Stellwegen, zum Head-, Face- und Eye-Tracking sowie zum Lipreading, das als Ergänzung der Sprachverarbeitung gedacht war.

Die Computerbedienung mit den Augen schien dabei besonders vielversprechend, da Augenbewegungen schneller als alle anderen Eingabemedien sind. Seit der Verfügbarkeit von Eye-Tracking-HMDs ab 1948 gab es daher viele Bemühungen, Blicke für Bombenabwürfe und Gerätesteuerungen, aber auch für Marktforschungen und HCI-Usability-Studien zu nutzen (Jacob/Karn 2003). Für die Computerbedienung kamen Eye-Tracking-Systeme erst seit den 80er Jahren infrage, als die Aufzeichnung weniger störend per Videokamera erfolgte (zum Pioniersystem siehe Bolt 1982). Doch die ständigen unbewussten Lidbewegungen und die schwierige Realisierung von Maustasten-Funktionen machen die Blick-

15 Die 2003 als eine die Internetwelt umwälzende Massenanzahl konzipierte virtuelle Multiuser-Plattform »Secondlife« muss gar wegen der geringen Verbreitung von 3D-Positioniergeräten für das Teleportieren von Avataren noch auf Pfeiltasten zurückgreifen.

auswertung am Ende aufwendiger, langsamer und unzuverlässiger als die klassischen manuellen Eingabetechniken. Ähnlich wie bei den Speech Interfaces tauchten auch beim Eye-Tracking in jedem Jahrzehnt seit 1950 neue Probleme auf, dadurch wurde auch diese Technik zu einem »Phoenix raising from the ashes again and again with each new generation of engineers« (J. W. Senders 2000, zit. nach Jacob/Karn 2003, S. 575). Gleichwohl gilt Eye-Gaze noch immer als großer Hoffnungsträger: »Scanning light beams to the retina could revolutionize displays for everything from cellphones to games.« (Lewis 2004 S. 16)

Noch größer sind die technischen Probleme bei der direkten Übersetzung von Hirnsignalen in Computerbefehle in den sogenannten Brain-Computer-Interfaces (BCI). Da bei ihnen die Maschinenbedienung ohne Stellwege und daher mit kürzester Reaktionszeit erfolgt und dabei, so hofft man, auch komplexere Denkinhalte abgetastet werden können, gelten sie bei manchen MCI-Forschern als ideale Lösung für die Aufhebung der Asymmetrie zwischen Ein- und Ausgabemedien oder gar als das Endstadium der MCI: »The ultimate HCI is the ability to control devices through the modulation of one's brain signals.« (Watson/Luebke 2005) Ray Kurzweil und Hans Moravec sehen im Scanning ganzer Hirninhalte und der Direktkopplung von Hirn und Computer sogar den Endzustand der Mensch-Computer-Beziehung.

Die in den 1970er Jahren einsetzende und seit den Neunzigern expandierende Forschung auf diesem Gebiet hat jedoch gezeigt, dass nicht-invasive BCI-Systeme nur eine geringe Bandbreite der Hirn-Computer-Verbindung ergeben. Die Asymmetrie zwischen Input und Output wird dadurch also noch erhöht. Invasive BCI-Systeme dagegen verfügen zwar über höhere Bandbreiten, doch sie stellen aufgrund der erforderlichen Hirnimplantate eine derartige Beeinträchtigung dar, dass sie sich allenfalls bei Schwerstbehinderten rechtfertigen lassen (Lebedev/Nicolelis 2006, S. 536 ff.). Insgesamt machen lange Abtast- und Auswertungszeiten für EEG-Daten BCIs aber noch aufwendiger, langsamer und unzuverlässiger als Eye-Tracking-Systeme.¹⁶ Doch erste experimentelle Erfolge bei Cursor-Steuerungen und »mental typewriters« nährten um 2000 die Erwartung, »that such a future is just around the corner.« (ebda., S. 535) In MCI-Zukunftsvisionen stehen BCIs ungeachtet aller Schwierigkeiten als Überwinder der WIMP-Welt sogar oft an vorderster Stelle: »Ultimately I prefer the computing environment would simply carry out my wishes via

16 Dass hierbei offenbar grundsätzliche Grenzen bestehen, zeigen Ebrahimi u.a (2003, S. 9): »One of the first problems to address is the limitation of the information transferrate, which is at best currently 20 bits/min. It seems dubious that BCI protocols based on mental task classification can improve this figure by much.«

a form of telepathy - cogito ergo fac - (I think therefore do it).« (Van Dam 2001, S. 50)

Der Anspruch von Forschern und Entwicklern, die Mensch-Computer-Interaktion so weit wie möglich natürlichen Kommunikationsweisen anzunähern, bewirkte eine fortschreitende Anthropomorphisierung der Interfaces. Sie entwickelten sich von stationären über portable zu am Körper getragenen Kleidungsstücken (Handschuhe, Helme, Gürtel) Prothesen (Shutterbrillen, Hörgeräte) und Sensoren. Bei diesen soll einerseits der physische Aufwand durch Verkürzung der Stellwege minimalisiert werden, andererseits sollen möglichst viele Sinne bzw. Kommunikationskanäle einbezogen und kombiniert werden: Sprache, Hand- und Körpergesten, Gesichtsausdruck, Blicke und am Ende Hirnaktivitäten. Die Erfassung und algorithmische Interpretation dieser verschiedenen Zeichenprozesse sowie deren Umsetzung in Computerprozesse ist bislang nur mit semantisch vereinbarten Befehlssprachen wirklich gelungen, doch diese sind schwerer zu erlernen und komplizierter zu bedienen als herkömmliche GUI-Techniken. Andererseits stecken die Versuche, die Realisierungsprobleme der Interaktion mit Natural Interfaces durch einen Verzicht auf die direkte Manipulation und eine deutliche Reduktion des Interaktionsaufwandes zu lösen, noch ganz in den Anfängen. Denn die Verarbeitung wirklich natürlicher Kommunikationsformen (natürliche Sprechweise, spontane Gestik, Gesichtsmimik und Körpersprache usw.) ist bislang nur ansatzweise gelungen, da alle »Recognition-based Interfaces« (vgl. Myers/Hudson/Pausch 2000, S. 18 ff.) auf unscharfen, interpretationsbedürftigen Eingaben beruhen, die sehr fein strukturiert, mit einander verwoben und höchst kontextabhängig sind:

»The difficulty, of course, is that we generally do not understand humans well enough to model their behavior accurately. This difficulty has forced most attempts at human-machine interface to adopt the simplest possible model of the human. [...] Such simple models, however, also make it impossible to build a system that takes real advantage of the human's abilities. The general approach we have developed is to instead model the human as a Markov device with a (possibly large) number of internal mental states, each with its own particular behavior, and interstate transition probabilities.« (Pentland 1995, S. 72)

All dies hat bislang auch die Integration der verschiedenen NUI-Techniken zu einem konsistenten multimodalen Gesamtsystem verhindert. Der erste Integrationsansatz, die VR, schuf mit ihrem Versuch einer radikalen Abkehr vom Desktop Computing neue *unnatürliche* Kommunikationsformen: Sie trennte den Benutzer von der Alltagswelt und versenkte ihn in virtuelle Räume mit schwer manipulierbaren schwerelosen Objekten. Die-

se Defizite an Natürlichkeit trieben innerhalb der VR-Community die Entwicklung von Force-Feedback-Techniken und vor allem des CAVE voran, der die Bewegungsfreiheit nicht mehr durch »bulky gloves and heavy electronic packs« einschränken sollte (Burdea 1996; DeFanti/Sandin/Cruz-Neira 1993). Doch der extrem hohe Preis und Raumbedarf verhinderten bislang eine breitere Einführung als »home- or business-based Caves«.

»Back to the Reality«: Die Alltagswelt als Interface

Die aufgrund ergonomischer Mängel misslungene Ablösung der GUI durch die VR ließ in der MCI-Community Anfang der 90er Jahre erneut eine Interfacelücke offenbar werden. Die Folge war, dass um 1990 gleichzeitig mehrere grundlegende Alternativen zur Virtualisierung der Welt und zum Desktop-Computing auf dem Plan erschienen. Die wie ein Schumpeterscher Schwarm auftretenden Ansätze für »next-generations computing environments« (Weiser 1993, S. 75) stellten sich im Juli 1993 erstmals gemeinsam unter der Parole »Back to the Reality« der breiteren Fachöffentlichkeit vor (Wellner/Mackay/Gold 1993 und unten Hornecker). Allen Gegenkonzepten zur VR war gemeinsam, dass sie den User nicht mehr in eine vollkommen künstliche Computer-generierte Umgebung versetzen, sondern ihn »in der Welt« belassen und die Arbeits- bzw. Alltagsobjekte lediglich mit Rechenkapazität anreichern wollen.

Zu den Alternativansätzen gehört zum einen die von Steven Feiner 1991/92 begründete »*Augmented Reality*« (AR), die reale Objekte mit wissensbasierten Grafikinformatoren überblendet. In ihr wird die Welt nicht mehr in ein feststehendes Computerdisplay geholt, sondern die reale Welt mithilfe halbtransparenter HMDs oder Shutterbrillen wird selber zum Interface (siehe unten Encarnação/Brunetti/Jähne). Als Erweiterung des AR-Ansatzes entwickelte sich seit Mitte der 90er Jahre die »*Mixed Reality*«, die auf verschiedene Weise fließende Übergänge zwischen Realität und Modellwelt herstellen will (siehe Rügge/Robben u.a. 1998; Brauer 1999). Besonders radikal gegen die VR tritt hierbei ab 1995 die Forschungsrichtung der »*Graspable*« bzw. ab 1997 der »*Tangible Interfaces*« (TUI) auf. Sie möchte anstelle einer Virtualisierung aller Objekte und Räume dem Benutzer durch gegenständliche Bedienschnittstellen und Modelle realer Gegenstände das rechnergestützte Arbeiten in der gewohnten haptisch erfahrbaren Welt ermöglichen (»*physical computing*«). Die Aufspaltung der Computerbenutzung in analoge Operationen und Denkweisen an der Bedienschnittstelle und algorithmische Formalisierung und Berechnungen in dem im Hintergrund aktiven Rechner gelten dabei als idealer Weg der

Überwindung der Bedienkomplexität und der Kooperationsdefizite der herkömmlichen MCI (siehe unten den Beitrag von Hornecker).

Auch bei diesem Ansatz wurde versucht, die »tangible« und »embodied interfaces« als logische Entwicklungsschritte einer Paradigmen-Evolution zu definieren, die auf »Keyboard UI«, »Graphical UI« und »Gestural UI« folgen und selber die Vorstufe eines endgültig in die Alltagsumgebung integrierten »invisible user interface« bilden (Fishkin/Moran/Harrison 1998). Doch entgegen dieser Zuversicht hat eine ganze Reihe technischer Probleme diese Richtung bisher noch nicht über eng begrenzte Nischenpositionen hinausgelangen lassen. So ermöglicht die unmittelbare Verbindung von Informationen mit physikalischen Alltagsobjekten oder gegenständlichen Symbolen (»physical icons« bzw. »phycons«) zwar vielfach eine intuitive Benutzung, doch die erweiterten Freiheitsgrade bei den Interaktionsmöglichkeiten sind dem User nicht immer transparent, so dass doch wieder ein Befehlssatz entwickelt und vom User gelernt werden muss. Ebenso steigt bei »TUIs« bzw. »Real« und »Mixed Realities« mit zunehmendem Grad der Annäherung an die reale Welt der Dedizierungsgrad, die Spezialisierung der Bedientechnik und damit der Realisierungsaufwand für die Interfaces, so dass man mit den Massenprodukten der traditionellen PC-GUI-Welt nur schwer konkurrieren kann.

Zu dem Philosophiewechsel am Anfang der 90er Jahre gehört auch das »Wearable Computing«, das sich zwar vieler von der VR geschaffener Interfaces bedient, jedoch auf die Immersion verzichtet. In ihm werden auch traditionelle Ein- /Ausgabemedien wie »pens«, »tablets« und Einhandtastaturen einbezogen und für den mobilen Einsatz angepasst. Zusätzlich entstand hier eine Fülle neuer funktions spezialisierter »wearable devices«, die an den Armen oder auf dem Kopf placiert, in Kleidungsstücke integriert oder als Accessoires getragen werden: »smart clothes«, »smart shoes« sowie Komponenten in Gürtel- und Schmuck-Form. Lange Zeit waren tragbare Computer mangels entsprechender Integrationsdichten nur Entwicklungsziel von Außenseitern wie Steve Mann, der mit einem am Körper getragenen Computer ein wirklich »persönliches«, die eigene Datenhoheit wahrendes Computing anstrebte (Mann 1997). Seit der Etablierung als MCI-Forschungsthema Anfang der 90er Jahre zielt WearComp darüber hinaus auf einen »intelligent assistant«, ein »visual memory prosthetic« und generell auf eine »natural extension of the user« (Mann 1998; Starner 2001). Doch die Erwartungen, mit WearComp ein ultimatives Interface zu schaffen, das den stationären PC ersetzt und endlich Lickliders Vision der »Man-Computer Symbiosis« verwirklicht, ist bisher nicht eingetroffen. Dazu war die Anwendungsentwicklung bisher zu technikzentriert, die Ergonomie für den Alltagsgebrauch oft zu unzureichend, die Interface-Landschaft zu heterogen und proprietär. Chancen haben da-

her eher an Bedürfnissen mobiler Tätigkeiten orientierte pragmatische Konfigurationen von Hard- und Softwarekomponenten (siehe dazu Rügge 2007 und ihren Beitrag unten).

Alle diese mit natürlichen Analogien operierenden neuen MCI-Integrationskonzepte vermochten bisher die Komplexität der Computerbedienung nicht radikal zu vereinfachen. Die Ausuferung der Funktionalität und die starke Kontext- und Situationsabhängigkeit der Anwendungen erfordert auch bei den vermeintlich natürlichen Interfacetechniken ausdifferenzierte Bediensemantiken, die als neue Programmiersprache gelernt werden müssen. Der propagierte schnelle Generationswechsel von den GUI- zu den NUI-Interfaces blieb daher im ersten Anlauf stecken. Und so hörten trotz aller Bemühungen um natürlichere Bedientechniken die Klagen der Community über Asymmetrien in der Mensch-Computer-Kommunikation nicht auf:

»It's an interesting paradox, computers and applications grow increasingly sophisticated but they still demand to be ›spoken‹ to in ways only they can understand: a mouse, keyboard or touchpad. Why shouldn't we reverse that. Why shouldn't computers understand our distinctly human ways of communicating? We speak, we gesture, we write: Why can't they get it.« (Accenture Technology Labs 2003)

Die NUIs der 80er und 90er Jahre zielten auf eine Ablösung der Desktop-Metapher und WIMP-Welt, sie blieben aber zunächst noch weitgehend innerhalb des Paradigmas der Direkten Manipulation. Dadurch unterlagen auch *sie* dem Problem, daß bei immer komplexerer Anwendungssoftware die Schwierigkeiten des vom Benutzer geleiteten kleinschrittigen Vorgehens ständig zunehmen. Dieses Dilemma setzte bald auch Bemühungen in Gang, über die Suche nach natürlicheren Kommunikationskanälen hinaus auch die Arbeitsteilung zwischen User und System durch einen Übergang von *interaktiven* zu *proaktiven* Interfaces grundlegend zu ändern. Die neuen Leitbilder hießen nun »Intelligent«, »Perceptive« und »Intuitive Interfaces« sowie »Proactive« und »Invisible Computing«. Damit deutet sich eine neue Entwicklungsspirale in der MCI-Geschichte an. In der ersten Phase des Computing waren die Probleme des interaktiven manuellen Computing durch die Automatisierung der Rechenprozesse gelöst worden. Die Insuffizienz des Automatic Computing führte zur Wiedereinbeziehung des Benutzers in einen interaktiven Prozeß. Die zunehmende Funktionalität und Komplexität der Interaktion wurde durch die Verlagerung in die Software-Interfaces zunächst aufgefangen, längerfristig aber verschärft. Jetzt soll ein Netz lernender Automaten im Hintergrund den User wieder von der lästigen Interaktionsarbeit befreien.

9 Proactive Computing als Lösungsansatz für die Interface-Krise?

Die Suche nach intelligenteren Bedientechniken wurde in den 90er Jahren neben den »Natural Interfaces« ein Hauptschwerpunkt der MCI-Forschung. Doch bereits Mitte der 80er Jahre, als die Diffusion der Benutzer-gesteuerten »Direkten Manipulation« in den Massenmarkt gerade begonnen hatte, tauchten erste Ideen auf, die Computerbedienung durch das aus der KI stammende Konzept von Software-Agenten radikal zu vereinfachen. Alan Kay schlug bereits 1984 (S. 58 f.) vor, die von ihm selber maßgeblich entwickelte schrittweise ikonische Arbeitsweise der direkten Objekt-Manipulation bei komplexen Eingaben aufzugeben und durch intelligente Assistenten bzw. Agenten zu ersetzen: »The question of software's limitations is brought front and center, however, by my contention that in the future a stronger kind of indirect leverage will be provided by personal agents: extensions of the user's will and purposes, shaped from and embedded in the stuff of the computer«. Kay bezog sich bei seinen »soft robots« ausdrücklich auf John McCarthy, der bereits in den 50er Jahren begrenzte Aufgaben an kleine autonom arbeitende Programme delegieren wollte, für die Oliver G. Selfridge 1958/59 den Agentenbegriff einführte.¹⁷ Die Psychologin Susan Brennan prägte ebenfalls 1984 für den neuen MCI-orientierten Agententyp den Begriff »interface agent« (Laurel 1990, S. 359) In der zweiten Hälfte der 80er Jahre begannen dann systematische Entwicklungsanstrengungen für spezialisierte Agenten zur Informationsbeschaffung und -filterung, vor allem bei der E-Mail-Verwaltung, sowie zur Kooperationsunterstützung (Groupware). In den 90er Jahren verschob sich der Entwicklungsschwerpunkt auf Informationsagenten für das Internet und Transaktionsagenten für E-Commerce, wobei zunehmend im Hintergrund arbeitende »autonome Agenten« bzw. Multiagentensysteme ein größeres Gewicht erhielten.

Ende der 80er Jahre erhielt das Agenten-basierte Computing dann den Rang eines generellen Lösungsansatzes für die »conversational interaction«, ja als Next-Generation-Technologie der Software überhaupt. Für Kay (1990, S. 206) war der Übergang vom interaktiven Manipulationswerkzeug zu einem vom Computer gesteuerten Prozess »the next big direction in user interface design«, ein größerer Schritt als die Entwicklung vom Teletype zum Macintosh-Interface: »The creation of autonomous processes that can be successfully communicated with and managed is a

17 Auch Licklider und Taylor hatten schon 1968 (S. 38 ff.) erwogen, bei der Abwicklung der E-Mail und bei Online-Transaktionen Intermediäre einzusetzen, die sie zu Ehren von Selfridge *OLIVER* (Online Interactive Expediter and Responder) nannten (siehe hierzu Pflüger 2004, S. 394).

qualitative jump from the tool – yet one that must be made.« Im gleichen Jahr verkündete Michael C. Dertouzos (1990, S. 1) vom MIT bei der CHI'90 eine generelle Abkehr der MCI-Entwicklung von der GUI-Welt zugunsten von »smarter interfaces«, die partiell die Intentionen der Benutzer erkennen »instead of brutally forcing us through a litany of repetitive and dumb rituals [...]. But to do this, we need to redefine the profession, from its current narrow confines of dealing with windows, mice and ergonomic factors to a far broader discipline that encompasses and seeks to improve everything that humans and computers do together, including purpose - in other words the total user interface of and for tomorrow.«

Das »totale Interface« bzw. das »intelligente Interface« soll durch die Ausstattung des Computers mit Programm-Intelligenz endgültig die bestehende, ja inzwischen noch verschärfte Disparität zwischen Mensch und Computer aufheben und am Ende eine »Konversation« auf annähernd gleichem Intelligenzniveau ermöglichen. Während beim »concept of dialogue« der Computer immer erst auf die äußerst schmalbandigen Eingaben des Users warten muss und die Ergebnisse visuell rückkoppelt, erfassen die aktiven intelligenten Interfaces nun permanent alle sprachlichen Äußerungen, Gestik, Mimik, Emotionen, Körperbewegungen sowie Zustände des Körpers und der Umgebung, interpretieren sie und reagieren proaktiv. Der Benutzer ist hierbei nicht mehr an den PC und seinen Bildschirm gekettet, denn nun dienen der menschliche Körper selber und »real-world objects« als »the total UI«: »Good-bye keyboard, so long keyboard. Hello smart rooms and clothes that recognize acquaintances, understand speech, and communicate by gesture. And that's just the beginning.« (Pentland 2000, S. 35) Seit 1991-93 setzten sich für dieses Technologiebündel dann die Begriffe »Noncommand User Interfaces«, »Perceptual Interfaces«, und noch mehr der Sammelbegriff »Intelligent User Interfaces« durch (Sullivan/Tyler 1991; Nielsen 1993, S. 98 f.; Roth/Malin/Schreckenghost 1997; Maybury/Wahlster 1998).

Bei den Überlegungen für intelligente Intermedien und Assistenzsysteme deuteten sich schon Ende der 80er Jahre zwei grundlegende Entwicklungsrichtungen an. Auf der einen Seite standen menschenähnliche Interface-Agenten, die an alltägliche Anthropomorphismen anknüpfen und ein partnerschaftliches Verhältnis suggerieren (Laurel 1990). Der Computer erscheint hier als ein »old friend«, der den Benutzer immer besser kennt und selber zum Träger sozialer Rollen wird. So verallgemeinerte Nicholas Negroponte vom MIT-MediaLab 1989 die bereits 1985 von Dannenberg und Hibbard entwickelte Idee eines Agenten-basierten »Butler Process« für die Verwaltung von Netzressourcen zu einem Butler-Agenten als personalisierter Bedienschnittstelle, an die die Benutzer ganze Aufgabenbündel delegieren: »The best metaphor I can conceive of for a human-

computer interface is that of a well-trained English butler.« Doch autonom agierende Agenten sollen auch als Assistenten, SekretärInnen, Tutoren oder Makler fungieren, die jeweils mit ›ihrem‹ User kooperieren und kommunizieren und für ihn aktiv werden. (Negroponte 1989; 1991, S. 190; 1997, S. 59) Aus derartigen Überlegungen entwickelte sich seit den 90er Jahren eine breite Skala anthropomorpher Interface-Agenten, denen aufgrund ihrer laiengerechten, intuitiven Bedienbarkeit, emotionalen Ansprache und Vertrauen schaffenden Wirkung in Zukunft ein großes Feld persönlicher Dienstleistungen zugewiesen wird. Die Spannweite reicht dabei von vage angedeuteter Persönlichkeit bis zu detailgetreuer Menschenähnlichkeit in Aussehen und Kommunikationsformen. Dieses personifizierende Agentenkonzept geht einher mit einer generellen Häufung anthropomorpher Metaphern, humanoider Interface-Konzepte und erneuten Annahmen einer schnellen Erreichbarkeit natürlichsprachiger Kommunikation mit Computern.

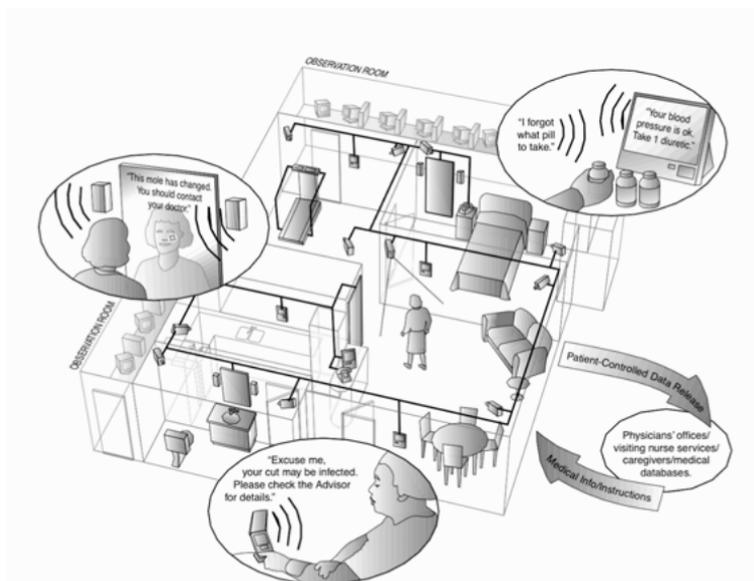
Die andere Richtung zielt nicht auf eine *Vermenschlichung* der Computer-Beziehung, sondern auf eine *Verdinglichung*. Da menschenähnliche »intelligent agents« und Spracheingabe als Interface noch immer eine zu hohe Aufmerksamkeit des Benutzers erforderten, soll die Bedienkomplexität hier über eine »world full of invisible widgets« beseitigt werden: »A good tool is an invisible tool. By invisible, I mean that the tool does not intrude on your consciousness; you focus on the task, not on the tool.« (Weiser 1994) Geräte und Interfaces werden deshalb in der alltäglichen Dingwelt verborgen, sie verschwinden in »all knowing rooms« und in einer interagierenden »society of objects«, die eine »heimliche Intelligenz« im Hintergrund entwickeln und aus Handlungserkennung selber Aktivitäten ableiten: »Computers will go underground. My refrigerator will know when it is out of milk, but take appropriate action only after a conversation with my calendar or travel planning agent.«¹⁸

Um den Benutzer von den Mühen des interaktiven Computing mit diesen unzähligen »smart objects« zu entlasten und die physikalische Welt mit der virtuellen Datenwelt zu verbinden, bedarf es jedoch nicht nur autonom agierender Software-Agenten, sondern auch einer intelligenten Sensorik. Diese soll permanent Zustände registrieren, Befindlichkeiten wahrnehmen und den Computern melden, Personen, Mobilgeräte und Dinge identifizieren und lokalisieren, und schließlich die erhobenen Da-

18 Negroponte (1991, S. 184) führt fast dieselben »smart appliances« auf, die der Architekt und Futurist Roy Mason bereits Anfang der 80er Jahre in einer Vision von »computer-controlled homes« entwarf und in dem intelligenten Musterhaus »Xanadu« prototypisch entwickelte, und die trotz ihrer technikzentrierten Alltagsferne noch immer leitmotivisch durch die UbiComp-Szenarien geistern (Halfhill 1982).

ten mit den gespeicherten Präferenz- und Kontext-Informationen in Beziehung setzen, um daraus sinnvolle Aktionen abzuleiten. Als neue Interface-Gattung entstehen so seit den 90er Jahren intelligente Sensor- und Aktornetze, drahtlose automatische Identifikationssysteme (RFIDs) und neue Tracking- und Lokalisierungs-Technologien, die zur technischen Basis des »Internets der Dinge« und von »Location-based Services« werden (siehe dazu unten den Beitrag von Encarnação/Brunetti/Jähne sowie Fleisch/Mattern 2005).

Abb 8.: Aware Home-Scenario des Smart Medical Home Research Laboratory (http://www.futurehealth.rochester.edu/smart_homel/smh.pdf): ständige Überwachung der Körperfunktionen, Handlungen und Bewegungen mit automatischer Ermahnung zu Medikamenten-Einnahme oder Arztbesuch.



Wie bei den Natural Interfaces konkurrieren auch bei den proaktiven Hard- und Software-Techniken eine Reihe von Integrationskonzepten und übergeordneten Leitbildkomplexen. Die ersten entstanden um 1990 überwiegend im akademischen Bereich, so »Ubiquitous Computing« und »Calm« bzw. »Invisible Computing«. Ihnen folgten seit Ende der 90er Jahre die besonders von Herstellerverbänden propagierten Konzepte »Pervasive Computing«, »Ambient Intelligence« und »Pro-active Computing« (ausführlich dazu unten Friedewald und Encarnação/Brunetti/Jähne). Wohl als erster formulierte Abraham Peled von IBM 1987 das Leitbild einer auf vernetzten PCs und vor allem »embedded computers« beruhen-

den *Ubiquitous Intellectual Utility*, die über »visual and other natural interfaces« Computer-Intelligenz allgegenwärtig machen sollte. Am Xerox PARC hatte sich ab 1988 *Ubiquitous Computing* nach Anfängen als Kooperations-unterstützendes Arrangement von Walldisplay, Notebooks und Handhelds (Weiser/Gold/Brown 1999) zunächst als natürlichere Alternative zur VR und zum Desktop-Computing placiert.¹⁹ Man setzte sich zugleich von den nomadischen bzw. monadischen WearComp-Visionen ab, indem man das Computing nicht an den Körper bringen, sondern durch Einbettung von Programm-Intelligenz in Alltagsgegenstände in die Prozesse der Alltagswelt integrieren wollte. An die Stelle von »virtual environments« treten hier »ubiquitous smart spaces«, »smart rooms«, »smart buildings« und »aware homes«, in denen die Dinge »sehen«, »denken« und fürsorgend für die verschiedenen User/Bewohner agieren.

Am Ende führt die Vernetzung aller Personen, intelligenten Gegenstände und Prozesse zu einem integrierten programmgesteuerten Datenraum, in dem die Computer selber nicht mehr sichtbar sind, denn »The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.« (Weiser 1991, S. 94) Die Computer verlieren damit ihren Werkzeug- und Mediencharakter, sie werden in der Peripherie der Alltagsdinge verborgen und treten nur bei »Bedarf« in Erscheinung: »The computer is really an infrastructure, even though today we treat it as the end object. Infrastructures should be invisible [...] a user-centered, human-centered humane technology where today's personal computer has disappeared into invisibility.« (Norman 1998, S. 6; Weiser/Brown 1996) Interfaces und Intermedien werden in diesem »Calm«, »Hidden« bzw. »Invisible Computing« kaum noch benötigt, die Aufgaben der MCI verlagern sich in das Design der Auswertungssysteme der intelligenten Sensorik und vor allem der hochkomplexen Netze von Software-Agenten. Nach dem Chef der Intel-Forschung David Tennenhouse (2000, S. 43, 48) erfordert der anstehende Paradigmenwechsel von der bisherigen *interaktiven* zur künftigen *proaktiven* »Man-Computer Symbiosis« deshalb eine radikale Umwälzung der gesamten Forschungs-Agenda der MCI: »Given a few billion human users, each of whom is able to generate a sizable agent constituency, we should anticipate interaction spaces involving many trillions of agents. Since these agents will interact with each other as they go about our business, we need to invent technologies that sustain human control over agent-based systems, yet allow agents to autonomously negotiate with each

19 »Perhaps most diametrically opposed to our vision is the notion of »virtual reality«, which attempts to make a world inside the computer. [...] Virtual Reality focuses an enormous apparatus on simulating the world rather than on invisibly enhancing the world that already exists.« (Weiser 1991, S. 95).

other in ways that honor overall systems objectives and constraints. [...]« Es stellt sich so, wie Tennenhouse einräumt, das große Problem, wie die Menschen mit Interfaces interagieren, die über bedeutend schnellere Antwortzeiten verfügen und wie angesichts von Tausenden von Computern und Millionen von »Knowbots« pro Person das »human-supervised computing« überhaupt realisiert werden könne.

Ungeachtet der exponentiell ansteigende Komplexität und der unbewiesenen Alltagsgauglichkeit erscheint für Tennenhouse der Übergang vom »Human-in-the loop computing« zum »Human-supervised computing« aber als logisch-zwingende Entwicklungsstufe, die sich unmittelbar aus der Moore's-Law-getriebenen Vermehrung der »embedded computers and microprocessors« ergibt. Auch bei Ubiquitous Computing wurde die breite Durchsetzung der Technik vorwiegend aus Entwicklungsmodellen abgeleitet. So definierte Peled (1987, S. 36) ganz in der Tradition des Bell's Law den »embedded computer« nach der Ära des Mainframe, des Minicomputers und des Personal Computers als »next computer revolution«, die zur »ubiquitous intellectual utility« führen werde. Nach Weisers späterem, sehr ähnlichem Modell folgt auf die »mainframe era«, in der sich viele Nutzer einen Computer teilen mussten, und der »personal computing era«, in der der User über einen Computer verfügte, als dritte Welle die »UC Era«, in der jedem Benutzer Hunderte, ja Tausende Computer und Prozessoren zur Verfügung stehen (Weiser 1998). Schon aus diesem Mengenwachstum der dem Menschen dienenden Computer ergibt sich für Weiser ein Zwang, die Interfaces in »augmented artifacts« verschwinden zu lassen.

Der Rekurs auf die Entwicklungslogik überspielt die massiven Realisierungs- und Nutzungsprobleme des proaktiven ubiquitären Computing, so die uneinheitlichen Schnittstellen zu Alltagsgegenständen, die Probleme des Zusammenspiels multimodaler Kontrollinformationen und der Programmierung heterogener und kontextsensibler Anwendungsprozesse. Durch den weitgehenden Verzicht auf interaktive Eingaben wird nämlich die gewachsene Komplexität in intelligente Hintergrundprozesse verlagert, denn diese müssen nun die Vielfalt wechselnder situativer Anforderungen und die disparaten Nutzungskontexte vorab berücksichtigen, um den Usern die richtigen Informationen zur richtigen Zeit zu liefern. Dem 1994 von Bill L. Schilit am Xerox PARC geschaffenen MCI-Forschungsgebiet der »Context Awareness« ist aber bisher nur eine relativ grobe Berücksichtigung von wechselnden örtlichen, zeitlichen und personenspezifischen Kontextinformationen gelungen, denn besonders in Alltagssituationen geraten proaktive Anwendungsprogramme in eine kaum zu beherrschende Komplexitätsfalle: »The sophistication of commonsense reasoning and context awareness that is required is daunting, given the current state

of our understanding of these fields. [...] No matter how hard the system designer tries to program contingency plans for all possible contexts, invariably the system will sometimes frustrate the home occupant and perform in unexpected and undesirable ways. A learning algorithm would also have difficulty because a training set will not contain examples of appropriate decisions for all possible contextual situations.« (Schilit/Adams/Want 1994; zur »Context Awareness« ausführlich unten Pflüger).

Ebenso werden mögliche Gefahren ausgeblendet wie das Entstehen unkontrollierbarer Wechselwirkungen zwischen Nutzer- und Umgebungssystem, Konflikte zwischen dem spontan agierenden Benutzer und seinem im System gespeicherten Benutzermodell bzw. Verhaltensprofil sowie ein verstärkter kommerzieller Zugriff auf Alltagsabläufe der Privathaushalte. Hinzu kommen gravierende Datenschutzprobleme bei der durchgängig erforderlichen Preisgabe von Nutzerpräferenzen und Kontextinformationen an die allgegenwärtigen »invisible computers«. Dabei geht es nicht mehr nur um Bewegungsdaten, sondern auch um die dauerhafte Registrierung von Nutzer-Emotionen und -Befindlichkeiten. Hinter dem unter dem Siegel der erhöhten Benutzungsfreundlichkeit propagierten MCI-Wandel von »computer-interfaces for people« zu »people-interfaces for computers« verbirgt sich somit ein radikaler Wandel der informationellen und politischen Kultur (siehe dazu unten Friedewald).

Die gewaltigen Probleme der Informatik bei der Bewältigung des intendierten Paradigmenwechsels lassen die Frage aufkommen, ob *Vermenschlichung* und *Verbergung* der Intermedien und Interfaces wirklich die geeigneten Wege zur Lösung des derzeitigen Komplexitätsproblems der MCI sind oder ob hier nicht unerfüllbare Versprechungen gemacht werden. Aus technikhistorischer Perspektive entsteht sogar der Eindruck, dass es sich bei »Humanoiden Interfaces« und beim »Invisible Computing« um die typischen Vereinfachungen der Bedienproblematik handelt, die auch frühere Einstiegsphasen in neue MCI-Paradigmen kennzeichnen. So glaubte man am Beginn der Entwicklung der Informationstechnik maschinelles Sprechen, Schreiben und Musikhören einfach durch mechanischen Nachbau des Sprechenden, Schreibenden und Klavierspielenden Menschen technisch realisieren zu können, bevor man einsehen musste, dass das Problem nur durch spezielle Mechaniken und nicht-anthropomorphe Medienkonstrukte zu lösen war. Auch am Anfang des Automatic und des Interactive Computing behinderten Giant-Brain-Visionen bzw. KI-Phantasien einer natürlichen Konversation mit dem Rechner zeitweise eine realistische Entwicklung von Intermedien und Interfaces für die Interaktion mit dem Computer.

Neben der anthropomorphen Vereinfachung scheinen sich in den Visionen einer umfassenden »Informatisierung des Alltags« (Mattern,

2005) und ubiquitärer Computerintelligenz frühere Illusionen einer vollständigen Berechenbarkeit und raschen Automatisierung von Alltagsabläufen zu wiederholen, die seit den 60er Jahren wellenartig immer wieder auftauchen. So sehr auch überzogene Erwartungen die Informatikentwicklung angespornt haben, so wenig haben sie tatsächlich für die Überwindung von MCI-Krisen beigetragen. Hier waren es die medienkombinatorischen Inventionen und Innovationen der Pioniere des Graphikorientierten Computing, die die Computerbedienung vereinfachten und so für breitere Schichten immer weitere Computeranwendungen erschlossen. Methoden der Medienkombinatorik, d.h. eine analytische und empirische Exploration neuartiger Medienkonstrukte und Medienanwendungen aus bestehenden Medienkomponenten und Interfaces unter Einbeziehung fortschrittlicher technischer Wirkprinzipien – werden m. E. in Zukunft dringend gebraucht. Ebenso Methoden eines alltagsnahen Szenario-Writing, das konsequent von den Anwendungen und nicht von der Technik her denkt. Denn entgegen den Auffassungen der Vertreter des *Calm, Disappearing and Invisible Computing* erzeugt nämlich die fortschreitende Diffusion des Computing in Alltagsgegenstände und -prozesse sowie die vielfältigen mobilen Informationssysteme und Unterhaltungsmedien gerade einen großen Bedarf an neuen alltagstauglichen Medien- und Interface-Formen.

Legt man die Erfahrungen des erfolgreich bewältigten MCI-Paradigmenwechsels in den 60er Jahren zugrunde, so bedarf es in der Gegenwart erneut

- einer Absage an überzogene KI- und NUI-Visionen,
- einer realistischen Bescheidung auf machbare Entwicklungskorridore,
- der Entwicklung neuartiger Intermedien, die für den User sichtbar und kontrollierbar bleiben,
- einer Bündelung der universitären Forschung mit einer massiven öffentlichen und privaten Förderung,
- der Schaffung einer explorativen Medienkombinatorik, die die historischen Erfahrungen der Medien-, Intermedien- und Interfacegeschichte aufarbeitet und systematisiert,
- eines langen Atems, wie er der langfristigen Entwicklungsdynamik der MCI gemäß ist,
- eines Verzichtes auf Entwicklungsmodelle, die eine zwanghafte technische Logik unterstellen und dadurch argumentativ Wahlmöglichkeiten blockieren und schließlich
- einer Absage an alle Final Interface-Illusionen entsprechend der historischen Einsicht von Biocca, Kim und Levy (1995, S. 13):

»The ultimate display will never arrive. It is a moving target.«

Literatur

- Aarts, E./Hartwig, R./Schuurmans, M. (2002): »ambiente intelligence«. In: Denning, P. J. (Hrsg.) (2002): »The Invisible Future«. New York, Chicago, San Francisco, S. 235-250.
- Accenture Technology Labs (2003): »Communicating with the Online World in Human Terms«. Juni 2003. (URL: <http://www.accenture.com/xd/xd.asp>).
- Aford, R. (1990): »The Mouse that Roared«. In: BYTE 15, 11, S. 395-401.
- Aiken, H. H. (1956): »The Future of Automatic Computing Machinery«. In: Hoffmann, W.; Walther, A. (Hrsg.), »Elektronische Rechenmaschinen und Informationsverarbeitung« (Nachrichtentechnische Fachberichte, Bd. 4), Braunschweig, S. 31-35.
- Alberts, G. (2005): »Das Verschwinden der Konsole und die Vorläufer des interaktiven *User*«. In: A. B. Cremers, R. Manthey, P. Martini, V. Steinhage (Hrsg.), »Informatik 2005«, 2 Bde. Bonn, Bd. 1, S. 205-209.
- Alexander, S. N. (1948): »Input and Output Devices for Electronic Digital Calculating Machinery«. In: Proceedings of a Symposium on Large Scale Calculating Machinery (1947), Cambridge, MA; wiedergedr. in: Charles Babbage Institute (Hrsg.), Reprint Series for the History of Computing Bd. 7, London, Los Angeles, San Francisco 1985, S. 248-253.
- Alexander, W.F. (1961): »Tracking Control Apparatus«, US Patent 3013441, 19.12.1961.
- A Perspective on SAGE (1983): »Discussion«. In: Annals of the History of Computing, 5, 4, S.375-380.
- Astheimer, P./Böhm, K. u.a. (1994): »Die Virtuelle Umgebung - Eine neue Epoche in der Mensch-Maschine-Kommunikation. Teil I: Einordnung, Begriffe und Geräte«. In: Informatik-Spektrum 17, S. 281-290.
- Babbage, Ch. (1832): »On the Economy of Machinery and Manufacture« (1832), zit. nach 4. Aufl. (1835). In: Campbell-Kelley, Martin (Hrsg.), »Works of Babbage«, Bd. 5, New York 1989.
- Babbage, Ch. (1837): »On the Mathematical Powers of the Calculating Engine«. In: Ders., »Works of Babbage«, hrsg. von M. Campbell-Kelley, 11 Bde. New York 1989, Bd. 3, S. 15-61.
- Babbage, Ch. (1864): »Passages from the Life of a Philosopher«. In: Campbell-Kelley, M. (Hrsg.), »Works of Babbage«, Bd. 11, New York 1989.
- Baecker, R./Grudin, J./Buxton, W. A. S./Greenberg, S. (Hrsg.) (1995), »Readings in Human-Computer Interaction: Toward the Year 2000«. 2. Aufl. San Francisco.
- Balzert, H. (1988): »E/A-Geräte für die Mensch-Computer-Interaktion«. In: Balzert, H./Hoppe, H. U. u.a. (Hrsg.), »Einführung in die Software-

- Ergonomie. Mensch Computer Kommunikation. Grundwissen«, Bd. 1, Berlin, New York, S. 67-98.
- Barnes, S. B. (2007): »Alan Kay: Transforming the Computer into a Communication Medium«. In: *Annals of the History of Computing* 29, 2, S. 18-30.
- Basalla, G. (1988): »The Evolution of Technology«. Cambridge, New York.
- Beauchair, W. de (2005): »Rechnen mit Maschinen. Eine Bildgeschichte der Rechentechnik«, Braunschweig (1968) 2. Aufl. Berlin, Heidelberg, New York.
- Bell, C. G./Chen, R./Rege, S. (1972): »The Effect of Technology on Near Term Computer Structures«. In: *IEEE Computer* 5, 2, S. 29-38.
- Bennett, J. L. (1985): »The Concept of Architecture Applied to User Interfaces in Interactive Computer Systems«. In: Shackel, B. (Hrsg.): *Human-Computer Interaction - INTERACT '84*, Amsterdam 1985, S. 865-870.
- Bernstein, M. I. (1974): »Toward Natural Man-Machine Dialogue«. In: Rosenthal, P. H.; Mish, R. K.: »Multi-Access Computing. Modern Research and Requirements«. Rochelle Park, N. J., S. 178-183.
- Biocca, F./Levy, M. R. (Hrsg.) (1995): »Communication in the Age of Virtual Reality«. Hillsdale, NJ.
- Blackwell, A. F. (2006): »The Reification of Metaphor as a Design Tool«. In: *ACM Transactions on Computer-Human Interaction*, 13, 4, S. 490-530.
- Blundell, B. G./Schwarz, A. J. (2005): »Creative 3D Display and Interaction Interfaces. A Transdisciplinary Approach«. Hoboken, NJ.
- Bobrow, D. G. (1967): »Problems in Natural Language Communication with Computers«. In: *IRE Transactions on Human Factors in Electronics*, HFE-8, Mrz, S. 52-55.
- Bolt, R. A. (1979): »Spatial Data-Management«. Architecture Machine Group, MIT, Cambridge, MA.; http://www.media.mit.edu/speech/sig_papers1.html (zuletzt gesehen: 10.5.2007).
- Bolt, R. A. (1980): »Put-that-there: Voice and gesture at the graphics interface«. *ACM SIGGRAPH Computer Graphics*, Proceedings of the 7th annual conference on Computer graphics and interactive techniques, 14, 3; http://www.media.mit.edu/speech/sig_papers1.html (zuletzt gesehen: 10.5.2007).
- Bolt, R. A. (1981): »Eyes at the Interface«. In: *Proceedings of the ACM Human Factors in Computer Systems Conference*, S. 360-362.
- Booth, A. D. (1956): »Input-Output for Digital Computing Machines«. In: Hoffmann, W.; Walther, A. (Hrsg.), »Elektronische Rechenmaschinen und Informationsverarbeitung« (Nachrichtentechnische Fachberichte, Bd. 4), Braunschweig, S. 15-20.
- Booth, A. D. (1960): »The Future of Automatic Digital Computers«. In: *Communications of the ACM* 3, 6, S. 339-341, 360.

- Bos, W. E. A. (1993): »Easier said or done? Studies in multimodal human-computer interaction«. Diss. Leiden.
- Bourne, Ch. P./ Engelbart, D. C. (1958): »Facets of the Technical Information Problem«. In: *Datamation*, 4, 9/10.
- Bowman; D. A./Kruijff, E. u.a. (2005): »3D User Interfaces. Theory and Practice«. Boston, San Francisco, New York.
- Brand, St. (1990): »MediaLab. Computer, Kommunikation und neue Medien«. Reinbek.
- Brauer, V. (1999): »Gegenständliche Benutzungsschnittstellen für die Mensch-Computer-Interaktion«. Diss. Bremen.
- Brooks Jr., F. P. (1965): »The Future of Computer Architecture«. In: *Information Processing 1965, Proceedings of IFIP Congress '65*, 2 Bde. Washington, D. C., London 1965, Bd. 2, S. 87-91.
- Brooks Jr., F. P. (1977): »The Computer ›Scientist‹ as Toolsmith - Studies in Interactive Computer Graphics«. In: *Information Processing 1977, Proceedings of IFIP Congress '77*, Amsterdam, London, S. 625-634.
- Burdea, G. C. (1996): »Force and Touch Feedback for Virtual Reality«. New York, Chichester, Brisbane.
- Burks, A. W./Goldstine, H. H./Neumann, J. von, (1946): »Preliminary Discussion of the Logical Design of an Electronic Computing Instrument«. Part I, Vol. 1, Institute for Advanced Study Princeton, N. J. Juni 1946. In: Taub, A. H. (Hrsg.), »Complete Works of John von Neumann«, 6 Bde. Oxford (UK), New York 1961-63, Bd. 5, S. 34-79.
- Buxton, W. A. S. (1983): »Lexical and pragmatic considerations of input structures«. In: *Computer Graphics* 17, 1, S. 31-37.
- Buxton, W. A. S. (1990): »The ›Natural‹ Language of Interaction: A Perspective on Nonverbal Dialogues«. In: Laurel, Brenda (Hrsg.): »The Art of Human-Computer Interface Design«. Reading, MA, Menlo Park, CA, New York, S. 405-416.
- Canny, J. (2006): »The Future of Human-Computer Interaction«. In: *ACM Queue*, Juli/August.
- Card, St. K./Mackinlay, J. D./Robertson, G. G. (1990): »The Design Space of Input Devices«. In: *CHI '90 Conference Proceedings*. Seattle, Washington April 1-5, 1990, New York, S. 117-124.
- Carr III, J. W. (1962): »Better Computers«. In: *Elektronische Rechenanlagen* 4, 4, S. 157-160.
- Carroll, J. M. (1983): »Presentation and Form in User-Interface Architecture«. In: *BYTE*, Dez. 1983, S. 113-122.
- Carroll, J. M./Mack, R. L./Kellogg, W. A. (1991): »Interface Metaphors and User Interface Design«. In: Helander, M. (Hrsg.), »Handbook of human-computer interaction«. Amsterdam, New York, S. 67-82.

- Caudill, M. (1992): »Kinder, Gentler Computing«. In: BYTE 17,4, S. 135-150.
- Chapin, N. (1962): »Einführung in die elektronische Datenverarbeitung«. Wien, München (Übersetzung von: An »Introduction to Automatic Computers. A Systems Approach for Business«. New York 1957.
- Cruz-Neira, C./Sandin, D. J./DeFanti, Th. A. (1993): »Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE«. In: Computer Graphics, Proceedings of SIGGRAPH '93, ACM SIGGRAPH, August, S. 135-142.
- Dannenberg, R. B./Hibbard, P. G. (1985): »A Butler process for resource Sharing on Spice Machines«. In: ACM Transactions on Office Information Systems 3, 3, S. 234-252.
- Darrell, T./Pentland, A. P. (1995): »Attention-driven Expression and Gesture Analysis in an Interactive Environment«. In: Bichsel, M. (Hrsg.): Proceedings International Workshop on Automatic Face- and Gesture-Recognition, Zürich, S. 135-140.
- Davies, D. W. (1954): »Input und Output«. In: International Symposium on Automatic Digital Computation (National Physical Laboratory, 1953), London; wiedergedr. in: »The Early British Computer Conferences«, edited and introduction by M. R. Williams and M. Campbell-Kelly, Charles Babbage Institute (Hrsg), Reprint Series for the History of Computing, Bd. 14, London, Los Angeles, San Francisco 1989, S. 310-324.
- Davis, M. R./Ellis, Th. O. (1964): »The RAND-Tablet: A Man-Machine Graphical Communication Device«. In: American Federation of Information Processing Societies, (AFIPS), Conference, Proceedings Bd. 26, FJCC, S.325-331.
- Davis, R. M. (1965): »A History of Automated Displays«. In: Datamation, 9, 1, S. 24-28.
- DeFanti, Th. A./Sandin, D. J./Cruz-Neira, C. (1993): »Room with a View«. In: IEEE Spectrum, 30, 10. S. 30-33.
- Denning, P. J. (Hrsg.) (2002): »The Invisible Future. The seamless integration of technology into everyday life«. New York, Chicago u.a.
- Dertouzos, M. L. (1990): »Redefining Tomorrow's User Interface«. In: CHI '90 Proceedings, New York April, S. 1.
- Dertouzos, M. L. (1999): »What Will Be. Die Zukunft des Informationszeitalters«. Wien New York.
- Dix, A./Finlay, J./Abowd, G./Beale, R. (1995): »Mensch-Maschine-Methodik«. New York, London, Toronto.
- Douglas, S. A./Mithal, A. K. (1997): »The Ergonomics of Computer Pointing Devices«. London.

- Ebrahimi, T./Vesin, J.-M./Garcia, G. (2003): »Human-Machine Interface in Multimedia Communication Communication – A Definition«. In: IEEE Signal Processing Magazine, Jan. 2003, S. 14-24.
- Ellis, St. R. (1995): »Origins and Elements of Virtual Environments«. In: Barfield, W./Furness, T. A.: »Virtual Environments and Advanced Interface Design«. New York, S. 14-57.
- Engelbart, D. C. (1960): »Augmented Human Intellect Study«. Proposal for Research to Air Force Office of Scientific Research, SRI No. ESU 60-251, 13 December 1960 (Internet-Version, gesehen 22.3.2000: http://sloan.stanford.edu/mousesite/EngelbartPapers/B6_F2_Augm Prop 1.html).
- Engelbart, D. C. (1961): »Special Considerations of the Individual as a User, Generator, and Retriever of Information«. In: American Documentation, April, S. 121-125.
- Engelbart, D. C. (1962): »Augmenting Human Intellect: A Conceptual Framework«. AFOSR-3233 Summary Report, Oct. 1962 (Internet-Version, gesehen zuletzt Januar 2008: <http://www.bootstrap.org/augdocs/friedewald030402/augmentinghumanintellect/ahi62index.html>).
- Engelbart, D. C. (1963): Brief an Robert Taylor, 5. 4. 1963, Engelbart Papers, Department of Special Collections, Stanford University Libraries, Box 6, Folder 15, 11638.
- Engelbart, D. C. (1987): Oral History Interview with Douglas Engelbart conducted by Henry Lowood and Judy Adams, Interview 3, March 4, 1987, S. 7; unter der URL (gesehen 22.3.2000): <http://www.histech.rwth-aachen.de/www/quellen/engelbart/ARCRFC.html>.
- Engelbart, D. C. (1988): »The Augmented Knowledge Workshop«. In: A. Goldberg (Hrsg.): »A History of Personal Workstations«. Reading, Menlo Park, New York, S. 185-232.
- English, W. K./Engelbart, D. C./Berman, M. L. (1967): »Display-Selection Techniques for Text Manipulation«. In: IEEE Transactions on Human Factors in Electronics, HFE-8, 1, S. 5-15.
- Erickson, Th. (1993): »From Interface to Interplace: The Spatial Environment as a Medium for Interaction«. In: Proceedings of the Conference on Spatial Information Theory COSIT '93, Berlin, Heidelberg, New York, S. 391-405.
- Evans, D. R. (1969): »Computer Graphics Hardware Techniques«. In: Parslow, R. D./Prowse, R. W./Green, R. E. (Hrsg.), »Computer Graphics. Techniques and Applications«. London, New York, S. 7-15.
- Everett, R. R. (1951): »The Whirlwind I Computer«. In: Proceedings of the Joint AIEE-IRE Computer Conference, 2 Bde. New York, Bd. 1, S. 70-74.
- Ewing, D. H./Smith, R. W. K. (1946/47): »TELERAN. Air Navigation and Traffic Control by Means of Television and Radar«. In: RCA Review 7 (1946), S. 601-621; 8 (1947), S. 612-632.

- Farber, J. M. (1989), »The AT&T User-Interface Architecture«. In: AT&T Technical Journal Sept./Okt., S. 9-16.
- Feiner, St./MacIntyre, B./Seligman, D. (1993): »Knowledge-Based Augmented Reality«. In: Communications of the ACM 36, 7, S. 51-63.
- Fisher, S. S. (1982): »Viewpoint Dependent Imaging: An interactive stereoscopic display«. In: S. Benton (Hrsg.): »Processing and Display of Three-Dimensional Data«. Proceedings. SPIE 367, 1982
- Fisher, S. S. et al. (1986) »Virtual Environment Display System«. ACM Workshop on 3D Interactive Graphics, Chapel Hill, NC., October 23-24.
- Fisher, S. S. (1990): »Virtual Interface Environments«. In: Laurel, Brenda (Hrsg.): »The Art of Human-Computer Interface Design«. Reading, MA, Menlo Park, CA, New York, S. 423-438.
- Fishkin, K. P./Moran, Th. P./Harrison, B. L. (1998): Embodied User Interfaces: Towards Invisible User Interfaces. In: Proceedings of Engineering for HCI '98, Heraklion, Crete, September 13-18, S. 1-18.
- Fleisch, E./Mattern, F. (Hrsg.) (2005): »Das Internet der Dinge«. Berlin, Heidelberg, New York.
- Foley, J. D. (1987): »Interfaces for Advanced Computing«. In: Scientific American 257, 4, S. 83-90.
- Foley, J. D./Van Dam, A./Feiner, St. K./Hughes, J. F. (1990): »Computer Graphics: Principles and Practice«. 2. Aufl. Reading, MA.
- Friedewald, M. (1999): »Der Computer als Werkzeug und Medium: Die geistigen und technischen Wurzeln des Personal Computers«. Berlin, Diepholz.
- Frohlich, D. M. (1992): »The Design Space of Interfaces«. In: Kjelldahl, L. (Hrsg.), »Multimedia. Principles, Systems, and Applications«. Berlin, Heidelberg, New York, S. 53-69.
- Furness, Th. A. (1986): »Configuring Virtual Space for the Super Cockpit«. In: Ung, M. (Hrsg.) Proceedings of Aerospace Simulation II, San Diego, CA, S.. 103-110.
- Furness, Th. A. (1988): »Harnessing Virtual Space«. In: Proceedings of Society for Information Display International Symposium, Digest of Technical Papers 16, S. 4-7.
- Gentner, D. R./Grudin, J. (1996): »Design models for computer-human interfaces«. In: IEEE Computer, 29, 6, S. 28-35.
- Gilmore, J. T., jr. (1989): o. T. In: »Retrospectives II: The Early Years in Computer Graphics at MIT, Lincoln Lab and Harvard«. SIGGRAPH '89 Panel Proceedings, S. 39-55.
- Grassmuck, V. R. (1995): »Die Turing-Galaxis: Das Universal-Medium auf dem Weg zur Weltsimulation«. In: Lettre International, 28, Frühjahr 5, S. 48-55.

- Grudin, J. (1990): »The computer reaches out: The historical continuity of interface design«. In: CHI '90 Proceedings, New York April 1990, S. 261-268.
- Grudin, J. (1993): »Interface. An evolving concept«. In: Communications of the ACM, 36, 1, S. 103-111.
- Gurley, B. M./Woodward, C. E. (1959): »Light-Pen Links Computer to Operator«. In: Electronics, 28,11, S. 85-87.
- Halbach, Wulf R. (1994), »Reality Engines«. In: N. Bolz, F. A. Kittler, Chr. Tholen (Hrsg.), Computer als Medium«. München, S. 231-244.
- Halfhill, Thomas R. (1982): »Computers in the Home: 1990«. In: Compute !, 15.4.1982; siehe die Internet-Version unter : http://www.commodore.ca/history/other/1982_Future.htm.
- Hellige, H. D. (1995): »Leitbilder, Strukturprobleme und Langzeit-dynamik von Teletex. Die gescheiterte Diffusion eines Telematik-Dienstes aus der Sicht der historischen Technikgeneseforschung«. In: M.-W. Stoetzer, A. Mahler (Hrsg.): »Die Diffusion von Innovationen in der Telekommunikation«, (Schriftenreihe des WIK, Bd. 17). Berlin, Heidelberg, New York, S. 195-218.
- Hellige, H. D. (1996a): »Technikleitbilder als Analyse-, Bewertungs- und Steuerungsinstrumente: Eine Bestandsaufnahme aus informatik- und computerhistorischer Sicht«. In: Ders. (Hrsg.), »Technikleitbilder auf dem Prüfstand. Das Leitbild-Assessment aus Sicht der Informatik- und Computergeschichte«. Berlin, S. 13-36.
- Hellige, H. D. (1996b): »Leitbilder im Time-Sharing-Lebenszyklus: Vom 'Multi-Access' zur 'Interactive On-line Community'«. In: Ders. (Hrsg.), »Technikleitbilder auf dem Prüfstand«. Berlin 1996, S. 205-234.
- Hellige, H. D. (1998): »Der 'begreifbare' Rechner: Manuelles Programmieren in den Anfängen des Human-Computer Interface«. In: I. Rügge, B. Robben, E. Hornecker, W. Bruns, (Hrsg.), Arbeiten und Begreifen: Neue Mensch-Maschine-Schnittstellen«. Münster, Hamburg, S. 187-200.
- Hellige, H. D. (2003): »Zur Genese des informatischen Programmbegriffs: Begriffsbildung, metaphorische Prozesse, Leitbilder und professionelle Kulturen«. In: K.-H. Rödiger (Hrsg.), »Algorithmik - Kunst – Semiotik«. Heidelberg, S. 42-73.
- Hellige, H. D. (2004a): »Sichtweisen der Informatikgeschichte: Eine Einführung«. In: Ders. (Hrsg.), »Geschichten der Informatik. Visionen, Paradigmen und Leitmotive«. Berlin, Heidelberg, New York, S. 1-28.
- Hellige, H. D. (2004b): »Die Genese von Wissenschaftskonzepten der Computerarchitektur: Vom 'system of organs' zum Schichtenmodell des Designraums«. In: Ders. (Hrsg.), »Geschichten der Informatik. Visionen, Paradigmen und Leitmotive«. Berlin, Heidelberg, New York, S. 411-471.

- Hinckley, K./Sinclair, M. (1999): »Touch-Sensing Input Devices«. In: CHI 99 Papers, S. 223-230.
- Hirose, M. (1995): »Foreword«. In: Chorafas, D. N./Steinmann, H., »Virtual Reality: Practical Applications in Business and Industry«. Upper Saddle River, N. J., S. XX.
- Hogan, D. L. (1983): »Speech Interfaces: Session Introduction«. In: Applied Natural Language Conferences. Proceedings of the First Conference on Applied Natural Language Processing, Santa Monica, CA, S. 178-179.
- Holloway, R. (1992): »Virtual Worlds Research Today«. In: BYTE 15, 4, S. 180.
- Hornbuckle, G. D. (1967): »The Computer Graphics User/Machine Interface«. In: IEEE Transactions on Human Factors in Electronics 8, 1, S. 17-20.
- Huang, Th. S./Pavlovic, V. (1995): »Hand Gesture Modeling, Analysis, and Synthesis«. In: Bichsel, M. (Hrsg.): Proceedings International Workshop on Automatic Face- and Gesture-Recognition, Zürich, S. 73-79.
- Hutchins, E. L./Hollan, J. D./Norman, D. A. (1986): »Direct Manipulation Interfaces«. In: D. A. Norman, S. W. Draper (Hrsg.), »User Centered System Design: New Perspectives on Human-Computer Interaction« Hillsdale, N. J., S. 87-124.
- Jacob, R. J. K. (1996): »Human-Computer Interaction: Input Devices«. In: ACM Computing Surveys 28, 1, S. 177-179.
- Jacob, R. J. K./Karn, K. S. (2003): »Eye tracking in Human-Computer Interaction and usability research: Ready to deliver the promises«. In: J. Hyönä, R. Radach, H. Deubel (Hrsg.), »The Mind's Eye: Cognitive and applied aspects of eye movement research«. Amsterdam, S. 573-605.
- Jacobson, B. (1992): »The Ultimate User Interface«. In: BYTE Apr. 1992, S. 175-182.
- Johnson, J. u.a. (1989): »The Xerox Star: A Retrospective.« In: IEEE Computer, 22, 9, S. 11-26.
- Johnson, St. (1999): »Interface Culture. Wie neue Technologien Kreativität und Kommunikation verändern«. Stuttgart.
- Kato, Y. (1995): »The future of voice-processing technology in the world of computers and communications«. In: Rabiner, L. R. (Hrsg.), »Human-Machine Communication by Voice«. Proceedings of the National Academy of Sciences, Bd. 92, pp. 10060-10063, October.
- Kay, A. (1972): »A Personal Computer for Children of All Ages«. In: Proceedings of the ACM National Conference, Boston, August 1972.
- Kay, A. (1975): »Personal Computing«. In: Meeting on 20 Years of Computing Science.. Istituto di Elaborazione della Informazione, Pisa, Italy,
- Kay, A. (1977): »Microelectronics and the Personal Computer«. In: Scientific American 237, 3, S. 231-244.

- Kay, A. (1984): »Computer Software«. In: Scientific American 251, 3, S. 53-59.
- Kay, A. (1989): »Predicting the Future«. In: Stanford Engineering, 1, 1, S. 1-6 (im Internet: <http://www.ecotopia.com/webpress/futures.htm>).
- Kay, A. (1990): »User Interface: A Personal View«. In: Laurel, Brenda (Hrsg.), »The Art of Human Computer Interface Design«. Reading, MA, S. 191-207.
- Kay, A./Goldberg, A. (1977): »Personal Dynamic Media«. In: IEEE Computer 10, 3, 31-42.
- Kay, A./Postman, N. (1999): »The Last Word. Distant Thunder«. In: Contextmagazine Juli 1999; (URL: <http://www.contextmag.com/setFrameRedirect.asp?src=/archives/199907/TheLastWord.asp>).
- Kistermann, F. W. (2000): »The DEHOMAG D11 Tabulator - A Milestone in the History of Data Processing«. In: Rojas, R./Hashagen, U. (Hrsg.), The First Computers - History and Architectures«. Cambridge, MA, London, S. 221-235.
- Konrad, K. (2004): »Prägende Erwartungen. Szenarien als Schrittmacher der Technikentwicklung«. Berlin.
- Krueger, M. W. (1991): »Artificial Reality II«. Reading, MA, Menlo Park, CA, New York.
- Krueger, M. W. (1993): »The Artistic Origins of Virtual Reality«. In: T. Lineham (Hrsg.): SIGGRAPH Visual Proceedings. New York, S.148-149.
- Kurzweil, R. (2002): »Fine Living in Virtual Reality«. In: Denning, P. J. (Hrsg.) (2002): »The Invisible Future«. New York, Chicago, San Francisco, S. 193-215.
- Laman, G. M. (1966): »Position Control Ball Assembly«, US Patent 3269190, 30.8.1966.
- Lange, W. (1986): »Buchungsmaschinen. Meisterwerke feinmechanischer Datenverarbeitung 1910 bis 1960« (Gesellschaft für Mathematik und Datenverarbeitung (Bonn), Berichte, Nr. 162), St. Augustin.
- Lanier, J. (1989): »The Arrival of Virtual Reality in the Real World«. Abstract of talk presented at NATO Workshop on Multi-Media Interface Design in Education, Castelvechio, September.
- Lanier, J. (1999): »Virtual Reality. A Techno-Metaphor with a Life of its Own«. In: Whole Earth Magazine, 98 (Fall 1999), S. 16-18 (URL: <http://www.wholeearthmag.com/ArticleBin/268.html>).
- Lanier, J. (2001): »Virtually There. Three-dimensional tele-immersion may eventually bring the world to your desk«. In: Scientific American, April 17.
- Laurel, B. (1990): »Interface Agents: Metaphors with Character«. In: Dies. (Hrsg.): »The Art of Human Computer Interface Design«. Reading, Mass., S. 355-365.

- Lebedev, M./Nicoletis, M. A. L. (2006): »Brain-machine interfaces: past, present and future«. In: Trends in Neurosciences 29, 9, S. 536-546.
- Lewis, J. R. (2004): »In the Eye of the Beholder«. In: IEEE Spectrum Mai 2004, S. 16-20.
- Lewis, J. R. /Potosnak, K. M./Magyar, R. L. (1997): »Keys and Keyboards«. In: Helander, M./Landauer, Th. K./Prabhu, P. V. (Hrsg.) (1997) Handbook of Human-Computer Interaction, 2. Aufl., Amsterdam, New York, S. 1285-1315.
- Licklider, J. C. R. (1960): »Man-Computer Symbiosis«. In: IRE Transactions on Human Factors in Electronics, Bd.1 März, S. 4-11.
- Licklider, J. C. R. (1965a), »Libraries of the Future«. Cambridge, Mass. 1965.
- Licklider, J. C. R., (1965b) »Static and Dynamic Models«. In: Information Processing 1965. Proceedings of IFIP Congress 65, 2 Bde Washington, D. C., London, Bd. I, S. 476-478.
- Licklider, J. C. R. (1965c), »Principles and Problems of Console Design«. In: Information Processing 1965. Proceedings of IFIP Congress 65, 2 Bde Washington, D. C., London 1965, Bd. I, S. 508-509.
- Licklider, J. C. R. (1967): »Graphic Input – a survey of techniques«. In F. Gruenberger (Hrsg.), »Computer Graphics. Utility, Production, Art«. Washington, S. 39-69.
- Licklider, J. C. R. (1968): »Man-Computer Communication«. In: Annual Review of Information Science and Technology 3 (1968), S. 201-240.
- Licklider, J. C. R. (1976): »User-oriented Interactive Computer graphics«. In: Proceedings of the ACM/SIGGRAPH Workshop on user-oriented design of interactive graphics systems, Oktober, S. 89-96.
- Licklider, J. C. R./Clark, W. E. (1962): »On-line Man-Computer Communication«. In: AFIPS Bd. 21, SJCC, S. 113-123.
- Licklider, J. C. R./Taylor, R. W. (Apr. 1968): »The Computer as a Communication Device«. In: Science and Technology, 76, 2, S. 21-31, wiedergedruckt in: In Memoriam Joseph C.R. Licklider 1915-1990, (Digital SRC Research Report 61), Digital Research Center August 1990; Internet-Version: <http://gatekeeper.dec.com/pub/DEC/SRC/research-reports/abstracts/src-rr-061.html>
- Lindgren, N. (1965): »Machine Recognition of Human Language. Part I: Automatic speech recognition«. In: IEEE Spectrum, Bd. 2, 3, S. 62-72.
- Lindgren, N. (1966): »Human Factors in Engineering. Part II: Advanced Man-Machine Systems and Concepts«. In: IEEE Spectrum, Bd. 3,4 S. 62-72.
- Lovelace, A. A. (1843): »Sketch of the Analytical Engine Invented by Charles Babbage Esq. By L. F. Menabrea«. In: Scientific Memoirs 3, S. 666-731; wiedergedruckt in: »Works of Babbage«, hrsg. von M. Campbell-Kelley, 11 Bde. New York 1989, Bd. 3, S. 89-170.

- Machover, C. (1978): »A Brief Personal History of Computer Graphics«. In: Computer 11, 11, S. 38-45.
- Machover, C. (1999): »Computer graphics pioneers«. ACM SIGGRAPH, 33, 3 (auch: www.es.com/about_eands/history.asp).
- Machover, C./Tice, St. E. (1994): »Virtual Reality«. In: IEEE Computer Graphics and Applications, 14, 1, S. 15-16.
- MacKenzie, I. S. (1995): »Input Devices and Interaction Techniques for Advanced Computing«. In: Barfield, W./Furness, T. A.: »Virtual Environments and Advanced Interface Design«. New York, S. 437-470.
- Maisel, A./Engelbart, D. (1996): »Father of the Mouse«. Interview for SuperKids Educational Software Review« (unter der URL: <http://www.superkids.com/aweb/pages/features/mouse/mouse.html>).
- Martin, E. (1925): »Die Rechenmaschinen und ihre Entwicklungsgeschichte. 1. Band: Rechenmaschinen mit automatischer Zehnerübertragung«. Pappenheim, Reprint Leopoldshöhe 1986.
- Mambrey, P./Paetau, M./Tepper, A. (1995): »Technikentwicklung durch Leitbilder. Neuere Steuerungs- und Bewertungsinstrumente«. Frankfurt a. M., New York.
- Mann, St. (1997): »Wearable Computing: A first step towards personal imaging«. In: IEEE Computer 30,2, S. 25-32.
- Mann, St. (1998): »Humanistic Computing: 'WearComp' as a Framework and Application for Intelligent Signal Processing«. In: Proceedings of the IEEE 86, 11, S. 2123-2151
- Marguin, J. (1994): »Histoire des instruments et machines à calculer. L'épopée de la mécanique pensante 1642-1942«. Paris.
- Marill, Th. (1961): »Automatic Recognition of Speech«. In: IRE Transactions on Human Factors in Electronics, HFE-2, Mrz, S. 34-38.
- Martin, E. (1925): »Die Rechenmaschinen und ihre Entwicklungsgeschichte. 1. Band: Rechenmaschinen mit automatischer Zehnerübertragung«. Pappenheim, Reprint Leopoldshöhe 1986
- Mattern, F. (2005): »Acht Thesen zur Informatisierung des Alltags«. In: Acatech Symposium, 28. Juni 2005, S. 54-59.
- Maybury, M. T./Wahlster, W. (1998): »Readings in Intelligent User Interfaces«. San Francisco.
- Meadow, Ch. T. (1970): »Man-Machine Communication«. New York, London, Syney, Toronto.
- Mooers, C. N. (1959): »The Next Twenty Years in Information Retrieval: Some Goals and Predictions«. In: AFIPS, Bd. WJCC, S. 81-86; wiedergedr. in: American Documentation 11 (Juli 1960), S. 229-236.
- Mori, S./Suen; Ch. Y./Yamamoto, K. (1992): »Historical Review of OCR Research and Development«. In: Proceedings of the IEEE 80, 7, S. 1029-1057.

- Myers, B. A. (1998): »A Brief History of Human Computer Interaction Technology«. In: ACM Interactions 5, 2, S. 44-54.
- Myers, B. A./Hudson, S. E./Pausch, R. (2000): »Past, Present, and Future of Interface Software Tools«. In: ACM Transactions on Computer-Human Interaction, 7, 1, S. 3–28.
- Nake, F. (1995): »Vom Batch Processing zu Direct Manipulation: ein Umbruch im Umgang mit Computern«. In: Hurrell, G./Jelich, F.-J. (Hrsg.), »Vom Buchdruck in den Cyberspace? Mensch – Maschine – Kommunikation«. Dokumentation einer Tagung im DGB-Bildungszentrum, Juni 1993, Marburg, S. 28-44.
- Negroponte, N. (1989): »An Iconoclastic View beyond the Desktop Metaphor«. In: International Journal on Human-Computer Interaction, 1, 1, S. 109-113. f
- Negroponte, N. (1991): »Beyond the Desktop Metaphor«. In: Meyer, A. R./Guttag, J. V.u.a. (Hrsg.): »Research Directions in Computer Science«. Cambridge, Mass., London 1, S. 183-190 (erweiterte Version eines Aufsatzes von 1989).
- Negroponte, N. (1997): »Agents: From Direct Manipulation to Delegation«. In: Bradshaw, J. M. (Hrsg.): »Software Agents«. Menlo Park, CA, Cambridge, MA, London.
- Netravali, A. N. (1999): »A Response to 'Wide-Band Communication into the Home'«. In: Proceedings of the IEEE, 87, 1, S. 201-204.
- Newell, A. F. (1985): »Speech - The Natural Modality for Man-Machine Interaction?«. In: Shackel, B. (Hrsg.), Human-Computer Interaction, INTERACT '84, Amsterdam, S. 231-235.
- Neumann, J. v. (1945): »First Draft of a Report on the EDVAC«. Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia 30. Juni 1945; korrigierte Fassung nach dem Originalmanuskript, hrsg. von M. D. Godfrey. In: Annals of the History of Computing 15 (1993) 4, S. 27-67.
- Neumann, J. v. (1958): »The Computer and the Brain«. 2. Aufl. New Haven, London (2000).
- Nielsen, J. (1993): »Noncommand user interfaces«. In: Communications of the ACM 36, 4, S. 83-99.
- Norberg, A. L./O'Neill, J. (1996): »Transforming Computer Technology: Information Processing for the Pentagon, 1962-1986«. Baltimore, MD.
- Norman, D. A. (1983): »Design Principles for Man-Computer Interfaces«. In: CHI '83 Proceedings, Human Factors in Computer Systems Conference, Boston 12-15. Dez. 1983, S. 1-10.
- Norman, D. A. (1998): »The Invisible Computer«. Cambridge, MA, London.

- Nyce, J. M./Kahn, P. (1991): »A Machine for the Mind: Vannevar Bush's Memex«. In: Dies. (Hrsg.): »From Memex to Hypertext: Vannevar Bush and the Mind's Machine«, Boston, San Diego, New York, S. 39-66.
- Ogden, W. C./Bernick, Ph. (1998): »Using Natural Language Interfaces«. In: Helander, M./Landauer, Th. K./Prabhu, P. V. (Hrsg.) (1997): »Handbook of Human-Computer Interaction«. 2. Aufl., Amsterdam, New York, S. 137-161.
- Oinas-Kukkonen, H. (2007): »From Bush to Engelbart: ›Slowly, Some little Bells Were Ringing‹«. In: *Annals of the History of Computing* 29, 2, S. 31-39.
- Olson, H. F. (1975): »An Interview Conducted by Mark Heyer«. IEEE History Center July 14, 1975 Interview 026 For the IEEE History Center, The State University of New Jersey.
- Oviatt, S. (1999): »Ten Myths of Multimodal Interactions«. In: *Communications of the ACM* 42 (1999) 11, S. 74-81.
- Peled, A. (1987): »The Next Computer Revolution«. In: *Scientific American* 257, 4, S. 35-42.
- Pentland, A. (1995): »Interactive Video Environments and Wearable Computers«. In: Bichsel, M. (Hrsg.): *Proceedings International Workshop on Automatic Face- and Gesture-Recognition, Zürich 1995*, S. 71-72.
- Pentland, A. (2000): »Perceptual Intelligence«. In: *Communications of the ACM* 43, 11, S. 35-44.
- Perry, T. S./Voelcker, J. (1989): »Of mice and menus: designing the user-friendly interface«. In: *IEEE-Spectrum*, 26, 9, S. 46-51.
- Peterson, R. H. (1955): »Mechanical Movement«, US Patent Nr. 2939332, 31.5.1955.
- Pflüger, J. (2004): »Konversation, Manipulation, Delegation. Zur Ideengeschichte der Interaktivität«. In: Hellige, H. D. (Hrsg.), »Geschichten der Informatik«. Berlin, Heidelberg, New York, S. 367-408.
- Rabiner, L. R. (1995): »Voice Communication between Humans and Machines - An Introduction«. In: ders. (Hrsg.): »Human-Machine Communication by Voice«. National Academy of Sciences at The Arnold and Mabel Beckman Center in Irvine, CA, February 1993, *Proc. Natl. Acad. Sci. USA* Bd. 92, Oct., S. 9911-9913.
- Rabiner, L. R./Juang, B.-H. (1993): »Fundamentals of Speech Recognition«. Englewood Cliffs, N. J.
- Ramo, S. (1961): »Man-Machine-Communications in the Coming Technological Society«. In: Hollitch, R. S./Mittman, B. (Hrsg.): »Computer Applications-1961 Proceedings of the Computer Applications Symposium, Chicago Okt. 1961«. New York, London, S. 45-54.
- Reinhardt, A. (1991): »Touch-and Feel Interfaces«. In: *BYTE* 16,2, S. 223-226.

- Rheingold, H. (1991): »Virtual Reality. Exploring the Brave New Technologies«. New York.
- Rice, R. (1959): »Computers of the Future«. In: AFIPS 16, EJCC, S. 8-14
- Rider, R. E. (1974): »Position Indicator for a Display System«, US Patent Nr. 3853464, 10.9.1974.
- Robben, B. (2006): »Der Computer als Medium. Eine transdisziplinäre Theorie«. Bielefeld.
- Roberts, L. G. (1964): »Graphical Communication and Control Languages«. Lincoln Laboratory, MIT Graphical Communication and Control Languages, Second Congress on Information System Science, Hot Springs, Virginia.; Netzversion unter der URL (zuletzt April 2007): <http://www.packet.cc/files/graph-comm-con.html>.
- Roberts, L. G. (1989): o. T. In: »Retrospectives II: The Early Years in Computer Graphics at MIT«. Lincoln Lab and Harvard, SIGGRAPH '89 Panel Proceedings, S. 56-65.
- Ross, D. T. (1988): »A Personal View Of the Personal Workstation. Some Firsts in the Fifties«. In: A. Goldberg (Hrsg.): »A History of Personal Workstations«. Reading, Menlo Park, New York, S. 54-111.
- Roth, E. M./Malin, J. T./Schreckenghost, D. L. (1997): »Paradigms for Intelligent Interface Design«. In: Helander, M./Landauer, T. K./Prabhu, P. V. Hrsg.) (1997) »Handbook of Human-Computer Interaction«. 2. Aufl., Amsterdam, New York, S. 1177-1201.
- Rubenstein, D. A. (1975): »The Status and Potential of OCR«. In: »Input/Output«. Infotech State of the Art Report 22, Maidenhead, S. 159-173.
- Rügge, I. (2007): »Mobile Solutions. Einsatzpotentiale, Nutzungsprobleme und Lösungsansätze« (Advanced Studies Mobile Research Center Bremen), Wiesbaden.
- Rügge, I./Robben, B./ Hornecker, E./ Bruns, W. (Hrsg.), (1998): »Arbeiten und Begreifen: Neue Mensch-Maschine-Schnittstellen«. Münster, Hamburg.
- Sackman, H. (1967): »Computer, System Science and Evolving Society. The Challenge of Man-Machine Digital Systems«. New York, London, Sydney.
- Schelhowe, H. (1997): »Das Medium aus der Maschine. Zur Metamorphose des Computers«. Frankfurt a. M., New York.
- Schilit, B. N./Adams, N. I./Want, R. (1994): »Context-aware Computing Applications«. In: Proceedings of the 1st International Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, IEEE. Dec. 8-9, S. 85-90.
- Schröter, J. (2004): »Das Netz und die Virtuelle Realität. Zur Selbstprogrammierung der Gesellschaft durch die universelle Maschine«. Bielefeld.

- Shackel, B. (Hrsg.) (1985): »Designing for People in the Age of Information«. In: INTERACT 84 - 1st IFIP International Conference on Human-Computer Interaction, London Sept. 1984, Amsterdam, S. 9-18.
- Sheridan, Th. B. (1992): »Musings on Telepresence and Virtual Presence«. In: Presence-Teleoperators and Virtual Environments, 1,1, S. 120-126.
- Shneiderman, B. (1991): »Touchscreens now offer compelling uses«. In: IEEE Software 8,2, S. 93-94, 107.
- Shneiderman, B. (2000): »The Limits of Speech Recognition«. In: Communications of the ACM 43, 9, S. 63-65.
- Skramstad, H. K./Ernst, A. A./Nigro, J. P. (1957): »An Analog-Digital Simulator for the Design and Improvement of Man-Machine System«. In: Proceedings of the Eastern Computer Conference Dez., S. 90-96.
- Starner, Th. (2001): »The Challenges of Wearable Computing: Part 1«. In: IEEE MICRO, 21, 4, S. 44-52; Part 2. In: Ebda., 21, 4, S. 54-67
- Strehl, R. (1952): »Die Roboter sind unter uns. Ein Tatsachenbericht«. Oldenburg.
- Streitz, N. A. (1988): »Mental models and metaphors: Implications for the design of adaptive user-system interfaces«. In: Learning Issues for Intelligent Tutoring Systems, New York, Heidelberg.
- Sturman, D. J./Zeltzer, D. (1994): »A survey of glove-based input«. In: IEEE Computer Graphics and Applications 14, 1, S. 30-39.
- Sullivan, J. W./Tyler, S. W. (Hrsg.) (1991): »Intelligent User Interfaces. Elements and Prototypes«. New York.
- Sutherland, I. E. (1962): »Sketchpad A Man-Machine Graphical Communication System«. Phil. Diss. 1962, M.I.T. Lincoln Laboratory Technical 296 Report, Januar 1963; Reprint als Technical Report 574 Cambridge, MA 2003.
- Sutherland, I. E. (1963): »Sketchpad. A Man-Machine Graphical Communication System«. In : American Federation of Information Processing Societies, (AFIPS), Conference Proceedings Bd. 23, SJCC, S. 329-346.
- Sutherland, I. E. (1965a): »The Ultimate Display«. In: Information Processing 1965. Proceedings of IFIP Congress 65, 2 Bde Washington, D. C., London 1965, Bd. I, S. 506-508.
- Sutherland, I. E. (1965b): »The Future of On-Line Systems«. In: Proceedings of the Symposium Sponsored by the University of California at Los Angeles and Informatics, Inc., Los Angeles, California, February 1965.
- Sutherland, I. E. (1967): »Ein- und Ausgabe bei elektronischen Rechenanlagen«. In: Information, Computer und künstliche Intelligenz, Frankfurt a. M., S. 51-67 (Scientific American, Sept. 1966).
- Sutherland, I. E. (1968): »A Head-mounted Three Dimensional Display«. In: American Federation of Information Processing Societies, (AFIPS), Conference Proceedings Bd. 33, FJCC, S. 757-764.

- Sutherland, I. E. (1970): »Computer Displays«. In: Scientific American, 222, 6, S. 56-81.
- Sutherland, I. E. (1971): »Introduction«. In: Prince, M. D.: »Interactive Graphics for Computer Aided Design«. Reading, Mass. S. IX-XI.
- Taylor, N. (1989): o. T. In: »Retrospectives I: The Early Years in Computer Graphics at MIT«. Lincoln Lab and Harvard, SIGGRAPH '89 Panel Proceedings, S. 20-25.
- Taylor, R. W. (1968): »Man-Computer Input-Output Techniques«. In: IEEE Transactions on Human Factors in Electronics, 8, 1, S. 1-4.
- Tennenhouse, D. (2000): »Proactive Computing«. In: Communications of the ACM, 43, 5, S. 43-50.
- Thompson, T. (1996): »The Elements of Design. An inside view of how innovative individuals produce technology breakthroughs«. In: BYTE, 21, 8, S. 81.
- Trimble, G. R. (1986): »The IBM 650 Magnetic Drum Calculator«. In: Annals of the History of Computing 8, 1, S. 20-29.
- Turing, A. M. (1947): »Lecture to the London Mathematical Society on 20 February«, wiedergedruckt in: Carpenter, B. E./Doran, R. W. (Hrsg.), »A. M. Turing's ACE Report of 1946 and other Papers« (Charles Babbage Institute, Reprint Series for the History of Computing, Bd. 10), London, Los Angeles, San Francisco 1986, S. 106-124.
- Turing, A. M. (1950): »Computing Machinery and Intelligence«. In: Mind 59, S. 433-460; wiedergedruckt in: Ders.: »Collected Works of A. M. Turing«. 4 Bde. Amsterdam, London, New York 1992; Bd. 3 (Mechanical Intelligence, hrsg. von D. C. Ince), S. 133-160.
- Turner, J./Krueger, M. (2002): »Myron Krueger Live, Interview«. In: ctheory.net, 23.1.2002; unter: [www.ctheory.net/articles.aspx?id= 328](http://www.ctheory.net/articles.aspx?id=328) (zuletzt: 10. Juni 2007).
- Van Dam, A. (1966): »Computer Driven Displays and their Use in Man/Machine Interaction«. In: Alt, F. L./Rubinoff, M. (Hrsg.): »Advances in Computers«. Bd. 7, New York, S. 239-290.
- Van Dam, A. (1999): »Post-WIMP 3D User-Computer Interaction, HCI«. Distinguished Colloquium Series, Center for Human Computer Interaction, 3. 5. 1999, unter: <http://www.cse.ogi.edu/CHCC/Series/lecture99.htm>.
- Van Dam, A. (2001): »User Interfaces: Disappearing, Dissolving, and Evolving«. In: Communications of the ACM 44, 3, S. 50-52.
- Wagner, K. (2006): »Datenräume, Informationslandschaften, Wissensstädte: Zur Verräumlichung des Wissens und Denkens in der Computermoderne«. Freiburg i.Br., Berlin.

- Walker, J. (1990): »Through the Looking Glass«. In: Laurel, B. (Hrsg.): »The Art of Human-Computer Interface Design«. Reading, MA, Menlo Park, CA, New York, S. 439-447.
- Watson, B./Luebke, D. (2005): »The Ultimate Display: Where Will All the Pixels Come From?« In: IEEE Computer 38, 8, S. 54-61.
- Weiser, Mark (1991): »The Computer for the 21st Century«. In: Scientific American 265 (Sept.), S. 94-104.
- Weiser, M. (1993): »Some Computer Science Issues in Ubiquitous Computing«. In: Communications of the ACM 36, 7, S. 75-84.
- Weiser, M. (1994): »Creating the Invisible Interface.« In: Symposium on User Interface Software and Technology New York.
- Weiser, M. (1998): »The Future of Ubiquitous Computing on Campus«. In: Communications of the ACM 41, 1, S. 41-42.
- Weiser, M./Brown, J. S. (1996): »Designing Calm Technology«. In: PowerGrid Journal, v 1.01, July 1996; revid. Fassung, 5.10.1996. unter: <http://www.ubiq.com/hypertext/weiser/acmfuture2endnote.htm>.
- Weiser, M./Gold, R./Brown, J. S. (1999): »The Origins of Ubiquitous Computing Research at PARC in the late 1980s«. In: IBM Systems Journal, 38. Jg., H. 4, S. 693-696.
- Wellner, P./Mackay, W./Gold, R. (1993): »Computer-Augmented Environments - Back to the real world«. In: Communications of the ACM 36, 7, S. 24-26.
- Wildes, K. L./Lindgren, N. A. (1985): »A Century of Electrical Engineering and Computer Science at MIT, 1882-1982«. Cambridge/Mass., London.
- Wolf, T. V./Rode, J.A./Sussmann, J./Kellog, W. A. (2006): »Dispelling Design as the 'Black Art' of CHI«. In: CHI 2006 Conference Proceedings. Montréal, Québec April 22-27, New York, S. 521-529.
- Zemanek, H. (1983): »Otto Schäffler (1838-1928). Pionier des Telephons, der Telegraphie und der Lochkarte sowie Erbauer der ersten Wiener Telephonzentralen«. In: Blätter für Technikgeschichte, Heft 41-43, Wien, S. 81-118.
- Zimmerman, Th. G./Lanier, J./Blanchard, Ch./Bryson, St./Harvill, Y. (1987): »A Hand Gesture Interface Device«. In: Proceedings of SIGGHI '87, S. 189-192.

**INTERFACES DER MAINFRAME-ÄRA:
VOM AUTOMATISCHEN ZUM INTERAKTIVEN
COMPUTING**

DIE ERGONOMISCHEN ERFINDUNGEN DER ZUSE-MASCHINEN IM INTERNATIONALEN KONTEXT

HORST ZUSE

1 Einleitung

Die Ergonomie ist die Wissenschaft von der Gesetzmäßigkeit menschlicher Arbeit, zentral ist dabei die Verbesserung der Schnittstelle zwischen Benutzer (= Mensch) und Objekt (= Maschine), also der Benutzerfreundlichkeit. Benutzerfreundlichkeit (auch »Benutzungsfreundlichkeit«) bezeichnet die vom Nutzer erlebte Nutzungsqualität bei der Interaktion mit einem System. Eine besonders einfache, zum Nutzer und seinen Aufgaben passende Bedienung wird dabei als benutzerfreundlich angesehen.

Es war ein enorm hoher technischer Aufwand, vollautomatische programmgesteuerte Rechenmaschinen, die wir heute als Computer bzw. Digitalrechner bezeichnen, Ende der 30er und Anfang der 40er Jahre zu konstruieren. Die Bedienungsfreundlichkeit der frühen Rechenmaschinen war sehr unterschiedlich ausgeprägt. Bei einigen Rechenmaschinen war die Programmierung bzw. Konfigurierung zeitlich sehr aufwendig, da sehr unterschiedliche Erstellungsprozeduren und Speichermedien für die Programme verwendet wurden. Die von Konrad Zuse konstruierten Rechenmaschinen Z1-Z4 (1936-1950) verfügten über speziell konstruierte ergonomische Schnittstellen, die eine direkte Interaktion zwischen dem Nutzer und der Rechenmaschine im Dialog ermöglichten. Diese ergonomischen Erfindungen Konrad Zuses im internationalen Kontext sind bisher nur wenig beachtet worden.

In diesem Beitrag werden zunächst die ergonomischen Erfindungen Konrad Zuses bei den Maschinen Z1-Z4, dem Planfertigungsteil der Maschine Z4 und dem geplanten Planfertigungsgerät beschrieben. Dann werden dann die ergonomischen Konzepte der frühen Rechenmaschinen aus den USA und England betrachtet und anschließend beide Richtungen miteinander verglichen.

2 Konrad Zuses Rechenmaschinen Z1-Z4 und seine ergonomischen Erfindungen

Die Architektur der Zuse-Maschinen (Z-Maschinen) ist schon in vielen Artikeln und Büchern beschrieben worden. Beispiele sind: Der Artikel von Alex (1997) im Spektrum der Wissenschaft, der Buchklassiker über die Entwicklung der Rechenmaschinen von de Beauclair (1968) und das Buch von Paul Ceruzzi (1983) mit einer sehr detaillierten Abhandlung der Zuse-Maschinen. Das Buch von Karl Steinbuch (1966), heute ebenfalls ein Klassiker, beschäftigt sich durchaus kritisch mit der Rolle der deutschen Computerindustrie um 1965. Das Buch von Raul Rojas (1998) analysiert und beschreibt die Rechenmaschinen von Konrad Zuse im Detail. Der Band von Hans Dieter Hellige (2004) enthält umfangreiche Abhandlungen über Konrad Zuses Rechenmaschinen. In Horst Zuse (2007a) ist eine detaillierte Beschreibung der Z3 zu finden und die CD des Autors (2007b) ist der Versuch, auf einem Multimedia-Medium das Werk Konrad Zuses, aber auch anderer Computerpioniere, darzustellen und mit umfangreichem Bild- und Filmmaterial sowie Simulationen von Z-Maschinen zu illustrieren.

Wenig beachtet wurden bisher die ergonomischen Erfindungen Konrad Zuses bei der Konstruktion seiner frühen Rechenmaschinen. In den dem Autor vorliegenden internen Berichten von Konrad Zuse aus den Jahren 1936 bis ca.1947 finden sich sehr viele Anmerkungen bezüglich der Handhabung der von ihm geplanten Rechenmaschinen. Die prägnanteste Formulierung ist aus unserer Sicht die Beschreibung der idealen Rechenmaschine für den Ingenieur. Am 30.1.1936 schreibt er:

»Die Anforderungen, die an die ideale Rechenmaschine des Ingenieurs gestellt werden müssen, gehen aus den folgenden Überlegungen hervor: Der Ingenieur hat viel mit festen Formeln zu arbeiten, die immer wiederkehren. Man hat gewisse Ausgangswerte, und die Arbeit besteht nun darin, durch bestimmte, für eine Formel immer gleiche Aufeinanderfolge von Grundrechenarten zwischen bestimmten Zahlen das Resultat zu berechnen.

Der Ingenieur braucht Rechenmaschinen, die diese Rechenoperationen automatisch ausführen, indem der Rechenplan auf einem Lochstreifen festgehalten wird, der die Befehle für die einzelnen Rechenoperationen selbsttätig und nacheinander an die Maschine gibt.

Die Maschine muß auf Befehl des Lochstreifens jede verlangte Grundrechnung vollautomatisch ausführen. Ferner muß die Maschine über ein Speicherwerk verfügen, in welchem die während der Rechnung auftretenden Zahlen der Nummer nach geordnet werden können, und aus denen durch ein mechani-

sches Wählwerk jede gewünschte Zahl abgelesen werden kann. Das Speichern und Ablesen der Zahlen wird ebenfalls durch den Befehlslochstreifen dirigiert.«

Durch die Spezifizierung der idealen Rechenmaschine für den Ingenieur leitet Konrad Zuse folgende Möglichkeiten seiner Maschine für den Ingenieur ab:

»Durch die Maschine wird dem Ingenieur die mechanische Rechenarbeit nicht nur abgenommen, sondern ihr Umfang kann enorm gesteigert werden. Man kann die Möglichkeiten der Maschine wie folgt abstufen:

Rationalisierung der Rechnungen ohne im wesentlichen von dem bisherigen Rechenverfahren abzuweichen.

Entwicklung neuer Methoden zur Lösung technischer Probleme.

Erschließung von Gebieten, die bisher der Rechnung nicht zugänglich waren« (Zuse 1936, S. 12 ff.).

Bemerkenswert sind die Anmerkungen von Konrad zum Zahlensystem, welches in der Maschine verwendet werden soll und der Bezug zum Ingenieur. Wir zitieren dazu aus, Zuse (1936):

»Es ist bisher für selbstverständlich gehalten worden, dass Rechenmaschinen nach dem Dezimalsystem arbeiten müssen. Das Dezimalsystem hat sich derart tief in unserm Denken festgesetzt, dass wir geneigt sind, es für ›das‹ Zahlensystem überhaupt zu halten. Hat man Maschinen, die lange technische Rechnungen automatisch ausführen, bei denen der Bedienende die einzelnen Zahlen der Zwischenrechnung gar nicht zu Gesicht bekommt, so ist die Art des benutzten Zahlensystems gleichgültig. Ist das Dezimalsystem unpraktisch, so kann man ein geeigneteres wählen, vorausgesetzt, dass es keine Schwierigkeiten macht, die Anfangs- und Endwerte der Rechnung ins Dezimalzahlensystem zu übersetzen.«

Wir fassen Konrad Zuses Ideen des einfachen Rechnens für den Ingenieur zusammen:

- Die Rechenmaschine soll ideal für den Ingenieur sein.
- Der Ingenieur soll von der bisher gewohnten Rechenart nicht abweichen müssen.
- Die Eingabe und Ausgabe der Zahlen soll im gewohnten Dezimalsystem erfolgen.
- Die Maschine soll frei programmierbar sein, d.h. mit wenigen sinnvollen Befehlen, die sich auf dem Speichermedium eines Lochstreifens befinden, zu bedienen sein.

Wir sehen hier die ersten wichtigen ergonomischen Konzepte von Konrad Zuse zur Konzeption seiner frühen Rechenmaschinen Z1-Z4. Da die Maschinen Z1 und Z3 sich von der Architektur her fast gleichen (Das Rechenwerk der Z3 verfügte zusätzlich über die Quadratwurzel und die arithmetische Ausnahmebehandlung, die Z2 war nur ein Versuchsmodell zum Testen der Telefonrelais mit dem mechanischen Speicher der Z1) betrachten wir nun die Realisierung der obigen Konzepte bei der vollautomatischen Rechenmaschine Z3.

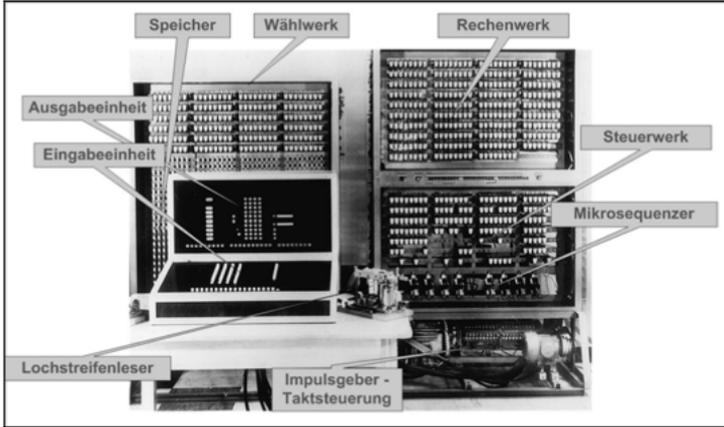
Die Rechenmaschine Z3

Die vollautomatische programmgesteuerte Rechenmaschine Z3, die am 12. Mai 1941 in Berlin-Kreuzberg in allen Komponenten Wissenschaftlern der Deutschen Versuchsanstalt für Luftfahrt (DVL) funktionsfähig vorgestellt wurde, rechnet mit binären Gleitkommazahlen. Die Ein- und Ausgabe erfolgt aber mit dezimalen Gleitkommazahlen, um dem Ingenieur einen einfachen Umgang mit der Ein- und Ausgabe von Zahlen zu ermöglichen. Dazu ist eine Übersetzung von dezimalen Gleitkommazahlen in binäre Gleitkommazahlen notwendig, die auch schon von der Maschine Z1 durchgeführt wurde.

Abbildung 1 zeigt den Nachbau der Maschine Z3 von 1961 mit den wichtigsten Komponenten. Es ist verblüffend zu beobachten, dass die logische Grundarchitektur moderner Computer eine große Ähnlichkeit zu der Maschine Z3 aufweisen. Die Z3 wurde am 12. Mai 1941 fertig gestellt und gilt heute als der erste funktionsfähige, frei programmierbare, auf dem binären Gleitkommazahlensystem basierende Rechner der Welt. Verkürzt kann gesagt werden: Die Z3 ist der erste funktionsfähige Digitalrechner.

Die Maschine Z3 konnte in zwei Betriebsmodi betrieben werden, und zwar in dem Programmmodus und dem manuellen Modus. Konrad Zuse beschreibt diese Betriebsmodi selbst bei der Konstruktion der Z4 (siehe unten). Ist kein Rechenplan auf dem Lochstreifen vorhanden, dann arbeitet die Maschine immer im manuellen Modus, d.h. im Dialogbetrieb. Ein laufender Rechenplan kann durch den Befehl Lu (Eingabe einer Zahl) unterbrochen werden. Dieser Befehl bewirkt den Stop des Ablaufes des Rechenplanes und fordert die Eingabe eines Zahlenwertes über die Tastatur. Die Maschine Z3 befindet sich jetzt im manuellen Modus. Wird ein Rechenplan ausgeführt, dann befindet sich die Maschine im Programmmodus.

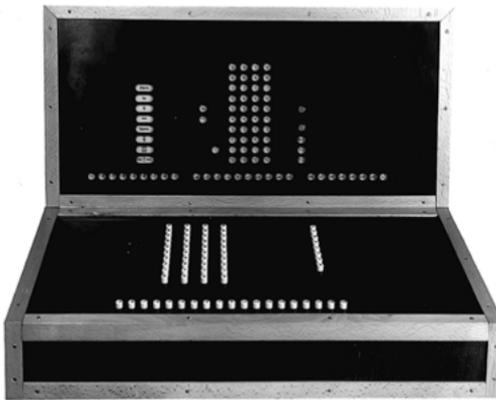
Abb. 1: Der Nachbau der Maschine Z3 von 1961 mit den Komponenten: Speicher, Wählwerk, Rechenwerk, Steuerwerk, Eingabe- und Ausgabeeinheit, Lochstreifenleser, Mikrosequenzer und dem Impulsgeber mit der Taktsteuerung. Die Original-Z3 Maschine enthielt noch einen zweiten Speicherblock, während der Nachbau nur einen Speicherblock mit 32 Worten mit je 22 Bits enthält (Konrad Zuse Archiv von Horst Zuse in Berlin).



Für Rechenoperationen, unabhängig davon, ob im Programmmodus oder im Manuellen Modus gearbeitet wird, stehen die beiden Register R1 und R2 als Kurzspeicher (22 Bits) für die Operanden der arithmetischen Operationen zur Verfügung. Gerechnet wird in der umgekehrten polnischen Notation, wie z.B. bei dem Taschenrechner HP 45 (1972) oder HP11 (1998). Über das Tastenfeld in der Mitte kann die Mantisse mit einer Genauigkeit von vier Dezimalziffern eingegeben werden, die vordere Tastenreihe erlaubt die Eingabe des Exponenten von -8 bis 8 . Mit den Tasten auf der rechten Seite können folgende Maschinenoperationen durchgeführt werden: Ld: Anzeige des Inhaltes von Register R1 auf der Ausgabeeinheit (oberer Teil der Ausgabeeinheit); R1R2: Speichern der eingegebenen Zahl in Register R1 oder R2; Ls1: Addition der Inhalte der Register R1 und R2: $R1 := R1 + R2$ (Zu lesen: die Inhalte der Register R1 und R2 werden addiert und das Ergebnis wird R1 zugewiesen); Lm: Multiplikation der Inhalte von Register R1 und R2: $R1 := R1 * R2$; Ld: Division der Inhalte von Register R1 und R2: $R1 := R1 / R2$; Lw: Quadratwurzel des Inhaltes von Register R1: $R1 := \text{SQR}(R1)$; A: der Rechenplan (Programm), falls vorhanden, wird fortgesetzt. Die Umschaltmöglichkeit vom Programmmodus in den Dialogmodus ist für den Nutzer sehr hilfreich und ist ein bei vielen programmierbaren Taschenrechnern beibehaltenes Prinzip.

Die vertikale Lampenreihe oben links zeigt nicht erlaubte arithmetische Operationen an, wie z.B. die Division durch 0 (Nicht realisiert bei den Maschinen Z1 und Z2). Diese Anzeigen sind ein weiterer Aspekt der Benutzerfreundlichkeit, denn Konrad Zuse wollte vermeiden, dass seine Maschine inkorrekt rechnet und dies dem Nutzer verborgen bleibt. Wir beschreiben dieses Konzept genauer bei der Rechenmaschine Z4.

Abb. 2: Die Ein- und Ausgabeeinheiten der Z3, Nachbau von 1961, (Konrad Zuse Archiv von Horst Zuse in Berlin).



2.2 Ergonomische Konstruktionsideen bei der Rechenmaschine Z4

Mit der Konstruktion der Maschine Z4 begann Zuse im Jahr 1942 in Berlin-Kreuzberg. Der Auftrag zum Bau dieser Rechenmaschine erfolgte 1942 durch die Henschel-Flugzeugwerke in Berlin Adlershof. In dieser Zeit bezeichnete er seine Maschinen noch mit V1-V4 (Versuchsmodelle 1-4). Erst nach dem Zweiten Weltkrieg erfolgte die Umbenennung in Z1-Z4. Die Z4 sollte die Maschine sein, die in Serie produziert und an Ingenieurbüros ausgeliefert werden sollte. Während Zuse die Maschinen Z1-Z3 noch als Prototypen betrachtete, wurden bei der Konstruktion der Z4 weitere Ideen verwirklicht, die es dem Anwender bzw. Ingenieur erleichtern sollten, mit einer solch komplexen Maschine zu arbeiten.

In einem Brief vom 11. November 1945 an die Eltern in Berlin aus Hinterstein finden wir Ausführungen, die die Bedeutung der Maschine Z4 für ihn belegen. »Aus unserer ersten Göttinger Zeit haben wir ja noch Nachricht geben können. Kurz vor Ostern war ich mit der Aufstellung

des Gerätes fertig und konnte vor den Professoren der AVA (Autor: Aerodynamische Versuchsanstalt) die Maschine einwandfrei arbeitend vorführen. Es war dies der Moment, auf den ich etwa 10 Jahre gewartet hatte, wo meine Arbeit endlich den gewünschten Erfolg brachte« (Konrad Zuse, 1945).

Im Jahr 1943 fasste Zuse (1943b) die Ideen zum Bau der Maschine Z4 zusammen. Zuerst wird der Begriff des Rechnens neu gefasst:

»Mit Rechnen wird in diesem Bericht ganz allgemein das Rechnen mit Umständen und Bedingungen bezeichnet. Das Rechnen mit Zahlen ist nur eine spezielle Art dieses allgemeinen Rechnens, das durch folgenden Satz definiert wird: Rechnen ist die Ableitung von Resultatangaben aus irgendwelchen gegebenen Angaben nach einer Vorschrift«.¹

Wie schon die Z1, Z2 und Z3 wird auch die Z4 durch einen Rechenplan gesteuert. Konrad Zuse führt dazu aus:

»Längere oder sich häufende wiederholende Rechengänge werden durch Rechenpläne, das sind Lochstreifen, die von den Rechengeralten abgetastet werden, selbsttätig gesteuert. Ein Speicherwerk dient zur Aufnahme von Angaben, die errechnet und zu einem späteren Zeitpunkt wieder benötigt werden; die Abgabe der Werte an das Speicherwerk und ihre Entnahme aus demselben wird ebenfalls durch die Rechenpläne selbsttätig gesteuert. Die Geräte, die schon geliefert oder im Bau sind, arbeiten mit Rechenplänen, die vom Menschen aufgestellt werden. Die Wege, die zur Entwicklung von Rechengeralten führen, die für ein begrenztes Aufgabengebiet den Rechenplan, der für die Lösung einer Aufgabe benötigt wird, selbst aufstellen und herstellen, sind schon geklärt. Solche Rechengeralten werden nicht nur die Zahlenrechnungen durchführen, sondern auch selbst die Formeln ableiten, nach denen gerechnet werden muss. Damit eröffnet sich in noch viel weiterem Ausmass wie bei den algebraischen Rechengeralten die Möglichkeit, technische und wissenschaftliche Fachkräfte von schematischen Denk- und Rechenarbeiten zu befreien und dadurch deren Arbeitskraft zu vervielfachen.«

1 Konrad Zuse (1937) verwendete des öfteren den Begriff: *Angaben*. In einem Bericht von 1937 beschreibt er sehr detailliert, was er unter Angaben versteht: »Diese können sehr verschiedene Bedeutungen haben, z.B. Zahlen, Aussagen, Namen, Kennziffern, Dienstgrade, Daten, Befehle, Nachrichten, Schlussfolgerungen, usw. Er schreibt dann weiter: Gemeinsam ist allen die Variabilität ihrer Aussage, denn wenn jede Ausgangsangabe nur eine Möglichkeit zuließe, so wäre das Rechnen überflüssig, da ja dann nur noch ein Resultat in Frage käme. So ist z.B. das Vorzeichen einer Zahl zweifach variabel, eine Dezimalzahl 10-fach, ein Buchstabe 26-fach, die Angabe über die Batallionszugehörigkeit dreifach variabel.«

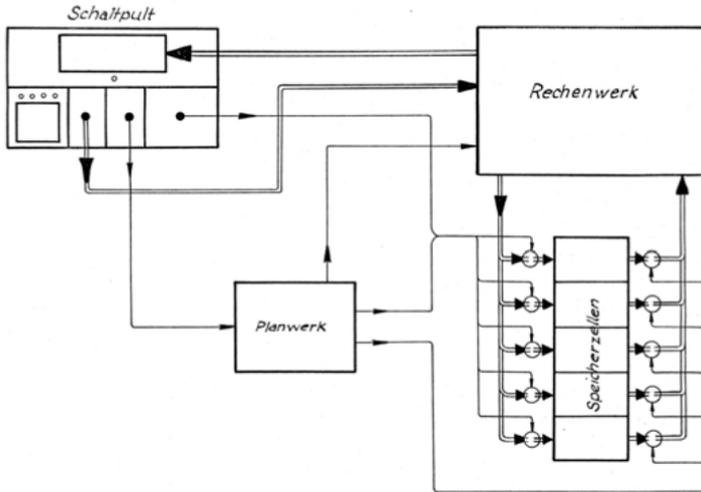
Zuse bezeichnet die Rechenmaschinen Z1-Z4 als algebraische Rechengereäte und führt dazu aus:

»Die algebraischen Geräte sind überall dort mit bester Wirkung einzusetzen, wo die gleichen Rechnungen mit verhältnismäßig großem Zahlenaufwand durchgeführt werden müssen. Die Leistungen der Geräte liegen etwa bei 30 bis 50 Rechenoperationen pro Minute. Es können also z.B. Gleichungssysteme mit einer großen Zahl von Unbekannten in sehr kurzer Zeit numerisch durchgerechnet werden. Aber auch in Fällen, in denen der Rechenplan oft wechselt, sind die Geräte mit Erfolg anwendbar. Die Aufstellung der Rechenpläne erfordert nur etwa die gleiche Zeit wie sonst, die erste Rechnung erfolgt also ebenfalls sehr schnell. Die Geräte sind dabei wesentlich universeller als die bekannten Spezialgeräte anderer Hersteller, etwa zum Lösen von Gleichungssystemen, Determinanten usw., welche zum Teil ausgeführt und zum Teil entworfen sind. Man kann z.B. mit den hier beschriebenen Geräten nicht nur die Gleichungen an sich lösen, sondern auch die gesamte vorbereitende Rechenarbeit, welche zur Bestimmung der Koeffizienten der Gleichung dient, damit bewältigen. Je nach Aufbau des Gleichungssystems können diese Rechnungen bekanntlich das Mehrfache des Rechenaufwandes erfordern, der zur Lösung der eigentlichen Gleichungen erforderlich ist. Ein weiterer Vorteil ist der, dass der einmal aufgestellte Rechenplan beliebig lange in Form von Lochstreifen aufbewahrt werden kann, ferner können zusammenhängende Rechnungen so aufgebaut werden, dass sie aneinander anschliessen und Zahlenübertragungen nicht nötig sind. Nur bei verwickelten Fällen, wie sie in der Praxis häufig vorkommen, ist eine weitgehende Mitarbeit des Menschen erforderlich.«

Er führt dann weiter aus, dass die Leistung des algebraischen Rechengereätes erweitert werden kann auf das Rechnen mit komplexen Größen, dabei rechnen die Rechengereäte selbsttätig komplex weiter, sobald komplexe Zahlen auftreten, ohne dass der Rechenplan geändert werden muss. Trigonometrische, Hyperbolische Funktionen und ähnlichen werden erweitert auf die Bestimmung der Wurzeln algebraischer Gleichungen 3. und höheren Grades. Größen, die durch rechnerisch nicht erfassbare Kurven gegeben sind, können bestimmt werden. Funktionswerte können gespeichert werden und Interpolationen der Zwischenwerte sind möglich. Numerische Integration von Differentialgleichungen sind ebenfalls vorgesehen.

Im Jahr 1943 beschreibt Zuse (1943b) die konstruktiven Einzelheiten der Rechenmaschine Z4. Wegen der Bedeutung der ergonomischen Konzepte der Maschine Z4 zitieren wir hier eine längere Passage:

Abb. 3: Schematische Darstellung der Rechenmaschine Z4
(Konrad Zuse Archiv von Horst Zuse in Berlin).



»Es handelt sich um eine algebraische Rechenmaschine, welche zur Aufgabe hat, längere schematische Rechenabläufe der Zahlenrechnung selbsttätig durchzuführen. Das Gerät besteht aus folgenden Teilen:

A) Tastatur und Anzeige-Vorrichtung zum Eintasten der Ausgangswerte, zum Ablesen der Resultatwerte und zum Herstellen der Rechenpläne. Das Eintasten der Werte erfolgt durch eine sogenannte Zehnertastatur mit 10 Tasten für die Ziffern 0 - 9, dazu kommen die Tasten »«, »Vorzeichen« »imaginär«, »Irrtum«, »fertig«, »0«, »,« »∞«, $\times 10^6$, $\times 10^{-6}$. Die Zahl wird wie bei einer Schreibmaschine einschliesslich des Kommas getastet. Also z.B. - 0,00347 $\times 10^{-6}$. Es können jedoch nur Zifferfolgen von 5 von 0 verschiedenen Ziffern getastet werden. Also z.B. 35 431 000 aber nicht 05 431 026. Die Anzeige der Resultatwerte erfolgt entsprechend so, dass die Folge der von 0 verschiedenen Ziffern angezeigt wird und relativ dazu die Lage des Kommas. Eventuell wird noch ein Potenzfaktor angezeigt. Wir haben ferner die Anzeigen für »0«, »∞«, »unbestimmt«, »imaginär«, »Vorzeichen«. Die Anzeige der Zahlen erfolgt zunächst durch Aufleuchten von Lampen, später sind andere Mittel vorgesehen. Die Kommandotastatur zerfällt in die Kommandotasten für die einzelnen Operationen und in die Kommandotasten für die Speicherzellen. [...]

Mit Hilfe der Speicherwerttasten können die einzelnen Speicherzellen geschaltet werden und zwar sowohl zum Ablesen als auch zum Speichern. Wir haben 16 Gruppen von Speicherzellen zu je 64 Zellen, also im Ganzen 1024 Speicherzellen.«

Es ist hier auffallend in welcher Größenordnung der Speicher für die Maschine V4 (Z4) geplant war. Tatsächlich war die Maschine Z4 bis zum 16. März 1945 (Tag der Flucht aus Berlin) in Berlin nicht funktionsfähig, und der Speicher umfasste nach einem mündlichen Bericht von ihm nur 12 Speicherworte zu je 32 Bits. Er führt dann weiter aus:

»B) Rechenwerk: Dem Rechenwerk fällt die Durchführung der eigentlichen Rechenoperationen zu. Zunächst werden die von der Tastatur kommenden Werte aus dem Dezimalsystem ins Sekundalsystem übersetzt. (Zahlensystem mit der Basis 2, auch Dualsystem oder Dyadik genannt). Dieses hat den Vorteil, dass die Relaischaltungen besonders einfach ausfallen. Die Übersetzung erfolgt vollautomatisch, ebenso die Rückübersetzung der Resultatwerte ins Dezimalsystem. Die bedienende Person merkt von diesem Vorgang überhaupt nichts. Die möglichen Rechenoperationen sind oben (Kommando-Tastatur) aufgeführt.

Das Rechenwerk hat 2 Bereitschaftsspeicher (Autor: Register mit 32 Bits), auf denen die beiden Operanden zunächst gespeichert werden. Das Resultat kann auf einem dieser Speicher abgesetzt werden, bzw. gleich ins Speicherwerk gegeben werden. Das Rechenwerk arbeitet mit einer Genauigkeit von etwa 7 - 8 Dezimalstellen.«

Die Genauigkeit von 7-8 Dezimalstellen ist aus unserer Sicht etwas zu hoch angesetzt. Die Wortlänge betrug 32 Bits, davon waren 1 Bit für das Vorzeichen der Mantisse und sieben Bits für den Exponenten reserviert. Es verbleiben also 24 Bits für die Mantisse. Für die Darstellung einer Dezimalziffer werden vier Bits benötigt, d.h. die Genauigkeit liegt bei sechs Dezimalziffern.

»Die Größenordnung der Zahlen kann sich jedoch zwischen 10^{+20} und 10^{-20} bewegen. Zu diesem Zweck werden die Zahlen in der Form $y = 2a \times b$ verschlüsselt, wobei a ganzzahlig ist und b zwischen 1 - 2 liegt. Die Größenordnung der Zahlen wird durch die Maschine selbsttätig bestimmt. Ebenso erfolgt die Bestimmung des Vorzeichens selbsttätig.

Die Verwendung der halblogarithmischen Form [Autor: heutzutage als Gleitkommazahlen bezeichnet] ist bei technischen Rechnungen angebracht, da hier innerhalb der gleichen Rechnung die Zahlen von sehr verschiedener Größenordnung sein können. Da die Genauigkeit aber nur beschränkt zu sein braucht, so wird selbsttätig nicht jedes Mal der ganze Stellenbereich der Zahlen durch die Rechnung hindurch geschleppt. Besonders vorteilhaft wirkt sich dies auf den Umfang der Speicherwerke aus. Treten Zahlen auf, die die angegebene Größenordnung ($10^{20} < x < 10^{-20}$) überschreiten, so gibt die Maschine den Bereich an, in dem das Resultat liegt, z.B. »Sehr gross« + »sehr gross« =

»sehr gross«; »sehr gross« - »sehr gross« = »unbestimmt«, 0/0 = unbestimmt. Wird dann z.B. das Zeichen »unbestimmt« mit weiteren Zahlen kombiniert ist das Resultat wieder unbestimmt. Auf diese Weise wird vermieden, dass die Maschine Fehlresultate errechnet, wenn sie längere Rechengänge ohne Aufsicht des Menschen macht.

C) Speicherwerk: Das Speicherwerk dient der Annahme der Ausgangs- und Zwischenwerke einer Rechnung. Es besteht aus mehreren Zellen (einige Hundert bis Tausend, je nach Ausbau) und kann je Zelle eine Zahl aufnehmen. (Einschließlich Vorzeichen und Angabe über die Lage des Kommas). Die einzelnen Speicherzellen können durch ein Wählwerk W angerufen werden und es können jederzeit vom Rechenwerk her Zahlen auf jede beliebige Speicherzelle übertragen werden und umgekehrt.

D) Planwerk: Dem Planwerk fällt die Aufgabe der Steuerung der Gesamtanlage durch den Rechenplan zu. Von hier aus werden einerseits die Befehle an das Wählwerk und das Speicherwerk gegeben, um die Operanden einer Operation auf das Rechenwerk zu übertragen bzw. das Resultat einer Operation zu speichern und andererseits Befehl an das Rechenwerk über die auszuführenden Rechenoperationen. Dazu kommen die Befehle, die das Übertragen von Zahlen von der Tastatur auf das Rechenwerk und zum Anzeigen der Resultate bewirken. Die Rechenpläne sind auf Lochstreifen festgehalten und werden durch Abtaster schrittweise abgetastet. Die Speicherzellennummern und Operationsbefehle sind verschlüsselt, so dass jedem einzelnen Befehl eine bestimmte Lochkombination zugeordnet ist.

Zum Planwerk gehört noch eine Vorrichtung zum Herstellen der Lochstreifen [Planfertigungsteil]. Diese besteht in der Hauptsache aus einem Locher, durch welchen die Papier- bzw. Filmstreifen mit den erforderlichen Lochkombinationen versehen werden. Die Steuerung erfolgt durch die Kommandotastatur [Siehe unten die ausführliche Beschreibung des Planfertigungsteils].

E) Arbeitsweise der Maschine: Beim Arbeiten der Maschine hat man folgende Möglichkeiten. Arbeiten ohne Rechenplan. In diesem Falle werden sämtliche Kommandos von Hand am Kommandopult eingetastet. Man tastet nacheinander die Ausgangswerte in die Maschine und gibt nach jeder Eintastung das Speicherkommando, welches angibt, auf welcher Zelle die Zahl gespeichert werden soll. Darauf gibt man die Kommandos für die einzelnen Operationen.

F) Herstellung des Rechenplanes: Soll nur der Rechenplan hergestellt werden, so werden die Kommandos entsprechend A) gegeben, jedoch ohne Eintastung der Ausgangswerte. Die Herstellung des Rechenplanes erfolgt in diesem Falle

also noch schneller als entsprechend E). Allerdings muss eine Kontrolle des Rechenplanes mit einem einfachen Zahlenbeispiel erfolgen.

G) Arbeiten mit Rechenplan: Beim Arbeiten mit Rechenplan braucht nur der Rechenplan in den Abtaster eingesetzt zu werden und die Ausgangswerte in der richtigen Reihenfolge eingetastet zu werden. Der weitere Ablauf ist dann voll automatisch. Zur Erleichterung des Eintastens haben wir ein Lampenfeld, auf welches das Zahlenprotokoll für die betreffende Durchrechnung des Rechenplanes aufgelegt wird. In den Rechenplan lassen sich nun Lampenfeldschaltbefehle einfügen, welche die Lampen so schalten, dass jeweils die zu tastende Zahl aufleuchtet. Die Lampenfeldschaltbefehle werden bei der Herstellung des Rechenplanes durch Tasten gegeben. Durch dieses Lampenfeld werden falsche Tastungen weitgehend vermieden, da Irrtümer über die zu tastende Zahl nicht möglich sind.

Resultatwerte der einen Rechnung als Ausgangswert, der nächste in einer Gruppe zusammengehöriger Rechnungen können die einzelnen Rechenpläne so aufgebaut werden, dass es lediglich zu Kontrollzwecken nötig ist, einzelne Zahlen herauszunehmen und anzuzeigen. Im übrigen können die Resultatwerte der einen Rechnung, welche als Ausgangswert der nächsten dienen, sofort im Speicherwerk verbleiben, ohne dass eine Übertragung durch den Menschen erforderlich ist. Auf diese Weise werden zahlreiche Fehler vermieden, welche durch das fehlerhafte Arbeiten des Menschen eintreten könnten.« (Zuse, Konrad, 1943b).

Für die Auslieferung der Z4 an die ETH-Zürich, die am 12. Juli 1950 erfolgte, wurde die Eingabe der Zahlen verändert, und zwar dahingehend, dass diese direkt auf dem Lochstreifen kodiert werden konnten. Das Planfertigungsteil, Konrad Zuses Konzept von 1942, erlaubte den Mathematikern an der ETH-Zürich die Handhabung der Z4 in ca. drei Stunden zu erlernen. Wir betrachten daher das Planfertigungsteil der Z4 im Detail.

2.3 Planfertigungsteil der Z4

Oben wurde bereits das Erstellen der Rechenpläne an der Maschine Z4 erläutert. Dieses Gerät zum Erstellen der Rechenpläne bezeichnete Zuse als Planfertigungsteil. Aus dem Planfertigungsteil entsteht im Jahr 1943 die Konzeption für das geplante Planfertigungsgerät, welches aber nie mehr realisiert wurde.

Die Maschine Z4 war bis 1945 in Berlin nicht betriebsbereit. Statt der acht geplanten Lochstreifenleser konnten bis 1945 nur zwei Exemplare eingebaut werden. Der mechanische Speicher hatte 1945 nur 12 Worte

Speicherkapazität zu je 32 Bit. Erstmals einwandfrei arbeitete die Z4 im April 1945 in Göttingen (Flucht aus Berlin am 16. März 1945 mit Ziel Göttingen).

Abb. 4: Abschiedsrede von Konrad Zuse im Jahr 1950 kurz vor der Auslieferung der Maschine an die ETH-Zürich (Konrad Zuse Archiv von Horst Zuse in Berlin).

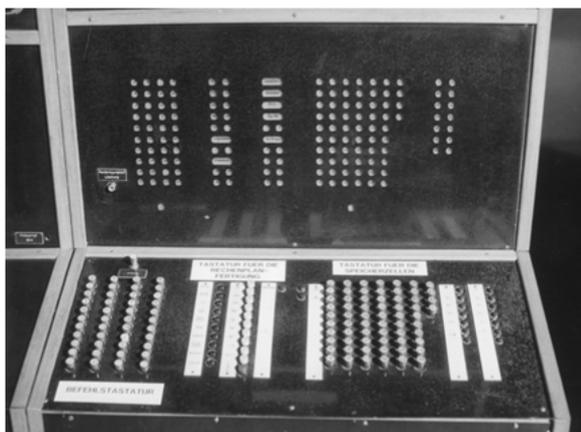


Im Juli 1950 wurde die Maschine Z4 an die ETH-Zürich für fünf Jahre vermietet, nachdem zuvor die Kapazität auf 64 Speicherstellen erhöht worden war. Vor Abschluss des Vertrages wurde sie im Jahr 1949 von Wissenschaftlern der ETH-Zürich in Hopferau einem ausführlichen Test unterzogen. Wir zitieren hier auszugsweise aus dem Abschlußbericht der Herren Böhm und Laett (1981):

»Im folgenden gebe ich einen Bericht über den Vergleich der Z4 mit dem amerikanischen Geräten im Jahr 1949 wieder. Unten ist von einem besonderen Gerät zur Erstellung des Befehlsstreifens [gemeint ist das Programmfertigungsteil] die Rede. Dieses Gerät ist geeignet, Programme sehr einfach und effizient zu erstellen: Vorteile: Ein hervorragender Vorteil der Z4 ist das besondere Gerät zur Erstellung der Befehlsstreifen. In Amerika werden solche ›coding-machines‹ erst geplant. Dies gestattet eine denkbar einfache Herstellung des Streifens (Programmstreifens), die nicht die Kenntnis eines besonderen Code erfordert. Demgegenüber brauchen die amerikanischen Rechen-Institute speziell ausgebildetes Chiffrierpersonal. Irrtümlich gegebene Befehle werden von der Z4 teilweise automatisch aufgedeckt, und der Befehlsstreifen kann dann leicht korrigiert werden. Die Möglichkeit, die Befehle von Hand zu geben, ohne Befehlsstreifen, ist sehr bequem für Rechnungen, die nur einmal ausgeführt werden müssen, und für die Durchführung von Prüfrechnungen. Die

oben angeführte Hilfseinrichtung zur Erstellung von Programmen ist sehr übersichtlich. Es können z.B. die Elemente einer Matrix in der richtigen quadratischen Anordnung eingegeben werden. Die Maschine Z4 arbeitet etwa 100 mal langsamer als eine betriebsichere amerikanische Maschine. Dieser Nachteil fällt aber nur ins Gewicht, wenn sehr viele Rechnungen mit demselben Streifen nacheinander ausgeführt werden. In allen anderen Fällen ist die Herstellung des Streifens sowieso die zeitraubende Arbeit.«

*Abb. 5: Planfertigungsteil der Maschine Z4
(Konrad Zuse Archiv von Horst Zuse in Berlin).*

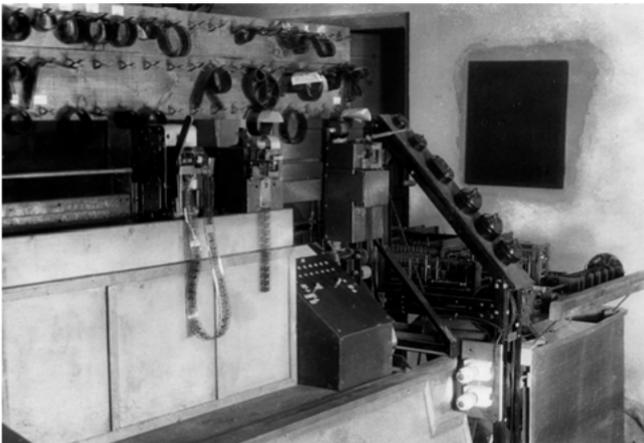


Wie schon ausgeführt wurde, konnten mit dem Planfertigungsteil auf einfache Weise Programme über symbolische Adressen und Symbole für arithmetische Operationen anfertigt werden. Mit Hilfe dieses Gerätes war es möglich, die Programmierung der Z4 (29 Rechenplanbefehle) in weniger als drei Stunden zu erlernen. Auch war es möglich, Programme zu korrigieren bzw. zu kopieren.

Die linke vertikale Reihe von Tasten dient der Ausführung von Rechenoperationen, die mittlere Reihe zur Rechenplanfertigung und die rechte Reihe zur Anwahl der Speicherstellen. Ambros Speiser (2003), der selbst an der Z4 arbeitete, schreibt über die Programmierbarkeit der Z4: »Die Programmierung erfolgte an einem Schaltpult, das überaus bedienungsfreundlich ausgestaltet war; es war klar, dass Zuse im Betrieb solcher Maschinen über große Erfahrung verfügte. Erst mit der MARK III (1948) von Howard Aiken wurde ein Gerät ähnlich dem Planfertigungsteil verwendet.« Ähnlich positiv bewertete der Mathematiker Eduard Stiefel (1954), der Leiter des Institutes, die Erfahrungen bei der Nutzung der Z4 an der ETH-Zürich:

»In der Schweiz hat das Institut für angewandte Mathematik der Eidgenössischen Technischen Hochschule seit 1950 seine wichtigste Aufgabe darin gesehen, das programmgesteuerte Rechnen für die schweizerische Industrie und Technik dienstbar zu machen. Sie [die Programmierung] ist natürlich nur dann durchzuführen wenn die Organisation des Automaten so klar und einfach ist, dass das Programmieren für den von außen kommenden Ingenieur nicht als eine komplizierte Meta-Mathematik erscheint, die er ablehnt und zwar mit Recht. Glücklicherweise erfüllte der von der Firma Zuse im Jahre 1950 gelieferte Relais-Automat [die Z4] diesen Wunsch im beträchtlichen Ausmaß, wie überhaupt Zuse seine Rechengерäte im Hinblick auf die Anwendung in der täglichen Praxis des Ingenieurs konzipiert hat, und sich nicht nur darauf beschränkt hat, eine Einrichtung zu bauen, die schnell addiert und multipliziert, aber sonst nicht viel mehr leistet. Die Rechenautomaten haben uns das numerische Rechnen abgenommen, uns aber dafür die noch langweiligere Arbeit des Programmierens gebracht. Meine Erfahrungen, die ich mit dem Zuse-Automaten gemacht habe, zeigen eindeutig, dass die Vereinfachung der letzteren Tätigkeit von großer Wichtigkeit ist, wenn die Automaten in Industrie und Technik Eingang finden sollen. Ein erster Schritt auf diesem Weg wäre, dass die verschiedenen Rechen-Institute über die Terminologie des Programmierens auf internationaler Basis entsprechende Vereinbarungen treffen.«

Abb. 6: Programme (Rechenpläne nach Konrad Zuse) für die Z4 waren an Nägeln in einem ehemaligen Mehllager in Hopferau (1949) aufgehängt. Es handelte sich um Testprogramme für die Z4, z.B. einen Speichertest, aber auch um die Berechnung von Determinanten und dem Runge-Kutta Verfahren (Konrad Zuse Archiv von Horst Zuse in Berlin).

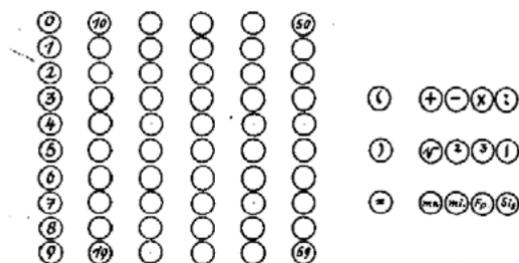


2.4 Planfertigungsgeräte

Konrad Zuse (1943a, 1943b) schrieb in einem Bericht über Planfertigungsgeräte. »Aufgabe dieser Geräte war es, die Herstellung der Rechenpläne für das algebraischen Rechengesetz Z4 wesentlich zu erleichtern. Der Ausbau der Geräte sollte in mehreren Stufen erfolgen: Eintastung einer Formel an einer Tastatur in der üblichen mathematischen Schreibweise, Ableitung von Rechenplänen spezieller Struktur aus Rechenplänen allgemeiner Struktur (Einschmelzen von Rechenplänen), und Zusammensetzen von Rechenplänen aus Teilplänen und Unterplänen (Montageverfahren)«.

Zuse hatte die Absicht, ein Planfertigungsgerät bauen, bei dem es lediglich nötig ist, die Formel so in die Maschine einzutasten, wie sie geschrieben wird. Die Tastatur dazu sollte wie folgt aussehen:

Abb. 7: Tastatur des Planfertigungsgerätes
(Konrad Zuse Archiv von Horst Zuse in Berlin)



»Links haben wir die Tasten für die Variablen. Rechts haben wir die sonstigen Zeichen einschließlich hoch 2 und hoch 9, Absolutstrich und die Hilfsoperationen wie Min und Max. Verschiedene Klammerzeichen sind nicht nötig, da die Kennzeichnung einer Formel mit nur einer Klammerart bereits eindeutig ist. Bruchstriche werden mit Hilfe der Klammerzeichen und das Divisionszeichen dargestellt

Die Maschine kontrolliert dann selbsttätig, ob eine sinnvolle Zeichenfolge getastet wird. Ausgelöst durch diese Befehlsfolge wird der Rechenplan selbsttätig entwickelt und auf ein Lochgerät übertragen. Zur Kontrolle des fertigen Rechenplans kann dann das Planfertigungsgerät mit einem Abtaster versehen werden und die vorliegende Befehlsfolge wieder rückwärts in eine Formel zurückgebildet werden. Zu diesem Zweck wird das Gerät mit einem Anzeigefeld versehen, an dem Lampen in derselben Anordnung angebracht sind wie die Tasten der Tastatur.« (Zuse 1943).

Konrad Zuses Vision lief darauf hinaus, ein Planfertigungsgerät mit einem Montagverfahren zu konstruieren, bei dem mehrere Rechenpläne aus einem Vorrat bereits vorhandener Rechenpläne zusammengesetzt werden. Im Jahr 1986 zog er Bilanz über sein Konzept der Planfertigungsgeräte:

»Die Idee der Planfertigungsgeräte ist typisch für den Stand der Technik um 1940. Die zur Verfügung stehende Speicherkapazität war gering (geplant waren mechanische Speicher mit 1000 Worten). Dagegen war die Schaltungsmathematik für Relaischaltungen bereits völlig ausgebaut und die Idee des allgemeinen Rechnens mit Bedingungen usw. völlig entwickelt. Aus dieser Situation erschien die Idee eines gesonderten Planfertigungsgerätes damals eine günstige Lösung. Heute werden für diese Aufgaben keine besonderen Geräte benötigt. Jedoch war die heutige Architektur der Geräte erst sinnvoll, als Speicher mit mindestens 8000 Worten Kapazität zur Verfügung standen.« (Zuse 1986)

3 Die Bedienung anderer Pioniermaschinen

Im Folgenden soll nun ein Überblick über die wichtigsten frühen funktionsfähigen Rechenautomaten bis zum Jahr 1945 gegeben und deren Ergonomie bei Bedienung und Programmierung betrachtet werden.

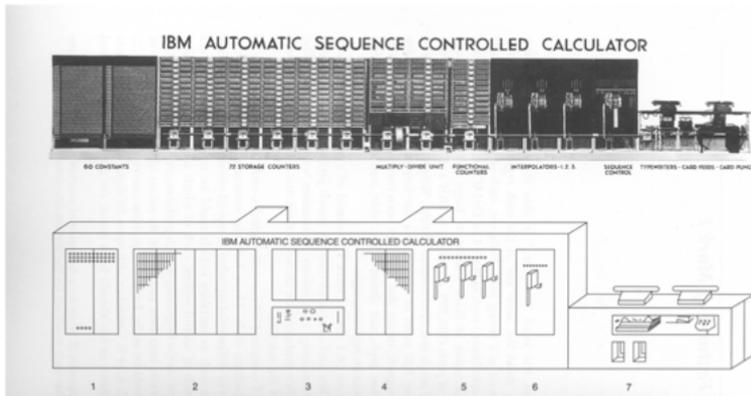
Im Jahr 1939 entwarf George Stibitz (1904-1994) bei den Bell-Laboratories (Bell-Labs) in New York den Drei-Excess-(Stibitz) Code und baute als erstes einen *Complex-Number-Calculator*, der nicht programmierbar war und daher als ein Spezialrechner zum Multiplizieren und Dividieren von komplexen Zahlen angesehen werden muss. Es folgten die Modelle II bis V, die von 1943-1947 für die Landesverteidigung der USA entwickelt wurden. BELL V arbeitete in der binären Gleitkommandarstellung. Stibitz ist damit - wie Konrad Zuse schon 1936 - ebenfalls ein Pionier der Gleitkommarechnung im Binärsystem. Das Besondere an ihm war, dass er der erste »Remote« betriebene Spezialrechner war: »One of the greatest claims to fame the Bell Complex Number Calculator is the fact that it was not only the first machine to service more than one terminal, but that it was also the first machine to be used from a remote location.« (Williams 1985, S. 228)

Der um 1942 von Vincent Atanasoff gebaute Rechner *ABC* (Atanasoff-Berry-Computer) war ein nicht programmierbarer Spezialrechner in Röhrentechnik. Er basierte auf dem Binärprinzip, arbeitete mit einer einfachen Integerrechnung und für die Gaußsche Elimination ausgelegt (Lösung von linearen Gleichungssystemen). Der ABC kann als Prototyp des Parallelrechners angesehen werden. Die Zahlen wurden zwar über

Lochkarten eingelesen, aber die gesamte Steuerung erfolgte über Tasten und Schalter.

Am 7. August 1944 vollendete die Firma IBM zusammen mit Howard Aiken die *MARK I* (Automatic Sequence Controlled Calculator), die ein dezimales Rechenwerk verwendete und die Trennung von Speicher, Steuereinheit und Rechenwerk nicht kannte. Die *MARK I* war wie die *Z*-Maschinen frei programmierbar ².

*Abb. 8: Schematische Darstellung der MARK I.
(Entnommen aus Cohen, 1999).*



Die Maschine *MARK I* bestand aus 765299 Teilen, so berichten die offiziellen IBM Unterlagen. Es waren 3300 Telefonrelais installiert, dazu kamen 2200 Zahnräder. Die Maschine wog 9445 amerikanische Pfund und es waren 530 Meilen Kabel verlegt. Es war eine der größten und komplexesten Maschinen, die zu dieser Zeit jemals gebaut wurden. Wir beschreiben nun die Funktion der einzelnen Gestelle (»Panels«).

Gestell 1, bestehend aus zwei Teilen, enthielt sechzig Konstantenregister. Jede Reihe besteht aus 24 Drehschaltern mit je zehn Ziffernpositionen. Diese Schalter definierten die Konstanten, die in irgendeiner algebraischen oder Differentialgleichung verwendet wurden. Zu Beginn einer jeden Rechnung gab ein Operator per Hand die Konstantenwerte ein.

- 2 Unter freier Programmierbarkeit wird die Fähigkeit eines Rechners verstanden, eine beliebige (zulässige) Folge von Befehlen durch das Steuerwerk zu interpretieren und damit das Rechenwerk, den Speicher oder die Aus- und Eingabeinheiten zu steuern. Die Befehle sollen dabei auf einem Speichermedium vorliegen, welches die Maschine lesen kann. Das Programm muss dazu nicht im Hauptspeicher der Maschine untergebracht sein.

Durch die Drehschalter konnten die Zahlen 0 bis 9 für jede Ziffer eingegeben werden. Jede Zahl enthielt 23 Ziffern und ein Vorzeichen. Jede Reihe einer Zahl von 23 Ziffern plus dem Vorzeichen war über den Lochstreifen adressierbar. Die Programmierer hatten ein »code book«, mit der jede Zahl adressiert werden konnte. Es konnten also 1440 Konstantenziffern eingegeben werden.

Bernhard Cohen (1999, S. 169) berichtet über die dabei auftretenden Probleme: »Grace Hopper told me that very often the 23 digits were more troublesome than helpful. For example, suppose a calculation were being made in which there were only seven significant figures. In performing such a calculation, MARK I would take all calculation out to 23 digits, 14 more than could possibly be useful – a tremendous waste of machine time.« Der Grund für 23 Ziffern für eine Zahl ist bis heute nicht geklärt. Cohen (S. 170) erhielt dazu von Howard Aiken die Erklärung, dass der damit Berechnungen in der Himmelsmechanik durchführen wollte. Aber die MARK I wurde niemals für solche Berechnungen eingesetzt, und es ist nicht klar, ob Aiken überhaupt Kenntnisse in Himmelsmechanik hatte.

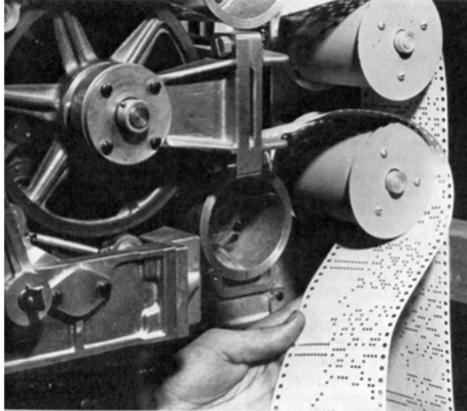
Gestell 2 enthielt 72 Register, die als Akkumulatoren (»Accumulators«) bezeichnet wurden. Jedes von ihnen hatte 24 elektromagnetische Zählräder (»counter wheels«), um eine Zahl mit 23 Ziffern und dem Vorzeichen behandeln zu können. Dieses Gestell mit den Akkumulatoren war der Speicher der MARK I (»store« oder »storage«). Eine Instruktion auf einem Lochstreifen konnte entweder einen Akkumulator oder ein Konstantenregister ansprechen. Wenn schon eine Zahl in dem Akkumulator vorhanden war, dann wurde die neue Zahl dazu addiert. Ein separates Gestell 3 war für die Multiplikation vorgesehen.

Gestell 4 enthielt die funktionalen Einheiten, d.h. die Konvertierung von Zahlen in den Logarithmus zur Basis 10, den Sinus, Cosinus usw. Diese Funktionen waren als eingebaute Funktionen (»built-in subroutines«) implementiert. Die ersten Programmierer der MARK I, Richard Bloch, Robert Campell und Grace Hopper, reduzierten jedes mathematische Problem in eine Sequenz von Befehlen. Grace Hopper berichtete später, dass Sie private Bibliotheken von Programmen hatte. Jahre später wurde solche Bibliotheken als Unterprogramm-Bibliotheken (»libraries of subroutines«) bezeichnet.

Gestell 5 bestand aus drei Lochstreifenlesern, die die Daten für die Interpolation von Funktionen bereitstellten. In Gestell 6 waren die sequentiellen Lochstreifenleser, die das Programm für die MARK I enthielten, welches Schritt für Schritt ausgeführt wurde. Rechts, am Ende der MARK I waren zwei Lochkartenstanzer und Schreibmaschinen (Typewriter) installiert, die die Ergebnisse in einer tabellarischen Form ausgaben. Numerische Eingaben wurden über Drehschalter eingegeben. Die Maschine

MARK I arbeitete mit Dezimalzahlen und einem festen Komma, sie konnte noch keinen bedingten Sprung, dieser wurde erst später hinzugefügt.

Abb. 9: Lochstreifen der Mark I (aus de Beauclair 1968, S. 70).

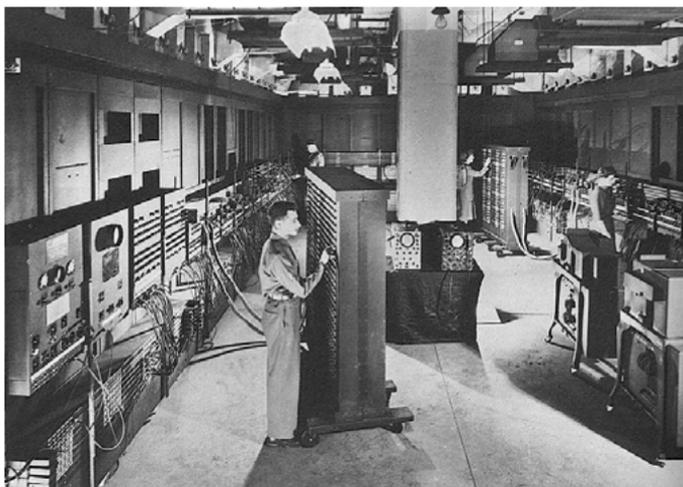


Der am 14. Februar 1946 der Öffentlichkeit vorgestellte *Electronic Numerical Integrator and Computer* (ENIAC) wurde von J.Presper Eckert und John W. Mauchly konzipiert. Die ENIAC mit ihren 17468 Elektronenröhren war nicht frei programmierbar. Es war eine hardwaremäßig konfigurierbare Maschine und arbeitete ebenfalls mit einem Dezimalrechenwerk. Die Steuerung der Maschine (Konfigurierung) wurde durch das Setzen von Hunderten von Drehschaltern und dem Stecken von Kabelverbindungen erreicht. Bis zur Abschaltung diente die Maschine der Berechnung ballistischer Tabellen.

Der ENIAC bestand aus 40 parallel arbeitenden Komponenten, von denen jede 60cm breit, 270cm hoch und 70cm tief war. Die komplette Anlage war in U-Form aufgebaut, beanspruchte eine Fläche von 10x17m und wog 27t. Der Stromverbrauch der 17.468 Elektronenröhren, 7.200 Dioden, 1.500 Relais, 70.000 Widerstände und 10.000 Kondensatoren lag bei 174kW. Der Bau des ENIAC kostete 468.000 \$, ein Betrag, der nur aufgrund des hohen Bedarfs an Rechenleistung seitens der US-Armee zur Verfügung stand.

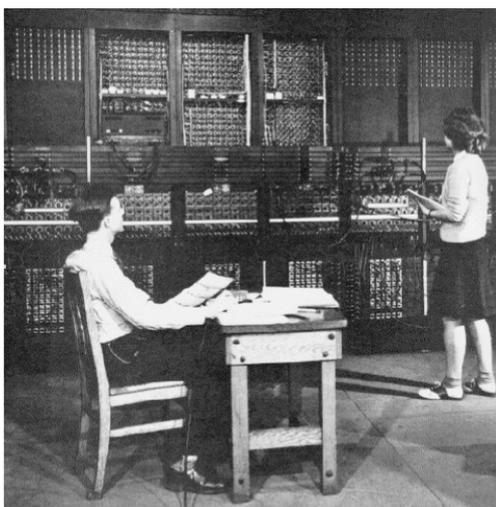
Der ENIAC konnte addieren, subtrahieren, multiplizieren, dividieren und Quadratwurzeln ziehen. Er wurde programmiert, indem man die einzelnen Komponenten mit Kabeln verband und die gewünschten Operationen auf Drehschaltern einstellte. Eine Addition/Subtraktion brauchte 0,2 Millisekunden, eine Multiplikation bis zu 2,8 ms, eine Division bis zu 24 ms und eine Quadratwurzel mehr als 300 ms.

Abb. 10: Der ENIAC (Konrad Zuse Archiv von Horst Zuse in Berlin).



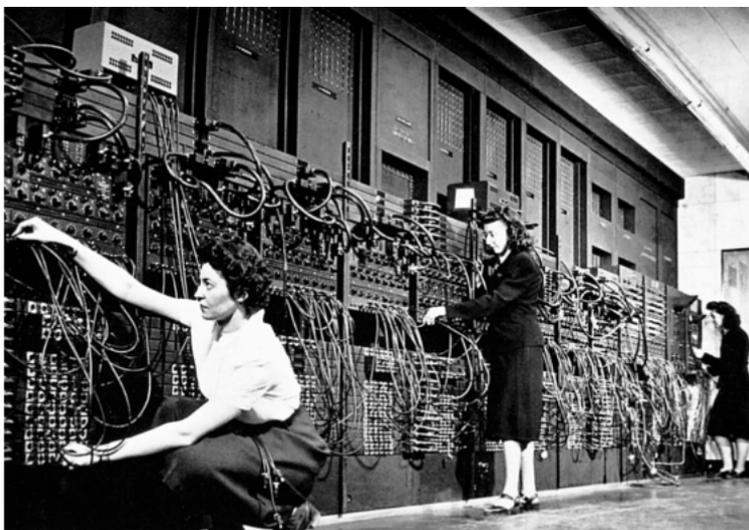
Der ENIAC wurde weitgehend von Frauen konfiguriert, den »Girls of the ENIAC«, die zuvor an der Heimatfront ballistische Berechnungen an mechanischen Tischrechnern durchgeführt hatten. In einem höchst komplizierten mehrstufigen Verfahren mussten für jede Rechenaufgabe die Komponenten des Rechners neu verschaltet werden.

Abb. 11: Konfigurieren des ENIAC nach Anweisungen (National Archives and Records Administration Still Picture Branch, College Park, Maryland).



Ein großes Problem bei der Entwicklung des ENIAC war auch die Fehleranfälligkeit der Elektronenröhren. Wenn nur eine der 17.468 Röhren ausfiel, rechnete die gesamte Maschine fehlerhaft. Um die Kosten dieser unvermeidlichen Ausfälle gering zu halten, wurden in den ENIAC eigens Diagnoseprogramme eingebaut, die das Auffinden einer auszutauschenden Röhre erleichterten. Eine Gegenmaßnahme bestand darin, stärkere Röhren einzubauen, als man eigentlich gebraucht hätte, und diese nur mit etwa 10% ihrer Nennleistung zu betreiben. Außerdem wurde bemerkt, dass mehr Röhren beim Ein- und Ausschalten kaputt gingen als während des laufenden Betriebes. Als Konsequenz ging man dazu über, den ENIAC einfach nicht mehr auszuschalten. Die Ausfallzeit konnte so auf nur wenige Stunden je Woche reduziert werden, ein den Umständen entsprechend hervorragendes Ergebnis.

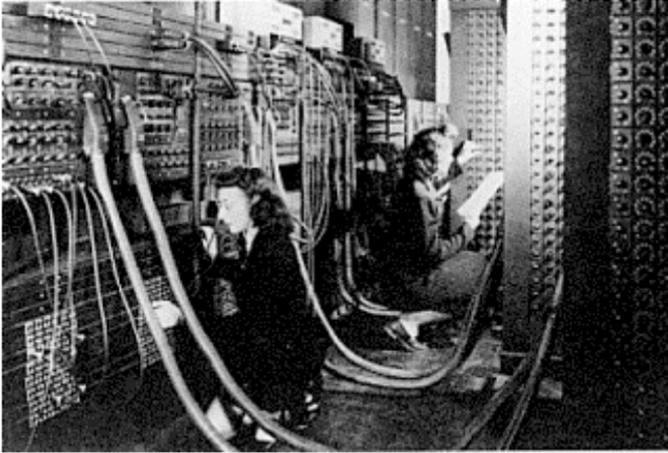
Abb. 12: »ENIAC-Girls« beim Konfigurieren der Maschine (Konrad Zuse Archiv von Horst Zuse in Berlin).



Während Lochstreifen als Programme leicht wieder verwendet werden konnten, war dies mit den Kabeln der ENIAC nicht so einfach möglich. Es wurden ganze Tablettts mit den Kabeln gelagert, die dann bei Bedarf wieder eingesetzt werden konnten. Die ENIAC war eine sehr leistungsfähige Maschine, aber die Verkabelung muss als komplex bezeichnet werden. Eine unmittelbare visuelle Ausgabe war in die Akkumulatoren integriert: Im oberen Bereich der Komponente gab es 102 Neonbirnchen zur

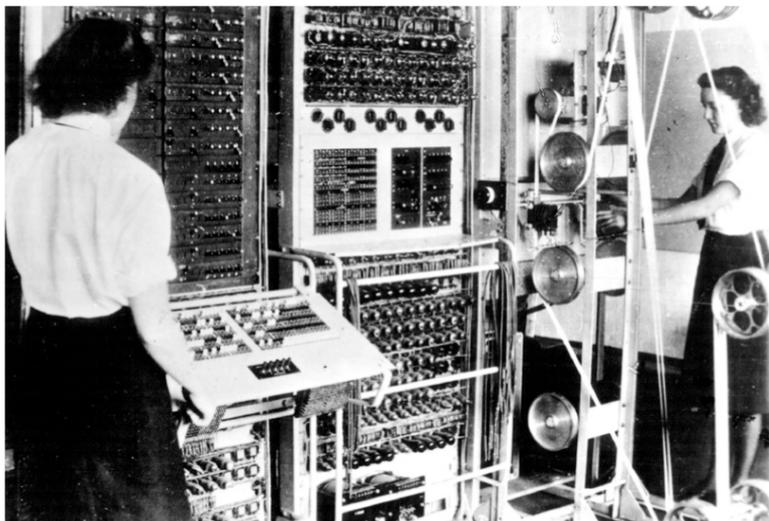
Anzeige der aktuell gespeicherten Zahl (je zehn für jede der zehn Ziffern, zwei für das Vorzeichen).

Abb. 13: »ENIAC-Girls« beim Einstellen des Main Control Panel (National Archives and Records Administration Still Picture Branch, College Park, Maryland).



Die in England von 1943-1945 gebauten zwölf *COLOSSUS*-Rechner waren wiederum Spezialrechner mit Röhren im Binärprinzip, sie waren nicht frei programmierbar und wurden aber erfolgreich zur Entschlüsselung von Funkprüchen der deutschen Wehrmacht (Chiffriermaschine SZ42) eingesetzt. Im Jahr 1946 wurden auf Befehl Churchills fast alle Maschinen und Unterlagen zerstört, aber es existiert heute im Bletchley Park, nördlich von London, ein Nachbau, der in 10-jähriger Arbeit von Tony Sale bewerkstelligt wurde. Interessant ist hier, dass bis 1977 das *COLOSSUS*-Projekt Computerhistorikern vollkommen unbekannt war, da der Britische Geheimdienst erst 1977 die Unterlagen frei gab. Konfiguriert bzw. gesteuert wurden die *COLOSSUS*-Rechner durch die Hardware, d.h. es existierte eine feste Hardwareprogrammierung. Die nicht frei programmierbaren Colossus-Maschinen entsprachen somit dem 1942-1943 von Konrad Zuses gebauten Spezialrechner S1. Dieser war eine reduzierte Z3, deren einziges Programm durch Schrittschalter hardwaremäßig realisiert war.

Abb. 14: Colossus-Rechner mit Einstellwerk
 (nach www.computerhistory.org/timeline/?year=1944)



4 Abschließende Bemerkungen

Die Konstruktion früher Rechenmaschinen war sehr aufwendig und kostenintensiv. Alleine der Complex-Number-Calculator von George Stibitz kostete 20.000 US-Dollar und die Firma Bell überlegte ernsthaft, keine weiteren Maschinen dieser Art zu bauen. Im Jahr 1942 bestellten die Henschel-Flugzeugwerke bei Konrad Zuse die Maschine Z4. Dazu wurde ein Etat von ca. 120.000 Reichsmark vorgesehen, aber die Z4 hat bis 1945 niemals für die Henschel-Flugzeugwerke gerechnet. Es ist daher nicht verwunderlich, dass bei den meisten Maschinen wegen des hohen finanziellen Aufwandes auf die Bedienungsfreundlichkeit wenig geachtet wurde.

Die Rechenmaschine MARK I war speziell konzipiert, um Funktionstabellen zu erstellen. Daher wurden spezielle Konstantenspeicher verwendet. Die Rechenoperationen wurden durch einen Lochstreifen gesteuert, aber die interne Rechnung wurde im Dezimalsystem durchgeführt, und zwar mit 23 Stellen mit einem festen Komma. Für die Berechnungen wurden immer alle 23 Ziffern einbezogen, was zu einer Verschwendung von Maschinenzeit führte, wenn z.B. nur ein Problem mit sechs Ziffern zu berechnen war.

Die ENIAC musste permanent neu konfiguriert werden. Dies war sehr aufwendig. Es zeigte sich schon damals, dass die Rechnungen nur

Sekunden dauerten, aber die Konfigurierung einer ballistischen Berechnung zwei Wochen in Anspruch nehmen konnte. Auch die ENIAC rechnete intern mit Dezimalzahlen, die binär repräsentiert waren, d.h. eine Röhre wurde für jede Dezimalziffer verwendet.

In Konrad Zuses Rechenmaschinen Z3 und Z4 finden sich erwähnenswerte ergonomische Konzepte, die einerseits die Handhabung der Maschinen Z3 und Z4 im Dialogbetrieb unterstützten und andererseits bei der Maschine Z4 die Erstellung von Rechenplänen so einfach wie möglich gestalten sollten. Das Planfertigungsteil der Z4 hat sich im Praxisbetrieb an der ETH-Zürich in den Jahren 1950-55 bei der Erstellung von Rechenplänen bestens bewährt, wie Eduard Stiefel ausführte. Zur Konstruktion des Planfertigungsgerätes von Konrad Zuse ist es wegen des 2. Weltkrieges nicht mehr gekommen.

Der Unterschied in der Zahlendarstellung von Konrad Zuses Maschinen, der MARK I und der ENIAC ist auffallend. Konrad Zuse verwendete in den Maschinen Z1, Z3 und Z4 ausschließlich binäre Gleitkommazahlen. Die Ein- und Ausgabe erfolgte mit dezimalen Gleitkommazahlen, um dem Ingenieur die übliche Arbeitsweise bei umfangreichen Rechnungen zu ermöglichen. Den beiden US-Maschinen MARK I und ENIAC fehlt die Eleganz der Gleitkommazahlen, die sich in heutigen modernen Computern zur Berechnung vielfältiger Probleme bewährt haben. Nur die Maschine Z2, es war eine Maschine zum Testen von Relais, verwendete ein Festkommarechenwerk mit 16 Bits. Konrad Zuses Konzepte zur einfachen und ergonomischen Handhabung seiner »Rechenmaschinen für den Ingenieur« wurden bereits von Zeitgenossen gewürdigt. Speiser und Stiefel führten die Bedienungsfreundlichkeit als Vorzug der Z4 auch noch ins Feld, als diese in Rechengeschwindigkeit und Funktionsumfang mit den amerikanischen Großrechnern bei weitem nicht mehr mithalten konnte.

Literatur

- Alex, J./Mons, W./Flessner, H./Pauli, K./Zuse, H. (2000): »Konrad Zuse - Der Vater des Computers«. Fulda.
- Beauchlair, W. de (1968): »Rechnen mit Maschinen«. Braunschweig, 1968.
- Cohen, B. (1999): »Howard Aiken – Portrait of a computer pioneer«. Cambridge, MA.
- ETH-Zürich (1981): »Konrad Zuse und die Frühzeit des wissenschaftlichen Rechnens an der ETH« - Dokumentation, ETH-Zürich.
- Hellige, H. D. (Hrsg.) (2004): »Geschichten der Informatik. Visionen, Paradigmen, Leitmotive«. Berlin, Heidelberg, New York.

- Rojas, R. (Hrsg.) (1998): »Die Rechenmaschinen von Konrad Zuse. Berlin, Heidelberg, New York.
- Speiser, A. (2003): »Ein Apparat, dicht bepackt mit Hebeln, Blechen, Federn, Stahlstiften. Wie Konrad Zuses Z4-Computer 1950 nach Zürich gelangte«. In: Neue Züricher Zeitung, 16.5.2003.
- Stiefel, E. (1984): »Rechenautomaten im Dienste der Technik. Erfahrungen mit dem Zuse-Rechenautomaten Z4.« In: Arbeitsgemeinschaft für Forschung des Landes Nordrhein-Westfalen 4, 45, S. 29-65.
- Williams, M. (1985): »A History of Computing Technology«. Englewood Cliffs, N. J.
- Zuse, H. (2007a): »Die Rechenmaschine Z3«. In: »Konrad Zuse«. Ernst Freiberger Stiftung.
- Zuse, H. (2007b): »Konrad Zuse Multimedia Show 2007«. Erhältlich bei: www.zuse.de
- Zuse, K. (1936): »Die Rechenmaschine des Ingenieurs«. Interner Bericht 1936. (Konrad Zuse Archiv von Horst Zuse in Berlin-10719, Schaperstr. 21, und »Zuse-Internet-Archiv« 0234 (<http://www.zib.de/zuse/>), dort auch die folgenden ungedruckten Zuse-Manuskripte).
- Zuse, K. (1936a): »Verfahren zur selbsttätigen Durchführung von Rechnungen mit Hilfe von Rechenmaschinen«. Patentanmeldung Z 23 139, 1936. (»Zuse-Internet-Archiv« 0230)
- Zuse, K. (1936b): »Das Zahlensystem«. Interner Bericht, 1936.
- Zuse, K. (1937): »Aus mechanischen Schaltgliedern aufgebautes Speicherwerk«. Patentschrift Nr. 924 107, 3. Juli 1937. (»Zuse-Internet-Archiv« 0114)
- Zuse, K. (1937a): »Einführung in die allgemeine Dyadik«.
- Zuse, K. (1943): »Planfertigungsgeräte. Bericht der Zuse Apparatebau Berlin (interner Bericht), 1943.
- Zuse, K. (1943a): »Angebot auf Entwicklung und Bau eines Planfertigungsgerätes. Notizen, Interner Bericht, ca. 1943.
- Zuse, K. (1943b): »Rechenplangesteuerte Rechengeräte für technische und wissenschaftliche Rechnungen, Anlage 1: Beschreibung des im Bau befindlichen algebraischen Rechengerätes (V4)« (»Zuse-Internet-Archiv« 0107).
- Zuse, K. (1945): Brief an die Eltern. 11. November 1945 (»Zuse-Internet-Archiv«)

ZEIGEN, ZEICHNEN UND ZEICHEN. DER VERSCHWUNDENE LICHTGRIFFEL

FRIEDER NAKE

*»The shift from algorithms to interaction
should have happened long ago in human-
computer systems [...]«*

Michel Beaudouin-Lafon 2006

An eine alte Geschichte soll hier erinnert werden, an die Geschichte, als die Interaktion von Mensch und Computer in einem engeren Sinne anfing. Der Umgang mit dem Rechner begann, eine gänzlich andere Form anzunehmen, eine Form, die heute nahezu allen Menschen vertraut, und wo nicht eben vertraut, so doch wenigstens bekannt ist. Diese Geschichte nimmt ihren technischen Anfang gegen Ende der 1950er Jahre. Mit einem Paukenschlag tritt sie im Februar 1963 in Erscheinung, den damals zwar nicht allzu viele sogleich hören, der aber wiederhallt bis heute. Sie braucht gut zwanzig Jahre bis 1984, bevor sie Allgemeingut und jedermanns Erfahrung wird. Sie ist heute, als eine Geschichte, schon ein wenig vergessen. Denn sie ist zur Natur, wenn dieses Wort hier erlaubt ist, des menschlichen Umgangs mit dem Computer geworden.

Hier soll nun nicht, wie es der Historiker wohl unternehmen müsste, ein Feuerwerk von Ereignissen, Orten, Personen und Erfolgen entzündet werden. Da ich das Historiker-Handwerk ohnehin nicht gelernt habe, nehme ich mir heraus, nur den Kern dessen darzustellen, was 1963 in Erscheinung trat, und das dann lieber ein wenig hin und her zu wenden. Es ging ums *Zeichnen*, umfassender gesehen aber um die Geometrie.

Computer waren Maschinen zum Rechnen. Das Rechnen hatte sich mit Alan Turing aus einer lange vor ihm wurzelnden Geschichte des Verhältnisses zwischen Denken und Rechnen als eine ungeheuer allgemeine geistige Tätigkeit herausgeschält. Auf sie konnte vieles, alles wohl, zurück geführt werden, das nach Rechnen zu allerletzt roch, das aber genau und immer genauer gefasst wurde, wobei ihm freilich die eine oder andere Feder gerupft wurde. Geistesgrößen des vergangenen Jahrhunderts hatten

dazu den Grund gelegt, so etwa Kurt Gödel, Alan M. Turing, Alonzo Church, John von Neumann.

Die sichtbaren Formen nun, die grafischen Erscheinungen geometrischer Gegenstände und ihre Herstellung durch die händische Tätigkeit des Zeichnens, diese Arbeit stand jetzt an, wo manches mit Computern schon gelungen war. In einem der ersten Anläufe gleich führte ein junger Ingenieur, Ivan E. Sutherland (Abb. 1), das algorithmische Zeichnen zu einem Höhepunkt, dessen Weitungen und Wirkungen noch heute zu spüren sind. Wir können behaupten: Von der Gnade, *zeichnen* zu können, zur Mechanik, *zeigen* zu können, führt von Sutherland her unsere Geschichte. Ihr kultureller Hintergrund ist die Welt der *Zeichen* und die semiotische Art des Denkens. Den Zeichen wenden wir uns deswegen zuerst zu.

Abb. 1: Ivan Sutherland und David Evans, mit dem er 1968 die Computergrafik-Firma Evans & Sutherland gründete (Machover 1999; Bild 1969).



Das voran gestellte Motto von Michel Beaudouin-Lafon bettet unsere ansonsten vielleicht nur dem Bildungsbeflissenen ein Interesse abnötigende Geschichte in die aktuelle Strömung des *Interaction Design* ein. In seinem Umfeld sind Informatikerinnen und Designer¹ tätig, die die Mauern der Algorithmik übersteigen, um weites Feld zu gewinnen, ein Feld, dessen Unabsehbarkeit die akademische Disziplin der Informatik vielleicht auf ein ihr genehmes Maß zurechtstutzen wird.

1 Dem Problem der deutschen Sprache mit den menschlichen Geschlechtern nähere ich mich gemischt, wie im obigen ersten Fall.

Den Sammelband, dem unser Zitat entnommen ist, zielt als letzter ein Beitrag, in dem Lynn Andrea Stein (2006) dazu aufruft, Neuankömmlingen der *Computer Science* ihr Fach in völlig neuer Form anzutragen. Stein greift dazu auf die uralte Kultur der Erzählung zurück, die nebenan, in der postmodernen Philosophie, zur selben Zeit zur Methode erhoben wurde, als die Informatik sich etablierte. Anders, als unser Denken sich ab und an zu Begeisterung aufschwingen mag, geht es beim Erzählen zu. Und auch das hat mit dem ersten Wetterleuchten objektorientierter Programmierung bei unserem Heroen Sutherland 1963 seinen Ausgangspunkt. Gewusst hat er das wohl nicht. Der Lichtgriffel, das elektronische Werkzeug, den Sutherland schwang und von dem unser Beitrag handelt, ist längst verschwunden in rhizomartigen Wurzelgeflechten², die wir nicht mehr durchschauen. Der damals neue Umgang mit den gezeichneten Zeichen, die aus Zeigehandlungen entstehen, aber ist geblieben.

Des Ent- und Verschwindens auf historischen Pfaden sich anzunehmen, empfinde ich als ausgesprochen reizvoll. Es ist doch auf der einen Seite kaum zu denken, dass ein Technisches wirklich endgültig verschwindet, wenn es überhaupt einmal zum Produkt geworden war. Zu genau nimmt es die Zunft der Archivare, Dokumentare, Datensammler und Geschichte-Schreiber mit der jeweiligen Gegenwart, um sie aus allem zu verbannen, was sie aufzeichnet. Eher könnte man das von aus dem Amt gejagten Politikern oder korrupten Managern erwarten.

Andererseits ist es ein Faktum, dass ein junger Mensch, dem zum vierten Geburtstag ein Notebook auf den Geschenktisch gelegt wird, bass erstaunt, wenn er ein Foto sieht, auf dem hohe breite Stahlblechschränke und etliche klein erscheinende Menschen zu sehen sind, und es wird behauptet, dies sei ein Computer. Der leiste aber vielleicht nur ein Millionstel, wenn überhaupt, dessen, was das Notebook könne.

Das im Titel anklingende Verschwinden ist, im Marxschen Sinne, ein moralischer Verschleiß, im Hegelschen Sinne eine Aufhebung. Moralischer Verschleiß: neuere Technik leistet das Gleiche, aber billiger und vielleicht auch überlegener. Aufhebung: dialektisch gesehen wird eines überwunden, indem es der Form nach entschwindet, dem Inhalt nach aber bleibt.

2 Der Begriff des Rhizom spielt in der Welt der (französischen) Postmoderne eine große Rolle (Deleuze & Guattari 1977). Es mag eine ironische Tatsache sein, dass die Ringstrukturen, die das Sutherlandsche Programm Sketchpad schuf, in ihrem Wuchern durchaus als rhizomatisch betrachtet werden könnten.

Vorab: über das Zeichen

»All my notions are too narrow. Instead of
'Signs', ought I not to say Medium?«

Charles S. Peirce 1906³

Oft hat Charles Sanders Peirce über Zeichen geschrieben. Niemand, der nicht in ihm den Begründer der modernen Semiotik sähe, jenes theoretischen Zugangs zum Phänomen der Zeichen, ohne den konzise kaum noch etwas theoretisch zu erfassen wäre, das heute die digitalen Medien umtreibt⁴. Immer wieder hat Peirce Formulierungen dessen geliefert, was das Zeichen sei und was es ausmache⁵. Nicht immer sind seine Formulierungen einfach zu verstehen. Das trifft im Übrigen für einen amerikanischen *native speaker* im Falle des originalen Wortlautes kaum weniger gilt als für den deutschen Leser des übersetzten Textes⁶.

Acht Jahre vor seinem Tod aber fragt Peirce sich anscheinend völlig ernsthaft, ob er auf den ein Leben lang philosophisch gejagten Begriff des *sign* nicht vollständig verzichten und ihn durch *medium* ersetzen solle, wie das vorangestellte Motiv belegt. So zeigt uns der amerikanische Gelehrte seine Begriffsanstrengung als eine nimmer ruhende. Seine Begriffsarbeit mündet nicht in einer schwarz auf weiß auf Papier herumtragbaren endgültigen Definition, sie geht vielmehr fort und fort im lebendigen Geist. Dieser kann auch, will er sich nur auf das beziehen, was ist und wird, gar nicht anders.

Auf den Gedanken, trotz heftiger Arbeit den eigenen Begriff vom Zeichen aufzugeben zugunsten eines ihm ebenso attraktiv erscheinenden »Medium«, konnte Peirce wohl gerade wegen seiner stets verfochtenen Auffassung vom Zeichen verfallen. Diese Auffassung sei hier nicht mit einem Zitat belegt, sondern in den schlichten Worten eines Informatikers, dem seit den frühesten bundesdeutschen Diskussionen um die Einrichtung einer Disziplin Informatik (Ende der 1960er Jahre) nicht aus dem Sinn gehen will, dass eine recht verstandene Informatik auf der Semiotik

3 Diese Aussage Peirce' ist bisher unveröffentlicht. Sie wird als MS 339 in Richard Robin, *Annotated Catalogue of the Papers of Charles S. Peirce*, University of Massachusetts Press 1967, geführt. Mir scheint reizvoll, dass hundert Jahre zwischen den beiden vorangestellten Motti liegen.

4 Zur Einführung sei (Peirce 1993) empfohlen, dazu (Pape 2004).

5 Robert Marty hat 76 Definitionen des Zeichens bei Peirce gefunden und im Netz verfügbar gemacht: www.cspeirce.com/menu/library/rsources/76defs/.

6 Man mag das prüfen an Hand der von Kloesel und Pape besorgten Übersetzungen der semiotischen Schriften (Peirce 2000), die dem deutschen Leser das semiotische Werk von Peirce erschließen.

füßen müsse⁷: nämlich auf Zeichen und ihren Prozessen, ohne die die *Maschinisierung von Kopfarbeit* nicht zu begreifen wäre, um die sich so eifrig die Informatik bemüht.

Das *Zeichen* ist nun zu allererst einmal kein Ding, sondern ein Verhältnis. Draußen in der Welt kommt das Zeichen nicht vor, ohne dass ein Mensch es herstellte. Zeichen sind nicht einfach da. Sie werden gemacht und festgestellt. Vielleicht werden sie durch Feststellung *hergestellt*, wo diese Feststellung sehr oft, ja: meistens, ganz unbemerkt und ohne große Anstrengung vonstatten geht. Semiotische Tiere, die wir nach einem Wort von Felix Hausdorff⁸ sind, leisten diese Art von Prozess so ganz nebenher.

Der kleine Unterschied zwischen dem Herstellen und dem Feststellen scheidet die Dingwelt von der Zeichenwelt, den Stoff vom Verhältnis. Max Bense, der in der westlichen BRD der frühe Rufer nach den Peirceschen Zeichen war, hat das klar gesagt. Seine kleine Semiotik beginnt manifestartig: »Zeichen ist alles, was zum Zeichen erklärt wird und nur was zum Zeichen erklärt wird«. (Bense 1967, S. 9) Jedes beliebige Etwas, fährt Bense fort, kann im Prinzip zum Zeichen gemacht werden. In diesem Vorgang, und nur insofern, verliert ein Objekt, das zum Zeichen gemacht wird, seinen Objektcharakter und gewinnt Relationalität.

*Abb. 2.: Zeichnerischer Strich. Aus R. Bothner:
»Bosheiten und andere Kleinigkeiten« 2001.*



Damit es nicht zu Missverständnissen kommt: der Strich mit dem Kohlestift auf Papier ist und bleibt ein Element der stofflichen Welt (Abb. 2), in seinem Fall in Form einer Abreibung von Kohlepartikeln, die auf dem

7 Ende der 1960er Jahre setzte in der BRD eine Diskussion um die Etablierung einer akademischen Disziplin „Informatik“ ein. An der Universität Stuttgart bildete eine Gruppe wissenschaftlicher Assistenten am Rechenzentrum eine Fraktion dieser Debatte. Dort vertraten wir die Ansicht, dass Datenverarbeitung grundlegend semiotisch zu verstehen sei.

8 Unter seinem Pseudonym Paul Mongré (1897).

Blatt Papier haften und zusammen mit dem Papier eine neue Stofflichkeit eingehen. Es steht zu vermuten, dass die Zeichnerin, die den Stift geführt hat, so dass der Strich so entstand, wie er nun ist, damit einer Absicht, einer Empfindung oder einem Gedanken Ausdruck verleihen wollte. Die Kohlepartikel fungieren für sie als Teil einer Zeichenrelation. Deren erste Manifestation ist das Zusammenfinden, die Assoziation von Empfindung und Kohlestrich in einer Relation.

Das Vertrackte an der relationalen Daseinsweise des Zeichens ist nun, dass das Zeichen trotz seiner Leugnung der Dinghaftigkeit auf Dinge angewiesen ist. Wir haben allen Grund, davon auszugehen, dass Zeichen in Kommunikationsprozessen fungieren. Damit das funktionieren kann, müssen Zeichen zunächst einmal wahrgenommen werden. Das geht nur, wenn Dinghaftes anwesend ist. Denn nur *an* Dingen und durch *Dinge* können wir Zeichen wahrnehmen.

Das Wahrnehmbare am Zeichen nennt Peirce das Repräsentamen; das Bezeichnete nennt er Objekt; und das Bedeutete heißt bei ihm Interpretant.

Das Zeichen etabliert also ein Verhältnis zwischen einem wahrgenommenen Etwas und einer herbeigeführten Interpretation. Das Zeichen fordert mit seinem Auftreten den anwesenden Menschen zu einem Akt der Interpretation geradezu heraus. Das Ergebnis seiner Interpretationsleistung ist eine Bedeutung, die er dem Zeichenereignis zuschreibt. Diese Bedeutung ist völlig individuell und singulär, hier und jetzt in der gegebenen Situation und im Kontext gefunden, entschieden, zugeschrieben. Sie kann im nächsten Augenblick schon eine andere werden, was naturgemäß hieße, dass ein Zeichenprozess stattfinden müsste. So sehr das Zeichen selbst ein statisches Etwas von der Art einer Relation zu sein scheint, so existiert es doch nur in Prozessen. Der Zeichenprozess (die *Semiose*) ist daher der eigentliche Gegenstand der Semiotik.

In der Interpretationsleistung eines Menschen kommt auf undurchschaubare und gewiss nicht kausale Weise die gesamte Kultur zur Wirkung, der er angehört. Jene Kultur nun, als ein Abstraktum, ordnet der wahrnehmbaren Zeichenkomponente auch und schon immer eine Art von Bedeutung zu.⁹ Jene Art von Bedeutung nämlich, die in dieser Kultur einem Zeichenereignis zugemessen wird, ist eine übliche, konventionelle, mehr oder minder verbindliche, per Gesetz oder Lexikon festgeschriebene, in der Schule eingebläute, die sich darauf auswirkt, wie wir Zeichenereignisse gewöhnlich wahrnehmen.

9 »Kultur« ist hier das, was ein Liebespaar, eine Kleingruppe, eine Fangemeinde, eine Partei, den deutschen Sprachraum und was es sonst noch an Gemeinsamkeiten geben mag, zusammenhält.

Je stärker einer in seiner Interpretation mit dem übereinstimmt, was in seiner Kultur üblich ist, je geringer also der Unterschied zwischen (allgemeinem) Objekt und (individuellem) Interpretanten des Zeichens ist, desto wohler mag er sich als Vertreter herrschender Auffassungen oder Lesarten fühlen, umso mehr aber begibt er sich der Mächtigkeit und Kraft der Zeichen, seinen Gedanken, wie das Lied sagt, freien Lauf zu lassen.

Es sei angemerkt, dass Charles W. Morris später den drei Peirceschen Ingredienzien des Zeichens nicht nur eine leicht abweichende Deutung von etwas problematischer Art gegeben hat, sondern dass er auch die Dimensionen der Syntaktik, Semantik und Pragmatik für die gestufte Behandlung der Zeichenaspekte vorgeschlagen hat (Morris 1988). Diese Redeweise ist allgemein üblich geworden.

Untersuchen wir die Zeichen in ihrer Stofflichkeit und Struktur, so betreiben wir Syntaktik. Untersuchen wir sie in ihrer Fähigkeit, allgemein üblich und konventionell zu bezeichnen, so betreiben wir Semantik. Untersuchen wir schließlich Zeichen in ihrer Kapazität, uns zu eigenen Schlüssen zu veranlassen, so betreiben wir Pragmatik. *Bedeutung* erscheint hier als persönliche, *Bezeichnung* als kulturelle Interpretationsleistung. Wir könnten auch von zwei Aspekten der Bedeutung sprechen.

Diese Differenzierung in dem, wofür das Zeichen per Repräsentamen für ein Anderes, Abwesendes steht, unterscheidet tiefgehend die triadische Peircesche von der dyadischen Saussureschen Semiotik (Saussure 2001). Jeder von uns weiß aus Erfahrung, dass er oder sie ein Wort, eine Farbe, einen Klang, eine Geste anders auffassen und verstehen kann als alle Anderen. Diesen wesentlichen Aspekt der Zeichenverwendung erfasst der Interpretant.

Jeder von uns weiß andererseits aber auch erfahrungsgemäß, dass all unser Verstehen auf der Teilhabe am gemeinschaftlichen, gesellschaftlichen Leben, an der Kultur fußt. Umberto Eco's Zeichenbegriff ist deswegen tief kulturell geprägt (Eco 1977).

Ein letzter Hinweis schließlich sei gegeben, der die Wichtigkeit des Zeichenbegriffs für die Informatik anspricht. Der Interpretant im Zeichen – jene Komponente also, die das Zeichen zusammenhält, in der sich der Sinn des Ganzen ausdrückt, wo die mediatisierende (vermittelnde) Funktion wurzelt – der Interpretant nämlich, stellt sich dann, wenn ich ihn explizit machen will, selbst wieder als ein Zeichen heraus. Nur selbst als Zeichen kann er *gefasst* werden!

Zeichen also existiert stets und nur als Prozess, und dieser Prozess ist darüber hinaus rekursiv. Im Peirceschen Zeichen-Begriff zeichnet sich ein Denken ab, das erst um den Computer und die digitalen Medien herum aktuell wird.

Dies ist eine lange Vorbemerkung zu einem Beitrag, der sich ums Zeichnen auf oder mit dem Computer dreht, um die Computergrafik also, um ihr *Interface* und die Interaktion von Programm und Anwender. Doch diese Vorbemerkung ist notwendig. Sie liefert die begriffliche Ebene, auf der allein – meines Erachtens – sinnvoll über den Umgang mit der semiotischen Maschine Computer zu sprechen ist.

Peirce hat sein Leben lang am Begriff vom Zeichen (und dann Medium) gearbeitet, weil er verstehen wollte, wie Kommunikation stattfinden und gelingen mag. An Kommunikation sind heute viele Zeitgenossen mehr denn je professionell interessiert. Die neueren Entwicklungen um die Computertechnik herum, um die *digitalen Mittel* der Kommunikation oder um die Mittel der *digitalen Kommunikation*, liefern Anlässe hierfür. Wir wollen uns mit zwei Tätigkeiten befassen, die hierbei wichtig sind: mit dem Zeichnen und dem Zeigen. Sie waren unter der Kommunikations-Thematik schon Thema des jungen Mannes Ivan Sutherland. Mit ihm befassen wir uns kurz im nächsten Abschnitt.

Ivan E. Sutherland

»Sketchpad. A Man-Machine Graphical Communication System« lautet der Titel der bahnbrechenden Dissertation des knapp 25-jährigen Ivan Sutherland. Mit ihr erschafft er, fast aus dem Nichts, das Gebiet des graphischen *interactive computing*, das für jeden, der heute einen Computer verwendet, völlig selbstverständlich ist. Die Dissertation beginnt mit den einfachen, eingängigen und doch stolzen Worten:

»The Sketchpad system makes it possible for a man and a computer to converse rapidly through the medium of line drawings. Heretofore, most interaction between man and computers has been slowed down by the need to reduce all communication to written statements that can be typed; in the past, we have been writing letters to rather than conferring with our computers.« (Sutherland 1963, S. 329)

Nach seiner Promotion im Januar 1963 geht Sutherland sofort für zwei Jahre zur *Army*. Dort ist er in Computerdingen unterwegs. Als *Lieutenant* wird er entlassen. 1965 wird er als Nachfolger von J.C.R. Licklider Chef einer Agentur, die wichtige Forschungsgelder für wegweisende Projekte zur gezielten, auf wenige Institutionen konzentrierten Förderung der Informationstechnologie vergibt: von ARPA (Advanced Research Projects Agency).

Wie wäre es möglich gewesen, dass der weitsichtige *Lick* (wie Licklider im Freundeskreis und darüber hinaus genannt wurde) das *Greenhorn* Sutherland ausgewählt und auf dem führenden Posten (den er später selbst wieder bekleidete) durchgesetzt hätte, wäre er nicht von dessen Vision überzeugt, ja begeistert gewesen? Eine Erklärung kann allein die große Affinität zwischen der vorausschauenden Erwartung Lickliders¹⁰ und dem von Sutherland geführten Nachweis ihrer praktischen Tauglichkeit liefern.

Sutherlands Vision fußt mit Sicherheit auf Lickliders¹¹ Spekulation: Der pragmatische Ingenieur schreckt vor der prinzipiellen Asymmetrie oder auch Unmöglichkeit einer *Kommunikation* zwischen Mensch und Computer nicht zurück. Ihm geht es ums Machen. Die Sorge um Begriff und Moral lässt er nicht zu. Es ist die hohe Zeit des Kalten Krieges. Die Sowjetunion landet Erfolge, die die politisch-militärischen Eliten der USA zum Zittern bringen.

Mit Sketchpad – einem System, das außer seinem Autor kaum je einen Benutzer (aber viele Bewunderer) im heutigen Sinne des Wortes hatte – hat er der Vision eines völlig neuartigen Umgangs mit dem Computer eine konkrete und funktionierende Gestalt gegeben. Licklider hatte 1962 in San Francisco die wohl früheste Demonstration von Sketchpad gesehen (ein Jahr vor der Dissertation) und war seither – falls wir Waldrops Erzählung (2001, S. 255 f.) folgen dürfen – ein heimlicher Bewunderer Sutherlands. Es mag sein, dass er gegen alle Gepflogenheiten schon da den Plan fasste, Sutherland zu seinem Nachfolger bei ARPA zu machen¹².

Ob in Sutherlands Wort vom »Medium der Linienzeichnung« der Begriff des Mediums nur beiläufig auftaucht oder nicht: Sutherland stößt in der Sache das Tor zur Auffassung vom Computer als Medium auf. Das geschieht zu einer Zeit, als Computer noch durch und durch als *Automaten* gesehen werden und noch nicht einmal in der später so wichtigen *Werkzeug*-Welt angekommen sind. Ich gehe darauf weiter unten genauer ein.

Zu Anfang der 1960er Jahre davon zu sprechen, dass die Interaktion zwischen Mensch und Computer bisher vor allem dadurch verlangsamt worden sei, dass alles auf geschriebene und getippte Aussagen reduziert werden musste, ist nur vor dem Hintergrund zu verstehen, dass Ivan

10 Drei Jahre zuvor hatte Licklider einen bis heute berühmten visionären Aufsatz unter dem Titel »Man-computer symbiosis« verfasst (1960).

11 Beide kannten sich, waren aber noch nicht befreundet. Erstaunlich, dass Sutherland den Namen Lickliders zweimal mit dem gleichen Schreibfehler wiedergibt (»Lickleder«).

12 Der Direktor von ARPA hatte ein Vorschlagsrecht für seinen Nachfolger.

Sutherland am MIT in dieser Hinsicht königlich verwöhnt wird¹³. Ihm steht der am Lincoln Laboratory in Lexington entwickelte und einzig dort existierende Computer TX-2 zur Verfügung. Das ist zu der Zeit eine der technologisch fortgeschrittensten und leistungsstärksten Maschinen. TX-2 wurde aus dem von Wesley Clark, Kenneth Olsen und Benjamin Gurley konstruierten TX-0, einem der ersten voll-transistorisierten experimentellen Rechner entwickelt (im Einsatz ab 1956; McKenzie 1974)¹⁴.

Der TX-2 verfügt über einen Speicher etwa zweimal so groß wie der anderer damaliger Maschinen. Seine Peripherie ist reichhaltig ausgestattet, insbesondere mit Bildschirm und Lichtstift. Bildschirme als Ausgabegeräte an digitalen Computern sind zu der Zeit alles andere als üblich (Abb. 3). Ihre Vorläufer sind runde, nahezu horizontal liegende optische Anzeigen des SAGE Überwachungssystems¹⁵ der USA. Mit ihm sollten feindliche Flugzeug- oder Raketen-Angriffe frühzeitig erkannt werden. Der TX-0 Rechner erhielt 1957 eine Kathodenstrahlröhre mit 512 x 512 adressierbaren Bildpunkten als Ausgabegerät der Größe 7 x 7 inches (17,5 x 17,5 cm). 1958 kam ein Lichtstift für die Eingabe hinzu. Der TX-2 hatte von vornherein beide Geräte an der Peripherie.

Olsen verließ das MIT im Jahr 1957, um die legendäre Kult-Firma Digital Equipment Corporation (DEC) in Maynard, Massachusetts, zu gründen. Deren erste Konstruktion, der Rechner PDP-1 (Programmable Data Processor) nahm die Erfahrungen von TX-0 auf und wurde mit Bildschirm und Lichtstift zum ersten interaktiv benutzbaren Computer, der kommerziell verfügbar war (ab 1959)¹⁶.

13 Im Interview sagt er 1989, bei der Verleihung der Turing Award: »it would have been difficult to do the kind of things that I did with graphics at that time without the very strong facilities that were available at Lincoln Laboratory« (Frenkel 1989: 312).

14 TX-0 steht für »Zeroth Transistorized Computer«. Einen TX-1 hat es nicht gegeben.

15 SAGE steht für »Semi-automatic ground environment«, s. (Ceruzzi 2002, S. 51). Das war ein hochkomplexes System aus vielen technischen und organisatorischen Teilen. Die Bildschirme dienten der Radaranzeige.

16 Ceruzzi (2002, S. 128) betont die Bedeutung der PDP-1 mit den Worten: »[...] it deserves a place in the history of computing for its architectural innovations – innovations that were as profound and long-lasting as those embodied in John von Neumann's 1945 report on the EDVAC.« Die Standard-Ausrüstung mit Lichtstift, rundem Bildschirm und Tastatur ist eines dieser Elemente. 50 Exemplare des Rechners wurden verkauft. Sein Einstiegspreis war \$ 120.000, ein niedriger Preis zu der Zeit.

Abb. 3: Ivan Sutherland beim ›Zeichnen‹ mit Lichtgriffel am TX-2 Computer (Sutherland 1963, S. 11)



Flugzeug- und Autobauer beginnen um 1960 damit, Entwürfe und Konstruktionszeichnungen am Computer herzustellen (so bei Boeing und General Motors). Doch Sutherland kann als *graduate student* allein im Lincoln Laboratory des MIT mit dem TX-2 und seiner reichen Peripherie arbeiten. Was angesichts der heute vorherrschenden Maus kaum vorstellbar ist, ist für ihn alltägliche Praxis: ein so reichhaltiger Umgang mit dem Computer, dass bei ihm der Gedanke an das Tippen von Symbolketten kaum noch aufkommen muss und sich auf das Benennen von Objekten beschränken kann.

Sutherland zeichnet, indem er zeigt. Denn mit dem Lichtstift auf Bildschirm zeichnet es sich nicht allzu gut (dazu unten mehr). Wichtig ist an dieser Stelle der Umstand, dass Sutherland zu einem Zeitpunkt, als die Knappheit der Resource »Computing« spürbar wird, wie im digitalen Schlaraffenland an der Maschine sitzt, hierhin greift, dort etwas einstellt und so die Erfahrung macht, wie unmittelbar Computing sein kann.

Bald nachdem der Student seinem Promotionskomitee (hochrangig besetzt mit Claude E. Shannon, Marvin Minsky, Douglas Ross, Lawrence Roberts und Steve Coons) seine Dissertation vorgelegt hat, erscheint weiter nördlich in Canada Marshall McLuhans Buch *Understanding Media* (McLuhan 1964). Es wird Furore machen. Kurze Zeit später, während seiner Arbeit an der Harvard University ab 1965, befasst Sutherland sich als erster mit technischen Bedingungen der virtuellen Realität: dem

Datensichthelm (*head-mounted display*). Er schreibt den Aufsatz »The ultimate display« (Sutherland 1965), dessen Titel seherisch wirkt.

Für Sutherland als Ingenieur bedeutet der Gedanke an einen kommunikativen Umgang mit Computern, Datenstrukturen und Algorithmen zu entwerfen und zu implementieren, die die *numerische* Maschine zu einer *geometrischen* erweitern. Dieser Schritt mag aus heutiger Sicht trivial und selbstverständlich erscheinen. Er war es zu jener Zeit keineswegs. Weder waren die dafür notwendige Technik (als Hardware/Software-Koppelung), noch die algorithmische Geometrie, noch das allgemeine Denken darauf vorbereitet.

Interaktives Zeichnen macht zunächst aus dem Zeichnen ein Zeigen und entfernt damit den Zeichner von der Zeichnung. Was oberflächlich für den Menschen sichtbar gemacht wird, errechnet das Programm hinter der Szene in solch hoher Geschwindigkeit, dass unsere Sinne die kleine Verzögerung nicht bemerken können, während das Bild berechnet wird. Virilios rasender Stillstand (Virilio 1998) geschieht erstmalig auf dem Monitor. Denn das dort wie still stehend stabil erscheinende Bild ist nichts als in Hochgeschwindigkeit ständig aufgebautes und verschwindendes Licht.

McLuhan als Medien-Soziologe hingegen spekuliert zur gleichen Zeit in griffigen Metaphern und Slogans übers Fernsehen und alle Medien davor. Er entwirft eine Sprechweise von Medium und Botschaft, heiß und kalt, fern und nah, Dorf und Welt, Mensch und Technik, die sofort wie eine Rakete einschlägt. Trotz großer internationaler Popularität aber gerät sie mit seinem Tod bald schon wieder in Vergessenheit, ist heute jedoch erneut ein erster Bezugspunkt, wenn es um das Verständnis digitaler Medien geht. Das ist umso bemerkenswerter, als der Computer (Anfang der 1960er Jahre noch lange nicht das digitale Medium!) McLuhans Thema gar nicht war.

Die Gleichzeitigkeit kann uns im Rückblick über alle Maßen bemerkenswert und erstaunlich erscheinen, ganz als sei sie ein Beleg für die begriffliche und gedankliche Kluft zwischen den technischen und den literarischen Disziplinen, von der kurze Zeit vorher, 1959, Charles P. Snow gesprochen hatte, eine breite internationale Debatte auslösend (Kreuzer 1987). Während auf der einen Seite nach sprachlichen Formeln gesucht wird, um die noch ungeahnten Verwerfungen ausdrücken zu können, die sich in der Explosion der Massenmedien ankündigen, werden auf der anderen Seite mathematische Formeln in Algorithmen gefasst, die im Laufe der kommenden vier Jahrzehnte eine Revolution der Bildlichkeit und der Klangwelten von ungekanntem Ausmaß hervorrufen werden.

Sutherland scheint sich *dieser* Seite der Revolution, zu der er beitrug, in keiner Weise bewusst gewesen zu sein¹⁷.

In diesem Zusammenhang von einer Revolution zu sprechen, scheint aus heutiger Sicht gerechtfertigt zu sein. Im Oktober 2004 wurde am Zentrum für Kunst und Medientechnologie (ZKM) in Karlsruhe eine Ausstellung unter dem Titel *Die algorithmische Revolution* (Abb. 4) eröffnet, die viermal bis Januar 2008 verlängert wurde. Sie handelt vom algorithmischen Prinzip in der Kunst. An ihrem Eingang steht eine funktionstüchtige Zuse Z22. Peter Weibel, der Kurator der Ausstellung, spricht von der *algorithmischen Revolution* als einer, die niemand bemerkt habe, die aber allenthalben höchst wirksam sei.

Abb. 4: Titelblatt der Broschüre zur Ausstellung
»Die algorithmische Revolution«, ZKM 2004



17 Persönliche Mitteilung per Email. Wie das umgekehrt bei McLuhan gewesen ist, weiß ich nicht zu beurteilen. Gefragt werden kann er nicht mehr. Nachforschung in Schriften und Dokumenten müsste einsetzen.

Unser zentrales Thema ist die allgemeine Medialität nicht. Wir betrachten lediglich einen speziellen ihrer Aspekte, jedoch einen, der sich als zentral erwiesen hat: das algorithmische Zeichnen. Wenn er das Gerät Lichtstift (vor allem dessen algorithmische Einbettung) behandelt, macht Sutherland gelegentlich implizit Andeutungen zur medialen Frage, die ihn als eigenständige Frage aber kalt lässt. Er konstruiert eine Software und legt Rechenschaft über deren Struktur und mögliche Verwendung ab.

Zeigen: Lichtgriffel und Maus

Wir hatten gesehen: Das Zeichnen wird zum Zeigen. Vielleicht in kürzester Form sagt das, was *dem* Zeichnen geschieht, das der Maschinisierung unterworfen wird. Wir reden hier vom interaktiven Zeichnen, von einer Art des Zeichnens also, bei der der Mensch statt eines Bleistiftes auf Papier ein zur Interaktion geeignetes Gerät benutzt, um auf einem Bildschirm Linien zu erzeugen. Eine andere Art, ähnliche Ergebnisse zu erreichen, wäre das automatisierte Abtasten (*Scannen*) einer schon vorliegenden Zeichnung. Diese Technik ist hier offenbar nicht gemeint.

Auch wenn das interaktive Zeichnen mit Sketchpad in den Jahren 1961 bis 1963 schwerer gefallen sein mag als vierzig Jahre später mit einem heutigen Zeichenprogramm, so ist die komplexe Operation in ihrem Kern doch mehr oder weniger erhalten geblieben. Der Zeichen-Software muss signalisiert werden, von welcher Art das geometrische Gebilde sein soll, um das es geht (gerade Linie, Kreisbogen, etc.); die so festgelegte Geometrie verlangt dann zu ihrer Ausgestaltung im einzelnen evtl. nach weiteren Parametern; mit deren Werten kann das Zeichnungselement erzeugt werden.

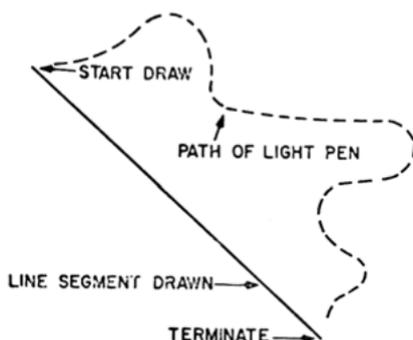
Nehmen wir ein gerades Liniestück als Beispiel. Es verlangt zur eindeutigen Festlegung einen Anfangs- und einen Endpunkt. Die Angabe »gerade Linie« legt die Klasse – und das heißt: die Form – fest, zu der das neue geometrische Objekt gehören soll. Die beiden Endpunkte bestimmen dieses Objekt als ein besonderes aus jener Klasse. Mit der Vorgabe der Klasse (»gerade Linie«) wird der Software implizit mitgeteilt, welcher Art die weiteren Daten sein müssen, die sie benötigt.

Sketchpad kannte hierfür die elegante, intuitiv wirksame Technik des *Rubberbanding* (elastisches Zeichnen). Ein Anfangspunkt wird festgelegt und bleibt fixiert. Der Lichtgriffel – um das Zeichnen mit ihm geht es – wird von diesem Startpunkt aus auf der Oberfläche des Bildschirms entlang einer in der Regel etwas wackeligen Bahn bis zum Endpunkt geführt. (Abb. 5)

Zu jedem Zeitpunkt während des Zeichenvorganges befindet sich der Lichtgriffel an einem bestimmten Ort. Dieser wird als der aktuelle, wenn auch vorläufige Endpunkt interpretiert. Um das sinnfällig zu machen, wird vom (festen) Startpunkt A zum aktuellen Endpunkt Z eine gerade Linie erzeugt. Der Vorgang wirkt so, als sei zwischen dem ersten Punkt und dem aktuell letzten ein elastisches Gummiband straff gespannt. Wird der Vorgang beendet, gilt die zu diesem Zeitpunkt erreichte Strecke als die beabsichtigte. Sie wird in Länge, Richtung und Lage fixiert und in die Zeichnung eingefügt.

Bei dieser Darstellung sind wir in einem Punkt noch nachlässig vorgefahren. Womit werden die Eingaben getätigt? Sketchpad erwartet sie vom *Lichtgriffel*. Mit diesem Gerät aber kann man gar nicht zeichnen, wenn wir unter »Zeichnen« verstehen, dass ein Material (das auch Licht sein mag) auf einen geeigneten Stoff abgerieben wird. Könnten wir mit dem Lichtgriffel in diesem Sinne auf dem Bildschirm zeichnen, so müsste er selbst hinter oder unter sich eine (bleibende) Lichtspur erzeugen. Das tut er jedoch nicht. Er *sendet* nämlich nicht, sondern *empfängt* Licht.

Abb. 5: Das Schema des Rubberbanding beim Zeichnen einer Strecke (Sutherland 1963, S. 14)

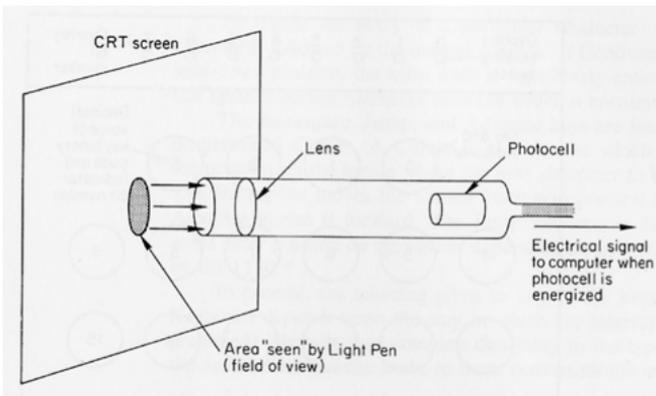


Im Wesentlichen umhüllt der Lichtstift eine Diode (Abb. 6). Sie besitzt ein »Blickfeld«. Wird während des zeilenweisen Bildaufbaus eines jener Pixel zum Leuchten angeregt, das sich im aktuellen Blickfeld des Lichtgriffels befindet, so registriert die Diode das als ein Ereignis. Dieses Ereignis liefert die Bildschirmkoordinaten des betreffenden Pixels. Die Software übernimmt die Koordinaten und sorgt dafür, dass (im Modus *Rubberbanding*) vom festgehaltenen Startpunkt zu diesem aktuellen Punkt eine gerade Linie gezogen wird. All dies muss im Rahmen eines graphischen Grundzyklus geschehen. Das verlangt vom Zeichner, dass er seine Zei-

chengeschwindigkeit bremst. Andernfalls käme die Software nicht schnell genug hinterher. Die elastische Linie bliebe irgendwo stehen, risse ab, während der Zeichner seinen Lichtgriffel längst entfernt hätte.

Das interaktive Zeichnen mit seiner elastisch an der Spitze des Lichtgriffels klebenden geraden Linie fußt also, wie wir sehen, auf einer klug organisierten Ausgabe-Operation. Die aktuell sichtbare gerade Linie ist eine interpretierende Eigenleistung der Software. Sie kommt als Reaktion auf eine Ortserkennung zustande, für die der Lichtgriffel geeignet ist. Nicht der Zeichner zeichnet, die Software zeichnet. Technisch: die Software sorgt für Eintragungen in einer Datenstruktur, die vom bilderzeugenden Prozessor in Sichtbares transformiert wird, dem Zeichner quasi zum Trost.

Abb. 6: Schemazeichnung des Lichtgriffels (Desmonde 1969, S. 64)

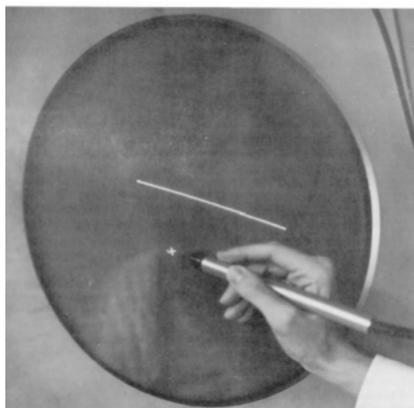


Das ist durchaus wörtlich zu nehmen. Alle Umstände sind so, dass er meinen soll und auch meint, er sei es, der da zeichnet. Er aber zeigt nur noch, wohin gezeichnet werden soll. *Das Zeichnen wird zum Zeigen.* Die mit dem Bleistift noch geschickte Hand verkümmert. Krumm und krakelig kann sie sich über die Glasfläche bewegen, die Software wird's richten. Sehen wir hier nicht überdeutlich, was *Maschinisierung* heißt? Gewinn und Verlust! Der geschickte Zeichner verliert. Der ungeübte gewinnt. Das wirkliche Ergebnis wird so aber nur kulturpessimistisch und saueröpfisch bezeichnet. Wirklich findet eine epochale Transformation auseres körperlichen, performativen Tuns in eine semiotisch-technische Operation statt. Das Zeichnen mit Stift auf Papier war beseelt – so wollen wir uns erlauben zu sagen –, weil die Sensation der von Hand auf dem Zeichenpapier erzeugten Linien, die wir als die Rückenlinie eines kräftigen Jünglings deuten mögen, uns ein erhebendes Empfinden von und Gefallen an eigener

Vollbringung verschafft. Wir sind es, die den Widerstand des Materials brechen, die Stift und Papier gefügig machen, die ihre Eigenbewegung analog, unmittelbar, unvermittelt, aber objektiviert vor uns sehen. Wir haben etwas, haben uns entäußert: wir staunen.

Schwelgen ließe sich so fort und fort. Doch lassen wir dem Gerechtigkeit zukommen, was neu geschieht, wenn das Zeichnen sich aus unserer Hand, unserem Arm, unserer geistig-körperlichen Leiblichkeit davon macht und sein kontinuierlich-ikonisches Wesen in sehr reduzierter Form der Maschine übereignet, die uns umgekehrt erlaubt und auffordert, ihr zu zeigen, wohin sie ihr elastisches Band Trost spendend ziehen soll. Wir werden sogleich sehen, dass und wie wir der Maschine auf die Sprünge verhelfen müssen.

Abb. 7: Bildschirm, Lichtstift, *Tracking Cross* (Prince 1971, S. 45)

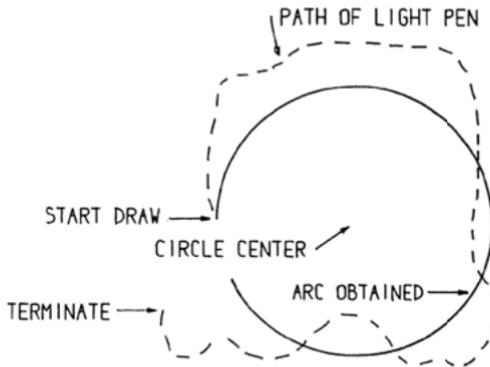


Da der Lichtgriffel Licht nur empfangen, es nicht aber senden kann, muss an seinem jeweils aktuellen Ort Licht vorhanden sein, wenn etwas geschehen soll. Er bliebe sonst blind. Ein kennzeichnendes Zeichenelement muss stets angezeigt werden, sagen wir etwa ein kleines Kreuz, das wir *tracking cross* nennen wollen (Abb. 7). Wird der Griffel im nächsten Augenblick fortbewegt, so darf ihn die Bewegung während eines gewissen Zeitintervalls nicht weiter weg führen, als die Ausdehnung des Kreuzes reicht. Denn andernfalls könnte es nicht mehr im Blickfeld der Diode liegen, von dieser also nicht mehr erkannt werden. Die Software könnte nicht »sagen«, wo sich der Lichtgriffel befindet.

Diese Darlegung zeigt, wie es möglich wird, mit einem Empfangsgerät zu zeichnen, also zu senden. Haben wir das Prinzip verstanden, können wir eine Vorstellung von komplizierteren Zeichenoperationen entwickeln

(z.B. davon, einen Kreisbogen zu fabrizieren, was Sketchpad als eine Variante des Gummiband-Zeichnens einprogrammiert war, Abb. 8). Die Lehre hieraus ist: auf dem total dunklen Bildschirm kann man mit dem Lichtgriffel nichts anfangen, noch nicht einmal zeigen lässt sich dann etwas (was auch klar ist, da ja nichts vorhanden ist).

Abb. 8: Zeichnen eines Kreises mit Sketchpad (Sutherland 1963, S. 14)



Bildschirme standen in den 1960er Jahren ungefähr senkrecht (sehr viel anders ist das im Jahr 2007 auch nicht geworden). Interaktives Zeichnen war ein Zeichnen mit ausgestrecktem Arm. Das kann zu rascher Ermüdung führen, vor allem dann, wenn wir gar nicht zeichnen, sondern immer nur zeigen. Vorn am Lichtgriffel befindet sich in der Regel ein kleiner Knopf, der mit dem ausgestreckten Zeigefinger leicht niedergedrückt werden muss, um das Ende einer Operation zu bestätigen. Der waagrecht gehaltene Griffel kann dabei durch den von oben nach unten gerichteten Druck auf der glatten Glasoberfläche leicht abrutschen, was zu Ungenauigkeiten oder Abbruch der Operation führen kann. Ergonomisch weist der Lichtgriffel also erhebliche Nachteile auf. Sie führten dazu, dass andere Geräte sich durchsetzten, als es um den massenhaft werdenden interaktiven Umgang mit der semiotischen Maschine ging. Zum fast uneingeschränkten Interaktionsgerät wurde bekanntlich die sog. Maus.

Halten wir aber kurz inne, bevor wir die Maus betrachten. Wozu eigentlich ist ein solches Interaktionsgerät nützlich, wofür wird es gebraucht? Welche Art von Daten kann es, muss es liefern? Die Diskussion hierum ist Teil der Anfangsgründe der *Human-Computer Interaction* (HCI) bzw. ihrer speziellen Ausprägung in WIMP-Schnittstellen: *Windows, Icons, Menus, Pointing*. Die drei ersten dieser Bestandteile des seit

langem vorherrschenden Interaktionsstiles stehen für Erscheinungen auf der Oberfläche des Bildschirms. Sie sind Lichterscheinungen bildlicher Art, Ausgaben der laufenden Software.

Pointing aber, die vierte Kennzeichnung, ist von anderer Art: Eingabe mit jenem Gerät, das dem interagierenden Menschen zur Verfügung steht. Radikal und minimalistisch gedacht, lässt sich die Operationsbreite des Menschen auf das Zeigen (*Pointing*) reduzieren, wohingegen der Computer sich reichhaltiger äußern darf: eben in den Formen von Fenstern (*Windows*), Auswahllisten (*Menus*) und Stellvertreterbildchen (*Icons*). Alle Einzelheiten hierzu sind allen ZeitgenossInnen von frühester Kindheit an vertraut, wir brauchen auf sie nicht einzugehen. Bemerkenswert ist aber doch, wie jene gefeierte Form der Interaktion (oder gar Kommunikation) in ihrer technischen Organisation den Menschen reduziert auf das wahrlich Rudimentäre, das Zeigen auf eines der scharenweise vorhandenen Angebote zur Betrachtung, die das laufende Programm liefert. Stumm und verlegen, alle ausschweifenden Gedanken unterdrückend, deutet der Mensch hierhin und dorthin – immer hoffend, dass etwas so geschehe, wie er oder sie es erwartet hat.

Erlaubt jedoch nicht die Tastatur eine breite Palette an Ausdrucksmöglichkeiten, wird eingewandt werden? Gewiss. Leicht wäre auch sie durch Licht zu ersetzen, auf das zu zeigen wäre. Doch das würde arg mühsam. Die Tastatur bleibt Teil der Peripherie. Was mit Tastatur bzw. Zeigergerät eingegeben wird, ist unterschiedlicher Art: symbolisch formulierter Inhalt bzw. deiktischer Umgang mit dem Programm. Wir müssen das Zeigergerät genauer betrachten, das im Zusammenspiel mit den drei anderen Komponenten von WIMP so vorherrschend geworden ist.

Das Zeigergerät weist eine höchst bemerkenswerte Doppexistenz auf. Zunächst existiert es als physikalisches Gerät bestimmter Form, das wir in die Hand nehmen und bewegen. Wir bewegen es, weil wir auf etwas zeigen wollen, das wir als Lichterscheinung auf dem Bildschirm sehen. Haben wir es unserer Absicht entsprechend positioniert, so drücken wir eine Taste am Gerät. Diese einfache Operation hat die Bedeutung von: »da« oder »das da« oder »na, los«, alles einfache, fast stumme Äußerungen, die auch als ein fragendes Hochziehen der Augenbraue oder als leichtes Neigen des Kopfes, als Wippen des Fußes gemacht werden könnten.

Eine Voraussetzung gilt es dabei für das Gelingen der simplen Zeiger-Operation zu beachten. Wohin der Mensch tatsächlich zeigt, davon hätte er ohne Vorsorge des Computers kaum eine Ahnung. Beim Lichtgriffel mag das noch einigermaßen klar sein, weil er in die nächste Nähe des Gegenstandes gebracht werden muss, der gemeint ist. Wenn wir aber das Mausekästchen auf einer horizontalen Tischfläche bewegen, fern des Bildschirms, und damit einen Lichtgegenstand auf dem Monitor meinen,

so ist die dabei geläufige Zuversicht der nie erlahmenden Unterstützung durch den Computer geschuldet. Unser Zeigen nämlich fände im Grunde im Dunkeln statt, führte die Software uns nicht die Hand, indem sie uns zeigt, wohin wir zeigen.

Beim Lichtstift spielt das *tracking cross* diese Rolle. Meist können wir es einigermaßen gut unter dem Zeigegerät erkennen, nicht immer jedoch ist das der Fall. Bei der Maus kommt die Fernsteuerung des *Cursors* auf. Genutzt wird unsere hoch entwickelte Fähigkeit der Hand-Auge Koppelung (vgl. Ceruzzi 2002, S. 261). Bei vielen handwerklichen Tätigkeiten sind wir auf diese feine Abstimmung angewiesen. Im Umgang mit dem WIMP-getriebenen Computer ist uns die Maus völlig zuhanden geworden: wir sehen sie auf dem Bildschirm in Form einer kleinen Licht-Anzeige in mancherlei Formen. Das Zeigegerät existiert als doppeltes: physikalisch in der Hand, semiotisch vor dem Auge.

Immer schon ist das physikalische Zeigegerät von den Gegenständen (Lichtgestalten) entfernt, auf die es zeigen kann und soll. Auch beim Lichtgriffel ist das so. Wenn wir mit ihm auf eine Lichterscheinung auf dem Monitor zeigen wollen, so gelingt das meist auch noch aus einer gewissen Distanz von der Glasoberfläche. Sogar auf der Glasoberfläche selbst aber befindet er sich ein wenig entfernt von den Phosphor-Beschichtungen, die der zeigende Interakteur meint, wenn er zeigt. Je feiner und kleiner der Gegenstand ist, auf den gezeigt werden soll, umso genauer muss das Zeigegerät in Anschlag gebracht werden. Dies ist – neben anderen Gründen – die Geburt des *Cursors* bzw. des *Tracking Cross*. Die Software (der Computer) meldet sich sozusagen mit dem lebenswürdigen Hinweis: »Solltest Du Dein Zeigen jetzt wirklich so meinen, wie Du zeigst, dann würdest Du hier treffen!« Das »Hier« ist dabei der vom *Tracking Cross* angezeigte Ort.

Das Zeigegerät, das die Bandbreite unserer Ausdrucks-Möglichkeiten so stark beschneidet und uns, wie wir meinen, nicht mehr sagen lässt als »da!«, traut uns tatsächlich noch nicht einmal das zu. Es verdoppelt sich in seine *Geräteform* und seine *Anzeigeform*. Nach einer Gewöhnungszeit an die kognitive Distanz, die wir überwinden müssen, ist es uns, als manipulierten wir selbst den Ort der Anzeigeform. Wir reden seit Ben Shneiderman (1983) von der »direkten Manipulation«. Das ist zwar falsch, aber gerechtfertigt.

Tatsächlich erweist sich das Zeigegerät als die eigentliche Schnittfläche, als das Interface. In diesem physikalisch-kognitiven Doppelpack findet durch ständig miteinander verschränkte Hand- und Augenaktivität alles seinen konzentrierten Ausdruck, was den Computer (das laufende Programm) vom Buch, vom Fernseher, ja gar vom Telefon unterscheidet.

Im Augenblick des Bruches der Hand-Auge-Koordination bemerken wir das sofort. Wir gestikulieren dann mit dem physikalischen Zeigegerät herum, drücken wiederholt auf seine Taste, nehmen aber keine Änderung der Cursor-Position wahr. Die unmerklich stattfindende Koppelung ist unterbrochen, das Interface zeigt sein wahres Gesicht.

Der Lichtgriffel wurde wohl 1948 erfunden¹⁸ und spielt in die 1960er Jahre hinein eine Rolle in Laboren der Forschung und Entwicklung. Die Maus geht auf erste Konstruktionen von Engelbart 1963 zurück. Sie taucht bei ihm 1967/68 öffentlich auf. Ihr bis heute anhaltender Siegeszug beginnt mit der Entscheidung der Gruppe am Xerox PARC, interaktive Operationen mit der Maus zu tätigen. Dazwischen liegen empirische Untersuchungen, so von English et al. (1967). Deren Ergebnis zeigt, dass Erfahrene die Maus favorisierten, während Unerfahrenen damals der Lichtstift lieber war. Eindeutig waren die Ergebnisse solcher Empirie nicht. Die faktische Entwicklung aber hat den Lichtstift hinweg gefegt.

Mit ihm ist noch vorsichtig und zurückhaltend ein Prinzip in die Mensch-Computer Interaktion eingeführt worden, ohne das sie nicht funktionierte. Als es etabliert war, ging es rasch daran, seine physikalische Form angenehmer zu gestalten. Diese Form beruht auf der Nutzung der äußerst effektiven Koppelung unseres Sehens und Handwerkens. Sie ermöglicht, dass quasi zwischen die Spitze des Zeichenstifts und die Fläche des Zeichenpapiers der Computer geschoben wird. So etwa drückt Ivan Sutherland den Kern dessen aus, was geschieht, wenn das Zeichnen zum Zeigen wird.¹⁹

Wir reduzieren so die Schnittstelle zwischen Mensch und Programm auf ihr nacktes Minimum. Ganz ähnlich wie Stecker des Stromkabels und Steckdose eine technische Schnittstelle zwischen zwei Systemen bilden, von denen eines gewaltig (der Stromerzeuger, das Stromnetz), das andere fast trivial ist (die Stubenlampe), so verhält es sich auch mit unserem Hintergrundthema, der HCI. Das komplexe System ist hier der Computer mit seinem Betriebssystem, all seiner Software und seinen Dateien. Das geradezu triviale System hingegen ist der Mensch. Über einen solchen Satz mögen wir erschrecken, wir werden die Aussage nicht zulassen. Doch wird der Mensch, damit die Interaktion mit dem Programm funktioniere, auf eine Lächerlichkeit reduziert: auf das Bewegen eines einfachen Gerätes, das Beobachten eines Fadenkreuzes und das Drücken einer Taste. Er tut all

18 In der Form der *light gun*, vgl. Friedewald 1999, S. 103 und oben den Beitrag von Hellige, Kap. 5.

19 »Such a drawing system is very different from ordinary drawing with pencil and paper. Because the computer is placed between the ›point of the pencil‹ and the ›paper‹ it can assist in every step of drawing.« (Sutherland 1966: 95)

das interessegeleitet, ohne Frage. Und mit seinem Interesse bleibt er jedem technischen System meilenweit überlegen, Gottlob.

Wir haben nun das, was werkzeugartig ist, in seiner durch Maschinisierung radikal reduzierten Form gesehen und wenden uns der Frage nach Werkzeug und Medium zu.

Reden: Werkzeug und Medium

Nicht ist hier der Ort einer eingehenden Erörterung dessen, was der Computer sei. Er ist zuallererst der Computer und sonst nichts, und wenn wir bei den nie verstummenden Diskussionen darüber, was er denn nun »wirklich« sei, nicht auf dieser Trivialität als Ausgangspunkt beharren, so verbauen wir uns den Weg zum Verständnis seiner Eigenart.

Einen Gegenstand, der in unserer Welt auftaucht und dort wirkt, wollen wir mit Fug und Recht verstehen, einordnen, vergleichen. All das, was wir interpretierend mit dem Gegenstand auf diese und jene Weise anstellen mögen, entfernt ihn notwendigerweise von sich selbst und untergräbt damit unser Verstehenkönnen. Wir haben andererseits gar keine Wahl, als uns von ihm zu entfernen. Das Beharren auf dem Betrachten des zunächst unverstellten Gegenstandes hat keinen anderen Zweck als den, die notwendigerweise eintretende Flucht in die Metaphysik hinauszuschieben. Haben wir erst einmal damit begonnen, den Gegenstand als einen anderen zu sehen, so können wir ihn bald schon nicht mehr anders als jenen anderen sehen.

Konkreter! Computer sind im allgemeinsten Sinne Produktionsmittel. Sie entstehen zu einer Zeit, als die Maschinenwelt hoch entwickelt ist: als jene Maschinen eben, mit denen der Angriff auf die Kopfarbeit beginnt (Nake 1992). In die Maschinenwelt einzuordnen sind Computer daher als Automaten. Das heißt als Maschinen, die innerhalb vorgegebener Grenzen quasi-selbständig operieren oder andere Maschinen kontrollieren und steuern. Sie müssen überwacht, aber nicht in jeder ihrer Operationen gesteuert werden.

Um die Mitte der 1960er Jahre herum entsteht die Computerkunst. Menschen veranlassen Computerprogramme dazu, Graphiken zu errechnen und zu zeichnen, die mit Kunstanspruch ausgestellt werden. Debatten kommen auf, wem der kreative Beitrag zuzuschreiben sei: dem Programmierer, dem Künstler (wenn dieser nicht selbst programmiert), dem Programm, dem Computer, gar der Zeichenmaschine? Oft genug hört man in dieser Debatte das Wort, der Computer sei für den Künstler wie ein Pinsel, er sei nichts anderes als »nur« ein Pinsel. In einem frühen Aufsatz vertritt z.B. A. Michael Noll (1967) diese Auffassung (der Computer

ist dort aber auch Partner und Medium für den Künstler). Sie lässt sich auch schon bei Leslie Mezei (1966) finden.

Das Wort »nur Pinsel« kommt auf, als der Computer-Künstler sich in der Defensive wähnt: Offensichtlich nicht zu leugnen ist, dass der Automat gehörig zum Ergebnis beiträgt. Arbeitet das Programm kräftig mit Zufallszahlen, wie es der Fall war, so beeindruckt der Anteil der Maschine am visuellen Ergebnis.

Die Schrumpfung des eben noch – in der BRD – als »Elektronengehirn« eingestuftes Computers (Lohberg/Lutz 1963) zum Pinsel hat in der Sorge um die eigene Kreativität ihre Ursache. Hielte der Künstler nicht daran fest, der Kreative in diesem ungleichen Gespann zu sein, wäre es um seine Kunst, also auch um ihn rasch geschehen. Behauptet er jedoch, den Computer so zu schwingen wie eben noch den Pinsel, so verstummt vielleicht der hinterhältige Frager. Dumm nur, dass die Farben vergessen worden sind. Viel später erst, die Nebeldebatte ist längst verfliegen, klärt Heinz Züllighoven (1998) die Sachlage mit der klaren und zulässigen Metaphorik von »Material und Werkzeug«, mit der er einen Weg zur objektorientierten Programmierung schafft.

Spannender als die oft verquaste BRD-Diskussion um das Werkzeug, das keines ist, sind unschuldig gemachte Nebenbemerkungen von Amerikanern. Deren grenzenloser Pragmatismus könnte eine Frage nicht langweiliger ansehen als die nach dem Computer als Werkzeug oder Medium, oder Partner gar? So lautet der erste Satz des *Abstract* zur Sutherlandschen Dissertation: »The Sketchpad system uses drawing as a novel communication medium for a computer«! (Sutherland 1980, S. 2).

Der Begriff des Mediums taucht auf, völlig beiläufig, nicht tragend, keinesfalls zur Diskussion auffordernd. Die Formulierung ist dennoch beneidenswert differenziert. Da gibt es das System namens Sketchpad. Es gibt das Kommunikations-Medium Zeichnung und es gibt den Computer, der offenbar das ist, was er ist, und nichts sonst, der sich aber der Zeichnung, jenes Mediums also, bedient.

Von Sketchpad wissen wir: es ist ein Software-System. Als solches ist es für seine dynamische Existenzweise auf einen Computer angewiesen. Dieses Software-System bemächtigt sich nun – vermittelt durch seinen Ziehvater, Sutherland – des alten Mediums Zeichnung. Dieses Medium wird dem Computer vom System Sketchpad in einer geradezu symbiotischen Beziehung verfügbar gemacht. Dem Computer wiederum verschafft das die Möglichkeit zur Kommunikation.

Ein kurzer Satz von anderhalb Zeilen Länge nur. Eine verhaltene, zu vermuten ist: unbeabsichtigte Klarheit in ihm. Der erste Satz des eigentlichen Dissertationstextes ist kaum länger. Er nimmt die eben erörterte Bemerkung auf: »The Sketchpad system makes it possible for a man and a

computer to converse rapidly through the medium of line drawings.« (Sutherland 1980, S. 8).

Kein Zweifel ist erlaubt, Linienzeichnungen sind das Medium, um das es hier geht. Es geht um dieses Medium insofern, als es bisher ganz in der Hand der Menschen lag. Damit Menschen mit dem Computer eine Konversation eingehen können, die sich auf Zeichnungen stützt, muss ein Software-System dafür sorgen, dass dem Computer Linienzeichnungen abgenötigt werden können.

Wir haben die Einfachheit und Klarheit dieser Position längst verlasen. Sutherland, der sich stets als Ingenieur versteht, soll nicht überinterpretiert werden. Was mir als Klarheit erscheint, dürfte nicht mehr sein als pragmatische Schlichtheit – eine Schlichtheit jedoch, die vom Machen, vom konstruktiven Schaffen zu kommen scheint, die von der Anstrengung des Begriffs nicht berührt, geschweige denn angekränkt wird.

Wenden wir uns nun aber der anderen Seite zu, die mit dem schlichten und oft so naiven, von der ARPA geförderten Tun nicht zufrieden ist (an dieses aber auch nicht heran kommt). Uns ist der Computer unmerklich, man konnte es nicht verhindern, zunächst zum Werkzeug, später zum Medium geworden. Wir sprechen nur selten noch davon, dass er, funktional betrachtet, eine automatische Maschine sei, die uns im Umgang als Werkzeug oder als Medium erscheinen mag, als beides auch gleichzeitig oder jedenfalls verschränkt (»instrumentales Medium« ist der treffende Begriff hierfür bei Schelhowe 1997). Die Rede vom Computer *als* Werkzeug, ebenso wie die *als* Medium«, macht explizit darauf aufmerksam, dass es sich um eine bloße Auffassung, eine Sichtweise, eine Zuschreibung handelt. Heute aber lesen wir oder reden auch selbst vom »Medium Computer«, was ja nur zu verstehen ist als »der Computer, das Medium« oder »das Medium, das der Computer ist«.

Keiner hat wohl die Vision vom Medium Computer so klar gesehen und so entschieden zu seiner Sache gemacht wie Alan Kay. 1977 schreiben er und Adele Goldberg, als sie von der epochal umstürzlerischen Arbeit bei Xerox PARC berichten (Kay/Goldberg 1977, S. 31):

»We design, build, and use dynamic media which can be used by human beings of all ages. [...] Although digital computers were originally designed to do arithmetic computation, the ability to simulate the details of any descriptive model means that the computer, viewed as a medium itself, can be all other media if the embedding and viewing methods are sufficiently well provided.«

Kay war von Sketchpad, dem Vorbereiter, und von Simula beeindruckt, dem entscheidenden Schritt hin zur objektorientierten Programmierung. Diesen Erkenntnis-Blitz verstärkte noch ein zweiter, der später Kay mit

seinen Leuten im PARC dazu brachte, sich all das auszudenken, was Millionen von Computernutzern heute lieb und wert und selbstverständlich und anders gar nicht mehr denkbar ist: eine an den Dingen orientierte Organisation *im* Computer, ihre visuelle Erscheinung *auf* dessen Peripherie und unser elegant-ästhetischer Umgang *mit* ihnen.

Dieser zweite Blitzschlag der Erleuchtung war die Begegnung mit Seymour Papert im Media Lab des MIT, mit den mit Logo dort spielenden Kindern und der Entwicklungs- und Lernpsychologie von Jean Piaget und Jerome Bruner. Das spielerische Lernen der Kinder wurde zu einer entscheidenden Maxime für den Wandel in der Auffassung des Computers vom Werkzeug zum Medium. Wir brauchen nur an die gnadenlos hierarchische Zurichtung jedes Dokumentes zu denken, die in Engelbarts Ansätzen und Systemen verlangt ist, und sie zu kontrastieren mit der offenen Art, in der Objekte eines Programms im Smalltalk-Stil mit einander in Austausch treten. Instrumental und medial!²⁰

Wir haben uns mittlerweile daran gewöhnt, das alles ganz beiläufig im Wort vom Medium zusammenzufassen. Gut 35 Jahre sind seit 1971, seit Beginn der Arbeiten Kays am PARC, verstrichen. Wir befinden uns wohl tatsächlich mitten im *Medium des Computing*, von dem wir gewöhnlich als dem digitalen sprechen.

Im Laufe einer technischen Entwicklung entfernt sich die zuerst sensationell erscheinende technische Gegebenheit oft meilenweit von ihrem Anlass. In dem Maße, wie dies geschieht, verliert sich der metaphorische Gehalt früher Zuschreibungen. Die Metapher stirbt ab, wird zur toten Metapher, deren Herkunft kaum noch im Bewusstsein vorhanden bleibt. Schließlich wandelt sie sich zum technischen Begriff, der jenes Neue bezeichnet ganz wie ein Eigenname.

Es mag sein, dass dem Medium im Sinne des digitalen Mediums Derartiges derzeit widerfährt. Die Frage danach, was das denn sei, das digitale Medium, wird dann immer seltener auch nur erhoben werden, weil der tägliche Umgang mit diesen und jenen Gegenständen und Bedingungen uns immer häufiger dazu führt, dass wir leicht erstaunt auf die eine oder andere Erscheinungsform des Computers verweisen. Das digitale Medium wird dann nicht Medium im niedrigen Sinne von »Linien-Zeichnung« oder »Fernsehapparat« sein, sondern eher im höheren Sinne eines komplexen, weit ausgreifenden und in unserem Falle mit Metamedialität ausgestatteten Megamediums: eine Medienmaschine, ein Medienmedium, die Medialität vielleicht schlechthin, die deswegen dann auch kaum noch einen besonderen Namen brauchte. Die Wurzel für eine solche Rede läge dann, das war die Behauptung, in dem kognitiv-physikalischen Doppel-

20 Vieles zu diesem Thema lässt sich vertiefen bei (Friedewald 1999), insbesondere dort Kapitel 6.

gerät, das mal als Lichtgriffel, mal als Maus physikalisch in Erscheinung tritt. Sein wesentlicher Aspekt ist die Bewegung und Platzierung eines indexikalischen Zeichens, das einen Ort markiert sowie die Auslösung einer Operation an einem so bedeuteten Ort. Die Maus hat sich schlicht als den menschlichen Fähigkeiten und Gewohnheiten besser angepasst erwiesen.

Einfach zu schlucken wäre der Wandel des Automaten Computer zum umfassenden Medium aus der Verschmelzung der HCI-Schnittstelle im Kästchen der Maus heraus wohl kaum. Ohne den Schritt zu einer semiotischen Auffassung der HCI zu tun, ist ein solcher Wandel auch kaum vorstellbar. Mit den folgenden drei Abschnitten werde ich versuchen, aus je unterschiedlicher Sicht dem näher zu kommen.

Zeichnen: von nah und von fern

Wenn eine zeichnet, sitzt sie mit einem Skizzenblock irgendwo in der Landschaft, im Atelier oder im Café, ein paar Bleistifte, Buntstifte, Kreiden oder was es sonst an Zeichengeräten gibt in Reichweite. Sie führt eines dieser Geräte auf vielfach variierende Art übers Papier. Dort hinterlässt sie (oder es) Spuren, die Abreibungen des Zeichenmaterials auf dem Zeichenpapier sind. Wie immer dieser erfreuliche Vorgang begonnen hat, sich fortsetzt oder auch beendet wird – er findet unter dem Begriff der Nähe statt. Die Künstlerin bearbeitet das Material mit ihrem Werkzeug und ist beiden dabei ständig so nah, wie es näher nicht geht. Auch muss sie Gerät und Material eng zu einander bringen. Als geschulter Zeichnerin ist ihr alles zuhänden: unmerklich verschmelzen Werkzeug und geschickte Hand. Der gestalterische Wille kann sich als solcher ausbreiten. Mangelnde handwerkliche Fertigkeiten kommen ihm nicht in die Quere.

Wie anders beim algorithmischen Zeichnen! Dann, wenn es tatsächlich geschieht, läuft ein Programm ab. Ob seine Ausgabe (heute) unmittelbar zu zeichnerischen Elementen führt oder (früher) sich erst in einer Zwischencodierung niederschlägt, die von einem Zeichenautomaten in sichtbare Formen und Farben gebracht werden muss, sei dahingestellt. Wesentlich ist, dass der ganzheitliche Akt des Zeichnens aufgelöst worden ist in die Programmierung einer Klasse von Zeichnungen und in die Realisierung eines Exemplars dieser Klasse. Der zeichnende Mensch aber täte sich Gewalt an, dächte er an eine Klasse von Zeichnungen statt an die eine Zeichnung.

Die zeichnende Hand hebt den Stift, also das Werkzeug des Zeichnens, vom Papier, dem Material des Zeichnens, ab. Sie legt den Stift zur Seite und schiebt auch den Zeichenblock weg, um den Gedanken an die

eine Zeichnung, die Realisat ihrer Klasse sein wird, umso konzentrierter, reiner und allgemeiner fassen zu können. Indem die Hand Werkzeug und Material zur Seite schiebt, fordert sie den Gedanken auf, sein Werk zu tun. Das Zeichnen aus der Ferne wird zu einem Zeichnen mit dem Kopf oder mit den Gedanken. Dem Schwung der Hand und des Armes, die den Stift über das Papier führen und eine einzelne bestimmte Linie erzeugen, entsprechen die Kühnheit des Denkens und die Abstraktheit der algorithmisch-allgemeinen Formalisierung aller Linienformen und Linienmuster, die in irgendeinem Sinne zusammenhängen. Eine Verwandtschaft wird gedacht und konkret formuliert. Ein Regelwerk entsteht, ein Geflecht von Bedingungen, die einzuhalten sind: von der ausführenden Maschine einzuhalten dann, wenn sie der Kontrolle des Zeichenprogramms unterworfen wird.

All das ist nicht sonderlich aufregend. Nichts daran ist irgendwie neu. Alles, was hier angedeutet wird, ist in die herrschende Kultur eingegangen. Täglich machen wir alle von solchen Programmen Gebrauch. Wir tun es mit Freude und mit Gewinn. Wir tun es mit Leichtigkeit und als Selbstverständlichkeit. Wir haben, als begeisterte Benutzer von Adobe Photoshop oder Corel Draw, Fertigkeiten erworben, die uns gestatten, passable Zeichnungen oder zeichnerische Elemente herzustellen, die wir uns »von Hand« nicht zutrauten.

Für Pädagogen und Kulturkritiker mag diese Gegebenheit gelegentlich noch immer problematisch erscheinen. Im Rahmen dieser Betrachtung lässt uns das relativ unberührt. Unser Hinweis nämlich besteht darin, dass wir den historischen Punkt relativ gut eingrenzen können, der zur Entfernung der zeichnenden Hand vom Zeichenpapier geführt hat. Dieser Punkt ist – in all der Ungenauigkeit, die ein gewolltes Setzen von Daten mit sich bringt – das Jahrzehnt zwischen dem Abschluss der Sutherlandschen Dissertation 1963 und dem Erscheinen der ersten Konzepte zur Hardware und Software bei Xerox PARC.

Die konzeptionellen Risse und Sprünge, die hierfür zu bewältigen waren, erscheinen auf dem Markt und werden zur Ware weitere zehn Jahre später mit dem Apple Macintosh (1984). Seitdem hat das Zeichnen seine Unschuld verloren und existiert in zwei Formen: der handwerklichen Form, die jeder schätzt, der sich der Mühe unterzieht, sie zu gewinnen; und der algorithmischen Form, die zu begreifen eine ähnliche Mühe verlangt, die zu verwenden jedoch erstaunlich einfach geworden ist.

Beide Formen bestehen neben einander und werden kaum noch ernsthaft gegen einander gestellt. Wer sollte auch Gefallen daran finden? Beide Formen gehören unterschiedlichen Epochen an: der des Handwerkes und der der Medialität. Die Kühnheit des algorithmischen Denkens

reißt die Verallgemeinerungen der algorithmisierten Regelwerke aus den Identität stiftenden Genügsamkeiten des still empfindenden Zeichners.

Manche jener Pioniere, die sich als erste daran machten, Computer zum Zeichnen zu zwingen, berichten davon, wie sie ein merkwürdiger Schauer ergriff, als sie die Zeichenmaschine dabei beobachteten, wie sie erstmals Linien auf Papier brachte, an die der Wissenschaftler oder Künstler in einem abstrakten Sinne, als pure Möglichkeit, gedacht hatte (Nees 2006, S. XI). Etwas war in die Welt gekommen, das nicht mehr verschwinden würde.

Ivan Sutherland hat sein Programm sehr schlicht »Sketchpad« genannt, »Skizzenblock«. Hat er daran gedacht, den Lichtgriffel zu nutzen, um Skizzen schnell und ungefähr auf den Bildschirm zu bringen, die Anlass für Mitteilungen, für ungefähre grafische Aussagen sein sollten, also Zwischenstufen bei der Entwicklung einer Konstruktionszeichnung vielleicht? Wir wissen es nicht. Möglich wäre es. Es wäre aber auch gleichgültig. Der Name des Systems gefiel ihm einfach.

Denn was er erreicht hat, ist die Auftrennung der Zeichnung in eine sichtbare und eine manipulierbare Schicht. Diese prinzipielle Errungenschaft betrachten wir im folgenden Abschnitt.

Verstehen: Unterfläche und Oberfläche

Die *Zeichnerin* schafft, ob sie das will oder nicht, mit ihren Linien, Formen, Flächen, Farben auf Papier: Zeichen. *Anlässe* für Zeichen, um genauer zu sein.

Der *Programmierer*, der seinen Computer dazu veranlasst, auf einem Plotterpapier oder einem Bildschirm eine Ansammlung von sichtbarem Material abzulegen, schafft ganz ebenso, ob er es will oder nicht: Anlässe für Zeichen.

Ich sage »Anlässe« und nicht gleich »Zeichen«, um anzudeuten, dass das, was auf den Papieren sichtbar wird, nicht selbst schon und umstandslos »Zeichen« ist. Zeichen werden jene Linien und Schwärzungen erst dann und dadurch, dass ein Mensch sich die Angelegenheit anschaut und, da er anders gar nicht kann, interpretierend zum Zeichen macht. Wir hatten oben darüber gesprochen. Hier kommt es uns auf einen besonderen Aspekt an, den die Zeichen auf dem Bildschirm aufweisen. Diesen besonderen Aspekt hatte ich in früheren Arbeiten (Nake 1986) die *Verdoppelung* des Computerdings genannt. Der dahinter aufscheinende Umstand lässt sich heute etwas präziser fassen.

Was wir auf dem Bildschirm als Ausgabe des laufenden Programms erblicken, ist – semiotisch betrachtet – das Repräsentamen eines Zeichens.

Das Repräsentamen ist die materielle, sinnlich wahrnehmbare Seite des Zeichens. Sie ist auch jener Aspekt, der Gegenstand der algorithmisch gesteuerten Bearbeitung durch den Computer wird.

Diese wahrnehmbare Seite ist Licht auf dem Bildschirm. Sie ist auf Innigste – auch das hatte ich schon erwähnt – mit den Inhalten des Bildspeichers verknüpft. Was im Bildspeicher steht, wird vom Display Prozessor gelesen und in Signale verwandelt, die die Beschichtungen des Monitors anregen bzw. – bei anderer Technologie – die LCD Elemente. Zwischen sichtbarem Bild und unsichtbarem Speicherinhalt besteht eine eindeutige Beziehung: sie ist eindeutig in beide Richtungen. Das ist auch der gute Grund dafür, dass wir oft ohne Unterscheidung von diesen beiden Erscheinungen zusammen als »dem Bild« sprechen.

Etwas genauer scheint mir jene Sprachregelung zu sein, die Bildern auf dem Computermonitor eine *Oberfläche* und eine *Unterfläche* zuschreibt. Die Oberfläche ist sichtbar. Sie ist für uns. Die Unterfläche ist unsichtbar. Sie ist für den Computer. Die Unterfläche kann der Computer verändern, er kann sie manipulieren. Die Oberfläche hat diese Eigenschaft nicht.

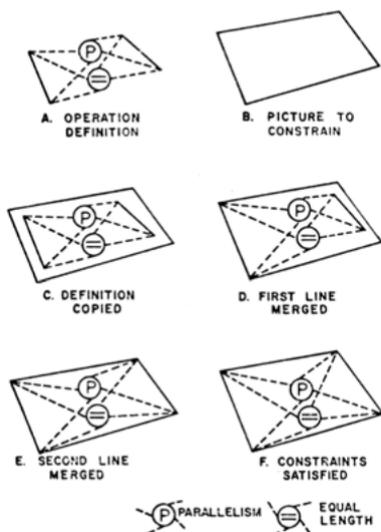
Die prinzipielle, nicht hintergehbare Doppelexistenz betrifft Gegenstände des Computers generell, jedenfalls dann, wenn ihre Existenz auch uns bekannt gemacht werden soll. Hier interessieren uns nur die Bilder. Bei ihnen liegen die Verhältnisse besonders klar. In ihrer Aufspaltung zur Ober- und Unterfläche, zur sichtbaren und bearbeitbaren Seite, erscheinen die Zeichen als *algorithmische Zeichen* (Nake 2001).

Solche distanzierten Betrachtungen haben Ivan Sutherland nie interessiert. Die Tatsache jedoch, die ich hier so fasse, war ihm vertraut. Eine seiner Großtaten in Sketchpad war, dass den zeichnerischen Elementen einer Grafik *Bedingungen* (*Constraints*) auferlegt werden konnten. Eine Bedingung konnte etwa lauten, dass zwei Seiten eines Vierecks parallel zu einander sein sollten. Eine andere, dass zwei Seiten orthogonal zu einander stehen sollten. Geben wir zu diesen zwei weitere Bedingungen an und verlangen, dass dadurch insgesamt ein geometrisches Ding festgelegt werde – was ja zweifellos der Fall ist –, so hätten wir ein allgemeines Viereck zu einem Quadrat gemacht. Wirklich *gemacht* hätten wir das nicht: wir hätten die geometrischen Bedingungen gesetzt, wir hätten das Ding, das wir haben wollten, nur beschrieben.

Diese Art der konstruktiven Beschreibung ist schon für sich genommen aufregend spannend (sie ist ein frühes Beispiel der späteren *Constraint-Programmierung*). Sutherland ging aber, seinem Programm der grafischen Kommunikation treu bleibend, einen Schritt weiter. Die Bedingungen können in Sketchpad zusätzlich als graphische Symbole gezeigt werden. Indem sie äußerlich erscheint, nicht innerlich nur gilt, kann eine Bedingung interaktiv manipuliert werden. Man kann in der ent-

stehenden Graphik etwa das Symbol der Bedingung »parallel« auftreten lassen. Es weist vier kurze Tentakeln auf. (Abb. 9) Je zwei davon können (mit dem Lichtgriffel) mit den Endpunkten der beiden Strecken verknüpft werden, die parallel zu einander sein sollen. Braucht man die Anzeige der Bedingung nicht mehr, kann man sie unsichtbar machen.

Abb. 9: Beispiel für den Umgang mit Constraints in Sketchpad.
(Sutherland 1963, S. 94)



So wird deutlich, dass die sichtbare Grafik in sich eine komplexe Beschreibung ihrer selbst enthält. Was wir zu Gesicht bekommen, ist nur eine geeignete Sicht auf das neuartige Gebilde „Grafik“. Durch seine Emigration vom Papier auf den Bildschirm ist ihm eine Unterfläche gewachsen, die all das aufnimmt, was die Zeichnung dem Computer gegenüber sein soll. In dieser Verdoppelung erscheint der interaktive Umgang mit einer algorithmischen Zeichnung weit jenseits der Werkzeugartigkeit. Insofern Sketchpad die Wurzeln dafür legte – wennzwar es einerseits im Denken an Werkzeug konzipiert wurde, im Agieren in Medien aber spät seine eigentliche Relevanz erst zeigte – sehe ich in Sketchpad speziell, in der frühen Computerkunst aber allgemein, das erste Wetterleuchten der digitalen Medien.

Computabilität, Interaktivität, Performativität

Mit einer erneuten Bemerkung zu Computer als Automat, Werkzeug und Medium möchte ich diese Betrachtung schließen. Sie ging um das Zeichnen mit Hilfe von Programmen, das algorithmische Zeichnen. Sie führte uns zu der Einsicht, dass diese Art des Zeichnens aus dem Zeichnen ein Zeigen macht.

In dieser gewaltigen Reduktion steckt eine entsprechend mächtige Öffnung. Die Zeichnung kann nun, wo sie dadurch entsteht, dass mit einem Gerät auf Konstruktionselemente, Operationen oder Bedingungen gezeigt wird, selbst (in einem Akt ihrer eigenen Verdoppelung) ihre konstruktive Beschreibung aufnehmen. Sie stülpt sich sozusagen, in einem historischen Akt der konstruktiven Dekonstruktion, nach außen um, verschlingt etliche der Kennzeichnungen ihrer Struktur und Erscheinung, die wir früher geistig abstrahieren und schließen konnten, die aber immer bei uns blieben und uns nur intellektuell zugänglich waren. Auf solche Weise angereichert und gesättigt, zieht sie sich in ihre Existenz als Objekt des Gezeigtwerdens zurück.

Dieser Akt hat die Zeichnung zum algorithmischen Zeichen werden lassen, einem Zeichen für zwei Herren, bescheidener: mit zwei Interpretanten. Der erste und alte Interpretant ist die Bedeutung, die ein Betrachter subjektiv der Zeichnung zukommen lässt. Der zweite und neue Interpretant ist die Menge all jener programmierten Operationen, die dem Prozessor abverlangt werden, wenn er die Zeichnung manipuliert.

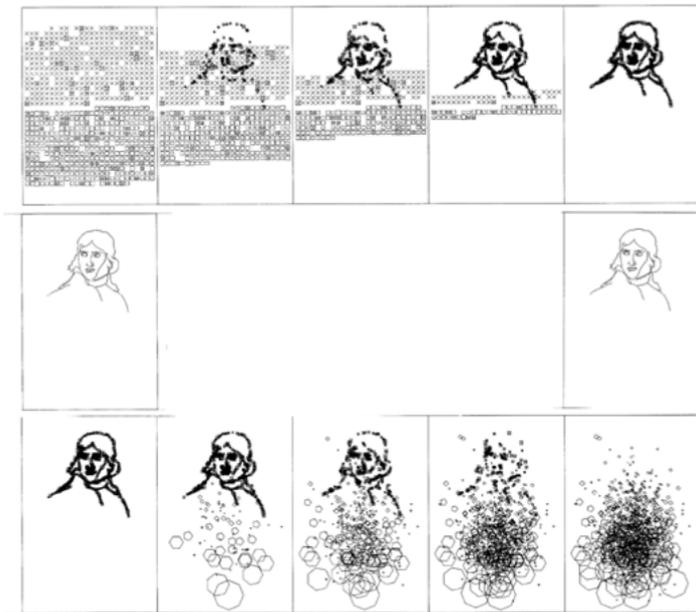
Drei Umgangsweisen mit dem Computer schwingen mit in diesen einigermaßen komplexen Verhältnissen. Die älteste und bleibende Umgangsweise ist die, die den Computer als *Automaten* nimmt. Wenn *Constraints* gesetzt sind, müssen sie mathematisch gelöst werden, damit die Zeichnung entstehen kann. Dazu tritt die automatische Funktion in Kraft (die auch sonst ständig virulent bleibt). Dieser Umgangsweise entspricht die *Berechenbarkeit*. Wir sind lediglich Beobachter.

Die *Interaktivität* bestimmt eine zweite Umgangsweise mit dem Computer: die *werkzeugähnliche*. Wir setzen eine Operation an, die wir aus einer größeren Auswahlmenge geholt haben und die oft relativ begrenzte Wirkung hat. Wir sind hier auf Hand und Auge eingeschränkt. Wenn wir eine Bedingung interaktiv auf Elemente der Zeichnung aufprägen, bewegen wir uns auf diesem Terrain.

Schließlich gewinnen wir das *Medium* der Zeichnung, dem ein *performativer Umgang* korrespondiert. Er ist im Entstehen. Alan Kay ist sein Prophet. Die ganze Zeichenhaftigkeit, die der Medialität entspricht, wird im Sinne algorithmischer Zeichen entfaltet. Wir begegnen der Zeichnung,

dem Bild als einem neuen Medium, jenem Doppel, von dem ich gesprochen hatte. Unsere Begegnung weist selbstverständlich implizit auf Berechenbares und Interaktives. Sie ist aber insofern mehr, als der Mensch im Umgang mit seiner Maschine, die er als Werkzeug denkt und die ihm Medium ist, in seiner gesamten Körperlichkeit gefordert ist. Die Formen, die das verlangt und annimmt, sind noch weitgehend unbekannt. In künstlerischen Umgängen werden sie gesucht, entdeckt, ergründet und geschaffen.

Abb. 10: Ästhetisch-mediale Umgangsweise mit dem Computer (Nake 1969).



Literatur

- Beaudouin-Lafon, M. (2006): »Human-computer interaction«. In Goldin, D./Smolka, S./Wegner, P. (eds.): »Interactive computation. The new paradigm«. New York, Berlin S. 227-254.
- Bense, M. (1967): »Semiotik. Allgemeine Theorie der Zeichen«. Baden-Baden.
- Bissell, D. (1990): »The father of computer graphics«. BYTE, 15, 6, S. 380-81.
- Brent, J. (1993): »Charles Sanders Peirce. A Life«. Bloomington, Indianapolis.
- Ceruzzi, P. E. (2002): »A history of modern computing«. Cambridge, MA (auch auf Deutsch erhältlich. Zuerst 1998).

- Deleuze, G./Guattari F. (1977): »Rhizom«. Berlin.
- Desmonde, W. H. (1969): »A conversational graphic data processing system: The IBM 1130/2250«. Englewood Cliffs.
- Eco, U. (1977): »Zeichen. Einführung in einen Begriff und seine Geschichte«. Frankfurt/Main.
- English, W. K./Engelbart, D. C./Berman, M. L. (1967): »Display-selection techniques for text manipulation«. In: IEEE Trans. Human Factors in Electronics HFE 8, 1, S. 5-15.
- Frenkel, K. (1989): »An interview with Ivan Sutherland«. In: Communications of the ACM 32, 6, S. 712-718.
- Friedewald, M. (1999): »Der Computer als Werkzeug und Medium«. Berlin, Diepholz.
- Kay, A./Goldberg, A. (1977): »Personal dynamic media«. In: IEEE Computer 10, 3, S. 31-41.
- Kreuzer, H. (Hrsg.) (1987): »Die zwei Kulturen. Literarische und naturwissenschaftliche Intelligenz. C.P. Snows These in der Diskussion«. Stuttgart.
- Licklider, J.C.R. (1960): »Man-computer symbiosis«. In: IRE Trans. Human Factors in Electronics, HFE-1, S. 4-11.
- Lohberg, R./Lutz, Th. (1967): »Was denkt sich ein Elektronengehirn?« Stuttgart (zuerst 1963 erschienen).
- Machover, C. (1999): Computer graphics pioneers. ACM SIGGRAPH vol. 33 no. 3 (auch: www.es.com/about_eands/history.asp).
- McKenzie, J. A. (1974): »TX-0 computer history«. MIT Research Laboratory of Electronics Technical Paper (Oct. 1, 1974), http://bitsavers.org/pdf/mit/tx-0/memos/McKenzie_TX-0_history_Oct74.txt (20.8.2007).
- McLuhan, H. M. (1964): »The extension of man. Understanding media«. New York.
- Mezei, L. (1966): »The electronic computer – a tool for the visual arts«. In: 5th National Conference Computer Society of Canada (als Manuskript).
- Mongré, P. (1897): »Sant’Ilario. Gedanken aus der Landschaft Zarathustras«. Leipzig.
- Morris, Ch. W. (1988): »Grundlagen der Zeichentheorie. Ästhetik der Zeichentheorie«. Frankfurt/Main.
- Nake, F. (1986): »Die Verdoppelung des Werkzeugs«. In: Rolf, A. (Hrsg.): »Neue Techniken alternativ«. Hamburg, S. 43-52.
- Nake, F. (1992): »Informatik und die Maschinisierung von Kopfarbeit«. In: Coy, W. et al. (Hrsg.) (1992): »Sichtweisen der Informatik« Braunschweig, Wiesbaden, S. 181-201.
- Nake, F. (2001): »Das algorithmische Zeichen«. In: Bauknecht, W./ Brauer, W./Mück, Th. (Hrsg.): »Informatik 2001. Tagungsband der GI/OCG Jahrestagung«. 2 Bde Wien, Bd. II, S. 736-742.

- Nees, G. (2006): »Generative Computergraphik.« Nachdruck der ersten Auflage von 1969, mit neuen Einleitungen von Hans-Christian von Herrmann und Georg Nees, Berlin.
- Noll, A. M. (1967): »The digital computer as a creative medium«. In: IEEE Spectrum 4,10, S. 89-95.
- Pape, H. (2004): »Charles S. Peirce zur Einführung« Hamburg.
- Peirce, Ch. S. (1993): »Phänomen und Logik der Zeichen«, hrsg. u. übersetzt von H. Pape, Frankfurt/Main, 2. Aufl.
- Peirce, Ch. S. (2000): »Semiotische Schriften«, 3 Bde. hrsg. von Chr. J. W. Kloesel und H. Pape. Frankfurt/Main.
- Prince, D. (1971): »Interactive Graphics for Computer-Aided Design«. Reading, Mass.
- Saussure, F. de (2001): »Grundfragen der allgemeinen Sprachwissenschaft«. Berlin, Heidelberg New York, 3. Aufl.
- Schelhowe, H. (1997): »Das Medium aus der Maschine. Zur Metamorphose des Computers«. Frankfurt.
- Shneiderman, B. (1983): »Direct manipulation: a step beyond programming languages«. In: IEEE Computer 16, 8, S. 57-69.
- Sutherland, I. E. (1963): »Sketchpad. A man-machine graphical communication system«. In: Proceedings of the Spring Joint Computer Conference 1963, AFIPS Conference Proceedings, Vol. 23, S. 329-345.
- Sutherland, I. E. (1965): »The Ultimate Display«. In: Information Processing 1965. Proceedings of IFIP Congress 65, 2 Bde Washington, D. C., London 1965, Bd. I, S. 506-508.
- Sutherland, I. E. (1966): »Computer inputs and outputs«. In: Scientific American, Sept. 1966, S. 86-96.
- Sutherland, I. E. (1980): »Sketchpad. A man-machine graphical communication system«. New York, London (zuerst intern bei MIT, 1963).
- Virilio, P. (1998): »Rasender Stillstand«. Frankfurt/Main (franz. Orig. 1990).
- Waldrop, M. M. (2001): »The dream machine. J.C.R. Licklider and the revolution that made computing personal«. New York.
- Züllighoven, H. (1998): »Das objektorientierte Konstruktionshandbuch nach dem Werkzeug & Material Ansatz«. Heidelberg.

**GESTALTUNGSPROBLEME UND DESIGNKONZEPTE
IN DER MENSCH-COMPUTER-INTERAKTION**

BENUTZERGERECHTE MCI IN EINER DYNAMISCHEN WELT - EINE GESTALTUNGSAUFGABE

HORST OBERQUELLE

1 Einführung

Die Verbreitung interaktiver Systeme in alle Lebensbereiche scheint unaufhaltsam. Vom Computerarbeitsplatz für wenige Spezialisten bis zu Bildschirmarbeit an etwa der Hälfte aller Arbeitsplätze hat es nur etwa 30 Jahre gedauert. In den letzten 10 Jahren haben sich mit Internet und Web, mit mobilen Computern und der Integration von Rechenleistung in Alltagsgegenstände die Zahl und die Vielfalt der Benutzungsschnittstellen exponentiell erhöht. Die Benutzergruppen entwickeln sich in Richtung auf *alle* in vielfältigsten Nutzungssituationen: Allgegenwärtige Mensch-Computer-Interaktion (MCI)!

Es stellt sich die Frage, wie und durch wen die ständig neu entstehenden Benutzungsschnittstellen eine benutzergerechte Gestalt bekommen. Werden sich technische Neuerungen schlicht durchsetzen oder bedarf es einer bewussten Gestaltung? Um dieser Frage nachzugehen, erscheint es lohnenswert, das Gestaltungsproblem näher zu charakterisieren und seine Historie zu betrachten, um Hinweise für die zukünftige Gestaltung zu gewinnen.

2 Das Gestaltungsproblem der MCI

Interaktive Systeme sind nur in dem Maße erfolgreich, wie sie sich im Nutzungskontext bewähren. Es geht also um die Kontextpassung. Die Gestalt eines interaktiven Systems, die ein Benutzer wahrnimmt, wird durch Menschen im Systemgestaltungsprozess unter Berücksichtigung ihres (Un-)Wissens über den Nutzungskontext und über technische Gestaltungsoptionen festgelegt. Zunächst soll dieser Zusammenhang näher beleuchtet werden.

Der Gestaltungskontext

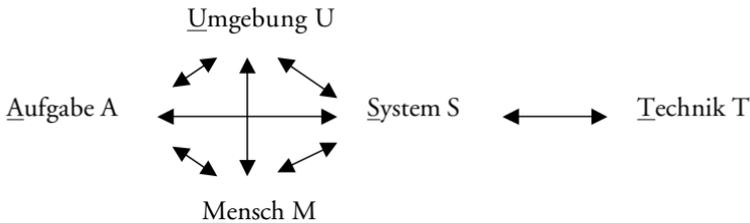
Es erscheint sinnvoll, beim Nutzungskontext drei Aspekte zu unterscheiden:

- die *Menschen* mit ihren spezifischen Fähigkeiten und Begrenzungen im Hinblick auf Wahrnehmung, Denken, Lernen, Gedächtnis, Emotionen und Handeln;
- die zu erledigende *Aufgabe* bzw. die verfolgten Ziele und Motivationen beim Einsatz interaktiver Systeme;
- die Randbedingungen der *Umgebung*, z.B. die Organisation, in der Aufgaben erledigt werden; die physische Situation, in der ein mobiles Gerät genutzt wird; oder die Gemeinschaft, in der Computer zur Kooperation eingesetzt werden.

Außerdem ist das Spektrum des *technisch Möglichen* zu berücksichtigen.

Der Zusammenhang zwischen diesen Aspekten kann gut durch ein Wechselwirkungsdiagramm veranschaulicht werden.

Abb. 1: MAUST-Diagramm



In der Abb. 1 werden bewusst Doppelpfeile verwendet. In Richtung auf das interaktive *System S* sollen sie bedeuten, dass Wissen über die Aspekte MAUT benötigt wird. Die von *S* ausgehenden Pfeile deuten darauf hin, dass der Einsatz von *S* im Kontext unweigerlich Wirkungen bei den Aspekten MAU entfaltet und auch die Technologieentwicklung befruchten kann. MCI ist deshalb als ein dynamisches Wechselwirkungsgefüge zu betrachten. Interaktive Systeme müssen für einen veränderlichen Kontext gestaltet werden und sich an ihn anpassen oder angepasst werden können. Es ist also prinzipiell von einer Art *Ko-Evolution* auszugehen.

Wenn im Weiteren von *Benutzern* gesprochen wird, sind damit Menschen gemeint, die ihre Aufgabe bzw. Ziele in einer bestimmten Umgebung unter Nutzung eines interaktiven Systems verfolgen.

Die *Gestaltungsaufgabe* besteht nun darin, einem interaktiven System eine solche Gestalt zu geben, dass es Benutzer unterstützt und eine Ko-Evolution von Nutzungskontext und System ermöglicht wird.

Wie und durch wen diese Gestaltungsaufgabe gelöst wird, ist eines der zentralen Probleme der MCI. Zunächst soll aber der Begriff der Interface-Gestalt geklärt werden.

Die »Gestalt« interaktiver Systeme

Interaktive Systeme sind für Benutzer komplexe künstliche Welten. Sie umfassen Räume und Orte, Gegenstände und Umgangsformen, ggf. aktive Komponenten wie Automaten und sog. Agenten sowie Verbindungen zu anderen Systemen und Benutzern. Wenn wir heute wie selbstverständlich von Schreibtischoberflächen mit Ordnern und Dokumenten sprechen, meinen wir eine solche virtuelle Welt. Diese Welt, die dem Benutzer gegenübertritt, wird seit Norman (1986) als *konzeptuelles Modell* bezeichnet. Benutzer machen sich davon ein mehr oder weniger korrektes *mentales Modell*, um sie zu verstehen, ihr Verhalten vorherzusagen und sie zu beeinflussen, d.h. sie zu kontrollieren. Das mentale Modell entspricht der wahrgenommenen *Gestalt* des Systems und muss nicht mit dem konzeptuellen Modell übereinstimmen.

Der Zustand einer solchen konzeptuellen Welt wird an der *Benutzungsoberfläche*, dem *Interface*, permanent (in Ausschnitten und Sichten) dargestellt und kann durch entsprechende Navigations- und Inspektionsoperationen erkundet werden. Navigieren durch Scrollen, Zoomen und Suchen ist inzwischen für viele eine Selbstverständlichkeit. Über passende interaktive Operationen können Veränderungen in der Welt ausgelöst werden. Das Verschieben eines Dokumentes von einem in einen anderen Ordner ist ein einfaches Beispiel.

Alle Operationen erfordern Eingaben, die durch unterschiedliche Interaktionsformen realisiert werden (z.B. Kommandos, Formulare, Direkte Manipulation, natürliche Sprache). Konkrete Ein-/Ausgabegeräte und -techniken dienen ihrer Realisierung.

Die hier vorgenommene Unterteilung in drei *Gestaltungsebenen* – konzeptuelle Welt, Interaktion/Visualisierung und Ein-/Ausgabe – ist rein analytischer Natur, um Gestaltungsfelder abgrenzen zu können. Aus Benutzersicht ergibt sich eine integrierte Gestalt, die sich während der Nutzung formt und im mentalen Modell des Benutzers ihren Niederschlag findet.

In wie weit das der Gestaltung zugrunde liegende konzeptuelle Modell für Benutzer verständlich ist, ist eine zentrale Frage der MCI. Wie es Gestalt annimmt, hängt vom Gestaltungsprozess ab.

Die »Gestaltung« interaktiver Systeme

Die *Gestaltung* eines interaktiven Systems geschieht im Entwicklungsprozess. Die Gestaltung füllt die sog. *Gestaltungslücke*, die dadurch entsteht, dass sich die Gestalt nicht einfach aus den Anforderungen des Nutzungskontextes, den technischen Möglichkeiten und den Rahmenvorgaben eines Projektes ableiten lässt. Vielmehr sind viele kreative Entscheidungen auf allen drei Gestaltungsebenen notwendig.

Schon beim konzeptuellen Modell ergibt sich ein Spannungsfeld. Dominiert eine technische Vorstellungswelt, so ergibt sich evtl. das Problem, die technische Gestalt den Benutzern zu vermitteln. Wird ein konzeptuelles Modell zugrunde gelegt, das nicht einfach technisch umzusetzen ist, wird die Realisierung eines Systems schwierig.

Bei der Gestaltung bewegen sich die Gestalter auch in einem *Spannungsfeld zwischen Konsistenz und Innovation*. Die Gestaltung interaktiver Systeme geschieht heute nicht mehr in einem »luftleeren Raum«. Es existieren bereits viele Systeme, die die Benutzer kennen und aus denen sie Wissen und Erfahrungen sinnvollerweise auf neue Systeme übertragen möchten. Dies kann für die Gestaltung neuer (Teil-)Systeme bedeuten, bekannte Teile von konzeptuellen Modellen, Interaktionsformen und Ein-/Ausgabetechniken zu übernehmen. Style Guides und Interaktionsmuster unterstützen hier die *konsistente Gestaltung*.

Innovative Gestaltung bricht in der Regel solche Konventionen. Konsistenz kann als Bremse der Innovation wirken. So wird inzwischen der Zugang zu Tausenden von Dateien über das gewohnte hierarchische Dateisystem eher als Problem betrachtet. Auf einem PDA wird der Zugang bereits bei wesentlich weniger Dateien zu einem Problem, weil durch den beschränkten Platz nur minimale Ausschnitte sichtbar gemacht werden können und somit die Orientierung erschwert ist. Auch die Gleichbehandlung von Programmen und Dokumenten, die mit ihnen bearbeitet werden, als Dateien ist für viele Benutzer schwer zu verstehen. Innovative Zugänge, z.B. über eine flexiblere datenbankgestützte Dateiverwaltung mit Such- und Filterfunktionen, haben sich aber noch immer nicht etablieren können. Technisch sind sie machbar; ein für Nichtinformatiker verständliches konzeptuelles Modell fehlt aber noch immer. Eine besondere Klippe scheinen hier z.B. das Fehlen eines klaren Aufbewahrungsortes von Dokumenten und Schwierigkeiten beim Verständnis der Unterscheidung von Objekten und Referenzen auf Objekte zu sein.

Die nächste interessante Frage ist, wer wann mit welchen Zielen diese Gestaltung vornimmt. Dies führt uns zu den GestalterInnen.

Die GestalterInnen

Gestaltung geschieht durch Menschen während des Entwicklungsprozesses. Je nach Organisation dieses Prozesses wird der Nutzungskontext genauer erkundet und analysiert sowie ein konzeptuelles Modell gestaltet, bevor die technische Realisierung beginnt. In modernen Ansätzen des *Usability Engineering* werden dazu besondere Methoden eingesetzt. Dabei spielt das Bild der Gestalter von den Benutzern eine wichtige Rolle. Es macht einen großen Unterschied, ob für Anwendungsexperten gestaltet wird, deren Facharbeit optimal unterstützt werden soll, oder ob die Gestalter für den DAU - den dümmsten anzunehmenden User - gestalten.

Auch die Ziele und Motive der Gestalter in Bezug auf ihre eigene Arbeit spielen eine Rolle.

- Gestalten sie so, dass sie selbst als potenzielle Benutzer zufrieden wären?
- Versuchen sie durch Wiederverwendung von Lösungen Aufwand einzusparen?
- Haben sie ein Lerninteresse und wollen innovative Technik ausprobieren? Derartige Motive haben im Web zu einer Flut von Flash-Einsätzen geführt, die inzwischen einer reflektierteren Nutzung dieser Technologie gewichen ist.
- Wollen sie auf jeden Fall etwas Neues machen? Wollen Sie sich durch ein einmaliges Design einen Namen machen?
- Oder sind sie schlicht gezwungen, nach der anspruchsvollen Realisierung der Funktionalität noch unter Zeitdruck ein User Interface zu basteln?

Prinzipiell stellt sich die Frage, in wie weit die Gestalter fachliche Gestaltungskompetenz haben und wer im Zweifelsfall die Entscheidungskompetenz in Gestaltungsfragen hat.

3 Das Gestaltungsproblem im Rückblick

In der Anfangszeit der interaktiven Systeme wurde das Gestaltungsproblem nicht als solches wahrgenommen, da Systementwickler und Benutzer sehr ähnlich waren. Mit den Interaktionsdiagrammen von Parnas (1969) wurde erstmals eine Modellierung möglicher Benutzungspfade aus Benutzersicht vorgenommen. Derartige Diagramme wurden von realen Benutzern aber kaum verstanden und genutzt.

Beim Aufkommen der Desktopsysteme (XEROX Star, vgl. Johnson et al. 1989; Apple Lisa und Apple Macintosh) Anfang der 80er Jahre wurde

den Benutzern ein mehr oder weniger eingängiges konzeptuelles Modell angeboten, das von den meisten Systementwicklern für die eigene Nutzung strikt abgelehnt wurde, da sie es eher für zu einfach und ihrer Kompetenz im Umgang mit Kommandosprachen unangemessen fanden. Bei der Ausgestaltung der Vermittlung an der Oberfläche wurden erstmals Graphikdesigner einbezogen, um Icons im Hinblick auf Wahrnehmbarkeit und Bildschirmauflösung zu optimieren.

Bei der Entwicklung von PCs dominierten zunächst die textbasierten konzeptuellen Modelle der Entwickler: ein Computer wurde über Kommandozeilen gesteuert. Mit dem Übergang von Windows auf eine graphische Oberfläche wurden bekannte Ideen für ein konzeptuelles Modell nachempfunden. Alan Cooper, der als Designer die Entwicklung zu beeinflussen versuchte, gab nach einigen Jahren auf und verfasste zwei interessante Bücher. In »About Face« (Cooper 1995; Cooper/Reimann 2003) werden Ideen zur Verbesserung der Gestalt von Desktopsystemen präsentiert, die sich nicht durchsetzen ließen. »The Inmates are running the Asylum« (Cooper 1999) beschreibt die Auswirkungen der Dominanz von technisch orientierten Entwicklern und die daraus resultierenden mangelhaften Benutzungsschnittstellen. Designer werden - wenn überhaupt - zu spät in die Entwicklung einbezogen, sollen oft nur die Oberfläche verschönern und werden bei der Gestaltung des konzeptuellen Modells - soweit hier überhaupt eine bewusste Gestaltung aus Benutzersicht stattfindet - nicht beteiligt.

In vielen Softwareentwicklungsprojekten dominierte eine Ingenieurs-sicht, die davon ausgeht, dass Benutzungsschnittstellen spezifiziert werden können und entsprechend zu realisieren sind und dass für die Gestaltung keine Spezialisten benötigt werden.

Die Entwicklungsphasen der MCI

Das Forschungsgebiet Mensch-Computer-Interaktion hat in spezieller Weise zur Gestaltung von Benutzungsschnittstellen beigetragen, zunächst eher durch Kritik, dann durch Ansätze zum Usability Engineering.

Die 1. Phase der MCI - häufig kurz mit Software-Ergonomie gleichgesetzt - wird durch das Bedürfnis nach Evaluation bestimmt: Wie kann festgestellt werden, ob ein interaktives System benutzergerecht ist? Die sorgfältige Differenzierung von Aspekten der Benutzergerechtigkeit und die Entwicklung von Evaluationsmethoden mit Bezug zu Arbeitskontexten waren charakteristisch. Erstere führte bis zur Formulierung von ISO-Normen für Produkte, von denen die Normenreihe ISO 9241 »Ergonomie der Mensch-System-Interaktion« (früher: »Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten«) der wichtigste Vertreter ist.

Diese Normen können als Mindeststandard betrachtet werden, der bei der Gestaltung wenigstens zu erreichen ist. Sie sind aber kein Patentrezept, um die Gestaltungslücke zu schließen. Die oft von Designern geäußerte Kritik, dass z.B. die Grundsätze der Dialoggestaltung (ISO 9241, Teil 10, neu Teil 110; DIN 2006), die Gestaltungsfreiheit zu sehr einengen, erscheint als unangemessen. Zum einen sind die Prinzipien zu allgemein, als dass sie eine bestimmte Gestalt determinieren. Zum anderen garantiert ihre Einhaltung nur ein Mindestmaß an Benutzergerechtigkeit. Einer darüber hinaus gehenden Gestaltung sind kaum Grenzen gesetzt.

In einer 2. Phase kam stärker zu Bewusstsein, dass die Gestaltung eine eigenständige Aufgabe ist, die im Softwareentwicklungsprozess ihren Platz finden muss. Bereits in ihrem grundlegenden Buch »Understanding Computers and Cognition. A New Foundation for Design« schreiben Winograd und Flores: »In order to understand the phenomena surrounding a new technology, we must open the question of design - the interaction between understanding and creation.« (Winograd/Flores 1986, S. 4). Damit gaben sie einen Anstoß für die Design-Wende der MCI.

Floyd (1987) forderte einen Paradigmenwechsel aus der Softwaretechnik heraus, weg von einer Produktsicht, die Software gemäß einer festgelegten Spezifikation ingenieurmäßig konstruiert und realisiert, hin zu einer Designsicht, bei der Software in Wechselwirkung mit dem Nutzungskontext gestaltet wird. Das von Floyd (Floyd et al. 1989) vorgeschlagene STEPS-Modell gibt dem Gestaltungsprozess einen neuen Rahmen und propagiert ein partizipatives, prototypgestütztes, iteratives Vorgehen. Es wendet sich allerdings an Softwareingenieure und sieht noch keine spezielle Rolle für Designer vor.

Im skandinavischen Raum dagegen wird die Gestaltung interaktiver Systeme zusammen mit Benutzern bereits seit dem bekannten UTOPIA-Projekt in den 80er Jahren zum Programm erhoben (Greenbaum/Kyng 1991).

Auch in den USA tritt die Forderung nach expliziter Gestaltung von Software in »Bringing Design to Software« prominent hervor: »Good design produces an object that works for people in a context of values and needs, to produce quality results and a satisfying experience.« (Winograd 1996, Introduction, S. XVI). Das *Software Design Manifesto* bringt den Ansatz auf den Punkt: »The most important social evolution within the computing professions would be to create a role for the software designer as a champion of user experience.« (Kapor 1996, S. 1) und weiter unten: »I would claim that software design needs to be recognized as a profession in its own right, a *disciplinary peer* to computer science and software engineering, a *first-class member* of the family of computing disciplines.« (Ebda., S. 5; meine Hervorhebungen)

Diese Forderung nach einem Vorrang des Designs vor der technischen Realisierung wird mit einer Parallele zur Architektur, die Vorrang vor dem Bauingenieurwesen und Bauhandwerk hat, begründet. Der Software-Designer soll passend zum Nutzungskontext gestalten, bevor die technische Umsetzung beginnt. Prototyping-Techniken können helfen, Gestaltungsalternativen zu erkunden, bevor die begonnene Realisierung ihre technischen und wirtschaftlichen Sachzwänge ausüben kann.

Auch Nike hat mehrfach darauf hingewiesen, dass neben der in der Software-Ergonomie dominierenden Evaluation die Gestaltung von großer Bedeutung ist (Nike 2003). Wenn er formuliert, dass »Software-Ergonomie im Design aufgeht oder untergeht«, macht er deutlich, dass der kreative Anteil der Gestaltung bisher eher vernachlässigt wurde. Nichtsdestotrotz muss jede neu gefundene Gestalt einer software-ergonomischen Evaluation in Bezug auf die Kontextpassung unterzogen werden. Die gut entwickelten Evaluationsmethoden sind hier ein nützliches Werkzeug.

Die Anstrengungen im Bereich des sog. *Usability Engineering* zielen auf die systematische Einbeziehung der Benutzbarkeitsproblematik in den Entwicklungsprozess ab. Hier wird zwar nicht von Gestaltung/Design gesprochen, aber es geht um die Gestaltung der MCI vor der technischen Realisierung. Das Label »Engineering« soll wohl den »harten« Software-Ingenieuren deutlich machen, dass Gestaltung auch einen systematischen Teil hat und zum Software Engineering passt. Beim Usability Engineering wird inzwischen in der Norm ISO 13407 (DIN 2000) ein multidisziplinäres Vorgehen gefordert, bei dem Designer ihren Platz haben.

Die Frage nach der Notwendigkeit von Gestaltungskompetenz in einer eigenständigen Gestalterrolle in Softwareentwicklungsprojekten blieb in der Praxis lange unbeantwortet, auch wenn es Befürworter und erste Anzeichen für eine Akzeptanz gibt. So wird in Winograd (1996) vom *Software-Designer* und in Winograd (1997) von *Interaction Design* gesprochen. Oberquelle (2002, 2003) entwirft das Bild eines *Useware-Designers*, der eine Brückenfunktion zwischen Nutzungskontext und technischer Konstruktion hat und Kompetenzen ähnlich denen eines Architekten haben sollte: Kommunikationsfähigkeit gegenüber den Kunden, benutzergerechte Konzeption unter Beachtung des technisch Möglichen und Aufsicht über die fachgerechte Umsetzung (»Bauleitung«).

Seit einiger Zeit hat eine 3. Phase begonnen, die mit der zweiten überlappt. Interaktive Systeme sind nicht mehr nur im Arbeitsleben im Einsatz, sondern verbreiten sich in alle Lebensbereiche, vor allem auf der Basis des Internet und neuer technischer Optionen. Gestaltungsfeld und Gestaltungsoptionen haben sich stark erweitert:

- Mobile Anwendungen erfordern beispielsweise ganz neuartige Lösungen, da die Geräte Platz- und Handhabungsprobleme mit sich bringen. Hier muss eine passende Synthese von physischer Gestalt und Software gefunden werden. Interessanterweise nimmt die neueste Fassung der ISO 9241 bereits auf solche Umstände Bezug, wenn die Benutzungsschnittstelle definiert wird als »Alle Bestandteile eines interaktiven Systems (Software und Hardware), die Informationen und Steuerelemente zur Verfügung stellen, die für den Benutzer notwendig sind, um eine bestimmte Aufgabe mit dem interaktiven System zu erledigen.« (DIN 2006, S. 7)
- Neue Möglichkeiten der Sensorik und Aktorik stehen für die Gestaltung zur Verfügung. Beispielsweise bietet die Positionssteuerung in 3D-Szenen über einen flexibel beweglichen Stuhl (ChairIO; Beckhaus et al. 2005) eine leicht erlernbare Steuerungsmöglichkeit, bei der die Hände für andere Aufgaben frei bleiben.
- Inzwischen werden interaktive Systeme in Kontexten genutzt, in denen es keine von außen vorgegebenen Aufgaben gibt, für die passende Funktionalität zu gestalten wäre, z.B. beim Surfen im Web oder bei kooperativen Spielen. Auch die Benutzer sind weitgehend unbekannt und heterogen.
- Heterogenität der Benutzer spiegelt sich auch in der Forderung nach *Barrierefreiheit (accessibility)* wieder.
- Die MCI hat die *Emotionen* der Benutzer als zusätzliches Phänomen entdeckt, dem durch Gestaltung Rechnung getragen werden muss. Norman (2004) spricht von »emotional design«; andere entwickeln Verfahren zur Messung der sog. hedonischen Qualität (Hassenzahl et al. 2003). Wenn heute von der Anmutung einer Website gesprochen wird, von Branding und User Experience, von »joy of use« dann wird deutlich, dass die Gestaltungsaufgabe nicht länger von Softwareentwicklern nebenbei erledigt werden kann.

Die Aufgabe der Gestaltung tritt immer deutlicher hervor. Bei der Gestaltung von Webseiten ist bereits eine berufliche Differenzierung zwischen Interface-Designern, Informationsarchitekten und Implementierern gang und gäbe. Beim Webdesign zeigt sich eine interessante Entwicklung. Zunächst wurde die ästhetisch aufregende Gestaltung durch die Gruppe der Grafikdesigner eher überbetont, Forderungen nach Usability eher als lästig empfunden. Die pointierte Kritik von Jakob Nielsen an Fehlentwicklungen im Webdesign (Nielsen 2000) wurde häufig zurückgewiesen. Inzwischen scheint sich eine sinnvolle Kombination von Usability und Ästhetik durchzusetzen.

Man spricht hier blumig auch von *user experience design*. Die Design-Disziplinen nehmen sich dieses Themas an und entwickeln eigene Ansätze, um interaktive Systeme ästhetisch ansprechend zu machen und die Aufmerksamkeit der Benutzer zu fesseln. Eine interessante Sammlung von Beiträgen aus dieser Richtung findet man in Buurman (2005).

4 Das Gestaltungsproblem der Zukunft

Betrachtet man die aktuelle Situation der MCI, so kann man sagen, dass es in Zukunft nicht mehr darum gehen wird, Benutzern ein technisches System wie den PC mit Desktop-Oberfläche und konsistenten Anwendungen zu »verkaufen«, d.h. verständlich zu machen. Die Vermittlung rein technisch orientierter Architekturen und Funktionsangebote gelingt kaum. Eine Aufblähung der Funktionalität birgt die Gefahr der *Featuritis*, wie beim Schweizer Armeemesser – es ist ein Notbehelf in vielen Situationen, aber in keiner Situation das Werkzeug der Wahl durch den Experten – und steht daher meistens im Widerspruch zu Nutzerinteressen.

Was ist zu gestalten?

Es wird darum gehen, Computerleistung in mannigfacher Weise für unterschiedlichste Kontexte passend verfügbar zu machen. Hierfür stehen heute vielfältige Mittel zur Verfügung. Einerseits sind Rechenleistung und Speicherkapazität keine Engpässe mehr. Andererseits stehen neue Möglichkeiten für Sensorik und Aktorik zur Verfügung, die über die übliche Visualisierung auf Bildschirmen und die Eingabe über Tastatur und Zeigergeräte hinausgehen. Gleichzeitig sind sie eine Verführung zu technikverliebter Gestaltung.

Ben Shneiderman fordert mit seiner Vision von *Leonardo's Laptop* eine kopernikanische Wende der Gestaltung, wie sie seiner Meinung nach auch Leonardo da Vinci gesehen hätte: »The old computing was about what computers can do; the new computing is about what people can do.« (Shneiderman 2002).

Für Benutzer relevante Ziele hat bereits Alan Cooper (Cooper 1996, S. 12) herausgearbeitet: Sie wollen nicht dumm erscheinen, keine großen Fehler machen, eine gute Portion Arbeit schaffen und Spaß dabei haben, zumindest nicht gelangweilt sein. Auf jeden Fall wollen sie sich nicht mit dem Interface an sich beschäftigen, sondern möglichst ungehindert ihre Ziele verfolgen. Dabei stehen Ihnen mehr Wahrnehmungs- und Aktionsmöglichkeiten zur Verfügung als bisher genutzt werden. Kognitive Möglichkeiten und Schranken (Gedächtnisleistungen, Vorstellungsvermögen,

Lerntempo, Beharrungsvermögen als Ausdruck von kognitiver Ökonomie, kaum Lernen auf Vorrat) sind zu berücksichtigen. Für die Gestaltung bedeutet dies vor allem, permanente Überforderungen und Unterforderungen zu vermeiden. Herausforderungen durch Lernen sind kein Problem, solange sich dies für routinierte Nutzung auszahlt.

Der visionäre John Seeley Brown hat bereits 1996 die Forderung nach Einfachheit erhoben (»Keeping it simple«, Brown/Duguid 1996).

Shneiderman (2002) und Raskin (2002) enthalten die Aufforderung zur Gestaltung nach menschlichem Maß: Interfacegestaltung wird nur dann erfolgreich sein, wenn sie sich an menschlichen Zielen, Fähigkeiten und Begrenzungen orientiert. D. Norman (2004) hat darauf hingewiesen, dass neben der bisher vor allem berücksichtigten Kognition auch der Aspekt der Emotionen eine Rolle bei der Interfacegestaltung spielt. Gestaltung nach menschlichem Maß bedeutet insbesondere, umfassendes Wissen über Ziele, Fähigkeiten und Grenzen von Benutzern einzubeziehen.

Bolter und Gromala (2003) weisen darauf hin, dass selbst bei einer Zielsetzung »*Through the Interface*« (Bødker 1991) mit Konzentration auf Aufgabenorientierung und Funktionalität die Benutzungsschnittstelle nicht verschwindet und der Benutzer sich in ihr widerspiegelt. Sie mahnen eine integrierte Gestaltung an. Eibl (2005) argumentiert, dass erst die Ästhetik die Gebrauchstauglichkeit spürbar macht. Die Idee des Branding von hinsichtlich Funktionalität sehr ähnlichen interaktiven Systemen kann als Aufgabe verstanden werden, die Ästhetik und Benutzbarkeit in einer einheitlichen attraktiven Gestalt zur Wirkung zu bringen. Im Zusammenhang mit Web-Anwendungen macht Bauer-Wabnegg (2006, S. 145) deutlich, dass der erste Eindruck entscheidend sein kann, niemals eine zweite Chance erhält und entsprechend attraktiv sein muss. Ohne Berücksichtigung der Ästhetik wird das kaum gelingen.

Ein spezielles Problemfeld stellen humanoide Komponenten von Benutzungsschnittstellen dar, sog. Avatare. Sobald ein Computer kommunikatives Verhalten zeigt, wird ihm von Benutzern eine gewisse Persönlichkeit zugeschrieben (vgl. Reeves/Nass 1996). Avatare verstärken diese Zuschreibung und rufen durch die Personalisierung spezielle emotionale Reaktionen hervor. Hier werden ggf. sogar Gestaltungs Kompetenzen benötigt, wie sie im Bereich von Film entwickelt wurden. Eine Gestaltung, die die Begrenztheit von Avataren sichtbar macht, ist eine besondere Herausforderung.

Zunehmend wird das Gestaltungsproblem auch als ein Problem der Semiotik gesehen. Sieckenuis de Souza (2005), Nake und Grabowski (2006) und auch Buurman (2006) betrachten interaktive Systeme als Systeme, die in den Dimensionen Zeichen, Bedeutung und Pragmatik und auch Ästhetik zu betrachten sind.

Wer soll gestalten?

Bessere Interfaces können nur durch ganzheitliche, disziplinübergreifende Gestaltung unter Berücksichtigung der Nutzung entstehen. Die Gestaltung muss bei den Werten und Zielen der Nutzer beginnen, die bereitzustellenden Dienste herausarbeiten und ein schlüssiges konzeptuelles Modell bereitstellen, welches über die Benutzungsoberfläche transportiert wird. Die Wirksamkeit der Gestaltung - inklusive emotionaler Aspekte - muss systematisch evaluiert und beobachtet werden. Die Gestaltung umfasst alle diese Aspekte und schließt die Gestaltungslücke - bis sich neue Anforderungen und Möglichkeiten ergeben. Aus heutiger Sicht ist die Idee von Winograd und Kapor, dass *Software*-Design notwendig ist, schon wieder zu eng, wenn man bedenkt, dass das Zusammenspiel von Hardware und Software in eingebetteten Systemen eine integrierende Sicht erfordert. Interaktionsdesign im Sinne der neuesten Fassung von ISO 9241 (DIN 2006) deckt auch diesen Bereich ab.

Die Notwendigkeit einer Design-Perspektive wird zunehmend vertreten (Löwgren/Stoltermann 2004). J. Thakara (2005) betont die Notwendigkeit, die computergestützten Dienste stärker in den Blick zu nehmen und von Betriebssystemoberflächen und Anwendungen wegzukommen sowie Design in einem breiteren Kontext zu betrachten. Nach Bauer-Wabnegg zielt »das Kommunikationsdesign [...] der Zukunft als integriertes Design vornehmlich auf soziale Interaktionsmuster in medialisierten und virtuellen Räumen« (Bauer-Wabnegg 2005: 151) und es geht auch um Aufmerksamkeitsbindung (»War of Eyeballs«).

Bessere Gestaltung ist auf jeden Fall ohne *Designkompetenz* kaum denkbar. Vermutlich werden fachübergreifende Designteams benötigt. Interfacegestaltung muss eine eigene verantwortungsvolle Aufgabe mit hoher Priorität und Entscheidungskompetenz im Systementwicklungsprozess werden, die Designkompetenz erfordert. Wo und wie Designkompetenz am besten vermittelt wird, ist eine offene Frage. Bereits Kapor (1996) weist darauf hin, dass Designkompetenz nicht in Vorlesungen gelehrt werden kann; vielmehr werden Werkstätten benötigt, in denen in einer Art Meister/Lehrlings-Verhältnis Kompetenz ausgeprägt werden kann. Interfacegestaltung sollte auch in der Informatikausbildung einen größeren Stellenwert erhalten und muss durch entsprechende interdisziplinär orientierte Forschungsgruppen abgesichert werden. In diese Richtung argumentiert auch Bauer-Wabnegg (2006). Hier besteht in Deutschland ein großer Nachholbedarf.

Wie sollte Gestaltung im Prozess der Systementwicklung positioniert werden?

J. Raskin fordert »Once the product's task is known, design the interface first; then implement to the interface design.« (Raskin 2000, S. 5). In diesem Zusammenhang spielen Ergonomie-Normen als Minimalanforderungen und formative Evaluationen als Handwerkszeug des Gestalters eine besondere Rolle. In diesem Sinne geht eine eng verstandene Software-Ergonomie im Design auf (Nake 2003). Aber Interfacegestaltung sollte nicht zum Spielfeld von solchen Designern werden, die sich primär als Künstler verstehen und Benutzer mit immer neuen, *coolen* und oberflächlichen Ideen überfordern. Wirksamkeit, Einfachheit, Transparenz und Ästhetik müssen zu einer Synthese gebracht werden.

Gibt es längerfristige »Lösungen« für das Gestaltungsproblem ?

Das Gestaltungsproblem kann nie endgültig gelöst werden, weil es Teil eines dynamischen Wechselwirkungsgefüges ist (vgl. Abb. 1). Wahrnehmungs- und Interaktionsgewohnheiten sind ebenso im Fluss wie technische Optionen. Interface-Gestaltung ist Teil einer schrittweisen Ko-Evolution von Benutzern, Nutzungskontexten und Systemen. Evolution braucht Zeit. Die Änderungsgeschwindigkeit wird allerdings nicht allein durch das Tempo der immer neuen technischen Möglichkeiten bestimmt. Kulturelle Metaphern und Organisationsformen zeigen Beharrungsvermögen. Ungeduld mit den »zu langsamen« Menschen wird nichts bewirken. Allerdings wird die Überwindung der seit vielen Jahren bekannten Grenzen der Desktop-Systeme von manchen Nutzern mit Ungeduld erwartet.

Bei der Vielfalt der Benutzer, Nutzungskontexte und technischen Möglichkeiten ergeben sich ständig neue Freiheitsgrade. Interaktive Systeme sollten eher offen gestaltet werden und den Benutzern Spielräume lassen, um sie anzupassen und auch in nicht vorgesehener Weise zu nutzen (»intelligenter Missbrauch«) - und auch so Anstöße für neue Interfaces zu ergeben. Überzeugende Ergänzungen und Verbesserungen werden sich mit einiger Verzögerung durchsetzen. Sie können durch Marktmacht vermutlich behindert werden.

5 Fazit

Computer liefern das digitale Material, mit dem flexibler als je zuvor gestaltet werden kann. Dass bewusste Gestaltung das reine Bauen ergänzen

muss, steht für viele außer Frage. Man kann sich hier Bauer-Wabnegg (2005, S. 154) anschließen: »Es ist an der Zeit, die bisher eher getrennten Fachkulturen Wissenschaft und Gestaltung enger zusammenzuführen und gemeinsame Ergebnisse anzustreben. Wobei es nicht darum geht, das jeweils Fremde zu assimilieren, sondern vielmehr darum, es arbeitsteilig zu erschließen.« Die Informatik wird sich mit dieser Verschiebung ihrer Rolle befassen müssen. Ob Informatiker zu Gestaltern werden können, hängt von der erworbenen Gestaltungskompetenz ab. Zumindest ist ein besseres wechselseitiges Verständnis notwendig.

Ob sich mit dieser Designorientierung die Anfänge einer neuen Designkultur zeigen, ist eine spannende Frage. Buurman (2005, S. 12) deutet an, dass Designer die »engineers of experience« werden, möglicherweise nicht nur beim Interface Design, sondern sogar beim »Design of Life«. Norbert Bolz wagt sogar einen noch allgemeineren Ansatz und propagiert in seinem *Design Manifest des 21. Jahrhunderts* (Bolz 2006), dass BANG-Design nicht nur die Bits (Interface Design), sondern auch Atome, Neuronen und Gene erfassen wird.

Ist die Gestaltung von Benutzungsschnittstellen nur erst ein Anfang? Ist Gestaltungskompetenz in allen Bereichen gefragt? Könnte das menschliche Maß auf der Strecke bleiben?

Ich bedanke mich für kritische Kommentare bei Timo Göttel, Matthias Müller-Prove, Hartmut Obendorf und dem Herausgeber.

Literatur

- Bauer-Wabnegg, W. (2005): »War of Eyeballs. Design in der Mediengesellschaft«. In: Eibl, M., Wolff, C., Womser-Hacker, C. (Hrsg.). »Designing Information Systems. Festschrift für Jürgen Krause«. Schriften zur Informationswissenschaft, Band 43, Konstanz, S. 145 - 155.
- Beckhaus, S./Blom, K./Haringer, M. (2005): »Intuitive, Hands-free Travel Interfaces for Virtual Environments«. In: Proc. IEEE VR2005, Workshop »New directions in 3D User Interfaces«. Aachen, S. 57-60.
- Bødker, S. (1991): »Through the Interface - a Human Activity Approach to User Interface Design«. Hillsdale, N J.
- Bolz, N. (2006). »Bang Design. Design Manifest des 21. Jahrhunderts«. Hamburg.
- Bolter, J.D./Gromala, D. (2003): »Windows and Mirrors: Interaction Design, Digital Art, and the Myth of Transparency«. Cambridge, MA./London.

- Brown, J.S./Duguid, P. (1996): »Keeping It Simple«. In: Winograd (1996), S. 129-145.
- Buurman, G.M. (Ed.) (2005): »Total Interaction. Theory and practice of a new paradigm for the design disciplines«. Basel.
- Cooper, A. (1995): »About Face. The Essentials of User Interface Design«. Foster City, CA.
- Cooper, A. (1999): »The Inmates are Running the Asylum. Why High-Tech Products Drive Us Crazy and How to Restore the Sanity«. Indianapolis, IN.
- Cooper, A./Reimann, R. (2003): »About face 2.0: The essentials of interaction design«. Indianapolis, IN.
- DIN (2000): DIN EN ISO 13407, Ausgabe:2000-11. »Benutzer-orientierte Gestaltung interaktiver Systeme« (ISO 13407:1999); Deutsche Fassung EN ISO 13407:1999, Berlin.
- DIN (2006): DIN EN ISO 9241-110, Ausgabe: 2006-08. »Ergonomie der Mensch-System-Interaktion - Teil 110: Grundsätze der Dialoggestaltung« (ISO 9241-110:2006); Deutsche Fassung EN ISO 9241-110:2006, Berlin.
- Eibl, M. (2005): »Natural Design. Some Remarks on the Human Nature and the Design of User Interfaces«. In: Eibl, M./Wolff, C./Womser-Hacker, C. (Hrsg.): »Designing Information Systems«. Konstanz, S. 157 - 170.
- Floyd, C. (1987): »Outline of a Paradigm Change in Software Engineering«. In: Bjerknes, G./Ehn, P./Kyng, M. (Eds.): »Computers and Democracy«. Aldershot, UK, S. 191-210.
- Floyd, C./Reisin, F.-M./Schmidt, G. (1989): »STEPS to Software Development with Users«. In: Ghezzi, C./McDermid, J.A. (Eds.): Proceedings European Conference on Software Engineering. ESEC '89 Berlin, Heidelberg, New York, S. 48-64.
- Greenbaum, J./Kyng, M. (Eds.): »Design at Work: Cooperative Design of Computer Systems«. Hillsdale, NJ.
- Hassenzahl, M./Burmester, M./Koller, F. (2003): »AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität«. In: Ziegler, J./Szwilius, G. (Hrsg.): »Mensch & Computer 2003. Interaktion in Bewegung«. Stuttgart, Leipzig, S. 187-196.
- Johnson, J./Roberts, T./Verplank, W./Smith, D.C./Irby, C.H./Beard, M./Mackey, K. (1989): »The Xerox Star: A Retrospective«. IEEE Computer 22,9, S. 11-28.
- Kapor, M. (1996): »A Design Manifesto«. In: Winograd (1996), S. 1-19.
- Löwgren, J./Stolterman, E. (2004): »Thoughtful Interaction Design. A Design Perspective on Information Technology«. Cambridge, MA, London, UK.
- Nake, F. (2003): »Informatik als Gestaltungswissenschaft. Eine Herausforderung an das Design«. In: Rödiger, K.-H. (Hrsg.): »Algorithmik - Kunst -

- Semiotik. Hommage für Frieder Nake«. Heidelberg, S. 142-163 (Manuskript eines Vortrages von 1998).
- Nake, F./Grabowski, S. (2006): »The Interface as Sign and as Aesthetic Event«. In: Fishwick, P.A. (Ed.) (2006): »Aesthetic Computing«. Cambridge, MA/London, S. 53 - 70.
- Nielsen, J. (2000): »Designing Web Usability. The Practice of Simplicity«. Indianapolis, IN.
- Norman, D.A. (1986): »Cognitive Engineering«. In: Norman, D.A./Draper, S.W. (Eds.): »User Centered System Design. New Perspectives on Human-Computer Interaction«. Hillsdale, NJ, S. 31-61.
- Norman, D. (2004): »Emotional Design. Why we love (or hate) everyday things«. New York.
- Oberquelle, H. (2002): »Useware Design and Evolution: Bridging Social Thinking and Software Construction«. In: Dittrich, Y./Floyd, C./Klischewski, R. (Eds.): »Social Thinking - Software Practice«. Cambridge, MA/London, UK, S. 391-408.
- Oberquelle, H. (2003): »Benutzergerechte Gestaltung der Mensch-Computer-Interaktion: Wissenschaft - Engineering - Kunst?«. In: Informatiktage 2002, Fachwissenschaftlicher Informatik-Kongress, 8. und 9. November 2002 im Neuen Kloster Bad Schussenried, Grasbrunn, S. 34-42.
- Parnas, D.L. (1969): »On the Use of Transition Diagrams in the Design of a User Interface for an Interactive Computer System«. Proc. ACM 24th National Conference, New York, S. 378 - 385.
- Raskin, J. (2000): »The Humane Interface. New Directions for Designing Interactive Systems«. Reading, MA.
- Reeves, B./Nass, C. (1996): »The Media Equation. How People Treat Computers, Television, and New Media Like Real People and Places«. New York.
- Shneiderman, B. (2002): »Leonardo's Laptop. Human Needs and the New Computing Technologies«. Cambridge, MA/London, UK.
- Sieckenuis de Souza, C. (2005): »The Semiotic Engineering of Human-Computer Interaction«. MIT Press, Cambridge, MA/London.
- Thakara, J. (2005). »In the Bubble. Designing in a Complex World«. Cambridge, MA, London, UK.
- Winograd, T. (ed.) with J. Bennett, L. De Young, B. Hartfield (1996): »Bringing Design to Software«. New York/Reading, MA.
- Winograd, T. (1997): »The Design of Interaction«. In: Denning, P. J./ Metcalfe, R. M. (Eds.): »Beyond Calculation. The Next Fifty Years of Computing«. New York, S. 149-161.

VOM PERSÖNLICHEN COMPUTER ZUM SOZIALEN MEDIUM. PARADIGMENWECHSEL DER MENSCH-COMPUTER-INTERAKTION

MATTHIAS MÜLLER-PROVE

Die Art und Weise, in der wir mit Computern umgehen, unterzieht sich einem ständigen Wandel. Der technische Fortschritt sorgt für immer neue Rechnergenerationen, die mehr können als die Vorgängermodelle. Zusammen mit der massenhaften Verbreitung des Internet in Form von E-Mail, Instant Messaging und World Wide Web für Informationsbeschaffung, Unterhaltung und Online-Handel, hat sich dadurch der Charakter der Anwendungen von Forschung und Wirtschaft hin zu Kommunikation und Freizeit verschoben. Die Computer sind auch in Form von Notebooks so leicht geworden, dass man sich in seiner Mobilität kaum noch eingeschränkt fühlt.

Die Bedienung ist hingegen nicht ganz so leicht, obwohl der Einführung des Personal Computers umfangreiche Forschungen voraus gegangen sind, die zu den bekannten grafischen Benutzungsschnittstellen mit überlappenden Fenstern, Menüs und der Schreibtischmetapher geführt haben.

Seither sind die Interaktionsmodelle in ihrem Kern nicht mehr hinterfragt worden, was insbesondere seit dem Erfolg des World Wide Web ein Konglomerat aus Desktop- und Browser-Metapher zur Folge hatte.

Es ist daher dringend geboten, die jeweils zu Grunde liegenden Paradigmen zu untersuchen und unter dem Gesichtspunkt ihres Zusammenwirkens neu zu positionieren.

1 Blick zurück nach vorn

Eine Betrachtung der Entwicklungsgeschichte der modernen grafischen Benutzungsschnittstellen und Webanwendungen ist nicht nur aus historisch-nostalgischen Gründen sinnvoll. Der Blick auf die Frühphase der interaktiven Computersysteme liefert ein unverstelltes Bild der ursprüng-

lichen Ideen und Konzepte. Damit wird es möglich, den Status Quo mit den ihm zugrunde liegenden Visionen zu vergleichen und eventuell Korrekturen für die immer komplexer werdende Schnittstelle zwischen Mensch und Computer vorzuschlagen.

Im Rahmen des folgenden Abschnitts werden stellvertretend für die zahlreichen Pioniertaten der Computerrevolution die Ideen und Konzepte von Vannevar Bush, Theodor Holm Nelson, Douglas Engelbart, Alan Kay und Tim Berners-Lee vorgestellt.

Memex

Vannevar Bush beschreibt in dem Artikel »As We May Think«, dass es Wissenschaftlern kaum noch möglich wäre, effektiv und effizient zu arbeiten, da sie ob der schier Masse an publizierten Fachartikeln die vielfältigen Bezüge und Querverweise unmöglich noch selbst erkennen und nachverfolgen könnten. Der Artikel erschien vor über sechzig Jahren in der Zeitschrift »The Atlantic Monthly« (Bush 1945).

Im zweiten Weltkrieg war Vannevar Bush (*1890 †1974) Direktor des Office of Scientific Research and Development und koordinierte in dieser Funktion die Forschungsaktivitäten von etwa 6.000 Wissenschaftlern, die unter anderem im Manhattan-Projekt die amerikanische Atombombe entwickelten. Zur Lösung des oben angeführten Informationsproblems schlug er die fiktive Maschine »Memex« (hergeleitet aus »memory extender«) vor, die dem Knowledge-Worker¹ seiner Zeit einen performanten Zugriff auf Dokumente geben sollte. Memex basierte auf dem Einsatz von Mikrofilmen. Ähnlich einem Zeitschriftenabo sollten Filmspulen mit den neuesten Veröffentlichungen den Dokumentenschatz des Memex kontinuierlich ergänzen. Im Inneren eines Schreibtisches waren Apparaturen untergebracht, die die Seiten auf mehreren Projektionsschirmen darstellten konnten.

Hatte der Wissenschaftler eine interessante Querverbindung gefunden, wählte er einen kurzen alphanumerischen Code, um die beiden Dokumente dauerhaft miteinander zu verbinden. Durch das Anhängen weiterer Dokumente zu dem jeweiligen Thema entstanden auf diese Art lange Ketten. Die sogenannten Trails entsprechen damit den Gedankengängen und Überlegungen des Forschers bei der Lektüre des Stoffs. Neben der Erstellung von themenbezogenen Trails sollten die Seiten auch mit Anmerkungen versehen werden dürfen, die dann vom Memex abfotografiert, und wie alle anderen Dokumente zur Lektüre und Verknüpfung zur Verfügung standen.

1 Der Begriff »Knowledge-Worker« wurde von Peter Drucker in »Landmarks of Tomorrow« 1959 eingeführt. (Stephan 2006)

Die Relation zwischen Trails und Dokumenten muss nicht eindeutig sein; unter verschiedenen Gesichtspunkten kann ein Dokument durchaus unterschiedlichen Trails angehören. Die sich daraus ergebende Struktur ähnelt verblüffend den Hypertext-Systemen, die ab Mitte der sechziger Jahre unter anderem von Ted Nelson konzipiert und entwickelt wurden.

Hypertext

Im Gegensatz zu Vannevar Bush rückt Ted Nelson (*1937) den Autor anstelle des Lesers in den Mittelpunkt. Er erklärt die Vorstellung für überholt, dass Texte genau so geschrieben werden müssen, wie sie im Allgemeinen auch gelesen werden: nämlich vom Anfang zum Ende. Die Erstellung eines Textes gleicht vielmehr der Erschaffung eines großen Geflechts aus einzelnen Gedanken, die in nahezu beliebiger Reihenfolge geschrieben werden können. Eine streng lineare oder hierarchische Anordnung dieser Gedankenteile sei der Komplexität des Beschriebenen nicht immer angemessen, so dass eine vernetzte Struktur die adäquate Repräsentation darstelle. Damit gibt es auch keine vom Autor vorgegebene eindeutige Reihenfolge mehr. Der Leser darf zwischen verschiedenen Pfaden auswählen und sich so den Korpus des Textes erschließen. Für diese nicht-sequentielle Textgattung prägt Ted Nelson den Begriff »Hypertext« (Nelson 1965).

Augmenting Human Intellect

1962 formuliert Douglas Engelbart (*1925) die Grundzüge seines Lebenswerks über den Einsatz des Computers zur Vergrößerung des menschlichen Intellekts (Engelbart 1962). Während seiner Armeezeit hatte er zuvor als Radar-Offizier erkannt, dass man auf Kathodenstrahlröhren im Prinzip auch Text projizieren konnte. Zusammen mit der formalen Logik eines Computers ergab dies ein neues symbiotisches System aus Mensch und Maschine, das sich über die kommenden Jahrzehnte in einer Co-Evolution weiterentwickeln sollte, um die wichtigen und drängenden Probleme der Welt zu lösen.

Diesen vollmundigen Zielen ließ Engelbart beeindruckende Taten folgen. Am Stanford Research Institute leitete er in den 1960er Jahren das Augmentation Research Center (SRI-ARC), dessen Resultate er im Dezember 1968 der Fachöffentlichkeit vorstellte. NLS/Augment² war

2 NLS leitet sich von oNLine System her. Im Gegensatz zum damals üblichen Offline/Batch-Betrieb, bei dem der Anwender einen Stapel Lochkarten beim Operator abgab, um nach einem Tag das Ergebnis des Programms zu erhalten, war der Anwender in direktem interaktiven Kontakt

Textverarbeitung, Outliner, Hypertext und E-Mail-System in einem. Darüber hinaus stand Anwendern, die zusammen am selben Dokument arbeiteten, eine parallele Videokonferenzschaltung zur Verfügung. Auch die Computer-Maus und ein spezielles 5-Tasten-Keyboard für Kommandos und kurze Texteingaben waren bahnbrechende Konzepte, die der Präsentation insgesamt den Titel »Mother of all demos« einbrachten. Douglas Engelbart hat es mit seinem Team geschafft, in weniger als 10 Jahren ein Computersystem zu entwickeln, das die wichtigsten Konzepte der modernen Systeme enthielt und in einigen Aspekten den heutigen Systemen immer noch überlegen ist.

Unter Douglas Engelbarts Ägide ist im September 2006 HyperScope 1.0 als aktuelle Nachfolgeversion von NLS/Augment fertiggestellt worden. Es ist der ambitionierte Versuch, die (Hyper-)Textfunktionen von NLS/Augment in einem Web-Browser zur Verfügung zu stellen.³

Personal Computing

Die Kopiererfirma Xerox gründete 1970 das Palo Alto Research Center (Xerox PARC), um für den Einzug der Computers in die Bürowelt gewappnet zu sein. Als ein Jahr später Alan Curtis Kay (*1940) zum Xerox PARC kam, prägte er mit seiner Vision des »Dynabook« auf unvergleichliche Weise die Forschungsagenda des Instituts (Kay 1972). Computer sollten persönlich werden, was bedeutete, dass jeder Anwender einen eigenen tragbaren Computer besitzen sollte. Eine Utopie zu einer Zeit, in der Großrechner im Timesharing-Betrieb genutzt wurden. Und sie sollen auch von Kindern zu benutzen sein, um ihnen damit die Welt der Mathematik und Programmierung spielerisch nahe zu bringen.⁴ Das Konzept des Dynabook ist mit den heutigen Notebook-Computern in Bezug auf Rechenleistung und Mobilität Realität geworden. In den 1970er Jahren mussten dazu aber erst noch die technischen Grundlagen geschaffen werden.

Der neu entwickelte Arbeitsplatz-Computer Xerox Alto von Butler Lampson und Chuck Thacker konnte dank der neuen Rastergrafik von Ron Rider verschiedene Zeichensätze auf dem Bildschirm darstellen – eine Voraussetzung für die Gestaltung ansprechender Dokumente am Computer. Diese Bildschirmdateien vermochte Gary Starkweather mit seiner

mit Engelbarts Computersystem. Die Bedeutungsverschiebung des Begriffs »online« hin zum Verbindungsstatus mit dem Internet hat mehr als eine Dekade später stattgefunden.

3 <http://hyperscope.org/>, März 2008

4 So luden Alan Kay und Adele Goldberg Schulkinder ans Xerox PARC ein, mit denen sie gemeinsam Smalltalk-Programme erstellten. Ein Teilnehmer in dem Projekt war Steve Case, der spätere Gründer von AOL.

innovativen Laser-Technik in hoher Qualität zu Papier zu bringen. Damit kam zum »What-You-See« das »What-You-Get«, die beiden Aspekte des WYSIWYG-Prinzips. Für die Datenverbindung zwischen den Alto Computern und dem Laserdrucker sorgte schließlich Robert Metcalfes Ethernet.

Aus all diesen Komponenten entwickelte Xerox ein kommerzielles System, das 1981 als »Xerox 8010 Information System« der Öffentlichkeit vorgestellt wurde. Der anfängliche Preis in Höhe von \$16.596 trug nicht gerade zur Verbreitung des auch Xerox Star genannten Systems bei, zumal wesentliche Funktionen ja erst im Netzwerkbetrieb genutzt werden konnten und man somit mehr als einen Computer kaufen musste.

Zwei Jahre danach kam Apple Lisa für \$9.995 auf den Markt. Der Computer passte zusammen mit einem 12"-Monitor in ein Gehäuse. Er stellte sich aber bald als zu langsam für die ausgefeilte Interaktion mit den Programmen heraus, so dass die Nutzung dem Anwender leider einiges an Geduld abverlangte.

Der Apple Macintosh löste dann 1984 endgültig die Desktop-Publishing-Revolution der achtziger Jahre aus. Er war bereits für \$2.500 zu haben, da er viel einfacher ausgestattet war, als das Apple Lisa System. Eine Festplatte war in der Basisausstattung nicht vorgesehen – stattdessen gab es nur ein 400 KB Diskettenlaufwerk. Innerhalb der ersten zwei Jahre wurden eine halbe Million Macintosh Computer verkauft. Der Mac wurde damit zum ersten kommerziell erfolgreichen Personal Computer mit einer grafischen Benutzungsschnittstelle.

Zu der mitgelieferten Software gehörte ab 1987 HyperCard, ein Hypertext-System von Bill Atkinson. Texte und Schwarzweißbilder wurden auf Karten gleicher Größe verteilt, die den 9" Monitor des Macintosh ausfüllten, und zwischen denen sich mit einfachen Befehlen Sprünge definieren ließen. Die Macintosh-Nutzer erstellten zu den verschiedensten Themen HyperCard-Stacks, die sie über Disketten und Online-Foren verbreiteten. Für Robert Cailliau sollte HyperCard zur Inspiration werden, zusammen mit Tim Berners-Lee das World Wide Web zu entwickeln.

World Wide Web

Fast alle bis 1985 entwickelten Hypertext-Systeme waren nicht netzwerkfähig.⁵ Danach gab es einige Systeme für LAN-Architekturen; aber erst 1989 stellte Tim Berners-Lee (*1955) am Kernforschungszentrum CERN in Genf den Projektantrag zum World Wide Web (Berners-Lee 1989), das die Infrastruktur des Internet nutzbar machte, um eine Hyper-

5 Die einzige nennenswerte Ausnahme ist NLS/Augment, das 1969 der zweite Knoten des ARPAnet wurde.

textseite auf einem Web-Server mit einer anderen Seite auf einem anderen Server über das Hypertext Transfer Protocol (HTTP) zu verbinden. Die Seiten mussten nur einer einfachen ASCII-basierten Syntax folgen, die von den Wissenschaftlern auf jedem Betriebssystem erstellt werden konnten. Mit der Öffnung und Standardisierung der HTML-Syntax umging Berners-Lee die proprietären Einschränkungen, die die heterogene Hardware-Situation am CERN mit sich brachte. Später sollte sich genau dieser Umstand als einer der Erfolgsfaktoren des Webs herausstellen, da jedes kommerzielle und nichtkommerzielle Entwicklungsteam einen eigenen Web-Server oder Browser entwickeln durfte, ohne Lizenzgebühren fürchten zu müssen.

Der Web-Client ist zunächst einmal ein normales Programm, das Webseiten, die mit einer eindeutigen URL⁶ identifiziert werden, in einem Fenster darstellen kann. Bei dem von Tim Berners-Lee und Robert Cailliau entwickelten Programm WorldWideWeb (später umbenannt in Nexus, um Verwechslungen mit dem World Wide Web an sich auszuschließen) handelte es sich noch um eine Kombination aus Web-Browser und Web-Editor. Dabei machte die Klassenbibliothek des NeXT-Computers⁷ es den beiden verhältnismäßig einfach, eine Ableitung der Klasse NSTextEdit mit Hypertext-Funktionalität auszustatten. Das Ergebnis war im Dezember 1990 ein funktionstüchtiger Prototyp, der HTML-Seiten nicht nur anzeigen, sondern auch im Layout-Modus editieren konnte.

Als in den folgenden Jahren Marc Andreessen Implementierungen unter der Windows- und Mac-Plattform in Arbeit hatte, hielt er die Nexus' Bearbeitungsfunktionen nicht für primär wichtig, so dass NCSA Mosaic ein reiner Browser wurde. Bis heute hat sich dieses Manko in der Browser-Technologie gehalten, da keine Nachfolgeversion von Mosaic und Netscape Navigator Webseiten bearbeiten kann. Auch Programme anderer Hersteller blieben reine Browser zum Betrachten der Webseiten.⁸

Die kurzen Einblicke können nur Schlaglichter in die Entwicklungsgeschichte der modernen grafischen Benutzungsoberflächen sein. Bushs Memex steht dabei für die Unterstützung des Wissensarbeiters, der versucht die Informationsflut zu beherrschen, indem er neue Verknüpfungen zwischen den Dokumenten anlegen darf. Das Hypertextkonzept von

6 Das ursprüngliche Adressschema des World Wide Web spricht im Gegensatz zu Uniform Resource Locator (URL) noch vom Universal Resource Identifier (URI). Gemeint ist in beiden Fällen eine Zeichenkette, die ein Protokoll, einen Server-Namen und Parameter zur Identifizierung eines Objektes enthält. Ein Beispiel wäre »<http://www.w3.org/History/1989/proposal.html>«.

7 NeXT war die zweite Computerfirma des Apple-Gründers Steve Jobs.

8 Die Geschichte des World Wide Web wird ausführlich in Gillies/Cailliau 2000 beschrieben.

Ted Nelson befreit zunächst den Autor von der Linearität des Werkes und damit auch den Leser von der Erwartung das Geschriebene in vorgegebener Reihenfolge zu lesen. Douglas Engelbart hat sich mit NLS/ Augment insbesondere der Unterstützung von Arbeitsgruppen gewidmet, während Alan Kay den Fokus etwas mehr auf das Verhältnis Anwender-Computer legt. Mit dem World Wide Web von Berners-Lee kommen dann vermeintlich alle Aspekte zusammen. Sein Ansatz war pragmatisch und daher erfolgreich. Auf der anderen Seite ist aber versäumt worden, die damals dominierenden Interface-Paradigmen für das Web zu erweitern.⁹

2 Der Persönliche Schreibtisch

Konstituierend für moderne grafische Benutzungsschnittstellen à la Mac OS X, Microsoft Windows XP und dem Nachfolgesystem Windows Vista oder KDE und GNOME auf dem Linux oder Solaris Betriebssystem sind die vier Komponenten Fenster, Icons, Menüs und Zeigegerät sowie fünftens die Schreibtisch-Metapher. Diese Schnittstellen seien hier unter dem Begriff *Desktop-Systeme* zusammengefasst.

Die Entwicklung der Desktop-Systeme reicht bis in die fünfziger Jahre zurück. Zuerst fand im militärischen Bereich die Lightgun, ein Vorläufer des Lichtgriffel, als Zeigegerät Verwendung (siehe oben *Nake*). Douglas Engelbart und William (Bill) English haben dann 1963 am SRI-ARC die Maus entwickelt. Geteilte Fenster zur Darstellung von grafischen Informationen auf einer Kathodenstrahlröhre wurden im gleichen Jahr von Ivan Sutherland für sein System Sketchpad benutzt (Sutherland 1963). Die ersten überlappenden Fenster und Popup-Menüs wurden von Alan Kay am Xerox PARC für das Smalltalk Entwicklungssystem eingeführt. Mit der Anlehnung an überlappende Papierseiten für die Fensterverwaltung kann nun mehr Inhalt auf begrenztem Bildschirmplatz dargestellt, sowie auf einfache Art zwischen Programmen hin und her gewechselt werden. Schließlich komplettierte David Canfield Smith Mitte der siebziger Jahre mit seiner Forschung über Icons das WIMP-Quartett.¹⁰

9 Für eine ausführliche historische Betrachtung sei auf Michael Friedewalds (1999) »Der Computer als Werkzeug und Medium« und auf »Vision and Reality of Hypertext and Graphical User Interfaces« (Müller-Prove 2002) verwiesen.

10 »WIMP« steht für die vier Aspekte Windows, Icons, Menus und Pointing Device. Die Abkürzung könnte gut von den Anhängern des Unix-Betriebssystems stammen, die als Benutzer eines Kommandozeilen-Interfaces auf die Konnotation zum englischen Begriff »wimp« für Mausbenutzer zielten.

Mit Icons, die sich aus der Bürowelt herleiten, kommt man dann zur »Schreibtisch-Metapher«, die ursprünglich als »Physical-Office-Metapher« eingeführt wurde. Man versteht darunter ein Paradigma der grafischen Benutzungsschnittstellen, über das die Dinge und Tätigkeiten des realen Büroalltags im Computer-Interface eine Entsprechung finden (Smith 1982). Dadurch sollte Benutzern, die im Umgang mit Computern noch unerfahren sind, die Scheu genommen und gleichzeitig auf ihr Alltagswissen rekurriert werden, um die Vorgänge im Computer gleichsam zu verbergen und vorstellbar zu machen. So wird die Systemsicht, die für die Entwickler eine wichtige Rolle spielt, durch ein anderes mentales Modell beim Anwender ersetzt. Die Bürowelt mit Dokumenten, Ordnern, Aktenschränken, Post-Ein- und Ausgangskörben, einem Papierkorb und sonstigen (Schreib-)Werkzeugen bot sich an, da hier auch die Kunden für die ersten kommerziellen Systeme Xerox Star und Apple Lisa gesucht wurden.

Interaktionskonventionen bei Desktop-Systemen

Alle Anwendungen auf Desktop-Systemen folgen den gleichen Grundprinzipien. Dabei sind die Darstellung von Objekten auf dem Bildschirm mittels Icons und Fenstern und deren Manipulation per Menübefehl oder direkt per Maus wesentlich. Aus Sicht des Nutzers ist die daraus folgende Konsistenz begrüßenswert, da er zügig mit unbekanntem Programmen umgehen und sogar mit geringer Einarbeitungszeit zu einem anderen Betriebssystem wechseln kann. Die Konventionen aus den Bereichen Mausinteraktion, Menüstruktur, Applikationsmodell und Dateistruktur werden im Folgenden besprochen, um sie später mit dem Interaktionsmodell des Webs zu vergleichen.

Für die Maus sind drei Interaktionsformen besonders hervorzuheben. Erstens selektiert ein einfacher Mausklick ein Icon für nachfolgende Befehle in der Objekt-Verb-Kommandosyntax. Zweitens lassen sich per Drag'n'Drop Icons an beliebige Orte verschieben, oder es werden spezielle Aktionen ausgelöst, indem man ein Icon auf ein anderes zieht. Drittens löst der Doppelklick die Öffnen-Funktion eines Icons aus.

Einen Ordner zu öffnen bedeutet, dass sein Inhalt in einem neuen Fenster angezeigt wird, während ein Dokument mit einem speziellen Werkzeug bearbeitet wird. Werkzeuge sind auf den Dokumenttyp spezialisierte Programme, die die Daten der Dateien interpretieren können und sie als Fensterinhalt darstellen. Wird das Dokument gespeichert, überschreibt das Programm die Datei mit den aktuellen Daten gemäß dem jeweiligen Dateiformat.

In der Menüstruktur bietet ein Programm dem Anwender einige standardisierte Aktionen an: Im Datei-Menü gibt es Kommandos zum Erzeugen neuer Dokumente, zum Sichern, zum Drucken und zum Beenden des Programms. Im Bearbeiten-Menü sind die Befehle zum Rückgängigmachen einer Aktion sowie die der Zwischenablage zu finden: Ausschneiden, Kopieren, Einfügen. Der weitaus größte Teil der Menüstruktur ist allerdings speziell auf den Dokumenttyp zugeschnitten. Hier drückt sich die Funktionsvielfalt des Programms aus.

Diese vorherrschende Systemarchitektur ist applikationszentriert; d.h. dass zunächst ein Programm gestartet werden muss, bevor darin das Dokument geöffnet und bearbeitet werden kann. Darum ist auch die Werkzeugmetapher nicht ganz zutreffend, da ein Werkzeug zwar für einen primären Zweck optimiert ist, aber dennoch flexibel für anderes benutzt werden kann. Mit einem Stift kann man zum Beispiel auf verschiedenen Materialien schreiben und ist auf kein spezielles Papier angewiesen.

Im Gegensatz zum applikationszentrierten Modell ist das dokumentenzentrierte Modell des Xerox Star für den Anwender leichter zu begreifen. Beginnend mit einem leeren Dokument entscheidet man erst im zweiten Schritt welche Elemente auf der Seite benötigt werden. Es ist damit möglich, Texte, Grafiken und Tabellenkalkulation auf einer Seite gemischt anzuordnen und die Inhalte an Ort und Stelle zu bearbeiten. Der Anwender muss sich auch nicht zu Beginn seiner Arbeit auf den späteren Verwendungszweck festlegen. Das Drucken eines gestalteten Dokuments oder das direkte Versenden per E-Mail ¹¹ ist mit jedem Dokument möglich.

Das Dateisystem bei heutigen PCs ist hierarchisch strukturiert. Diese Sicht wurde für die Ablagestruktur der Dokumente übernommen, sodass Dateien und Ordner beliebig tief verschachtelt in Ordnern liegen dürfen. Für die Ordnung ist der Anwender selbst verantwortlich, da er jedes Icon an den Ort seiner Wahl bewegen kann. Bei wenigen hundert Dokumenten, die vor 20 Jahren pro Anwender üblich waren, hat dieser Ansatz gut funktioniert – bei Festplattenkapazitäten im Gigabytebereich ist es für den Anwender nicht mehr möglich, den Überblick über mehrere Hunderttausend Dateien zu behalten und gezielt Dokumente wieder zu finden.

11 Im applikationszentrierten Modell erstellt man mit einem Textprogramm sein Dokument. Dann öffnet man das E-Mail Programm, erzeugt eine neue E-Mail und hängt das Dokument an. Im einfacheren dokumentenzentrierten Modell erzeugt man ein Dokument, schreibt seinen Text und schickt es an einen Empfänger, indem man das Dokument-Icon auf das Postausgangs-Icon des Schreibtisches zieht.

Seit der Markteinführung des Apple Macintosh hat es kein Hersteller geschafft, qualitative Neuerungen im Bereich der Desktop-Systeme zu etablieren. Zwei größere Anstrengungen von Apple und Microsoft, die ein radikales Umdenken bedeuten würden, seien hier erwähnt. Mit OpenDoc hatte Apple Mitte der 1990er Jahre versucht, sich vom applikationszentrierten Modell zu lösen. Kleine und leicht zu programmierende Module sollten die unterschiedlichsten Informationscontainer eines OpenDoc-Dokuments darstellen und bearbeiten können.¹² Das Projekt scheiterte, da es sich als zu ambitioniert herausstellte und nicht angemessen zu implementieren war. Zehn Jahre später arbeitete Microsoft an der Fertigstellung des neuen Betriebssystems Windows Vista. Es war eigentlich vorgesehen, das hierarchische Dateisystem durch eine relationale Datenbank zu ersetzen, mit der aus Sicht der Anwender viel flexibler auf die Dateien zugegriffen werden kann. Dokumente haben nicht mehr nur einen Ort in der Baumstruktur der Ordner, sondern sie können unter verschiedenen Gesichtspunkten allen Projekten zugeordnet werden, für die sie relevant sind. Die Pläne dazu wurden verschoben, um die Auslieferung des Produkts nicht weiter zu verzögern.

Neben diesen beiden größeren Projekten gab es in den vergangenen zwanzig Jahren nur marginale Weiterentwicklungen: Hierarchische Untermenüs, Kontextmenüs, hierarchische Ordnerstrukturen in der Ordneransicht, Windows-Taskbar und Apple-Dock, Farbe und die Unterstützung eines Mauseisens sind alles Erweiterungen, die das grundsätzliche Modell nicht in Frage stellen, sondern in einzelnen Aspekten versuchen, mit der gewachsenen Komplexität und den Ansprüchen der Nutzer Schritt zu halten.

Im Bereich der Schreibtisch-Metapher ist sogar zu beobachten, dass die ursprünglichen Konzepte erodieren. Dazu drei Beispiele. Beim Xerox Star, der Apple Lisa und lange Jahre auch beim Macintosh galt der Grundsatz, dass Dokument-Icon und Dokument-Fenster nur zwei unterschiedliche Darstellungsformen desselben Objektes sind und folglich das Icon sein Aussehen änderte, sobald das Dokument geöffnet war. Heute kann man einem Dokument-Icon im Allgemeinen nicht mehr ansehen, ob das Dokument gerade in einem Programm geöffnet ist, da es immer gleich aussieht. Noch konsequenter war die Umsetzung dieser Identität zwischen Dokument-Fenster und -Icon bei der Apple Lisa. Die Verwaltung von

12 Unter Microsoft Windows gibt es die Funktion »Object Linking and Embedding« (OLE), die den Versuch darstellt, die positiven Eigenschaften des dokumentenzentrierten Modells mit dem vorherrschenden applikationszentrierten Modell zu verbinden. In der Praxis stellt sich diese Technik allerdings als sehr fehleranfällig und verwirrend für den Anwender dar, da die Benutzungsoberfläche unvermittelt auf den neuen Datentyp umgeschaltet wird.

Fenster, Datei und Dokument wurde vollständig vom Betriebssystem übernommen, so dass der Anwender sich nicht über den Zustand der Datei im Verhältnis zum geöffneten Dokument kümmern musste. Ein manisches Sichern des Dokumentes war überflüssig, da Fenster und Dokument-Icon aus Nutzersicht stets ein und dasselbe waren.¹³

Eine weiteres Konzept, das zunächst aus der realen in die virtuelle Welt übernommen wurde, ist die Eindeutigkeit des Ortes. Objekte befinden sich zu jeder Zeit an genau einer Position. Diesem Grundsatz folgend verschwand ein Dokument vom Schreibtisch der Apple Lisa, nachdem man es auf ein Disketten-Icon gezogen hatte.¹⁴ Auch bei der Darstellung von Ordnern ist dieser Effekt zu beachten. Ein Ordner der realen Welt kann nur an einer Stelle geöffnet sein. Heute ist es auf dem Computer-Schreibtisch allerdings möglich, dass zwei verschiedene Fenster denselben Ordner darstellen. Ein eklatanter Verstoß gegen die Alltagserfahrung, die den Computer-Anwender verwirren kann.

Selbst die Persistenz des Ortes wird weniger penibel im Computer nachgebildet als bei den Desktop-Systemen der ersten Generation. Reale Gegenstände haben die Eigenschaft, an der Stelle zu verharren, an die sie gelegt wurden. Bei der Platzierung von Icons und Fenstern ist es inzwischen auf allen Plattformen aber reiner Zufall, ob man sie nach einigen Tagen oder Wochen noch in derselben Anordnung an der ursprünglichen Stelle vorfindet. Da solche Formationen für den Nutzer durchaus von Bedeutung sein können, ist diese Nachlässigkeit seitens der Entwickler der Systeme kaum zu entschuldigen.

3 Das weltweite Netz

Das World Wide Web besteht aus zahlreichen weltweit verteilten Web-Servern, von denen Web-Clients mittels HTTP Webseiten anfordern, um sie in einem Browser-Fenster darzustellen.

Obwohl das Web-Consortium W3C seit 1995 damit beschäftigt ist, einen Konsens über die verwendete Markup-Sprache herzustellen und Erweiterungen wie Cascading Style Sheets (CSS) und stärkere Strukturen mittels der Extended Markup Language (XML) zu definieren, ist das Web

13 Der Lisa Desktop Manager verwaltete – transparent für den Anwender – für jedes Dokument zwei Dateien. Die zweite Datei diente als persistente Sicherungskopie, die sogar dafür genutzt wurde, das Dokument nach einem System-Neustart wieder im ungesicherten Zustand zu öffnen.

14 Heutige Systeme legen bei dieser Aktion eine Kopie auf dem Zielmedium an; diese Designentscheidung ist sinnvoll, da das die weniger fehlerträchtige Operation ist. Wichtig ist in jedem Fall, dass der Anwender vom System visuelles Feedback erhält.

in seinen Grundprinzipien unverändert. Sein Charakter hat sich allerdings durch die Kommerzialisierung und neue Inhaltsangebote seit Mitte der 1990er Jahre sehr gewandelt. Stellvertretend für die ökonomische Seite sei hier auf Amazon und eBay hingewiesen.

Interaktionskonventionen im Web

Die Interaktionsmetapher des Web ist das Browsen – sinngemäß das Durchstöbern von Seiten nach interessanten Inhalten. Die wichtigsten Interaktionsformen sind dabei der Klick auf einen Hyperlink, der die Ziel-seite des Links im Browser anzeigt und dabei die aktuelle Seite ersetzt, und die reziproke Aktion, der Klick auf den Back-Button, der den letzten Schritt rückgängig macht.

Ein paar kleinere Innovationen sind im Bereich des Browsers hinzugekommen. So sind zum Beispiel in den letzten Jahren Tabbed-Windows sehr populär geworden, ein Interface-Element, bei dem mehrere Sessions wie Karteikarten in einem Fenster Platz finden. Der Vorteil ist eine bessere Ausnutzung des verfügbaren Platzes auf dem Bildschirm bei gleichzeitiger Minimierung des Aufwands die Fenster ständig hin und her schieben zu müssen, um die jeweils verdeckten Fenster sichtbar zu machen.

Für einige Browser gibt es auch Erweiterungen zur Gestenerkennung. Mit relativ einfachen Mausbewegungen, wie beispielsweise einer raschen Bewegung nach links oder oben, kann der Benutzer bestimmte Kommandos im Browser auslösen. Da die Aktionen frei definierbar sind, gibt es keinerlei Konventionen, welche Geste die vorige Seite anzeigt, einen neuen Tab anlegt oder die Schrift vergrößert.

Die größeren Veränderungen sind nicht bei der Browser-Applikation, sondern bei den Inhalten und Services im Browser-Fenster zu beobachten. Seit kurzem bekommen nämlich einige Sites immer mehr die Anmutung von Desktop-Anwendungen. Ein Klick auf einen Hyperlink oder ein grafisches UI-Element führt nicht mehr zum Neuaufbau der ganzen Webseite; statt dessen ist es möglich einzelne Bereiche nach Bedarf auszutauschen, ohne die gesamte Seite neu laden zu müssen. Das ausgefeilte Interaktionsmodell wird durch das Zusammenwirken mehrerer Techniken erreicht, die mit dem Schlagwort »Ajax« bezeichnet werden.¹⁵

Im Folgenden soll nun die Anwendungsperspektive betrachtet werden und damit die Bedeutung, die das neue Medium für den Nutzer erlangt

15 Jesse James Garrett prägte »AJAX« als Abkürzung für Asynchronous JavaScript and XML. Da Garrett inzwischen selbst einräumt, dass auch Cascading Style-Sheets eine ebenso wichtige Rolle spielt, ist Ajax heute mehr als Begriff denn als Akronym zu verstehen. (Garrett anlässlich der reboot8 im Juni 2006: <http://www.mprove.de/script/06/reboot8/#Garrett>, März 2008).

kann. Zunächst stellt sich die Frage nach dem Umgang mit Informationen im Web. Suchmaschinen haben eine Bedeutung gewonnen, ohne die der Nutzer der Datenflut hoffnungslos ausgeliefert wäre. Erst die Indexierung von Yahoo!, Google und weiteren Firmen macht die Informationen im Netz auffindbar – in gewissem Sinne kann man daher die Suchdienste als Äquivalent zum Dateisystem des Desktops sehen.

Neben dem weltweiten Stöbern nach interessanten Webseiten, erweist sich die mittelbare Interaktion mit anderen Nutzern und somit das Web als Kommunikationsmedium zwischen Menschen als bedeutender neuer Aspekt.

4 Das »soziale« Netz

Texte werden von Menschen geschrieben, damit sie von anderen Menschen gelesen werden. Geschieht dies bidirektional, erhält man eine Korrespondenz, bzw. einen schriftlichen Dialog zwischen den beiden Personen. Geschieht es öffentlich, haben weitere Leser die Möglichkeit, die Diskussion zu verfolgen und mit ihren eignen Beiträgen zu bereichern.

Das Web ist nun insbesondere dazu konzipiert worden, Textseiten plattform- und programmunabhängig über das Internet zu verbreiten. Jeder, der Webseiten erstellen kann, ist potentiell aktiver Teilnehmer am weltweiten Diskurs. Das Problem bestand über viele Jahre allerdings darin, dass die Einstiegshürden für viele zu hoch waren. Erst als sich dank einiger Tricks Browser auch zum Erstellen von Inhalten benutzen ließen, konnte sich der soziale Raum im Web entfalten.

Die folgenden drei Abschnitte beschreiben diese Entwicklung anhand von Wikis, Blogs und Tagging von Bookmarks.

Wikis

Mitte der 1990er Jahre entwickelte Ward Cunningham eine Methodik, die es ermöglicht, eine Webseite direkt im Browser zu bearbeiten, obwohl der Browser Webseiten eigentlich nur anzeigen kann. Indem Cunningham aber die Formular-Funktion des Browsers zum Bearbeiten des Inhalts der Seite zweckentfremdete, wurde es möglich, Webseiten mit jedem Browser direkt und schnell zu erstellen und zu verändern. »Schnell« ist auch die Bedeutung des hawaiianischen Wortes »wikiwiki«, dem die Technik ihren Namen verdankt.

Ein Wiki-Server versteht also eine spezielle Syntax, um mittels eines HTTP-Post-Kommandos Webseiten auf dem Wiki-Web-Server selbst zu modifizieren. Der Vorteil der Wiki-Technik besteht darin, dass die

Anwender keine speziellen Web-Editoren benötigen, um die Seiten zu bearbeiten und keine speziellen Protokolle benutzen müssen – FTP oder WebDAV – um die Seiten von ihren lokalen Computern zum Web-Server zu übertragen. Buchstäblich jeder, der einen Internet-Zugang besitzt, kann auf einem Wiki-Server schreiben und muss sich damit nicht mehr mit der Rolle des Lesers begnügen. Durch die Aufhebung der Grenze zwischen Autor und Leser ist ein Wiki sehr gut zur Unterstützung verteilter kooperativer Gruppen geeignet. Da alle Mitwirkenden in dieser Hinsicht gleichberechtigt sind, kommen Wikis insbesondere bei Open-Source-Projekten zum Einsatz. Beispielsweise wird die Kommunikation rund um die Planung der Software-Entwicklung bei OpenOffice.org durch ein Wiki unterstützt.¹⁶

Das prominenteste Beispiel eines Wiki-Projekts ist aber sicherlich die 2001 gegründete Wikipedia. Diese weltweite Initiative widmet sich der Aufgabe, eine freie Enzyklopädie zu erstellen, ein Online-Lexikon auf jeweils neuestem Stand zugänglich für alle Menschen. Die so genannten Wikipedianer verfassen in ihrer Freizeit Artikel für Wikipedia und redigieren und diskutieren die Artikel anderer. Der englischsprachige Teil verzeichnet Ende März 2008 über 2,3 Millionen Begriffe; auf Deutsch, Französisch und Japanisch gibt es immerhin jeweils mehr als 700.000, 600.000 bzw. 400.000 Einträge. Zum Vergleich: die Encyclopædia Britannica besitzt in ihrer gedruckten letzten Fassung etwas mehr als 65.000 Artikel.¹⁷

Die Dynamik, die sich hier entfaltet, war sicherlich bereits in der ersten Dekade des Web angelegt. Aber erst durch die Wiki-Server wurde die technische Einstiegsschwelle so weit herabgesenkt, dass dieser Aspekt von der Öffentlichkeit wahrgenommen wird.

Blogs

Der Begriff »Blog« ist ein Kurzform von Weblog, das sich aus den Bestandteilen Web und Log (im Sinne von Logbuch oder Tagebuch) zusammensetzt. Im Vergleich zu klassischen Web-Sites und Wikis ist das Publizieren von Texten für das Web noch einmal vereinfacht worden, da die Struktur der Site von der Blogging-Software vorgegeben ist. Es ist eine Web-Site mit einer primär chronologischen Struktur. Das heißt, wenn der Autor in einem Online-Formular einen kurzen Text schreibt und ihn per Klick auf einen Button veröffentlicht, erscheint der neue Eintrag mit

16 <http://wiki.services.openoffice.org/wiki/>, März 2008.

17 <http://en.wikipedia.org/wiki/Wikipedia>

http://en.wikipedia.org/Encyclopædia_Britannica, März 2008.

aktuellem Datum versehen am Anfang des Weblogs. Ältere Beiträge werden gleichzeitig auf der Seite nach unten verschoben.

Im Gegensatz zu Wikis hat jeder Blog meistens nur einen Autor. Die soziale Komponente ergibt sich, indem Blog-Autoren gerne und häufig aufeinander verweisen. Da einem Blog-Inhaber durch spezielle Mechanismen zeitnah angezeigt wird, dass sich jemand in einem Blog-Beitrag auf ihn bezieht, kann er seinerseits darauf reagieren. Es entspinnt sich ein Dialog und bei mehreren Beteiligten eine Diskussion über ein Thema.

Auf der Ebene der Anwender entsteht auf diese Art ein Geflecht sozialer Beziehungen, das sich auf der Hypertext-Ebene in einer hochgradig vernetzten Struktur zwischen den Blogs niederschlägt. Beides zusammen nennt man die Blogosphäre. Die Blogosphäre ist eine neue Dimension im Cyberspace, die sich durch neue Techniken und Konventionen definiert. Für den Außenstehenden ist diese Szene – bedingt durch den technischen Slang – schwer zugänglich. Begriffe wie Aggregator, Atom, Blogroll, Pings, Permalinks, Syndicatable RSS Feeds (in verschiedenen Versionen), Tags und Trackbacks tragen nicht zum anfänglichen Verständnis bei. Nichtsdestotrotz ist es im Vergleich zu einer eignen Web-Site viel einfacher und schneller, einen Blog bei einem Blog-Server einzurichten. Was bei einer Domain Tage bis Wochen dauert, ist bei Blogs in Minuten bis Stunden erledigt, da der Neublogger oft aus einer großen Palette von Designvorschlägen einen zu ihm passenden Stil auswählen kann.

Die Frequenz, mit der neue Blog-Beiträge erscheinen, ist im Allgemeinen sehr viel höher, als bei klassischen Webseiten. Das überfordert allerdings die Algorithmen, mit denen die Suchdienste ihren Index berechnen. Google braucht in etwa eine Woche, bevor eine neue Seite gefunden werden kann, da das PageRank-Verfahren für die Bewertung der Relevanz einer Seite alle eingehenden Hyperlinks betrachtet und relativ zum PageRank der Seite, von der der Link kommt, bewertet. Diese Methode liefert sehr gute Ergebnisse, ist aber zugleich sehr aufwändig und damit langwierig in der Berechnung. Neue Anbieter haben sich dieses Problems angenommen und schaffen es inzwischen, Blog-Beiträge bereits wenige Minuten nach Veröffentlichung auffindbar zu machen. Zu nennen sind hier exemplarisch Technorati, WebLogs.com und Bloglines. Diese Dienste bieten ebenfalls auf die verschiedensten Themen spezialisierte Nachrichtenkanäle an, die einen unterstützen, die interessanten Artikel aus dem Meer von Informationen heraus zu fischen. Aus Anwendersicht stellt sich aber die Frage nach dem Sinn zweier Kategorien von Suchdiensten. Schließlich handelt es sich bei klassischen Websites und bei Artikeln der Blogosphäre um einfache Webseiten.

Social Bookmarking

Ein anderer Aspekt, der von Browser-Applikationen unzureichend abgedeckt wird, ist das Verwalten von Bookmarks bzw. von Favoriten im Internet Explorer von Microsoft. Browser bieten die Funktion, für eine Webseite die zugehörige URL zu merken und sie in der Menüstruktur für späteren direkten Zugriff anzubieten. Die Listen dieser Lesezeichen werden schnell unübersichtlich und veralten teils, da die Webseiten nach einiger Zeit nicht mehr existieren oder ihr Inhalt stark verändert wurde. Außerdem kommt es zu Synchronisationsproblemen, wenn man den Browser wechselt oder gar mehrere Browser oder verschiedene Computer nutzt.

Für eine bessere Ordnung und für die Unabhängigkeit von einem konkreten Browser bieten Web-basierte Dienste eine Alternative. Dabei werden die Bookmarks an ein persönliches Konto – beispielsweise bei mister-wong.de, ma.gnolia.com oder del.icio.us¹⁸ – geschickt und zusätzlich mit einem oder mehreren Schlagworten versehen. Die Schlagworte nennen sich »Tags« (engl. für Etikett) und haben mehrere Funktionen. Wenn man einen der Tags auswählt, werden alle unter ihm klassifizierten Bookmarks aufgelistet. Damit entsprechen die Tags in Vannevar Bushs Sinne viel genauer den Trails des Memex als die Hyperlinks. Schon dort sollten nämlich – wie eingangs beschrieben – Artikel, die in thematischem Bezug zueinander standen, vom Benutzer durch die Eingabe eines kurzen Codes miteinander verbunden werden können.¹⁹

Zweitens bestimmt man durch die Vergabe von Tags für Einträge nicht einen eindeutigen Ort in einer Ordnerstruktur. Stattdessen bilden Tags eine nicht-hierarchische Ordnung über den Dingen, da eine Website ohne weiteres mehreren Tags zugeordnet sein kann. Damit entfällt der rigide Zwang zur eindeutigen Ablage. Hierin verbirgt sich ein Potential für den Umgang mit Informationen im Computer an sich, da Tagging als recht junges Element in der Mensch-Computer-Interaktion noch nicht voll zur Entfaltung gekommen ist (Müller-Prove 2007).

Wenn die Tagging-Daten öffentlich sind, ergeben sich weitere Vorteile. Es lassen sich Leute finden, die die gleichen Webseiten besuchen wie man selbst und mit denen man möglicherweise gemeinsame Interessen hat. Anders herum kann man auch die Masse für sich instrumentalisieren, indem man beobachtet, welche Webseiten sich hinter speziellen Tags

18 Als der Webdienst del.icio.us (<http://del.icio.us>, März 2008) 2005 von Yahoo! übernommen wurde, war die Zahl der Benutzer in den zwei Jahren seit Gründung auf 300.000 angewachsen.

19 Typisierte Hyperlinks, die vom Anwender frei gesetzt werden können, hätten einen ähnlichen Effekt. Dieser Mechanismus ist im Web mit seiner beschränkten Browser-Technik aber nicht entwickelt.

ansammeln. Die eigene Recherchearbeit zu einem Thema wird so durch die Aktivität der anderen unterstützt. Bei solcher Art von Informationsaufbereitung ist aber zu bedenken, dass die »Link-Empfehlungen« rein subjektiv sind. Es gibt keinerlei Autorität oder Redaktion, die für die Güte und Vollständigkeit der Quellen bürgt. Der Umgang mit kollektivem Filtern der Webseiten erfordert ein hohes Maß an Medienkompetenz, um nicht der Versuchung zu erliegen, Informationsquellen außer Acht zu lassen, nur weil sie nicht über einen Social-Bookmarking-Service oder eine elektronische Suchmaschine erschlossen sind.

5 Von der Gleichzeitigkeit des Unvereinbaren

Aus Sicht des Desktop-Paradigmas ist der Web-Browser lediglich ein Schreibtischwerkzeug unter vielen. Neben Textverarbeitung, Tabellenkalkulation und E-Mail gibt es eben auch noch ein Programm für das Browsen im Netz. Das Web hat nun aber eine Größe und Bedeutung angenommen, die bezüglich der Benutzungsschnittstelle insgesamt nicht mehr ignoriert werden darf. Die Interaktionsformen des Computer-Schreibtisches auf der einen Seite und des Web auf der anderen stehen nämlich konträr gegeneinander. Der wichtige Grundsatz der Erwartungskonformität der Schnittstelle ist schwer verletzt. Dazu vier Beispiele:

- (1) In einem Fenster genügt ein Einfachklick auf einen Hyperlink, wohingegen in dem anderen Fenster ein Doppelklick auf ein Dokument- oder Ordner-Icon für eine äquivalente Funktion notwendig ist. Anwender bringen das gerne durcheinander und erwarten eine Funktion mit einem Doppelklick auszulösen, obwohl ein Einfachklick genügt. Die Gestalter von Webseiten und Software-Applikationen müssen diese Unsicherheit einkalkulieren, damit der zweite Klick des Doppelklicks nicht gleich die nächste Funktion auslöst.
- (2) Der Browser merkt sich die Folge der aufgerufenen Webseiten in der History. Wenn man nun den Schritt zu einer uninteressanten Seite rückgängig machen möchte, hilft einem die Back-Funktion. Nicht so in Dokumenten der Desktop-Welt. Wenn man da eine Folge von Bearbeitungsschritten rückgängig machen möchte, steht einem (hoffentlich) die Undo-Funktion zur Verfügung. *Undo* und *History* sind also zwei Konzepte, die wegen ihrer semantischen Ähnlichkeit zu einer einzigen zusammengefasst werden sollten.
- (3) Immer mehr Web-Services haben die Anmutung von Applikationen. Beispiele dafür sind Web-Mail, Web-Kalender bis hin zu Anwendun-

gen, die auf Google Maps basieren.²⁰ Man verlässt die Webseite nicht mehr, sondern ändert lediglich direkt die Parameter auf der Webseite. Konventionen aus der Desktop-Welt gelten hier unvermittelt nicht mehr. Ein Command-S zum Sichern einer E-Mail löst in einer Web-Mail-Anwendung die unsinnige Funktion aus, die Webformularseite als HTML abzuspeichern. Und bei Bedienfehlern hilft einem weder die Back-Funktion der History (die beendet ohne Warnung die Webanwendung!), noch die Undo-Funktion des Browsers (keine Wirkung für den Inhalt von Webseiten) und ein spezielles Undo für den Web-Service ist nicht vorgesehen.

- (4) Auf Desktop-Systemen sind Icon und Fenster zwei verschiedene Darstellungsformen eines Dokuments. Das Dokumentfenster hat dabei eine sehr starke Bindung an eine bestimmte Datei, da der Speicherungszustand von zentraler Bedeutung ist. Unter dem Webparadigma gleicht ein Browser-Fenster vielmehr einem Wechselrahmen, der nacheinander verschiedene Dokumente darstellt. Ein Anwender, der sich auf die Eigenschaften der Desktop-Welt verlässt, die ihn davor bewahren, eigene Daten zu verlieren („Wollen Sie das Dokument jetzt sichern?“), wird beim versehentlichen Schließen eines Browser-Fensters entsprechende Warnung schmerzlich vermissen. Es wäre denkbar, die *History* selbst als schützenswerten Inhalt zu begreifen und sie quasi als *Trail* im Sinne des Memex abzuspeichern. Das ist aber in heutigen Browsern nicht vorgesehen.

Die Beispiele zeigen, dass das Nebeneinander verschiedener Interaktionsformen ein effektives und elegantes Arbeiten mit dem Computer verhindert. Der Benutzer wird durch ständige Kontextwechsel und einem daraus resultierenden Benutzungsschnittstellenwirrwarr von seiner eigentlichen Tätigkeit abgelenkt. Das User Interface sollte aber vertraut und selbstverständlich sein und dadurch im besten Sinne des Wortes transparent werden können. Erst dann lässt sich das Potential des Mediums Computer voll ausschöpfen.

20 Ein »Mashup« (engl. für Vermischung) ist eine Webanwendung, die aus den Angeboten mehrerer Websites einen neuen Service erstellt. So gibt es inzwischen zahllose Seiten, die auf den Landkarten und Satellitenbildern von Google eigene Daten, wie beispielsweise Restaurants oder Staumeldungen, einzeichnen. (vgl. auch <http://de.wikipedia.org/wiki/Mashup>, März 2008)

Objekte und Informationen

Auf grundlegender Ebene ist ein Desktop-System objektorientiert. Der Begriff bedarf einer Erklärung, da hier nicht die softwaretechnische Sicht im Sinne der Programmierung gemeint ist. Die Schreibtisch-Metapher induziert eine Menge realer Objekte – wie beispielsweise Ordner und Dokumente – in den Aktionsraum des Computeranwenders. Dort werden sie als Icons und Fenster repräsentiert. Der Anwender begreift diese Objekte als wären sie real. Er hantiert mit ihnen über das angeschlossene Zeigegerät, das den Mauszeiger über den Bildschirm bewegt. So entsteht die Interaktion des Anwenders mit den virtuellen Objekten des Computers.

Alan Kay drückte das in den siebziger Jahren mit dem Satz aus: »Doing with images makes symbols.« *Doing* ist das buchstäbliche Manipulieren der Dinge. *Images* sind die Objekticons und Fenster. Und *Symbols* verweist auf die abstrakte symbolische Ebene, auf der die für den Anwender relevanten Operationen stattfinden. Beispiele für solche Operationen sind das Verschieben eines Dokumentes an einen anderen Ort in der Ordnerstruktur, das Öffnen und Bearbeiten eines Dokuments, aber auch das Versenden eines Dokuments per E-Mail an einen Kollegen. Ben Shneiderman prägte für diese Interaktionsform den Begriff »Direkte Manipulation« (Shneiderman 1983).

Ein wichtiger Aspekt für alle Objekte des Desktops ist deren Verhältnis zum Anwender: die Objekte gehören ihm – es sind *seine* Ordner, in denen er *seine* Dokumente verwaltet. Der Zugriff anderer ist per se ausgeschlossen.²¹ Und das Gestaltungsprinzip der Ortspersistenz hat einen hohen Wert. Dinge bleiben an dem Ort, an die der Anwender sie gelegt hat. Damit ist die Entsprechung zur realen Bürowelt gewahrt, da man dort auch für Ordnung oder Unordnung selbst verantwortlich ist.

Anders im Web. Der Web-Surfer verlässt seinen Schreibtisch und reist durch fremde Datenräume. Dabei ist von den Entwicklern vergessen worden – oder es erschien nicht mehr notwendig – eine metaphorische Objektschicht einzuführen. Der Anwender agiert mit Informationen, die anderen gehören, bzw. die von anderen ins Internet gestellt wurden. Der Desktop-Metapher entspricht im Web noch am ehesten das Browsen, das Stöbern in Bibliotheken und Durchblättern von Büchern und Magazinen.²²

21 Wenn diese persönliche Grenze überschritten wird, spricht man folglich von Computerkriminalität.

22 Aus dem Oxford Dictionary: browse: [...] 2 read (parts of a book or books) without any definite plan, for interest or enjoyment: *browsing among books in the public library*. (Hornby 1984)

Die Gestalter der Online-Welt bedienen sich gern und häufig aus der Begriffswelt der Städteplaner und Architekten: *Home page, landing page, web site, site map, portal, chat room, online forum, online shop, community, domain, information highway, world wide web, cyberspace*. Diese Begriffe werden implizit verwendet, aber nicht forciert, um etwa die Nutzung des Internets zu veranschaulichen. So entstehen zahlreiche Brüche und Ungeheimheiten. Eine Sammlung von *Webseiten* wird *Web-Site (Platz)* genannt. Dabei müssten mehrere Seiten doch ein Buch ergeben! Und die »*Bookmarks*« eines Browsers dienen zum Merken von Orten im *Cyberspace* – Sprachmetaphern der Nautik und Navigation wären hier wohl angebrachter.

6 Ausblick

Bei dem hier diskutierten Thema verbietet sich eine abschließende Zusammenfassung. Zu viel ist noch im Fluss – zu viel ist noch unfertig und provisorisch.

Der Vergleich der Grundprinzipien von Desktop- und Webparadigma hat gezeigt, dass keine der beiden Herangehensweisen eine angemessene Antwort auf das Informationsproblem darstellt. Beide Systeme haben ihre unbestreitbaren Vorteile – die nur leider zum Nachteil der Nutzer in diametral unterschiedlichen Interaktionsmodellen gegeneinander stehen.

Die metaphorische Umsetzung des Hantierens mit realen Objekten im Desktop-Paradigma hat maßgeblich zur Verbreitung des Computers in nicht-akademische Bereiche des Arbeitsalltags beigetragen. Durch die Abstraktion von den elektrotechnischen und algorithmischen Prozessen durch die Schreibtisch-Metapher wurden die Vorgänge für den Nichtfachmann vorstellbar – die Daten wurden im Wortsinne sogar »begreifbar« gemacht, denn Douglas Engelbarts Erfindung der Maus ist zur selbstverständlichen Erweiterung der menschlichen Hand im Interaktionsraum des *Cyberspace* geworden.²³ Das nun direkte Manipulieren der virtuellen Objekte erzeugt ein Eigentumsverhältnis zwischen Anwender und Daten, das einen wesentlichen Beitrag zum persönlichen Charakter des PCs leistet.

23 In »*The Architecture of Intelligence*« benennt Derrick De Kerckhove den *Cyberspace* als die dritte Wahrnehmungsmodalität des menschlichen Geistes. Nach der Entwicklung der Sprache, mit der Menschen direkt miteinander Gedanken austauschten, und der Schriftkultur, in der das geschriebene und gedruckte Wort an nachfolgende Generation überliefert werden konnte, eröffnet sich hier eine globale, kognitive, immersive Dimension (De Kerckhove 2001, S. 33).

Im Web sind wir lediglich zu Gast. Wir navigieren durch Datenräume, die von anderen für uns geschaffen wurden. Mittels Suchdiensten sind Informationen im Netz inzwischen oft schneller auffindbar als Dokumente auf der eigenen Festplatte. Ein Begreifen im obigen Sinne gibt es aber nicht mehr. Weder lassen sich im Allgemeinen die Webseiten verändern oder verschieben, noch können, wie in einem Buch, Textstellen markiert und Bemerkungen an den Rand geschrieben werden. Wie oben dargestellt wurde, entspricht das weder Ted Nelsons Ideal vom Hypertext, noch dem der Entwickler des World Wide Web, die am CERN schon die Grundlagen für ein flexibles Medium zum Erstellen und Verbreiten von Informationen geschaffen hatten.

Erst in den letzten Jahren gewinnt dieser Aspekt wieder an Bedeutung, indem die Websurfer auch selbst schreiben. Blogs und Wikis sind dabei zwei einfach nutzbare Möglichkeiten, die hier vorgestellt wurden. Damit wird der Webkonsument endlich zum aktiven Teilnehmer und – eingeschränkt auf bestimmte Domains – selbst zum Autor. Online-Profile bei Social-Networking Sites wie LinkedIn und Xing²⁴ und eigene Seiten bei Diensten wie MySpace²⁵ sind weitere Beispiele dafür, dass auch hier die Technik inzwischen soweit vereinfacht wurde, dass nicht nur Teilchenphysiker und teuer bezahlte Webdesigner Inhalte im Web veröffentlichen können.

Das Web ist zum Massenphänomen geworden, dessen Seiten immer mehr auch von den Menschen aktiv gestaltet werden. So ist das Web auf gutem Wege, auch ein persönliches Medium zu werden. Auf das persönliche Medium Desktop-PC wird bei der Entwicklung aber leider kaum Rücksicht genommen.

Für die Zukunft könnte sich daraus eine abnehmende Bedeutung des Desktops ergeben. Denn durch immer mehr Anwendungen, die im Webbrowser vergleichbaren Nutzen zu den klassischen Applikationen anbieten, entfällt der Anschaffungs- und Lernaufwand für eben diese Programme. Der primäre Interaktionsmodus würde sich dadurch vom Computerschreibtisch zum Webbrowser verlagern. Eine konsequente Weiterentwicklung wäre dann ein System, das gar keinen Schreibtisch mit persönlichen lokalen Dokumenten und Ordnern mehr anbietet, sondern alles nur noch über den Webbrowser darstellt.

Wenn man die Schreibtischmetapher nicht aufgeben will, muss man sich um ein Zusammenführen beider Welten bemühen. Eine zentrale Rolle nimmt dabei der Browser ein, der für viele der Interaktionsinkonsis-

24 <http://www.linkedin.com> und <http://www.xing.com>, März 2008

25 MySpace.com wurde 2005 für 580 Millionen Dollar von Rupert Murdochs News Corp gekauft und verzeichnet heute mehr als 100 Millionen Nutzer.

tenzen verantwortlich ist; dessen Weiterentwicklung stagniert aber seit Jahren. Dazu wurden einige Beispiele im vorigen Abschnitt genannt.

Auch ist es für den Anwender nicht einsichtig, warum im Netz und auf seinem persönlichen Computer unterschiedliche Such- und Ordnungsmechanismen zum Einsatz kommen. Viel besser wäre eine Suche, die je nach Anwendungsfall auf der Festplatte und/oder im Netz nach Treffern sucht und die sich für den Anwender in einer einheitlichen Schnittstellenkomponente darstellt.²⁶ Da das Konzept des Taggings noch sehr jung ist, kann man hier auf eine Website-übergreifende Lösung spekulieren und eine spezielle Einbeziehung der Folksonomy-Daten für die Suche erwarten.²⁷

Auf Seiten der Desktop-Systeme sind offene und standardisierte Dateiformate ein wichtiger Schritt zur Integration beider Welten. So sorgt zum Beispiel im Office-Bereich das ISO-Format OpenDocument für eine Unabhängigkeit von Programmherstellern und Betriebssystemen. Genau wie der offene Standard HTML Voraussetzung für den Erfolg des Web war, kann der Standard für formatierte Textdokumente, Präsentationen und Tabellenkalkulationen einen Innovationsschub auslösen, der die Kluft zwischen Desktop und Web zu überbrücken hilft. Für den Anwender ergäbe sich daraus ein System, bei dem es möglich ist, ein Dokument zunächst mit einem Office-Programm zu erzeugen und später mit einer Webanwendung weiter zu verarbeiten, die den ISO-Standard unterstützt. Damit könnte die Schwachstelle der Webs ausgeglichen werden, nämlich das Erstellen neuer Inhalte.

Vannevar Bushs Idee des Memex war ein einheitliches System zur wissenschaftlichen Lektüre und Forschung. Ted Nelson ging es um das Verfassen neuer (Hyper-)Texte und um deren weltweite Vernetzung. Douglas Engelbart ergänzte diese Vision um den kollaborativen Aspekt. Zusammen ergibt das ein integriertes Recherche- und Autorenwerkzeug für weltweit agierende Teilnehmer. Die einzelnen Komponenten dazu sind heute durchaus in einer Qualität verwirklicht, die damals undenkbar war. Die Entwicklungsgeschichte hin zum Personal Computer und dem World Wide Web hat aber ein komplexes System hervor gebracht, das

26 »Google Desktop Search« ist ein Ansatz in dieser Richtung. Lokale Suchergebnisse in Dokumenten, E-Mails, Fotos usw. werden in Form von üblichen Online-Suchergebnissen präsentiert. Die Frage nach dem Datenschutz ist dabei nur unzureichend geklärt, da Google im Prinzip Einblick in die persönlichen Daten des Anwenders erhält.

27 Der Neologismus »Folksonomy« leitet sich von Taxonomie ab und bezeichnet die von den Anwendern durch Tags kollektiv definierte Ordnung der Dinge. Der Begriff wurde erstmals 2004 auf einer Mailingliste des Information Architecture Institute von Thomas Vander Wal verwendet (Merholz 2006, Müller-Prove 2007).

weit hinter den einzelnen Visionen zurückbleibt. Desktop- und Webparadigma ergänzen sich leider nicht zu einem homogenen und konsistent bedienbaren Ganzen, sondern konkurrieren miteinander in wichtigen Aspekten der Metapher und Interaktion.

Nur wenige Utopisten und Visionäre hatten eine Ahnung, wie sehr die Computerrevolution das Zusammenleben auf unserer Erde tatsächlich verändern würde. Der Computer ist in ein paar Generationen zu einem weltumspannenden Kommunikationsmedium geworden, dessen Folgen für unsere Gesellschaften sich gerade erst abzuzeichnen beginnen. Wie die Diskussion gezeigt hat, bietet die Interaktion mit dem Computer und dem Web noch viel Potential für Weiterentwicklungen – und daher stehen wir in dieser Phase noch ganz am Anfang.

Literatur

- Berners-Lee, T.-B. (1989): »Information Management: A Proposal. CERN«. Geneva <http://www.w3.org/History/1989/proposal.htm>, März 2008.
- Bush, V. (1945): »As We May Think«. In: *Interactions* 3, 2, S. 35-46, 1996. Nachdruck aus: *The Atlantic Monthly* 176.
- De Kerckhove, D. (2001): »The Architecture of Intelligence.«, Boston.
- Engelbart, D. C. (1962): »Augmenting Human Intellect: A Conceptual Framework«. Summary Report AFOSR-3223 under Contract AF 49(638)-1024, SRI Project 3578 for Air Force Office of Scientific Research. Stanford Research Institute, Menlo Park, CA. <http://www.bootstrap.org/augdocs/friedewald030402/augmentinghumanintellect/ahi62index.html>, März 2008
- Friedewald, M. (1999): »Der Computer als Werkzeug und Medium«. Berlin, Diepholz.
- Gillies, J./Cailliau, R. (2000): »How the Web was Born«. Oxford.
- Hornby, A. S. (1984): »Oxford Advanced Learner's Dictionary for Current English«. 19th ed. Oxford.
- Kay, A. C. (1972): »A Personal Computer for Children of All Ages«. In: *Proceedings of the ACM National Conference*, Boston. New York. <http://www.mprove.de/diplom/gui/Kay72a.pdf>, März 2008.
- Merholz, P./Starmer, S./Surla, St./McMullin, J./Reiss, E. (2004): »Annual Report«. 2004-05, The Information Architecture Institute. <http://iainstitute.org/news/000464.php>, März 2008.
- Müller-Prove, M. (2002): »Vision and Reality of Hypertext and Graphical User Interfaces«. University of Hamburg, Department of Informatics, Report FBI-HH-B-237/02. <http://www.mprove.de/diplom/>, März 2008.

- Müller-Prove, M. (2007): »Taxonomien und Folksonomien – Tagging als neues HCI-Element«. In: i-com 6, 1. <http://www.mprove.de/script/07/icom/>, März 2008.
- Nelson, Th. H. (1965): »A File Structure for the Complex, the Changing and the Indeterminate.« In: »The New Media Reader«, hg. v. N. Wardrip-Fruin/N. Montfort. S. 134-145. Cambridge, MA.
- Shneiderman, B. (1983). »Direct manipulation: A Step Beyond Programming Languages«. In: IEEE Computer, 16, 8, S. 57-69.
- Smith, D. C./Irby, Ch. H./Kimball, R./Verplank, W. L./Harslem, E. (1982): »Designing the Star User Interface«. In: »Integrated Interactive Computing Systems«. Proceedings of the European Conference on Integrated Interactive Computing Systems, ECICS 82. Stresa, Italy, hg. v. Degano, P./Sandewall. E., p. 297-313. Nachdruck aus Byte 7, 4, 1982.
- Stephan, P. F. (2006): »Nicht-Wissen als Resource sowie sieben Thesen zur künftigen Wissensarbeit«. In: i-com 5, 2, p. 4-10.
- Sutherland, I. E. (1963): »Sketchpad – A Man-Machine Graphical Communication System«. MIT, Lincoln Lab.: Report 296, Neuauflage 1965. <http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-574.pdf>, März 2008.

**ALTERNATIVEN ZUM DESKTOP-COMPUTING:
DER KÖRPER UND GEGENSTÄNDE ALS INTERFACES**

WEGE UND IRRWEGE DER MENSCH-MASCHINE-KOMMUNIKATION BEIM WEARABLE COMPUTING

INGRID RÜGGE

Die meisten Menschen unseres Kulturkreises kennen inzwischen die kleinen Hochleistungsnotebooks, PDAs (Personal Digital Assistants) und Smartphones – praktische Lösungen, wenn man unterwegs eine Fahrplanauskunft benötigt, schnell etwas mit jemandem abstimmen oder im Zug noch an einem Text arbeiten will. Computer und Telefone sind heute mobil einsetzbar. Handys sind als Kommunikationsmittel sowohl aus dem beruflichen, vor allem aber aus dem privaten Sektor kaum noch wegzudenken. Sie gehören zum Alltag von Menschen jeder Altersgruppe, wie schon seit geraumer Zeit die kleinen mobilen Unterhaltungselektronikgeräte. Die verwendeten technischen Komponenten werden immer kleiner und leistungsfähiger, inzwischen können sie bereits in Kleidung integriert werden (siehe Abb. 1). Ihre Einsatzmöglichkeiten reichen deutlich über die Benutzung im privaten Bereich sowie für mobile Tätigkeiten, die bisher am Schreibtisch ausgeübt wurden, hinaus. Mobile informations- und kommunikationstechnische Computerunterstützung wird zunehmend für andere Sektoren attraktiv.

Abb. 1: »Smart Electronics«-Lösung für den privaten Gebrauch



Beispiele für mobile Arbeitstätigkeiten jenseits des Schreibtisches sind die Wartung von Windkraftanlagen, die Dokumentation ambulanter Pflege, Notarzteinsätze oder die Kommissionierung und Auslieferung von Waren (siehe Abb. 2). Das sind jeweils Arbeiten, die in der »gegenständlichen, physischen Welt« und nicht am oder vom Computer ausgeführt werden (können), jedoch sehr wohl durch den Einsatz von Computertechnologie vereinfacht und verkürzt werden können. Eine solche mobile Lösung, die der BenutzerIn sehr nahe an den Körper rückt – und deshalb auch als Wearable Computing-Lösung bezeichnet wird –, muss dann aber auch wirklich »tragbar« sein, in jeder Hinsicht.

Abb. 2: Potenzielle Einsatzsituation einer Wearable Computing-Lösung in der Logistik (Bild: Metro 2007)



Die folgenden Ausführungen konzentrieren sich auf mobile Lösungen im oben genannten Sinne für die Unterstützung mobiler, professioneller Arbeitstätigkeiten jenseits des Schreibtisches. Anhand von Beispielen werden ihre Einsatzpotenziale, ihre Probleme bzgl. der Mensch-Computer-Interaktion sowie Lösungsansätze zur Überwindung dieser und zur Entwicklung tragbarer mobiler Lösungen vorgestellt.

Beispiele für Wearable Computing-Lösungen

Es existiert bereits eine Vielfalt konzipierter und prototypisch realisierter mobiler Lösungen. Sie resultieren zumeist aus einem technologischen Fachgebiet und zielen auf einen speziellen Anwendungsbereich. Case Studies und White Paper der Hersteller von Wearable Computing-Hardware oder von SoftwareentwicklerInnen zum Einsatz ihrer Programme auf mobilen Geräten sowie wissenschaftliche Untersuchungen verweisen auf ein großes Innovationspotenzial mobiler Lösungen. Die meisten dokumentierten Wearable Computing-Lösungen sind jedoch nach wie vor Prototypen, die Machbarkeitsstudien für Forschungsansätze oder Feldstudien dienen. Produkte sind bis heute kaum verfügbar (siehe Rügge 2007). Auf Erfahrungen mit einem langfristigen kommerziellen Einsatz einer mobilen Lösung kann nur im Fall des Wearable Scanning Systems der Firma Symbol Technologies verwiesen werden (siehe Symbol Technologies 2006, Stein et al 1998).

Abb. 3: Kommerzielles Wearable Scanning System von Symbol Technologies, Einsatzbereich Logistik (Bild: MRC)



Die Bandbreite der bereits in Erwägung gezogenen Einsatzgebiete umfasst die klassischen Wirtschaftssektoren – von der Dienstleistung über die Industrie bis hin zur Landwirtschaft – und den Konsumenten-Bereich¹. Umsetzungen wurden im professionellen Bereich jenseits der Schreib-

¹ Zu Einsatzbeispielen von mobilen Lösungen für Konsumenten oder im Büro-Bereich siehe z.B. Pichler 2005 oder Wichmann/Stiehler 2004

tischarbeit vor allem für Sparten wie Transport und Logistik, Produktion, Instandhaltung, Gesundheitswesen, Sicherheit und Journalismus entwickelt.

Als ein Beispiel sei hier der Wartungsbereich herangezogen: Die Instandhaltung von Produktionsanlagen leistet einen wichtigen Beitrag zum störungsfreien Ablauf einer Produktion. Sie ist aber auch in vielen anderen Sektoren Bestandteil der täglichen Arbeit, z.B. bei der Wartung von Haushalts- oder Gebrauchsgeräten oder von Gebäuden. Bereits seit den Anfängen der Entwicklung von Wearable Computing-Lösungen wird daran gearbeitet, diese Technologie für die Instandhaltung einzusetzen, da der Zugriff auf Informationen (Handbücher, Schaltpläne, Reparaturanleitungen etc.), eine multimediale Expertenkonsultation, eine papierlose Schadensberichtserfassung oder die systematische Steuerung des Arbeitsablaufs per interaktiver Checkliste zu einer deutlichen Prozessoptimierung beitragen können. Für die Wartung von Gebrauchsgeräten (z.B. Drucker) oder der Gebäudetechnik (Vernetzung), zur Visualisierung verborgener architektonischer Strukturen in Gebäuden, zur Inspektion von Fahrzeugen, Flugzeugen, Industriekränen, Schiffen und Postsortieranlagen, zur Instandhaltung von Wohnhäusern sowie für die Inspektion in Produktion und Montage, z.B. beim (verteilten) Bau von Schiffen, wurden bereits Prototypen entwickelt (siehe Rügge 2007, S. 171 f.).

Kommerziell breit einsetzbare Systeme sind jedoch auch hier nicht auf dem Markt verfügbar. Dies liegt vor allem daran, dass ein Modell aller zu inspizierenden Objekte sowie ein Modell des jeweiligen Inspektionsprozesses implementiert sein muss, um ein solches System produktiv einzusetzen. Daran mangelt es in dem meisten Fällen, denn die Digitalisierung eines Teilekatalogs reicht für diesen Zweck einfach nicht aus und die Modellierung jedes Wartungsprozesses würde einen erheblichen personellen Aufwand erfordern. Es liegt aber auch daran, dass die bisher verfügbare Hardware den Einsatzbedingungen der Anwendungsbereiche und dem Arbeitsprozess der BenutzerInnen nicht im erforderlichen Maße angepasst wurde, wie im Folgenden noch dargelegt werden wird.

Verfügbare Systemkomponenten für mobile Lösungen

Die Palette eingesetzter mobiler Endgeräte für mobile Lösungen reicht von PDAs und Smartphones über UMPCs² und Wearable Computer bis hin zu »computerisierten« Accessoires. Anders als bei DesktopPCs, die sich im Prinzip nur in Details wie Prozessorleistung, Qualität des Bildschirms oder in den integrierten Schnittstellen unterscheiden, gibt es für mobile Lösungen eine Vielzahl von mobilen Rechnern, die sich deutlich voneinander abheben und nur einige wenige grundlegende Merkmale gemeinsam haben. Gleiches gilt für mobile Interaktionsgeräte. Gemeinsam ist allen Geräten und Komponenten das oft vernachlässigte Problem des Stromverbrauchs bzw. der Stromversorgung und der durch Batterien/Akkus verursachten Belastung der BenutzerIn, z.B. durch Gewicht oder Wärmerwicklung. Im Folgenden wird eine kleine Auswahl von Geräten bzgl. ihrer Merkmale als Komponenten für mobile Lösungen vorgestellt.

Universell einsetzbare, kompakte Wearable Computer

Alle Hersteller bemühen sich bisher darum, ihre Hardware so universell einsetzbar wie möglich zu gestalten, damit Serien produziert und keine Einzelanfertigungen gemacht werden müssen. Ziel ist darüber hinaus immer noch »kleiner, stärker, schneller«. Neben den Standardgeräten, die am Markt verfügbar sind, gibt es Prototypen, Bausätze und Komponenten, die noch keine Marktreife erlangt haben, die sich jedoch in Abgrenzung zu den Massenprodukten an den individuellen Bedürfnissen spezieller Anwendungsbereiche orientieren. Jedes Produkt und jeder Prototyp hat seine speziellen Eigenarten, so dass man sie schwer vergleichen kann. Es lassen sich aber Rahmenbedingungen benennen, auf die bei der Auswahl eines Gerätes zu achten ist. Ein mobiles Endgerät

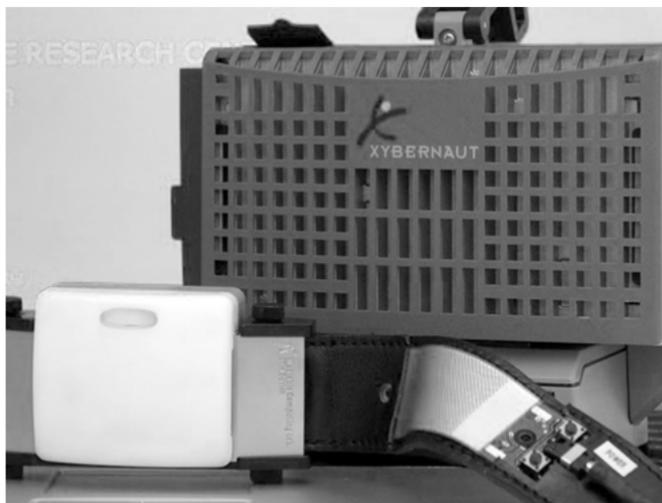
- ist während des Betriebs und der Benutzung am Körper zu tragen bzw. zu befestigen, ohne in der Hand gehalten werden zu müssen und ohne eine externe Ablage zu benötigen,
- muss klein und leicht sein und darf die Bewegungsfreiheit des Körpers und das Wohlbefinden der TrägerIn nicht oder nur geringfügig einschränken,

2 UMPCs sind ultra mobile PCs, die weniger als halb so groß wie übliche Notebooks sind, aber ähnliche Leistungsdaten haben. Die Bezeichnungen in diesem Gerätesektor und die Definition, was genau unter den Begriff UMPC fällt, sind nicht einheitlich; einen Eindruck über den aktuellen Stand vermittelt die Forum-Webseite www.umpcportal.com (zuletzt gesehen 1.3.2008).

- muss einen mehrstündigen Dauerbetrieb, auch unter extremen Umgebungsbedingungen, ermöglichen,
- benötigt eine autarke, langlebige, leichte, wiederaufladbare Stromversorgung, die mit einem Wechselmodus ausgestattet sein sollte, der einen unterbrechungsfreien kontinuierlichen Betrieb des Geräts ermöglicht,
- sollte vom Prinzip her möglichst universell einsetzbar und vor allem so programmierbar sein wie ein DesktopPC,
- sollte mit (Standard-)Schnittstellen für den Anschluss peripherer Geräte und eine Vernetzung ausgestattet sein, z.B. zum Anschluss von Ein-/Ausgabemedien, Sensoren, etc.

Welchen Stellenwert die jeweilige Eigenschaft hat, wird allerdings von der intendierten Einsatzsituation bestimmt, so dass mobile Endgeräte nicht generell bewertet werden können³.

Abb. 4: Größenvergleich Xybernaut MA V und QBIC (Bild: MRC)



Die speziell für den industriellen Einsatz gebauten »Mobile Assistants« (kurz MA) der Firma Xybernaut sind entsprechend robust ausgestattet. Sie sind relativ groß und klobig, können aber auch Stöße etc. vertragen. Auch das mobile zweiteilige Tough-Book von Panasonic ist für industrielle Umgebungen gebaut worden, es entspricht in seiner Baugröße der eines

3 Zu weiteren Merkmalen von Komponenten mobiler Lösungen siehe Rügge 2007, S. 63 ff.

Xybernaut-Rechners. Der QBIC (Amft et al 2004, ETH Zürich 2004) hingegen ist ein »Gürtelschnallenrechner«, der zum ergonomischen Tragen am Körper entwickelt worden ist. Er ist im Vergleich zu den beiden genannten Produkten winzig und schwach: Er findet in einer nur wenig überdimensionierten Gürtelschnalle und dem Gürtel selbst Platz. Er verfügt über einen Prozessor der Leistungsklasse eines PDAs und ist in Art und Anzahl seiner Schnittstellen beschränkt. Um diese Einschränkungen im systemischen Sinne auszugleichen, ist er mit drahtloser Konnektivität ausgestattet. Seine Größe, sein Gewicht und seine Ausmaße kommen dem angestrebten Ideal schon sehr nahe, so dass er als Zubehör von Arbeitskleidung für ÄrztInnen, Pflegekräfte, Stewards, Sicherheitspersonal oder IngenieurInnen kein Hindernis darstellt.

Der QBIC ist bisher noch nicht zum Produkt weiterentwickelt worden, so dass bei der Realisierung von Wearable Computing-Lösungen weiterhin auf Interims-Technik wie PDAs und UMPCs zurückgegriffen wird, die als Massenprodukte preiswert und verfügbar sind, auch wenn sie nicht die notwendigen Trageeigenschaften besitzen.

Aufgaben- oder umgebungsoptimierte mobile Rechner

Es gibt einzelne mobile Rechner, die für besondere Aufgaben entwickelt oder auf besondere Umgebungsbedingungen hin optimiert worden sind, häufig mit integrierten mobilen Interaktionskomponenten.

Der VuMan ist ein an der Carnegie Mellon University (CMU) in mehreren Generationen entwickelter Wearable Computer (Smailagic/Siewiorek 1999, Bass et al 2001). Er wurde für die Abarbeitung einer Checkliste entworfen. Eine hardwareseitige Optimierung auf einen Anwendungsbereich und auf eine zu unterstützende mobile Tätigkeit erfolgte vor allem bei den Ein-/Ausgabe-Möglichkeiten. Die Eigenschaften der Recheneinheit selbst sind unspektakulär, da sie aus Standard-Komponenten zusammengebaut worden ist.

Der kommerziell erfolgreichste Wearable Computer ist bis heute das Wearable Scanning System der Firma Symbol Technologies (siehe Abb. 3). Die erste und am häufigsten verkaufte Variante dieser mobilen Lösung wurde unter dem Namen »WWS 1000 – Wearable Wrist Computer« bekannt, das neueste Gerät kam 2006 unter dem Kürzel WT4000 auf den Markt. Dieser Wearable Computer ist ein mobiles, freihändig trag- und benutzbares System, das auf die beiläufige Datenaufnahme – das Lesen von Barcodes – hin optimiert ist. Die Hardware, bestehend aus dem Computer und einem miniaturisierten, am Finger getragenen Laserscanner, ist ergonomisch optimal an die primäre Aufgabe der BenutzerInnen angepasst (manueller Transport von Gütern). Sie ist so gestaltet, dass die

TrägerIn in ihrer körperlichen Bewegungsfreiheit nicht eingeschränkt wird und der Benutzung des Systems kaum Aufmerksamkeit schenken muss (Stein et al 1998). Das System lässt sich im Rahmen dieser primären Aufgabe an die Gegebenheiten der speziellen Anwendungsumgebung anpassen, z.B. durch den Einsatz eines anderen Scanners, und es lässt sich erweitern, beispielsweise durch die drahtlose Integration in ein Firmennetz bzw. ein umfassendes Prozesssteuerungssystem. Im Gegensatz zu vielen anderen Wearable Computern sind kaum Schnittstellen für den Anschluss anderer Geräte oder Sensoren vorgesehen, so dass die Variationsmöglichkeiten relativ eingeschränkt sind.

Smart Clothes: Integration in die Kleidung

Der Übergang von Wearable Computern zu Smart Clothes, d.h. die Integration von informations- und kommunikationstechnischen Komponenten in die Kleidung, ist fließend. Der oben genannte Gürtelschnallenrechner QBIC ist dafür ein gutes Beispiel: Er ist in einen Bestandteil herkömmlicher Kleidung integriert, wird aber über jeder Kleidung getragen und bleibt insofern separat und hat keine Bekleidungsfunktion. Andere Wearable Computer sind ebenfalls mit Tragesystemen ausgestattet – Gürtel, Holster, Rucksack –, die keine Bekleidungsfunktion erfüllen. Der kleine aber feine Unterschied ist, dass z.B. der MA von Xyberonaut als »DesktopPC am Gürtel« gestaltet wurde (der auch im Rucksack getragen werden kann) wohingegen der QBIC ein »Rechner im Gürtel« ist⁴. Ein anderes Beispiel sind die so genannten Watch-Computer (z.B. Miller 2002), die wie eine Uhr am Arm getragen werden. Darüber hinaus existieren Designstudien zur Integration von Computertechnologie in Schmuck und andere Accessoires.

Denkt man die Ergänzung der Kleidung um Computer konsequent weiter, liegt der Gedanke nahe, alle Bestandteile einer mobilen Lösung in einem geeigneten Kleidungsstück unterzubringen. Das MIThril-Projekt des MIT MediaLab (2003) stellt ein solches Konzept vor: Alle benötigten Komponenten werden »unsichtbar« in eine Weste eingebaut. Infineon⁵ und Rosner boten ab 2005 eine waschbare Jacke mit eingewebten Leiterbahnen an, in die auf den Jackenärmel eine Tastatur zur Steuerung von Unterhaltungselektronik aufgebracht war (siehe Abb. 1 und Jung/Lauterbach 2002, Pahl 2004). Bei diesen kommerziell angebotenen Produkten

4 Der QBIC dient im Europäischen Großprojekt wearIT@work als Ausgangspunkt für eine verteilte Hardware-Plattform für Wearable Computing in Produktion, Wartung, Krankenhaus und Katastropheneinsatz (siehe wearit@work 2004).

5 Infineon hat bereits länger an Textilintegration gearbeitet (siehe auch Marculescu et al 2003)

der Unterhaltungsbranche werden als mobile Endgeräte Mobiltelefone, PDAs oder MP3-Player verwendet, die meistens nicht zum Lieferumfang des Kleidungsstücks gehören und vor dem Waschen auf jeden Fall entfernt werden müssen. Einige dieser Produkte wurden mittlerweile jedoch bereits wieder vom Markt genommen (vgl. Pahl 2004, Günzel 2005).

Abb. 5: Integration einer Tastatur in die Bekleidung (Bild: MRC)



In der Forschung zu *Smart Clothes* gibt es jedoch deutliche Bestrebungen in Richtung einer tieferen Integration. Dabei handelt es sich um »Smart Fabrics«, d.h. die Integration von Drähten oder elektronischen Komponenten in Gewebe und den Stoff selbst. Seitens der Materialforschung wird an neuen Materialien und Fasern geforscht, die z.B. leitfähig sind und gleichzeitig eine Abschirmung gegenüber elektrischen Feldern liefern. Ein Beispiel dafür sind die metallisierten Garne der Firma Statex aus Bremen.

Ein Unterschied zwischen Wearable Computern und Smart Clothes ist die Kompaktheit der ersteren und die Verteiltheit der letzteren. Bei der Realisierung mobiler Lösungen werden zukünftig – abhängig von der zu unterstützenden mobilen Tätigkeit – sowohl Komponenten der einen als auch der anderen Gerätekategorie benötigt, diese Heterogenität wird die Qualität mobiler Lösungen positiv beeinflussen. Wie bereits in der Vergangenheit ist weiterhin damit zu rechnen, dass Chips kleiner, schneller und leistungsfähiger werden, was Smart Clothes entgegen kommt. Seitens der Textilintegration elektronischer Komponenten ist die Frage der Waschbarkeit noch nicht gelöst und eine dauerhafte physische Verbindung der »harten« Chips mit dem »flexiblen« Stoff muss noch bis zur Marktreife entwickelt werden (vgl. Mecheels et al 2004).

Interaktionsmedien

Eine mobile, tragbare Rechneinheit allein macht noch keine mobile Lösung aus, da die zentralen Qualitätskriterien das Primat der ›eigentlichen‹ Aufgabe und die Gebrauchstauglichkeit der unterstützenden mobilen Lösung sind. Um eine beiläufige Mensch-Computer-Interaktion zu realisieren, werden mobile Ein- und Ausgabemedien sowie eine mobile Sensorik eingesetzt, die als Mittel zur expliziten und impliziten Interaktion dienen.

Explizite und implizite Eingaben lassen sich folgendermaßen unterscheiden: Eine bewusste Eingabe erfolgt durch eine zielgerichtete Handlung der BenutzerIn mit dem Ziel der Interaktion zwischen Mensch und Computer und durch die Nutzung des Computers als Primärartefakt. Die implizite Eingabe besteht in der weitgehend automatischen sensorischen Erfassung und Auswertung von Messwerten durch das Computersystem, die durch das Verhalten der NutzerIn ausgelöst und vom Computer als Eingabe interpretiert werden (vgl. Schmidt 2000). Die implizite Interaktion erfolgt somit auf der Ebene der technischen Komponenten, d.h. zwischen dem Wearable Computer und der Infrastruktur der Umgebung, ohne bewusste Benutzung durch die TrägerIn. Die Realisierung einer impliziten Interaktion stellt eine große technische Herausforderung dar. Im Folgenden wird der Schwerpunkt jedoch auf die explizite Interaktion gelegt, die Interaktion der TrägerIn einer mobilen Lösung mit der Technik, wobei die implizite Interaktion auch der Unterstützung bzw. Vereinfachung der expliziten Interaktion dienen kann.

Eingabe

Im Gegensatz zu herkömmlichen Anwendungssystemen muss bei mobilen Lösungen immer berücksichtigt werden, dass die primäre Aufgabe der BenutzerIn in der realen Welt verortet ist und ihre Aufmerksamkeit deshalb ebenfalls dort liegt. Eingabetechnologien müssen daher so gestaltet sein, dass die ›eigentliche‹ Interaktion mit der realen Umgebung möglichst wenig durch die nachrangige Interaktion mit dem Computer behindert wird. Hierbei hängt es jedoch von den spezifischen Eigenschaften der Umgebung und der primären Aufgabe ab, inwieweit die parallele Bedienung z.B. eines Eingabegerätes als hinderlich empfunden wird. Für das bei Computersystemen für mobile Lösungen zu erwartende große Spektrum an Bediensituationen ist daher bereits jetzt ein reichhaltiges Repertoire an Eingabegeräten vorhanden:

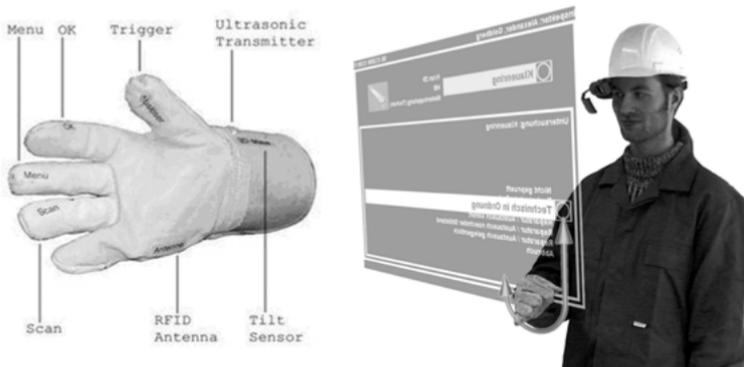
- Spracheingabe
- verschiedene Arten mobiler Tastaturen

- drahtlose Zeigemedien und Navigationskomponenten
- auf konkrete Aufgaben hin optimierte Eingabegeräte

Spracheingabe, d.h. das »natürliche Sprechen mit dem Computer« war schon seit den Anfängen der Computertechnologie das Ziel einiger Technologen und häufig auch erklärter Wunsch der BenutzerInnen. Doch in konkreten Anwendungssituationen stellte sich heraus, dass Spracheingabe seitens der BenutzerInnen letztlich doch nur selten genutzt wird. Darüber hinaus wurde nachgewiesen, dass sich eine vollständige Erkennung natürlicher Sprache technisch nicht realisieren lässt. Der praktische Einsatz beschränkt sich deshalb und wegen der übermäßig hohen Bedarfe an Rechnerkapazitäten für Realtime-Erkennung natürlicher gesprochener Sprache auf sehr eingeschränkte Systeme zur Schlüsselworterkennung.

Mobile Tastaturen und Zeigegeräte sind durch die Desktop-Metapher motiviert, doch ist die Eingabe von längeren Texten während einer mobilen Tätigkeit selten erforderlich. Interessant sind für innovative mobile Lösungen deshalb neu konzipierte Eingabekomponenten. Die Wearable Group im Laboratory for Interactive Computer Systems der Carnegie Mellon University (siehe Bass et al 2001) hat für den VuMan in einem mehrphasigen, partizipativen Design-Prozess ein mobiles Eingabegerät für die Handhabung einer elektronischen Checkliste in einer intuitiven und beiläufigen Weise entwickelt. Die Navigation erfolgt per Wählscheibe. Die Auswahl eines Menüpunkts wird über einen Knopf auf der Wählscheibe realisiert. Das Gerät wird am Gürtel getragen und kann wahlweise mit der rechten oder linken Hand benutzt werden.

Abb. 6: Winspect-Eingabehandschuh (1. Generation) und die intendierte Nutzung (Bilder: TZI)



Ebenfalls zur Checklistenauswahl ist der »Winspect-Handschuh« entwickelt worden (vgl. Herzog et al 2003). Er hat allerdings noch weitere integrierte Funktionen. In der ersten Version wurde ein handelsüblicher Arbeitshandschuh aus Leder mit Sensoren und Schaltern bestückt (siehe Abb. 6, links). U.a wurde ein Neigungssensor eingebaut, der das Drehen der Hand in Handgelenk und Unterarm nach rechts oder links registriert. Diese Bewegung wird mit dem Rollen eines Menüs verknüpft, so als ob auf einem kleinen Abschnitt des Arms eine Liste befestigt wäre, von der man je nach Drehung der Hand einen Eintrag im Vordergrund sieht (Bildschirmdarstellung siehe Abb. 6, rechts). Die Auswahl des hervorgehobenen Menüpunktes erfolgt ebenfalls per Hand durch die Berührung des Daumens mit dem Mittelfinger.

Abb. 7: Komponenten des optimierten Winspect-Eingabehandschuhs, 3. Generation (Bild: MRC)



Jeder Finger des Winspect-Handschuhs ist mit einer anderen Funktion belegt, der kleine Finger beispielsweise mit der Aktivierung eines mobilen RFID-Lesegerätes (Radio Frequency Identification). Diese Identifikationstechnologie wird dazu verwendet, die durch die BenutzerIn explizit zu tätigenden Eingaben auf die Elemente zu reduzieren, die für die Dokumentation der aktuellen, konkreten Inspektion unumgänglich sind. Zentrale Elemente beispielsweise einer zu inspizierenden Maschine sind aus diesem Grund mit RFID-Tags zu kennzeichnen. Der Interaktionsaufwand für die BenutzerIn reduziert sich so auf einen Scan, sie muss nicht mehr auswählen, welche Checkliste sie an dem jeweiligen Bauteil benötigt, dessen Auswahl erfolgt automatisch anhand der eingescannten Identifika-

tion (Kenn/Rügge 2006). In der dritten Version wurden die im Handschuh enthaltenen technologischen Komponenten optimiert (Lawo 2006). Für konkrete Einsatzsituationen wurden die Trageeigenschaften des Handschuhs aufgrund der Wünsche potenzieller BenutzerInnen angepasst, indem die Finger vom Handschuh befreit und Schaltsensoren an Stellen platziert wurden, an denen sie bei der eigentlichen Arbeit in der gegenständlichen Umgebung nicht stören. Es gibt mittlerweile einen Handschuh als Eingabegerät für Wartungsprozesse in der industriellen Fertigung oder bei der Inspektion von Flugzeugen und ein Armband als Eingabegerät für die sterile, berührungslose Interaktion im Krankenhaus.

Ein ganz anders geartetes Beispiel einer mobilen Eingabetechnologie, die von den potenziellen BenutzerInnen als wenig beeinträchtigend wahrgenommen wird, ist eine Papier-basierte Stifteingabetechnologie der Firma Anoto (2006). Dabei handelt es sich um ein speziell bedrucktes Papier, das mit einem Stift beschrieben wird, der sowohl über eine Kugelschreibermine verfügt als auch über eine Kamera und einen Speicher, mit dem die Schreib- und Zeichengesten, die mit dem Stift auf dem Papier ausgeführt werden, elektronisch aufgezeichnet und später als Eingaben weiterverarbeitet werden, ohne dass eine erneute Erfassung erforderlich ist. Die Benutzung dieser mobilen Eingabetechnologie referiert auf eine Kulturtechnik, die weit verbreitet und vielen Menschen geläufig ist und die darüber hinaus auch bisher Teil des mobilen Arbeitsprozesses war und von daher keine wesentliche Verhaltensänderung bei der BenutzerIn erfordert.

Ausgabe

Abhängig von der konkreten Ausprägung der mobilen Tätigkeit und im Verbund mit der Wahl der Eingabemedien und des Rechners sind ganz verschiedene Formen der Ausgabe denk- und realisierbar. Vom Desktop-Computing ausgehend ist die offensichtlichste Art eine visuelle Ausgabe, doch bieten sich für mobile Lösungen auch Sound und Sprache sowie eine taktile Ausgabe an. Der Fokus der EntwicklerInnen liegt jedoch auf der visuellen Ausgabe. Bevorzugtes Ausgabemedium für Wearable Computing Lösungen ist das *Head Mounted Display* (HMD), auf das hier deshalb besonders eingegangen wird.

Für eine mobile, freihändige Benutzung, bei der Arme und Hände zur Handhabung des Computersystems nicht zur Verfügung stehen, ist die Verwendung einer Halterung, z.B. am Kopf – ein so genanntes *Headset* – erforderlich. In ein solches Headset wird ein Mikrodisplay unter Verwendung von optischen Linsen-, Prismen- und/oder Spiegelsystemen eingebaut. Eine Kombination mit einem Kopfhörer als zusätzlichem Ausgabekanal bietet sich an. Je nach Bedarf kann ein Headset darüber

hinaus mit einem Mikrofon, einer Kamera und einem Trackingsystem ausgestattet werden. HMDs unterscheiden sich voneinander durch das Befestigungssystem sowie durch die daraus resultierenden Trageeigenschaften, die bestimmt werden durch Größe, Gewicht, Kompaktheit und Auffälligkeit des HMDs und dadurch einen hohen Einfluss auf die Akzeptanz dieser mobilen Technologie durch die BenutzerIn haben.

Die Firma MicroOptical Corporation entwickelte verschiedene Varianten so genannter Clip-on-HMDs, die z.B. an einer handelsüblichen Brille oder an einer Schutzbrille befestigt werden können (siehe Abb. 8). In Einsatzbereichen, in denen Schutzbrillen obligatorisch sind, wird diese Art der Befestigung nicht als Einschränkung gesehen. Eine Integration in eine Brille ist auch noch aus anderen Gründen zu begrüßen:

- Brillen werden individuell an die Physionomie einer Person angepasst.
- Das Brillengestell bietet viele Möglichkeiten, die erforderliche Technik unterzubringen. Nicht nur das Display und seine Elektronik, sondern auch die Stromversorgung kann so ohne Beeinträchtigung der BenutzerIn z.B. als »Gegengewicht« eingebaut werden.
- Eine Brille schützt die Augen ihrer TrägerIn.
- Dadurch, dass Brillen so dicht an empfindlichen Körperteilen platziert sind, werden sie automatisch von der BenutzerIn geschützt. Das bedeutet für die Technik, dass sie zum einen weniger robust und damit leichtgewichtiger gebaut werden kann und zum anderen, dass mit weniger Ausfällen durch Bruch zu rechnen ist.

Abb. 8: Head-Mounted Display an einer Schutzbrille (Bild: MRC)



Anfang 2007 kündigte MicroOptical⁶ jedoch die Einstellung der Produktion von monokularen HMDs an, was für die EntwicklerInnen von Wearable Computing-Lösungen einen herben Rückschlag bedeutet. Nächster Hoffnungsträger ist nun das Projekt »WearIT@work – Unterstützung mobiler Arbeiter durch Wearable Computing« (wearitatwork 2004), in dem die Firmen Lumus und Zeiss an der Entwicklung fortgeschrittener HMDs und vor allem an tragbaren Headsets arbeiten.

Es gibt vielfältige Kombinationsmöglichkeiten aller hier vorgestellten Bestandteile. Welche Kombination eine einsetzbare und für BenutzerInnen akzeptable mobile Lösung ergibt, kann nicht per se entschieden werden, da die Gebrauchstauglichkeit letztlich nur auf der Basis der Bedingungen im konkreten Anwendungsbereich und des Wohlbefindens der TrägerIn bewertet werden kann. Dazu ist eine Untersuchung der Einsatzbereiche und der Nutzungssituationen erforderlich.

Untersuchungsspektrum »mobile Arbeitstätigkeit«

Mobile Lösungen werden durch die Miniaturisierung von Computertechnologie, durch eine großflächige und vielfältige drahtlose Vernetzungsinfrastruktur, durch die Integration verschiedener Netztechnologien und durch eine Effektivierung mobiler Stromversorgung ermöglicht und auch getrieben. Dass sie nach mehr als 10 Jahren Forschung und Entwicklung noch keinen nennenswerten Marktanteil erreicht haben, sondern bisher weitestgehend nur als Prototypen existieren, ist aus technischer Perspektive verwunderlich, denn es gibt bereits zahlreiche technologische Komponenten für den mobilen Einsatz und auch Ansätze für komplette mobile Systeme. Doch wirft man einen Blick auf die intendierten Einsatzbereiche, dann werden die Ursachen offensichtlich: Die Technik ist weder dem mobil tätigen Menschen noch seiner primären Aufgabe jenseits der »Computerarbeit« angepasst, mobile Lösungen sind noch nicht tragbar.

Um die notwendige Anpassung zu erreichen, ist eine fundierte Vorstellung des Begriffs der »mobilen Tätigkeit« essentiell. Er ist Voraussetzung für das Verständnis der Besonderheiten, denen mobile und Wearable Computing-Lösungen gerecht werden und die sie realisieren müssen, wenn sie Innovationspotenziale für Arbeitsprozesse eröffnen und neue Einsatzbereiche angemessen bedienen sollen.

6 Das Unternehmen ist mittlerweile in MIVU umbenannt worden und hat die Zielgruppe geändert, siehe www.microoptical.com und www.myvu.com (zuletzt gesehen 1.3.2008)

und auch wenn Warenwirtschaftssysteme den Warenfluss auf der symbolischen Ebene vollständig begleiten, so ist es in der Praxis doch immer noch erforderlich, dass Menschen diese hochkomplexen Einrichtungen kontrollieren und instandhalten oder mit Waren bestücken. Zur Unterstützung dieser Arbeit können mobile Lösungen ein geeignetes Instrumentarium sein.

Mobile Arbeitstätigkeiten sind (Teil-)Aufgaben, die mehrere der folgenden Eigenschaften besitzen: Eine mobile Tätigkeit wird *in der Bewegung* ausgeführt, z.B. die Kommissionierung von Gütern oder die landwirtschaftliche Bewirtschaftung. Eine mobile Tätigkeit findet *an wechselnden Einsatzorten* statt, z.B. die Wartung und Instandsetzung von Schiffen, die ständig unterwegs sind, die weltweite Inbetriebnahme und Wartung von Industrieanlagen und auch ambulante Pflege gehört dazu. Oder die mobile Tätigkeit wird zwar an einem Ort, aber an wechselnden, oftmals *großen oder weitläufigen Objekten* durchgeführt, z.B. die Wartung eines Flugzeugs im Hangar oder der Service in einem Zug. Wesentliches Charakteristikum mobiler Arbeitstätigkeiten ist, dass die primäre Aufgabe und damit die Aufmerksamkeit der BenutzerIn *in der realen gegenständlichen Welt verortet* ist, wie z.B. beim Reparieren, Messen, Prüfen von mechanischen Teilen, beim Arbeiten in dynamischen Umgebungen und bei medizinischen und pflegerischen oder Service-Tätigkeiten.

Mobile Arbeitstätigkeiten sind fast ausnahmslos in eine Arbeitsorganisation eingebettet, die auf Zusammenarbeit beruht und von einem EDV-gestützten Prozessablauf begleitet wird. Beispiel Wartung: Bei Wartungsarbeiten wird zunächst eine Checkliste der zu prüfenden Teile und Funktionen erzeugt, die dann an der Anlage abgearbeitet wird. Die Befunde werden dokumentiert und zur Erstellung der Rechnung oder für die Versicherung benötigt. Im Laufe der Wartungsarbeiten sind Messungen notwendig, das Nachschlagen in der Technischen Dokumentation, der Blick auf eine Zeichnung oder eine Frage an eine KollegIn, die gerade anderswo beschäftigt ist.

Denkbar ist – und dies wurde in der Vergangenheit auch versucht –, versierte MitarbeiterInnen zur Unterstützung draußen vor Ort mit Notebook, PDA oder Smartphone auszustatten. Doch dieses Vorgehen hat sich als nicht praktikabel, in den meisten Fällen haben sich diese Geräte sogar als hinderlich erwiesen. Denn Arbeitsorte wie die auf einer Windkraftanlage, an der Turbine eines Flugzeugs oder den Leuchtsignalen einer Ampel haben ihre eigenen Gesetze: Es gibt nur wenige sichere Standflächen, es ist zugig, schmutzig, nass und vor allem beengt. Oft benötigt das Personal eine Hand zum Festhalten oder zum Hantieren, muss Arbeitsschutzbekleidung (inkl. Handschuhe) tragen und verfügt nicht über die Zeit, lange nach den Informationen zu suchen, die für den jeweiligen Arbeits-

schritt benötigt werden. Fazit: Notebook, PDA oder Smartphone und Software, gestaltet nach der Software-Metapher, sind für diese Praxis nicht geeignet.

Benötigt werden für mobile Arbeitstätigkeiten »mobile Assistenzsysteme«, die am Körper getragen werden, und die sich, wie (menschliche) Assistenten, z.B. wie ein Handlanger oder ein Tutor, verhalten: d.h., dass sie die NutzerIn nicht *be-*, sondern vielmehr *entlasten*, die richtige Information zur richtigen Zeit am richtigen Ort präsentieren und dank einer einfachen Handhabung keine besondere Aufmerksamkeit erfordern. Für folgende Teilaufgaben sind mobile Lösungen besonders geeignet:

- Bildung, Training, Instruktion »on the job«
- Kommunikation und Kooperation: synchroner oder asynchroner Dialog zwischen einzelnen Personen oder größeren Teams
- Dokumentation: Informationsbereitstellung durch Zugriff auf Pläne, Zeichnungen, Handbücher, Reparaturanleitungen usw. aber auch Berichtswesen, d.h. Protokollierung und Berichterstattung
- Messen, Erfassen, Auswerten, Vergleichen und Speichern von Daten jederzeit und an beliebigen Orten

Diese Art von Tätigkeiten ist in vielen verschiedenen Anwendungsbereichen anzutreffen. Eine Generalisierung hinsichtlich der Anforderungen an mobile Lösungen ist dennoch nicht möglich, da in den verschiedenen Einsatzbereichen wiederum sehr unterschiedliche Nutzungsbedingungen herrschen, die wiederum den Einsatz anderer technischer Artefakte erfordern.

Architektur und Problemstruktur mobiler Lösungen

Für mobile Lösungen werden annähernd die gleichen »Bausteine« wie für andere Computersysteme benötigt: Hardware, Software, Vernetzung, Ein-/Ausgabe-Geräte, Benutzungskonzepte. Hinzu kommen jedoch Sensoren und automatische Verfahren zu ihrer Auswertung, da z.B. die Ortsveränderung und der Wechsel der Umgebungsbedingungen genutzt werden können, um den »Informationsraum« ohne explizites Zutun der BenutzerInnen einzuschränken: Die TechnikerIn an der rechten Turbine braucht nur die technischen Details, die diese betreffen, es sei denn, ihre KollegIn an der linken Turbine hat eine Frage zu ihrer parallel auszuführenden Aufgabe. So etwas lässt sich durch die automatische Auswertung des so genannten »Kontextes« berechnen: Wo steht die TrägerIn im

Verhältnis zum Flugzeug, an welcher Stelle im Wartungsprozess ist sie gerade, mit wem muss sie sich koordinieren?

Die Architektur mobiler Lösungen lässt sich analog zur Terminologie des IEEE-Standards 1471-2000, wie sie z.B. Hasselbring verwendet, beschreiben:

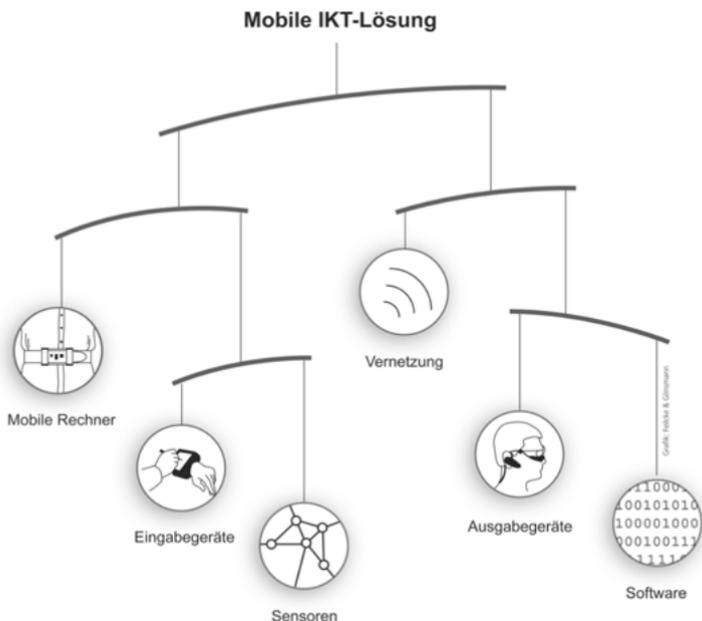
»Definition (Software-Architektur): Die grundlegende Organisation eines Systems, dargestellt durch dessen Komponenten, deren Beziehung zueinander und zur Umgebung sowie den Prinzipien, die den Entwurf und die Evolution des Systems bestimmen.« (Hasselbring 2006, S. 48)

Die Komponenten, die benötigt werden, und die Entscheidungen, die dafür gefällt werden müssen, bestimmen die Systematik der Gesamttechnologie. Festgelegt werden mit diesen Entscheidungen u.a. die ›Arbeitsteilung‹ innerhalb des gesamten Computersystems (Topologie, Vernetzung, Softwarearchitektur, Hardware) und auch die ›Arbeitsteilung‹ zwischen Mensch und Computer (explizite und implizite Interaktion). Es gibt eine große Ähnlichkeit der Architekturmuster für mobile Lösungen mit den bekannten Architekturmustern der Softwareentwicklung, z.B. Client/Server-, Peer-To-Peer-, Service-orientierte und Komponenten-orientierte Architekturen (ebda.). Doch gibt es auch spezifische Eigentümlichkeiten von mobilen und Wearable Computing-Lösungen, die sich aus den Merkmalen der Mobilität in den intendierten mobilen Tätigkeitsbereichen ergeben, auf die im Folgenden näher eingegangen wird.

Die Architektur eines Computersystems für eine mobile Lösung wird durch wenige Hauptstrukturen bestimmt, die sich jedoch mehr als das bei herkömmlichen stationären Lösungen der Fall ist, wechselseitig bedingen. Eine kleine Variation bei einem Element hat starke Auswirkungen auf alle anderen. Ein Beispiel soll diese enge Verzahnung verdeutlichen: Wählt man eine *monadische Architektur* – d.h. entscheidet man sich für einen autonomen Ansatz, der das mobile Gesamtsystem autark arbeiten lässt, ohne eine drahtlose Integration in ein Netzwerk –, dann müssen die am Körper getragenen Geräte so leistungsstark sein und über so viele Ressourcen verfügen, dass sie alle erforderlichen Daten, Programme und Dienste enthalten, die für den aktuellen Einsatzzweck erforderlich sind. Die mitgeführten Informationen und Dienste müssen vollständig sein, ein Zugriff auf externe Datenquellen und eine Kommunikation mit anderen darf nicht notwendig sein, und eine Sicherung der Daten muss im mobilen Endgerät selbst oder während des Ladens der mitgeführten Stromversorgung realisiert werden. Der Abgleich mit dem umgebenden EDV-System kann dann nur außerhalb der mobilen Tätigkeit, z.B. während des Feierabends erfolgen. Bei dieser monadischen Architektur wird der Wunsch

der BenutzerIn nach absolutem Schutz und Sicherheit des eigenen Systems vor Angriffen von außen erfüllt.

Abb. 10: Komponenten einer mobilen Lösung und ihre Abhängigkeit voneinander (Bild: MRC)



Im Vergleich dazu stellt eine *eingebettete Architektur* – ein nicht autark arbeitendes System, das drahtlos in ein Computer- und Kommunikationsnetzwerk integriert ist – deutlich geringere Anforderungen an die Prozessor- und Speicherleistungen, erzeugt aber auf der Ebene der Software und der Vernetzung eine wesentlich höhere Komplexität. Sie ermöglicht einen Zugriff auf verteilte hochaktuelle Informationen und Ressourcen (Rechnerleistung, Speicherplatz) sowie die Kommunikation mit kooperierenden ExpertInnen und Computersystemen. Dafür erfordert eine derartige eingebettete Architektur die Sicherstellung der zuverlässigen Verfügbarkeit der Vernetzung, eine geeignete Verschlüsselung des Datentransfers und Mechanismen zur verteilten Datenverarbeitung.

Aus der Sicht der BenutzerIn ist es allerdings unwesentlich, ob sich die Daten und Dienste auf dem mobilen Endgerät befinden oder über eine sichere (permanente oder ad-hoc) drahtlose Verbindung zu einem Server z.B. im mitgeführten Werkzeugkoffer, im Fahrzeug, auf dem Werksgelände oder am anderen Ende der Welt realisiert werden. Für sie sind nur die

Einfachheit und Beiläufigkeit der erforderlichen expliziten Interaktion mit dem technischen System sowie dessen Trageeigenschaften relevant.

Aus der Perspektive der EntwicklerInnen sind die zentralen Dimensionen der Architektur mobiler Lösungen folgende:

- *Hardware/Software*: Die physische Hardware-Konfiguration inklusive der Energieversorgung und der damit eng verbundenen Vernetzung sowie die Softwarearchitektur.
- *Umgebung*: Die Integration des technischen Systems in die Umgebung im weitesten Sinne, d.h. die Einbeziehung sowohl der physischen als auch der elektronischen und digitalen Umgebung sowie die Integration des mobilen Systems in die umgebende informations- und kommunikationstechnische Infrastruktur (nicht nur auf Hardware-Ebene, sondern insbesondere auch auf Ebene der Anwendungsprogramme).
- *Nutzungskontext*: Die Regelung der Interaktion zwischen Mensch und Computer; die Integration des eingesetzten mobilen Systems in den individuellen Arbeitsprozess bzw. Handlungsablauf sowie in die Kommunikationsabläufe von kooperierenden Teams.
- *Interaktion und Proaktivität*: Die Arbeitsteilung zwischen Mensch und Computer, die vor allem die Benutzung in der Bewegung berücksichtigt und eine Bedienung fördert, die nicht von einer kontinuierlichen und uneingeschränkten Aufmerksamkeit der BenutzerIn für das Computersystem ausgeht.

Da das vorliegende Buch den Engpass Mensch-Computer-Interface fokussiert, wird im Weiteren vor allem auf die Gestaltung der Interaktion zwischen Mensch und Computer für Wearable Computing-Lösungen eingegangen. Der Interaktionsstil für mobile Lösungen ist geprägt vom Übergang von der bisher bekannten kleinschrittigen Interaktion nach der Desktop-Metapher hin zu einer durch die Erfassung und Auswertung des mobilen Kontextes beförderten systemgenerierten Proaktivität, deren konkrete Ausprägung vom jeweiligen Anwendungsbereich und vom Nutzungskontext bestimmt wird. Für die Gestaltung der Interaktion kann zwar auf bekannte Richtlinien zurückgegriffen werden (z.B. DIN EN ISO 13407 und DIN EN ISO 9241), diese müssen aber an die Bedingungen der Mobilität angepasst werden. Das zentrale Bewertungskriterium hierfür ist die *Gebrauchstauglichkeit*. Sie ist das Nadelöhr, der keine Wearable Computing-Lösung bisher gerecht werden konnte.

Mobile Lösungen im Spannungsfeld übergeordneter Leitkonzepte und Interaktions-Gestaltungsparadigmen

Technik wird nach expliziten und impliziten Leitbildern gestaltet. Mit Leitbildern sind hier die Annahmen von EntwicklerInnen gemeint, wie ein zu modellierender Arbeitsprozess abläuft, wie Menschen in bestimmten Situationen handeln und welche mentalen Modelle die zukünftigen BenutzerInnen haben, aber auch die Vorgehensweise bei der Entwicklung von Technik. Aus diesen Leitbildern entstehen Paradigmen, das sind technologieorientierte Vorgaben, wie Anforderungen aus einem bestimmten Bereich zu modellieren sind. Mobile Lösungen werden vor dem Hintergrund vorhandener Konzepte gestaltet bzw. werden an ihnen gemessen, deshalb werden hier die für mobile Lösungen relevanten Aspekte dieser Konzepte beschrieben.

Desktop Computing und Mobile Computing

Die Desktop-Metapher ist das bekannteste und am weitesten verbreitete Paradigma der Gestaltung von Benutzungsschnittstellen von »Personal Computern (PCs)« und Notebooks. Die Ebene der expliziten Interaktion zwischen Mensch und Computer wird hier nach dem Leitbild des Schreibtisches gestaltet, Vorlage sind die Arbeitsprozesse des Büroalltags (Laurel 1990). Gegenstand der Arbeitstätigkeit ist vor allem Information: Beim Erstellen einer Rechnung handelt es sich z.B. darum, dass jemandem mitgeteilt werden soll, für welche Leistung er welchen Preis in welcher Form und zu welchem Zeitpunkt zu zahlen hat. Auch das Erstellen von Zeichnungen ist nichts anderes als die Materialisierung von Informationen darüber, wie z.B. eine Maschine oder ein Gebäude gebaut werden soll. Diese Arbeit wird heutzutage sozusagen »im Computer« erledigt. Für viele Unternehmen bedeutet das, dass es erforderlich ist, die Informationsflüsse zu beschleunigen und effektiver auszuwerten, um die Wettbewerbsfähigkeit zu erhalten. Hierbei lassen sich deutliche Parallelen zur Automatisierungstechnik entdecken, in der ganze Klassen von Aufgaben an Maschinen übertragen wurden, Frieder Nake (1992) spricht von der »Maschinisierung von Kopfarbeit«.

Die Bedienoberflächen dieser Computersysteme sind so gestaltet, dass sie die BenutzerIn bei der Erfüllung ihrer Aufgabe unterstützen: Die Tastatur zum Schreiben, der Bildschirm zum Sehen, die Maus zum Zeigen, der Lautsprecher zum Hören der Informationen. Die Desktop-Metapher verwendet als Bildelemente Windows, Icons, Menus, Pointing (WIMP), die mittels der so genannten Direkten Manipulation benutzt

werden. Die primäre Aufgabe der BenutzerIn ist die geschickte und kompetente Bedienung von Informations- und Kommunikationstechnik. Die herkömmliche, papierbasierte Büroarbeit wurde so auf eine symbolische Ebene gebracht. Die Aufmerksamkeit der BenutzerIn liegt auf der Benutzung des Computersystems; der Computer ist Mittel, Medium der Kommunikation und Information, Medium der Arbeit (Schelhowe 1997).

»Mobil« hat im Gestaltungsbereich des Desktop Computing eine bestimmte Bedeutung: Es bezeichnet Notebooks bzw. robuste Prozessoren mit geringem Energieverbrauch sowie Handheld-Geräte wie PDAs und Handys/Smartphones. Diese sind durch ihre autarke Stromversorgung an wechselnden Orten benutzbar und stellen zusammen mit der verfügbaren (evtl. drahtlosen) Konnektivität zum Inter- oder Intranet einen relativ vollständigen (Schreibtisch-)Arbeitsplatz zur Verfügung. Die ständige Erreichbarkeit und der ortsunabhängige Zugriff auf Informationen sind die wesentlichen Aspekte. Die Möglichkeit, Bürotätigkeiten überall auszuüben, wird als »mobiles Büro« bezeichnet; wenn der Zugriff auf Informationen im Vordergrund steht, wird die Bezeichnung »mobiles Internet« verwendet. Die Benutzungsoberfläche ist teilweise an die Hardwaregegebenheiten angepasst (z.B. Stiftbedienung bei PDAs und Tablet PCs), die Anordnung der Bedienelemente folgt dem Leitbild des Telefons oder der Desktop-Metapher, je nachdem, ob das eingesetzte mobile Endgerät ursprünglich aus der Computertechnik oder der Kommunikationstechnik stammt. Von einer freihändigen oder beiläufigen Benutzung kann nur selten die Rede sein (siehe Abb. 11).

Abb. 11: Mobile Lösung mit Standardgeräten, konzipiert für den privaten oder Business-Bereich (Bild: MRC)



Mixed Realities und Ubiquitous Computing

Beim Desktop-Computing ›verschwindet‹ die Arbeit im Computer, unter dem Leitbild von *Virtual Reality* ›verschwindet‹ sogar der Mensch in einer vollständig vom Computer generierten Welt. Die menschliche Wahrnehmung wird in der so genannten *immersiven* Virtual Reality gegenüber der realen Welt weitestgehend abgeschottet, so dass nur computergenerierte Effekte wahrgenommen werden können. Der Mensch wird mittels Sensorik ›beobachtet‹, so dass eine ebenfalls vollständig vom Computersystem gesteuerte Rückkopplung erfolgen kann. Als Gegenentwurf genau hierzu sind die Ansätze der *Mixed Realities* aufzufassen: Gegenständliche Objekte der physischen Realität werden mit virtuellen Objekten kombiniert, der Mensch wird in seiner realen Umgebung belassen. Hierzu gehören Entwicklungen aus dem Bereich Augmented Reality sowie auch Tangible Media und Graspable User Interfaces⁷. Im ersten Fall wird die Sicht auf die reale Welt mit computergenerierten Bildern durch punktgenaue Überlagerung ergänzt⁸, im zweiten Fall werden Gegenstände der realen Welt so mit Modellen im Computer verbunden, dass die physischen Gegenstände Repräsentanten der virtuellen Objekte sind und eine Handhabung oder Veränderung des realen Gegenstands eine entsprechende Anpassung im Virtuellen auslöst (siehe Abb. 12). Die realen Gegenstände und die Umgebung werden dazu um virtuelle Informationen und ›Computerfähigkeiten‹ ergänzt.

Unter dieser Perspektive soll eine allgegenwärtige elektronische Unterstützung gewährleistet werden, die dem Menschen im Idealfall verborgen bleibt (siehe Norman 1998), da allein die Umgebung und die Alltagsgegenstände mit Informations- und Kommunikationsfunktionalität ausgestattet werden sollen und der Mensch frei von computertechnischen Artefakten bleibt. Historisch ist dieser Ansatz mit unterschiedlichen Begriffen belegt⁹: »Ubiquitous Computing« wurde von Weiser (1993) als akademisches Forschungskonzept eingeführt; Firmen wie IBM verwenden »Pervasive Computing« als Bezeichnung für ein stärker praxis- und geräteorientiertes Konzept; in Europa wird »Ambiente Intelligence« für eine anwen-

7 Siehe den Beitrag von Eva Hornecker in diesem Band.

8 Ein ungelöstes Problem ist das Tracking der Bewegung und die Ermittlung der exakten Blickrichtung der BenutzerIn, das so genannte »Registrierungsproblem« (Barfield/Caudell 2001, S. 10-11). Darüber hinaus stellt die erforderliche Echtzeitvisualisierung extrem hohe Anforderungen an die verfügbaren Ressourcen, die von mobilen Endgeräten nicht erfüllt werden können. Ein pragmatischer Ansatz des Wearable Computing ist die Visualisierung von Informationen, die zwar den Ortsbezug haben, die Sicht der BenutzerIn auf die reale Welt jedoch nicht punktgenau überlagern.

9 Siehe dazu ausführlich Michael Friedewald in diesem Band.

dungsorientierte Software- und nutzungsorientierte Sichtweise auf das gleiche Ziel verwendet.

Die explizite Interaktion zwischen Mensch und Computer soll im Ubiquitous Computing durch den intensiven Einsatz und die Verfeinerung von Sensoren, durch die Kombination verschiedener Sensoren (sensor fusion) und durch eine intensive Nutzung maschineller Intelligenz unter dem Stichwort »context awareness« reduziert werden.

*Abb. 12: Real Reality-Objekte zur Fabriksteuerung
(Robben/Rügge 1998, S. 135)*



Wearable Computer und Smart Clothes

Während in der Vision vom Ubiquitous Computing von einer Allgegenwart der Computertechnologie ausschließlich in der Umgebung und in den benutzten Gegenständen ausgegangen wird, favorisiert das Paradigma des *Wearable Computing* als Ideal eine fast persönliche Beziehung zwischen der BenutzerIn und ihrem mobilen Computersystem, das völlig unabhängig von allen Infrastruktureinrichtungen der Umgebung funktioniert. Wearable Computing bedeutet zum einen den Einsatz »tragbarer« Rechner und meint damit die Integration der Technologie in die unmittelbare körperliche Nähe der TrägerIn. Es favorisiert ein »Anziehen« von Computertechnologie im Sinne von Kleidung, Schmuck oder im Sinne des Anlegens eines Tragesystems wie z.B. eines Gürtels, eines Rucksacks oder eines Holsters. Eingesetzt werden vor allem Wearable Computer,

Smart Clothes und Digital Accessoires. Zum anderen zeichnet sich Wearable Computing dadurch aus, dass es stärker durch interaktive Momente gekennzeichnet ist. Ziel ist die Realisierung einer ebenfalls allgegenwärtigen »intelligenten mobilen Assistenz«, die ihrer TrägerIn die richtige Information (oder Funktion) am richtigen Ort zur richtigen Zeit ohne expliziten Aufwand bereit stellt.

Wearable Computer sind leistungsfähige mobile Endgeräte, die von ihrer BenutzerIn so am Körper getragen werden, dass sie über das größtmögliche Maß an Bewegungsfreiheit für die Ausführung ihrer primären Tätigkeit verfügt. Bradley Rhodes (1997) fordert, dass ein Wearable Computer ständig »im Einsatz«, also immer angeschaltet und sofort benutzbar ist, auch in der Bewegung, und dass möglichst beide Hände frei bleiben für Aufgaben jenseits der Computerbenutzung. Ein Wearable Computer sollte den physischen Kontext der BenutzerIn mittels Sensoren erfassen und der BenutzerIn immer proaktiv Informationen liefern. Diese Charakterisierung verdeutlicht, ein Wearable Computer besteht nicht nur aus einer kleinen Recheneinheit mit einer bestimmten Leistungsfähigkeit. Die Ein-/Ausgabe-Geräte, ihre Anordnung am Körper der BenutzerIn, der Einsatz »intelligenter« Sensoren und insbesondere die Gestaltung einer »beiläufigen« Interaktion zwischen Mensch und Computer sind die wesentlichen Komponenten (siehe Rügge 2002, S. 2).

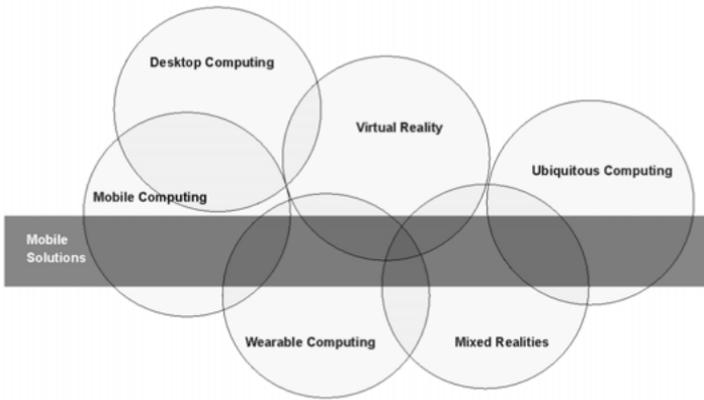
Abb. 13: Tragesystem mit Textilintegration für den industriellen Einsatz (Bild: MRC)



Smart Clothes stellen eine weitere Annäherung von Computertechnik an den Körper dar. Bisher beschränkt sich die Integration von informations- und kommunikationstechnologischen Komponenten noch auf modische Bekleidungsstücke, die mit Taschen und Kabelkanälen für die Unterbringung von Unterhaltungselektronik versehen sind (vgl. Abb. 1). Eine weitergehende Integration wird in Richtung der Entwicklung von leitenden Fasern und Stoffen und die Verbindung von elektronischen Bauteilen mit diesen forciert (siehe z.B. Mecheels et al 2004, Fackler 2006, Katragadda/Xu 2005, Linz et al 2005). *Smart Clothes* können so als nächste Generation von Wearable Computern aufgefasst werden.

Die primäre Aufgabe und damit auch die Aufmerksamkeit der BenutzerIn ist beim Wearable Computing wie auch bei den Mixed Realities in der realen Welt verortet. Computertechnologie wird nur unterstützend eingesetzt und ordnet sich den Gegebenheiten der Situation und den Anforderungen der realen Welt unter.

Abb. 14: Mobile Lösungen im Verhältnis zu den bekannten Leitbildern der Technikgestaltung (Bild: MRC)



Die Verwendung von Sensoren und die Interpretation des erfassten Kontextes spielt somit für Wearable Computing ebenfalls eine zentrale Rolle. Der wesentliche Unterschied zu Ubiquitous Computing ist die Platzierung der Technik ausschließlich am Körper der BenutzerIn. Diese Platzierung hat weitreichende Konsequenzen, die auch Auswirkungen auf die verwendbaren technologischen Komponenten, die Einsatzbereiche und vor allem auf die Akzeptanz seitens der BenutzerInnen haben. Zwar ist es technisch gleichermaßen möglich, eine TrägerIn aus der Umgebung heraus oder die Umgebung von der TrägerIn aus zu überwachen, doch

wird das Tragen der Technik am eigenen Körper hinsichtlich des Überwachungsaspektes als geringere Bedrohung empfunden. Steve Mann dreht mit seinem Wearable Computer-System dieses Machtverhältnis sogar um: er benutzt seinen persönlichen Wearable Computer als Überwachungsinstrument für seine Umgebung, um sich mit dieser Technik zu schützen und zu wehren (siehe unter der Bezeichnung »Sousveillance« Mann 2006).

So wie Ubiquitous Computing eine konzeptionelle Idealform hat – die Umgebung ist computerisiert, der Mensch bleibt völlig unbelastet – so beinhaltet auch Wearable Computing eine Vision: Der Mensch ist der alleinige Handelnde, er ist Träger und Steuerer der Technik. In dieser Absolutheit können jedoch beide nicht realisiert werden. Pragmatische und damit einsetzbare technische Lösungen liegen daher zwischen beiden Extremen, d.h. neue informations- und kommunikationstechnologische Komponenten und Computertechnik werden in die stationäre Umgebung sowie in Gegenstände integriert, und auch der mobil tätige Mensch wird mit technischen Komponenten ausgestattet. Ich bezeichne Lösungen mit einem solchen hybriden, meiner Ansicht nach pragmatisch motivierten Charakter ebenfalls als mobile Lösung (siehe Abb.14).

Technische, systemische und grundlegende Problemfelder

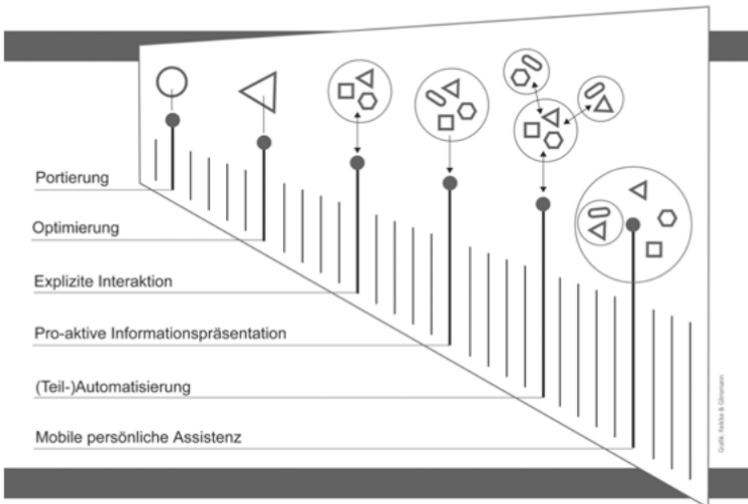
Die Untersuchung vorhandener mobiler Komponenten und Gesamtlösungen und ihre Gegenüberstellung mit dem intendierten Nutzungskontext hat eine Systematik der Lösungstypen gezeigt, die von der einfachen Portierung herkömmlichen Software auf ein mobiles Endgerät hin zur Realisierung einer automatisierten persönlichen Assistenz reicht, in der alle technologischen Möglichkeiten zur Vereinfachung der Interaktion für die BenutzerIn eingesetzt werden (siehe Rügge 2007, 202 f.).

Die Mehrzahl der realisierten mobilen Lösungen ist bisher am Mensch-Computer-Interface gescheitert. Die Gründe dafür sind vielfältig, sie gehen auf einer abstrakten Ebene konform mit den Kritikpunkten, die auch seitens der Usability-Forschung bereits für Desktop Computing-Lösungen festgestellt wurden, haben im Detail jedoch noch andere Probleme. Zu nennen sind hier u.a.:

- Die verwendete und bisher verfügbare Hardware hat keine akzeptablen Trageeigenschaften.

- Hardware und Software sind für den mobilen Einsatz nicht aufeinander abgestimmt und werden bei der Gestaltung der Interaktion nicht als ein aufeinander bezogenes Gesamtsystem aufgefasst.
- Die Gestaltung der Benutzungsoberfläche orientiert sich bisher an den Gestaltungskriterien für Desktop Computing-Lösungen und berücksichtigt nicht die Besonderheiten der mobilen Benutzung.
- Die Sensorik, die Kombination von Sensoren und ihre Interpretation zur Unterstützung der Interaktion sind noch nicht ausgereift.
- Die Gebrauchstauglichkeit mobiler Lösungen wird nicht hinreichend genug untersucht.

Abb. 15: Evolution der Entwicklung mobiler Lösungen (Bild: MRC)



Ein grundlegender Fehler, der bei der Entwicklung von Technik immer wieder gemacht wird, ist die technikzentrierte Perspektive auf die Lösung. D.h. es werden Technologien und Komponenten entwickelt, die aus Sicht von IngenieurInnen und InformatikerInnen wichtig für die Realisierung einer mobilen Lösung sind, ohne dass zuvor die Bedarfe der intendierten BenutzerInnen intensiv ermittelt wurden und ohne dass die potenziellen BenutzerInnen bei der Entwicklung beteiligt wurden. Forschungsgruppen, die 3D-Visualisierung und Virtual Reality als Forschungsthema haben, entwickeln so z.B. mobile Lösungen indem sie das, was sie für stationäre Systeme geschaffen haben, auf mobile Endgeräte übertragen. Dass der Impuls für eine mobile Lösung technologisch motiviert ist, stellt im Prinzip kein Problem dar. Die vorhandene technologische Kompetenz ist sogar unbedingt für die Entwicklung technischer

Komponenten und Gesamtsysteme erforderlich. Allerdings ist für die Entwicklung einer einsetzbaren mobilen Lösung die Kenntnis der Bedingungen im Anwendungsbereich ausschlaggebend.

Eine Technologie, die einer Person derart »zu Leibe« rückt, muss soweit maßgeschneidert sein, dass sie der TrägerIn und ihrer aktuellen Aufgabe optimal angepasst ist – was im Fall von Werkzeug oder Kleidung selbstverständlich ist. Niemand würde auf die Idee kommen, einer GoldschmiedIn den Hammer einer KFZ-MechanikerIn in die Hand zu geben oder einem Feuerwehrmann die Uniform einer Stewardess anzuziehen – das wäre weder sinnvoll noch angemessen. Doch schmälert eine Konfektionierung technischer Geräte – aus betriebswirtschaftlicher Sicht – vermeintlich den Gewinn.

Abb. 16: Begegnung zweier Wissenschaftler, ausgestattet mit Wearable Computing-Prototypen auf der Cebit 2002 (Bild: TZI)



Ebenso wenig Erfolg beschieden war dem Versuch, Software von großen Rechnern auf kleine mobile Endgeräte zu portieren. Denn, wie leicht nachvollziehbar, fühlt sich z.B. eine NotärztIn bei ihrer eigentlichen Arbeit – der Rettung von Menschenleben – behindert, wenn sie die ge-

setzlich vorgeschriebene Dokumentation auf einem winzigen Bildschirm machen soll, auf dem sie ständig scrollen muss und den sie zudem bei einer Eskalation der Situation nicht einfach fallenlassen darf. Auch die Verwendung eines größeren, robusten mobilen Endgeräts stellt nicht automatisch eine praktikable Lösung dar, da z.B. ein TabletPC, der wie ein elektronisches Klemmbrett benutzt werden kann, mit immerhin rund 1,5 kg Gewicht ebenfalls hinderlich ist.

Besonders auffällig ist, dass bisher so gut wie keine systematischen Benutzungstests mit einer aussagekräftigen Anzahl und Auswahl von Probanden sowie kaum (software)ergonomische Evaluationsergebnisse von mobilen Lösungen veröffentlicht worden sind. Es lassen sich nur einzelne Hinweise finden, z.B. ein Bericht über die Entwicklung des kommerziell erfolgreichen Wearable Scanning System von Symbol Technologies, die Begründung für das Scheitern des Augmented-Reality-Systems bei Boeing (Mizell 2001) und die Dokumentation der Erfahrungen mit der BenutzerInnen-Beteiligung bei der Entwicklung der Hardware für den VuMan. Zu einzelnen Komponenten oder Prototypen gibt es Nutzungsversuche mit kleinen BenutzerInnengruppen oder eine technische Evaluation des informations- und kommunikationstechnologischen Systems¹⁰. Doch sind die Ergebnisse für die Bewertung der Gebrauchstauglichkeit kompletter Lösungen nicht aussagekräftig genug, da komplexe oder reale mobile Arbeitssituationen dabei nicht berücksichtigt wurden. Für die Mehrheit der bisher entwickelten Systeme stehen systematische und repräsentative Usability-Tests und aussagekräftige Feldversuche noch aus.

Lösungsansätze für die Entwicklung tragbarer mobiler Lösungen

Die Gegenüberstellung von bisher entwickelten mobilen Lösungen bzw. ihren technologischen Komponenten mit den Anforderungen aus den Anwendungsbereichen zeigte Defizite auf Seiten der Technik und im Vorgehen beim Design mobiler Lösungen (Rügge 2007: 251 ff.). Folgende Desiderate wurden identifiziert, die für die Qualität einer mobilen Lösung entscheidend sind:

10 Siehe z.B. die jährliche Veröffentlichung der Beiträge des International Symposium on Wearable Computers (ISWC) der IEEE (iswc.net/) oder die Dokumentation der internationalen Tagung mobileHCI und des International Forum on Applied Wearable Computing (IFAWC) (www.ifawc.org/).

- Eindeutiger Verbesserungsbedarf bei den einzusetzenden mobilen Komponenten auf der Ebene von Hard- und Software, vor allem mit Blick auf ihre Wechselwirkung bzw. ihr Zusammenwirken.
- Enge interdisziplinäre Zusammenarbeit in Forschung und Entwicklung.
- Mehrdimensionaler Integrationsbedarf auf technischer, methodischer und arbeitsorganisatorischer Ebene.
- Partizipativer, NutzerInnen-zentrierter Technikgestaltungsprozess.
- Die Methodik der Anforderungsermittlung muss den Charakteristika mobiler Tätigkeiten angepasst werden.
- Die mobile Tätigkeit muss einer anwendungsorientierten Analyse unterzogen werden, die technische Umsetzungsmöglichkeiten mit berücksichtigt.
- Die Entwicklung und insbesondere auch die Untersuchung der Ergonomie mobiler Lösungen müssen auf mehreren Ebenen erfolgen.

Als zukünftiges Konzept für die Entwicklung der erforderlichen mobilen Technologien für tragbare mobile Lösungen wäre ein Komponentenmodell wünschenswert, wie es z.B. aus der Herstellung variantenreicher Produkte bekannt ist. Dafür ist ein hoher Grad an Normierung erforderlich und die Bereitstellung von kompatiblen Schnittstellen, wie sie in informations- und kommunikationstechnologischen Geschäftsbereichen zwar immer wieder gewünscht, aber bisher nicht annähernd realisiert wurden. Die vielen feinen Unterschiede bei den Anforderungen in den Anwendungsbereichen und bei den Bedürfnissen der einzelnen NutzerInnen weisen jedoch deutlich die Notwendigkeit einer Konfektionierung besonders von Wearable Computing-Lösungen hin.

Die Entwicklung und Umsetzung mobiler Lösungen erfordert mehr als eine gute Ingenieursleistung – hier müssen verschiedene Fachdisziplinen eng zusammenarbeiten: Für die Entwicklung mobiler Sensoren ist Mikroelektronik erforderlich, bei der Auswertung ihrer Messwerte kommen Berechnungs- oder sogar Lernverfahren aus Mathematik und Informatik zum Einsatz. Bei der Auswahl, Kombination und Platzierung der Sensoren sind Biologie, Medizin und nicht zuletzt Bekleidungsdesign gefragt. Für die Herstellung tragbarer leitfähiger und isolierender Materialien, ihre Kombination zu Stoffen und für eine belastbare Verbindung zwischen Sensoren und Auswertungssystem sind TextilentwicklerInnen die geeigneten Partner. Und um die steifen Elektronikkomponenten mit den flexiblen Stoffen zu verbinden, ist die Zusammenarbeit von Textilforschung und Elektrotechnik erforderlich. Nicht zuletzt erfordert die Notwendigkeit einer ausreichenden Energieversorgung der mobilen Systeme weitere, spezielle Kompetenzen und dies nicht nur auf der Ebene der

Hardware – auch die Software muss ressourcenschonend ausgelegt sein. Neben dem Zusammenwirken technischer Disziplinen ist aber insbesondere eine Verzahnung arbeitswissenschaftlicher, organisations- und individualpsychologischer und medizinischer Forschungsgebiete mit den technischen Realisierungen zu schaffen.

Unter ExpertInnen besteht Einigkeit darüber, dass die Akzeptanz einer mobilen Lösung bestimmt wird von ihrem Mehrwert für die AnwenderIn und vom subjektiven Wohlbefinden der BenutzerIn während der Verwendung. Dafür sind detaillierte Wirtschaftlichkeitsberechnungen erforderlich sowie eine passgenaue Integration in den Arbeitsprozess und in die umgebende informations- und kommunikationstechnische Infrastruktur. Doch einen herausragenden Stellenwert hinsichtlich Akzeptanz und Praxistauglichkeit von mobilen Lösungen nimmt unbestritten die maßgeschneiderte ergonomische Anpassung an den Bewegungsablauf und die Aufmerksamkeitsverteilung der BenutzerInnen ein.

Deshalb ist der Schluss naheliegend, dass für die Entwicklung wirtschaftlicher und nützlicher mobiler Lösungen eine frühe und sehr enge Zusammenarbeit aller Disziplinen mit den BenutzerInnen des anvisierten Einsatzbereichs angeraten ist. Ein Disziplin- und Interessenlagen-übergreifendes Mittel, ein gemeinsames Verständnis von den Anforderungen und den Umsetzungsmöglichkeiten herzustellen, ist die Nutzung von »Technologie zum Anfassen und Ausprobieren«. Im doppelten Sinne des Wortes »begreifbare« Prototypen konkreter mobiler und Wearable Computing-Lösungen sind hervorragend dazu geeignet, als Kommunikationsmittel in einer dialogischen, gleichberechtigten Form des Miteinanders zu dienen und so maßgeschneiderte Lösungen für mobile Tätigkeiten zu erarbeiten, die auch von den TrägerInnen akzeptiert werden. Das heißt für die zukünftige Entwicklung mobiler Lösungen, dass die Methoden der Anforderungsermittlung und der Evaluation einer Anpassung an die Bedingungen der Mobilität mobiler Aktivitäten und Tätigkeiten bedürfen, die im ersten Moment kostspielig und anstrengend erscheinen mag, doch essenziell für eine Verbreitung mobiler Lösungen im professionellen Einsatz jenseits der Schreibtischarbeit ist.

Literatur

- Amft, O./Lauffer, M./Ossevoort, S./Macaluso, F./Lukowicz, P./Tröster, G. (2004): »Design of the QBIC Wearable Computing Platform«. In: 15th IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP '04), 2004, S. 398-410.
- Anoto (2006), URL der Homepage der Firma: <http://www.anoto.com/> (zuletzt gesehen am 14.10.2006).
- Barfield, W./Caudell, T. (Eds.) (2001): »Basic Concepts in Wearable Computers and Augmented Reality«. In: Barfield, W./Caudell, T. (Eds.): »Fundamentals of Wearable Computers and Augmented Reality«. Mahwah, N J, London, 2001, S. 3-26.
- Bass, L./Siewiorek, D./Bauer, M. et al. (2001): »Constructing Wearable Computers for Maintenance Applications«. In: Barfield, W./Caudell, Th. (Eds.): »Fundamentals of Wearable Computers and Augmented Reality«. Mahwah, N J, London, S. 663-713.
- ETH Zürich (2004): »QBIC - Q-Belt Integrated Computer«. URL: <http://www.wearable.ethz.ch/research/groups/platforms/QBIC/index> (zuletzt gesehen 1.3.2008).
- Fackler, I. (2006): »Tragbare Elektronik macht uns immer mobiler«. In: Probleme und Lösungen 5/2006. URL: <http://www.produktion.de/article/0f485fcad88.html> (Zuletzt gesehen am 20.10.2006)
- Günzel, C. (2005): »Tragbare Computer – Der vernetzte Frack«. In: PC Magazin 1/2005, S. 16-18
- Hasselbring, W. (2006): »Software-Architektur«. In: Informatik Spektrum 29,1, S. 48-52.
- Herzog, O./Rügge, I./Boronowsky, M./Nicolai, T. (2003): »Potenziale des Wearable Computing in der Industrie – am Beispiel der Inspektion«. In: Gausemeier, J./Grafe, M. (Hrsg.): »Augmented & Virtual Reality in der Produktentstehung«. 2. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, 4. und 5. Juni 2003, Heinz Nixdorf MuseumsForum, S. 21-39.
- Jung, S./Lauterbach, C. (2002): »Elektronik zum Anziehen«. In: Elektronik 12/2002, S. 40-43.
- Katragedda, R.B./Xu, Y. (2005): »A Novel Intelligent Textile Technology Based on Silicon Flexible Skins«. In: Proceedings of 9th International IEEE Symposium for Wearable Computers (ISWC'05), S. 78-81.
- Kenn, H./Rügge, I. (2006): »Botschaften aus dem Handgelenk«. In: Sonderausgabe RFID in Bremen von »RFID im Blick«, Juli 2006, S. 36-38.
- Laurel, B. (1990): »What's an Interface?« In: Laurel, B. (Hrsg.): »The Art of Computer Interface Design«. Reading, Mass. 1990.

- Lawo, M. (2006): »Ein drahtloser Eingabehandschuh für Augmented Reality Anwendungen«. In: 5. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, 31.05.-01.06., Paderborn, hrsg. Gausemeier, J./Grafe, M., S. 199-208.
- Linz, T./Kallmeyer, C./Aschenbrenner, R./Reichl, H. (2005): »Embroidering Electrical Interconnects with Conductive Yarn for The Integration of Flexible Electronic Moduls into Fabric«. In: Proc. of 9th ISWC '05, S. 86-89.
- Mann, S. (2006): Homepage von Steve Mann: URL: <http://wearcam.org/> (zuletzt gesehen am 21.09.06).
- Marculescu, D./Marculescu, R./Park, S./Jayaraman, S. (2003): »Ready to Ware«. In: IEEE Spectrum, 40, 10, S. 28-32.
- Mecheels, S./Schroth, B./Breckenfeldern, C. (2004): »Smart Clothes – Intelligente textile Produkte auf der Basis innovativer Mikrotechnologie. Expertensicht – Beispiele – Empfehlungen«. Hohensteiner Institute.
- Miller, M. (2002): »Fossil Wrist PDA-PC FX2002«. In: PDA REVIEW (erstellt am 2002) URL: <http://geek.com/hwswrev/pda/fossil/> (zuletzt gesehen am 18.10.2006).
- MIT MediaLab (2003): Homepage des MITHril-Projekts (zuletzt aktualisiert 2003) URL: <http://www.media.mit.edu/wearables/mithril/> (zuletzt gesehen 21.09.06).
- Mizell, D. (2001): »Boeing's Wire Bundle Assembly Project«. In: Barfield, W./Caudell, T. (Eds.): »Fundamentals of Wearable Computers and Augmented Reality«. Mahwah, NJ, London, S. 447-467.
- Nake, F. (1992): »Informatik und die Maschinisierung der Kopfarbeit«. In: Coy, W. et al (Hg.): »Sichtweisen der Informatik«. Braunschweig, Wiesbaden, S. 181-207.
- Norman, D.A. (1998): »The Invisible Computer«. Cambridge, Mass.
- Pahl, C. (2004): »Kleidsame Elektronik«. In: Technology Review, Das M.I.T.-Magazin für Innovation, 12, Dez., S. 95-97.
- Pichler, M. (2005): »Identifying Mobile People's Information Needs«. SCCH-Technical Report S. 511.
- Rhodes, B.J. (1997): »The Wearable Remembrance Agent: A System for Augmented Memory«. In: Proc. of 1st ISWC, Cambridge, Mass, 13/14. Okt. S. 123-128. Siehe auch URL: <http://alumni.media.mit.edu/~rhodes/Papers/wear-ra-personaltech/index.html> (zuletzt gesehen 25.10.2006).
- Robben, B./ Rügge, I. (1998): »Mit den Händen beGreifen: Real Reality«. In: I. Rügge et al. (Hrsg.): »Arbeiten und begreifen: Neue Mensch-Maschine-Schnittstellen«. Münster, S. 133-146.
- Rügge, I. (2002): Studie »Technologische und anwendungsorientierte Potenziale mobiler, tragbarer Computersysteme«. Universität Bremen, TZI-Bericht Nr. 24/2002.

- Rügge, I. (2007): »Mobile Solutions. Einsatzpotentiale, Nutzungsprobleme und Lösungsansätze«. (Advanced Studies Mobile Research Center Bremen), Wiesbaden.
- Schelhowe, H. (1997): »Das Medium aus der Maschine. Zur Metamorphose des Computers«. Frankfurt a. M., New York.
- Schmidt, A. (2000): »Implicit Human Computer Interaction Through Context«. In: Personal Technologies 4, 2, S. 191-199.
- Smailagic, A./Siewiorek, D. (1999): »User-Centered Interdisciplinary Concurrent System Design«. In: ACM Mobile and Communications Review 3, 3, S. 43-52.
- Stein, R./Ferrero, S./Hetfield, M./Quinn, A./Krichever, M. (1998): »Development of a Commercially Successful Wearable Data Collection System«. In: ISWC'98, 2. International Symposium on Wearable Computers, Pittsburgh, PA. October 19-20, S. 18-24.
- Symbol Technologies: URL der Homepage der Firma: <http://www.symbol.com/>, URL des Wearable Scanning System WSS 1000/1060 <http://www.symbol.com/WS1000/1060>, URL des Wearable Scanning System WT 4000: <http://www.symbol.com/wt4000> (zuletzt gesehen 27.10.2006).
- wearitwork (2004): Homepage des europäischen Großprojekts (IP) wearIT@work (erstellt 2004). URL: www.wearitwork.com (zuletzt gesehen: 21.09.06).
- Weiser, M. (1993): »Some Computer Sciences Problems in Ubiquitous Computing«. In: Communications of the ACM, 36, 7, S. 75-85.
- Wichmann, T./Stiehler, A. (2004): Basisreport »Prozesse optimieren mit Mobile Solutions«. Berlecon Research, Berlin, März.

DIE RÜCKKEHR DES SENSORISCHEN: TANGIBLE INTERFACES UND TANGIBLE INTERACTION

EVA HORNECKER

»We live in a complex world, filled with myriad objects, tools, toys, and people. Our lives are spent in diverse interaction with this environment. Yet, for the most part, our computing takes place sitting in front of, and staring at, a single glowing screen attached to an array of buttons and a mouse.«

(Wellner/Mackay/Gold 1993, S. 24)

Der Bildschirm war immer nur ein Fenster, durch das hindurch wir versuchen, in die virtuelle Welt hinein zu greifen, und den wir dabei nicht als ein materielles Ding wahrnehmen. Desktop-Systeme ebenso wie virtuelle Welten trennen alltägliche Lebenswelt und digitale Welt voneinander. Ungeachtet dessen, ob wir Texte schreiben, malen, dreidimensionale Gegenstände entwerfen, Statistiken berechnen oder einen Hochofen warten, tun wir dies mit immer den gleichen Bewegungen – Tippen auf der Tastatur, Bewegen der Maus und angestrengtem Starren auf den Bildschirm. Verloren gehen dabei die Vielfältigkeit unserer Alltagswelt, der Reichtum an Umgangsmustern mit der realen Welt sowie die körperliche, alle Sinne einbeziehende Interaktion mit Gegenständen.

In den letzten Jahrzehnten entstand ein gegenläufiger Trend zur Virtualisierung, der sich in vielfältiger Form äußert. Hierunter fallen Tangible Interfaces sowie in die Umwelt eingebettete Schnittstellen (pervasive, ubiquitous computing). Diese sollen gewohnte Umgangsformen mit Artefakten sowie deren Situiertheit in der Alltagswelt erhalten und zugleich unser Leben mit neuen Funktionen bereichern. Die reale, materielle Welt wird nicht ersetzt, sondern bereichert und erweitert. Sie bleibt Ausgangspunkt und integraler Bestandteil, erweitert um digitale Eigenschaften.

Tangible (User) Interfaces (kurz: TUIs) sowie *Tangible Interaction* als Gestaltungsansatz für Schnittstellen wählen Greifbarkeit, ertastbarkeit (Engl.: tangibility) und körperliche Interaktion als grundlegendes Gestal-

tungsprinzip und verleihen digitalen Ressourcen materielle Form. Mit Tangible Interfaces agieren wir in unserer materiellen Umgebung und berühren das Interface selber. Die Faszination, die solche greifbaren Schnittstellen ausüben – erkennbar an ihrer wiederholten »Erfindung«, ihrer Medienwirksamkeit sowie dem regen Interesse bei Endanwendern – zeigt, dass sie einen willkommenen Gegenpol zur Virtualisierung aller Lebensbereiche bieten und ein menschliches Bedürfnis nach greifbarem Kontakt und körperlichem Umgang mit der Umwelt erfüllen.

Solche Systeme zu gestalten ist jedoch alles andere als einfach und stellt noch eine große Herausforderung dar. Computer und Software verschmelzen mit unserer Umwelt, verbinden sich mit allen möglichen Gebrauchsgegenständen. Die zunehmende Integration von Digitaltechnik in Haushaltsgeräte und andere Maschinen sowie der Trend zu immer kleineren, spezialisierten Computern machen es schon lange unmöglich, eine Grenzlinie zwischen mechanischen und digitalen Geräten zu ziehen. Alterprobtes Gestaltungswissen aus dem Bereich graphischer Schnittstellen und der Mensch-Computer-Interaktion lässt sich nicht ohne weiteres auf diesen neuen Bereich übertragen. Hardware und Software bilden eine Einheit und müssen entsprechend entworfen werden. Elektronik, Mechanik und Produktdesign sind folglich ebenso gefragt wie die Informatik – die Entwicklung erfolgreicher Systeme bedarf der Zusammenarbeit verschiedenster Disziplinen. Dabei sind Tangible User Interfaces nur ein Ansatz unter vielen Formen des »Beyond the Desktop« oder des »Pervasive und Ubiquitous Computings«, die den Computer in die Alltagswelt integrieren (Dourish 2001). In diesem Beitrag wird daher die umfassende Perspektive der *Tangible Interaction* eingenommen.

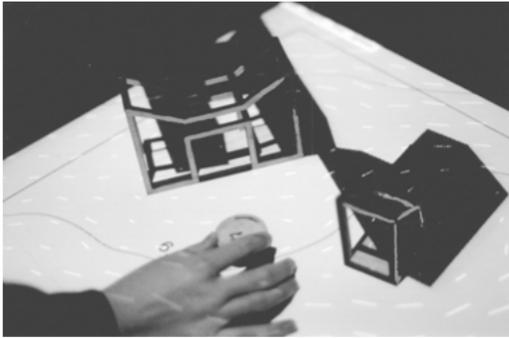
Im Folgenden werden zunächst einige einführende Beispiele für Systeme aus dem Bereich der Tangible Interaction vorgestellt, welche die Bandbreite des Gestaltungsspektrums verdeutlichen. Anschließend wird die noch junge Geschichte dieses Forschungsbereichs skizziert. Abschließend wird der Wechsel der Perspektive vom *Tangible Interface* zur *Tangible Interaction* diskutiert sowie eine Schlussbetrachtung gegeben.

Tangible Interaction: Einige einführende Beispiele

Kerngedanke der Beschäftigung mit Tangible User Interfaces ist, anfassbare, greifbare Gegenstände für die Interaktion mit digitalen Repräsentationen und Informationen zu verwenden. Dies ermöglicht eine direkte, händische Manipulation der Daten. Die Schnittstelle soll nicht wie bisher zweidimensional und rein visuell sein, sondern dreidimensional und

haptisch-taktil erfahrbar. Der vorliegende Beitrag befasst sich nicht allein mit Tangible User Interfaces, sondern auch mit dem umfassenderen Bereich der Tangible Interaction. Letzterer beruht auf einem erweiterten Verständnis von *tangibility*, fokussiert auf die Interaktion anstelle der Schnittstelle (zu einer bereits vorhandenen Maschine) und greift darin Gedanken aus dem Produktdesign sowie der interaktiven Kunst auf. Die folgenden Beispiele sind bewusst breit gewählt, um die verschiedenen theoretischen Perspektiven und den Stil der resultierenden Systeme zu veranschaulichen.

Abb. 1: Urp: Die Umrissmodelle zweier Gebäude werfen Schatten; weiße Linien visualisieren die Windströmung. Jemand prüft die Windstärke – die Werte sind auf und neben der Windsonde zu sehen.



URP - ein interaktiver Planungstisch für Architekten

Eines der bekanntesten Tangible-Interface-Systeme ist das von Hiroshi Ishii und seiner »Tangible Media Group« am MIT Media Lab entwickelte *Urp* (Underkoffler/Ishii 1999). Das *Urban planning*-System unterstützt Stadtplaner und Architekten durch eine Simulationssoftware, deren Berechnungsergebnisse in ein händisch manipulierbares Modell eingebunden werden. Die Planer positionieren Umrissmodelle von Gebäuden auf einem Tisch. Licht- und Schatteneffekte werden auf die Planungsfläche projiziert. Bewegt man die Modelle, bewegen sich auch ihre virtuellen Schatten. Zusätzliche Objekte dienen als Werkzeuge. So verwandelt die Berührung mit einem »Zauberstab« das Material einer Wand in Stein oder Glas. Etwaige Spiegelungen werden berechnet und in das Modell projiziert, ebenso wie Licht- und Schattenfall für die eingestellte Uhrzeit. So lässt sich beispielsweise prüfen, ob ein Innenhof genug Tageslicht erhält. Ferner wird die Windströmung durch kleine bewegte Striche dar-

gestellt, die anzeigen, ob sich zwischen und hinter Gebäuden Windwirbel oder Sog bilden. Die Windgeschwindigkeit wird in Zahlen neben eine »Wind-Sonde« eingeblendet. Eine »Windrose« gibt die Windrichtung vor. Auf diese Weise können die Konsequenzen von Designentscheidungen experimentell und spielerisch überprüft werden.

Dieses System ist vom Aufbau her recht typisch für viele TUIs. Dabei dient eine plane Fläche als Basis, auf der die Benutzer mit so genannten »Token« agieren (greifbaren Gegenständen, welche vom System registriert werden). Meist ist die Sensorik zur Erfassung der Token in die Arbeitsfläche integriert, oder Kameras filmen das Geschehen und speisen Bilderkennungssoftware. Urp ist ein gutes Beispiel für eine gelungene Aufteilung der Repräsentation in materielle und digitale Anteile – die Elemente, mit denen die Planer direkt interagieren wollen, sind durch greifbare Modellelemente und Werkzeuge verfügbar, während Simulationsergebnisse und veränderliche Werte projiziert werden.

Abb. 2: Zwei Personen spielen auf der Klaviatur, während ein Besucher hinter ihnen vorbeigeht und implizit in Interaktion tritt. (Hornecker/Bruns 2005)



Die Klaviatur - Körper agieren im Raum

Die *Klaviatur* wurde 2002 von Studierenden der Medieninformatik der Universität Bremen als Teil eines Ensembles von sieben Installationen erstellt (Hornecker/Bruns 2005). Ein ca. 800 qm großer Bereich im Bremer Wallanlagenpark verwandelte sich für drei Nächte in einen »sensorischen Garten«. Bei der »Klaviatur« handelte es sich um einen Pfad mit am Boden angebrachten, von den Besuchern unterbrochenen Lichtschran-

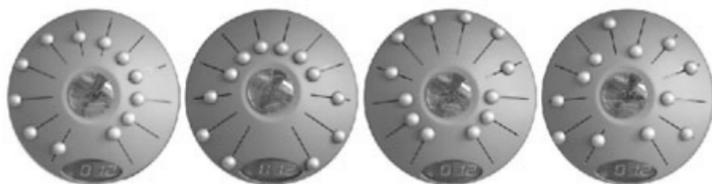
ken. Wer hier schlenderte, erzeugte ein Echo aus Farben, Licht und Klang. Bunte Scheinwerfer leuchteten, wo man den Fuß aufsetzte; verschiedene Drums und Beats erklangen. Viele Besucher tanzten, hüpfen von Licht zu Licht, und erzeugten allein oder gemeinsam Musik und Rhythmen. Vorbeigehende oder auf den Bänken ausruhende Besucher traten ungewollt in musikalische Interaktion mit den Akteuren. Eine Gruppe junger Leute tanzte nächtens lange ausgelassen auf der Klaviatur. Hier handelte es sich um eine ästhetisch ansprechende Installation, deren Interaktionsprinzip leicht zu verstehen war und die dabei soziale Interaktion herausforderte. Da eine einzelne Person nur auf einen Teil der Klaviatur wirken konnte, war man auf andere Personen angewiesen, um eine komplexere Soundlandschaft zu erzeugen.

Wie die Tänzer zeigen, enthielt die Interaktion Aspekte einer Performance – die Interaktion über das Bewegen des eigenen Körpers bietet die Option, dies nicht nur rein manipulativ-funktional zu tun, sondern den eigenen Körper bewusst einzusetzen. Während sich im vorangegangenen Beispiel die Benutzer über die Interaktionsfläche beugen und Token verschieben, wird hier der eigene Körper gleichsam zum Token. Greifbarkeit und Haptisch-Taktilis sind in den eigenen Körper verlagert, der im Raum der Schnittstelle navigiert, welche sich nun um ihn herum befindet. Die Klaviatur steht hier paradigmatisch für (zumeist künstlerische) Installationen, welche interaktive Räume schaffen, die auf Interaktion und Bewegungsmuster der Besucher reagieren.

Ein »ausdrucksvoller Wecker« - Interaktion gestalten

Der Prototyp eines »ausdrucksvollen« Weckers wurde im Rahmen einer Dissertation im Produktdesign an der TU Delft und TU Eindhoven konzipiert und entwickelt (Djajadiningrat/et al 2004). Ziel war es, auf ausdrucksstarke und emotionale Weise mit interaktiven, »intelligenten« Alltagsgegenständen zu interagieren, wobei die Art und Weise der Interaktion das Interaktionsresultat moduliert. Aussehen/Form und Interaktionshandlung sollten miteinander in Bezug stehen. Ein Wecker erschien als gutes Beispiel eines Produkts, das auf eine Vielzahl von Emotionen reagieren kann und eine relativ simple Funktionalität hat. Der Designer Wensveen entwarf einen halbkugelförmigen Wecker mit zwei Anzeigefeldern und zwölf Schieberegler. Das vordere Anzeigefeld zeigt die aktuelle Uhrzeit, das mittlere die eingestellte Weckzeit. Die Regler sind rund um das mittlere Anzeigefeld angebracht und befinden sich im Ausgangszustand am äußeren Rand. Schiebt man sie zur Mitte, wird fortlaufend die Weckzeit erhöht (in Differenz zur Uhrzeit). Jeder Regler entspricht dabei maximal 60 Minuten. Ist die erwünschte Weckzeit erreicht, drückt man das zentrale Anzeigefeld, und aktiviert so den Wecker.

Abb. 3: Der »ausdrucksstarke« Wecker: bei gleicher Uhrzeit (0:12) und Weckzeit (7:00) zeigen die Schieberegler völlig verschiedene Muster.



Was in der verbalen Beschreibung kompliziert wirken mag, wird in der eigenhändigen Nutzung rasch verstanden. Die Bedienung ist trotz der vielen Schieberegler intuitiv und schnell. Man kann mehrere Regler gleichzeitig bewegen, beispielsweise mit der ganzen Hand und zugleich mit einer leichten Bewegung eines Fingers die Minuten fein justieren, wobei das mittlere Anzeigefeld stets synchrones und eindeutiges Feed-back gibt.

Der Benutzer hat hier unzählige Möglichkeiten zur Eingabe der Weckzeit. Je nach Verlauf der Eingabe variiert das resultierende Muster der Regler. Diese »Freiheit des Verhaltens« ermöglicht es, Information zu vermitteln, sich dabei aber je nach Gemütszustand anders zu verhalten. Der Ausdruck von Emotion wird so zum Bestandteil der Interaktion mit dem Produkt. Experimente zeigten, dass Versuchspersonen je nach Stimmung eher regelmäßige oder zackige, asymmetrische Muster generierten. Die (nicht mehr realisierte) konzeptionelle Idee war, dass der Wecker je nach Art des Musters und der Interaktion (schnell oder langsam) verschieden klingelt.

Geschichte der Entwicklung

Als Trend oder eigene Forschungsrichtung innerhalb der Mensch-Computer Interaktion feststellbar sind Tangible Interfaces seit Mitte der 90er Jahre des letzten Jahrhunderts, markiert durch eine Reihe von Publikationen, welche TUIs oder »Graspable User Interfaces« als einen Gegenentwurf zur Virtual Reality positionierten. Jedoch gibt es etliche weniger bekannte Vorläufer und Vordenker, deren Ideen bis heute nachwirken. Diese verstanden ihre Arbeiten allerdings meist nicht als einen neuen (generellen) Ansatz der Schnittstellengestaltung, sondern gingen von konkreten Problemen ihres Anwendungsbereichs aus. Dies mag einer der Gründe sein, warum sie innerhalb der Mensch-Computer Interaktion lange Zeit relativ unbekannt blieben, obwohl ihre Arbeiten oft Inspiration für die

TUI-Entwicklung boten. Einige dieser Vordenker werden im Anschluss an den folgenden historischen Abriss vorgestellt.

Die Entstehung der Idee eines Tangible Interface ist eng mit den Strömungen der *Augmented Reality* und des *Ubiquitous Computing* verknüpft. Gemeinsame Ausgangsmotivation dieser neuen Konzepte der Mensch-Computer Interaktion war die wachsende Unzufriedenheit mit gängigen Computerschnittstellen. 1993 erschien ein Themenheft der »Communications of the ACM« mit dem programmatischen Titel »Back to the real world« (Wellner/Mackay/Gold 1993); ein Titel, der die vorgestellten Ansätze als eine Alternative sowohl zu graphischen Benutzungsschnittstellen als auch zur damals aktuellen Virtual Reality akzentuiert. Nach Ansicht der Autoren des Themenhefts trügen diese nämlich dazu bei, den Menschen von seiner realen Umwelt zu trennen, indem sie aufwendige Übersetzungs- und Transferarbeiten verlangten (z.B. zwischen Papierakten und Computer) oder die »reale Welt« durch graphische Parallelwelten ersetzten. Stattdessen komme es darauf an, beide Welten zu integrieren und fließende Übergänge zu schaffen, indem man gewohnte Umgangsweisen verwendet und die Menschen in ihrer alltäglichen Lebenswelt belässt. Die reale Welt soll nicht ersetzt oder dupliziert werden, sondern bereichert und erweitert. Sie bleibt Ausgangspunkt und integraler Bestandteil. Der gewohnte Umgang mit den Dingen ist die Basis des Umgangs mit den neuen, erweiterten Objekten. Diese werden Teil der Alltagswelt und fügen sich in alltägliche Praktiken ein.

Diese Ideen lagen offenbar »in der Luft«, denn unabhängig von der Entwicklung in den USA wurde auch in Europa mit dem *Real Reality*-Konzept (Bruns 1993, 1996) ein Ansatz entwickelt, der die Interaktion mit der realen Welt wieder ins Zentrum stellen will und die sinnliche Erfahrung betont. Während sich die MMK bislang eher an den Kognitionswissenschaften und dem Bild des Menschen als »Information Processor« orientierte, werden nun Einflüsse aus dem Forschungsbereich ethnographischer Work Studies, wie etwa die *Situated Cognition*¹, sowie der Lebenswelt-Philosophie und Phänomenologie wirksam: »[...] humans are of and in the everyday world.« (Weiser 1993). Dourish's Analyse der neuen Strömungen der MMK macht dies explizit, indem sie als deren Hauptmerkmal *Embodied Interaction*, die Situiertheit in materiellen und sozialen Kontexten, beschreibt (2001).

Die Grundgedanken für Tangible User Interfaces wurden im Rahmen des obigen Themenheftes eingeführt, traten aber noch nicht als eigenständiges Konzept in Erscheinung. 1995 beschrieben Fitzmaurice/Ishii/Buxton das Konzept eines *Graspable User Interface*. Holzblöcke als »greifbare Griffe« (Engl: handles), welche auf den Monitor gelegt werden, machen

1 Lucy Suchmans Buch »Plans and Situated Action« erschien 1987.

graphische Repräsentationen manipulierbar, die mit den Griffen verankert sind. Ein wesentlicher Bestandteil des Konzepts sind ferner funktionsbezogene Ein/Ausgabegeräte. Greifbare Griffe und spezifische Ein/Ausgabegeräte verteilen die Eingabe räumlich, desequentialisieren sie, verringern die Anzahl der Vermittlungsschritte zwischen Zielgegenstand, Werkzeug und Manipulationsmedium und erlauben beidhändige Manipulation.² Wenig später stellten Ishii/Ullmer (1997) eine umfassendere Vision als *Tangible Bits* vor. Bits sollen in der realen, materiellen Welt direkt zugänglich werden, indem letztere sowohl zum Display (Anzeigemedium) als auch zum Manipulationsmedium werden – die ganze Welt wird zum Interface. Daten werden mit Objekten und architektonischen Oberflächen verknüpft, Bits werden greifbar und »Ambient Displays« stellen Information durch Geräusche, Licht, Wind oder Wasserbewegung dar. Zu beachten ist der begriffliche Wechsel von *graspable* zu *tangible*. Während ersteres die manuelle Manipulation betont, umfasst das Bedeutungsspektrum von *tangible* die Berührbarkeit und das Tasten; es betont die multi-sensorische Interaktion.³

»GUIs fall short of embracing the richness of human senses and skills people have developed through a lifetime of interaction with the physical world. Our attempt is to change ›painted bits‹ into ›tangible bits‹ by taking advantage of multiple senses and the multimodality of human interactions with the real world. We believe the use of graspable objects and ambient media will lead us to a much richer multi-sensory experience of digital information.« (Ishii/Ullmer 1997).

Einer der ersten Prototypen war der *Tangible Geospace* – eine interaktive Karte des MIT Campus auf einem Projektionstisch. Legt man »physical icons« (wie die Token damals genannt wurden) auf den Tisch, z.B. das Plexiglasmodell des Campus Kuppeldoms, positioniert sich die Karte so, dass das Modell über dem entsprechenden Gebäude auf der Karte liegt. Legt man ein weiteres Gebäude hin, wird die Karte so gedreht und gezoomt, dass beide Modelle richtig liegen. Ein mit einem Schwenkarm

2 Rauterbergs Build-It System, interaktiver Planungstisch zur Interaktion mit CAD Modellen (Fjeld/Bichsel/Rauterberg 1997), verfolgt diesen Ansatz weiter. Zwei bis drei greifbare Blöcke dienen zur Selektion und Manipulation graphischer Elemente auf dem Planungstisch. Sie werden nur temporär an graphische Objekte gebunden und dienen ferner zur Selektion und Manipulation von Menüs (anders als in Fitzmaurice Ansatz spezifischer Werkzeuge). Die Blöcke sind somit ein universelles Eingabemedium.

3 Lat. ›tangere‹ erinnert an die ambivalente Funktion des Tastsinns – man wird immer auch selbst durch das Ding berührt, das man berührt. Im Tasten sind Wahrnehmen und Agieren eins (Mattenklott 1997). Weiterhin bedeutet »tangible« auch ungefähr »real, gewiß« (tangible evidence).

am Tisch befestigter kleiner Bildschirm dient als eine Art Lupe: schwenkt man ihn über die Karte, sieht man auf dem Monitor eine 3D-Darstellung des darunter liegenden Kartenausschnitts. Das Interaktionsprinzip beidhändiger, direkter Manipulation ist geblieben, aber an die Stelle der abstrakten und generischen Blöcke des »Graspable User Interface« treten ikonische und symbolische Stellvertreter. Erkennbar ist, wie stark diese ersten Prototypen noch GUI-Metaphern verhaftet sind – das graphische Interface wurde lediglich in eine greifbare, dreidimensionale Form transformiert. Spätere Projekte, wie das eingangs vorgestellte Urp, versuchen sich hiervon zu lösen und konzentrieren sich auf den Umgang mit greifbaren Token, die weiterhin auf Flächen positioniert und bewegt werden. Greifbare Objekte sollen gleichzeitig zur Manipulation und Repräsentation von mit ihnen verknüpften Daten dienen (Ullmer/Ishii 2000). Die Repräsentationsfunktion impliziert, dass Token langfristig mit Daten verknüpft sind und daher in Form und Aussehen auf deren Bedeutung verweisen können.

Nachdem Tangible Interfaces programmatisch als Alternative zu graphischen Schnittstellen proklamiert wurden, konzentrierte sich die Forschung mehrere Jahre lang vornehmlich auf die Entwicklung neuer Systeme (zumeist eher »proof-of-concept« als praktisch nutzbar) sowie die Erkundung der technischen Möglichkeiten neuer Hardware. Parallel hierzu entwickelten jedoch auch einige Forschungsgruppen *Mixed Reality* Ansätze, die eher empirisch und von praktischen Problemen des Anwendungsbereichs ausgingen. Feldstudien wiesen immer wieder darauf hin, dass die Computertechnik in vielen Anwendungsbereichen die althergebrachten Medien nicht direkt ersetzt, sie sehr wohl aber ergänzen und erweitern kann. Die Sensorisierung und digitale Erweiterung traditioneller Medien und Arbeitsobjekte resultierte vielfach in Systemen, die sich als *Tangible Interfaces* klassifizieren lassen. Hervorzuheben sind hier insbesondere die Arbeiten von Wendy Mackay, beispielsweise über die »flight strips« der Fluglotsen (Mackay/Fayard 1999).

In den letzten Jahren werden Tangible Interface-Systeme vermehrt in Anwendungsdisziplinen entwickelt. Diverse Werkzeugkästen zum Prototyping setzen die Einstiegsschwelle zur praktischen Systementwicklung herab. Dinge, die bislang nur Ingenieure oder Elektroniktüftler realisieren konnten, sind nun auch für Designer und Künstler (oder Studierende) ohne lange technische Ausbildung machbar.⁴ Dies verkürzt Prototyping-

4 Phidgets sind ein beliebtes Prototypingwerkzeug (<http://group.lab.cpsc.ucalgary.ca/phidgets/>). Wiring, basierend auf der Programmiersprache Processing, ist eine Do It Yourself Sprache sowie ein elektronisches I/O Board für Designer und Künstler (<http://wiring.org.co/>). Arduino ist eine Open-Source Plattform aus einem einfachen I/O Board und einer Entwicklungsumgebung für Processing sowie Wiring (<http://www.arduino.cc/>).

zyklen und erlaubt, sich auf die konzeptionelle Gestaltung zu konzentrieren, bereits frühzeitig Entwürfe zu realisieren und diese von Benutzern testen zu lassen. Das ›Proof-of-concept‹-Stadium des letzten Jahrzehnts scheint abgeschlossen. Jetzt wird einerseits vermehrt kritisch gefragt, ob dieser Gestaltungsansatz zielführend ist und praktischen Nutzen zeigt (z. B. Marshall 2007), andererseits werden zunehmend Arbeiten publiziert, die zu einem tieferen konzeptionellen Verständnis sowie zum Gestaltungswissen beitragen (siehe z.B. Dourish 2001, Hornecker/Buur 2006, Hurtienne/Israel 2007).

»Tangible Interface« wurde zum festen Leitbegriff auf Tagungen, doch es zeigte sich bald, dass dieser Begriff eine eingeschränkte Sichtweise widerspiegelt. Diese ergab sich aus der Abgrenzung gegen graphische Schnittstellen und Virtual Reality. Parallele Entwicklungen in anderen Design- und Forschungsbereichen erweitern die Perspektive. Daher sollten wir diese Entwicklung als Ganzes betrachten. Arbeiten aus dem Industrie- und Produktdesign hatten schon früh maßgebliche Inspiration für die Suche nach neuen Interaktionsformen geliefert. Wenn »die ganze Welt zur Schnittstelle wird« (Ishii/Ullmer 1997; Übersetzung der Autorin), werden umfassende Designfragen virulent, mit denen sich neben der Informatik und HCI-Forschung auch Architektur und Produktdesign befassen. Interaktionsdesign (oder IT Produktdesign) als neue transdisziplinäre Fachrichtung befasst sich auch mit der Gestaltung ›herkömmlicher‹ Geräte wie Wecker, Radios und Küchengeräte – oder auch völlig neuartiger mechatronischer Geräte, welche Digitaltechnik und Sensorik zum integralen Bestandteil haben.

Informatik und klassische Designdisziplinen gelangen aus verschiedenen Richtungen auf gemeinsames Gebiet. Dabei stehen sie in gleicher Weise vor Herausforderungen, wenn es gilt, sowohl materielle, dreidimensionale Gegenstände, als auch deren Graphik, ihr Verhalten und zugrunde liegende Algorithmen zu gestalten:

»While for people with an HCI background, the physical aspect is often new ground, the physical has of course always formed an essential part of product design. [...] In the world of software, augmented reality is often presented as a way of introducing real, tangible objects in an otherwise virtual world. [...] Clearly, unlike software, electronic consumer products are tangible to start with. From a product design perspective [...] what is new is not so much the tangibility of the interaction as the richness of the interaction.« (Djajadinigrat/Overbeeke/Wensveen 2000)

Neu ist, dass nicht nur *Form* und *Ausehen* der Geräte gestaltet werden, sondern auch ihr *Verhalten*. In den Blick geraten nun die Dynamik der

Interaktion sowie die körperlich-motorische Interaktion (Buur/Jensen/Djadiningrat 2004, Larssen/Robertson/Edwards 2007). Relevante konzeptionelle und technische Entwicklungen, die wesentlich zur Verbreitung und praktischen Umsetzung von *Tangible*-Ideen beitragen, finden somit in einer Reihe von Disziplinen statt. Diese Multidisziplinarität spiegelt sich beispielsweise in den Beiträgen zur neuen Tagungsreihe »Tangible and Embedded Interaction« wider (Ullmer/et al 2007). Bevor jedoch die Schwerpunktverlagerung von den *Tangible Interfaces* zur umfassenden *Tangible Interaction* erörtert wird, sollen einige einflussreiche Vordenker vorgestellt werden.

Die Vordenker

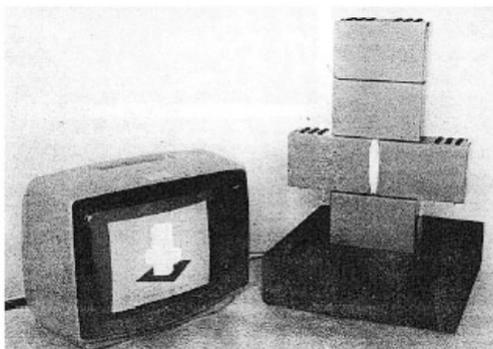
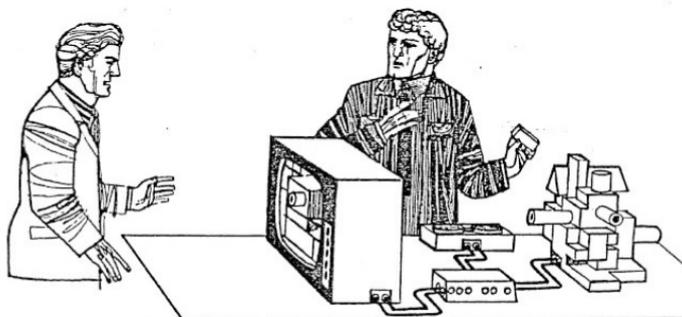
Viele TUI Systeme ziehen ihre Inspiration aus bestehenden Arbeitspraxen, wie beispielsweise der Nutzung von dreidimensionalen, händisch manipulierbaren Modellen zur Veranschaulichung von Entwürfen und als Hilfsmittel in der Konstruktion. Architekten, Stadtplaner und Ingenieure legen nach wie vor Wert auf dreidimensionale Modelle, um die räumliche Wirkung ihrer Entwürfe einzuschätzen, das räumlich-mechanische Zusammenspiel der Elemente zu verstehen, mit dem Entwurf zu experimentieren und die Diskussion in interdisziplinären Teams zu unterstützen. Systeme wie *Urp* erhalten solche Praxen und suchen sie digital zu unterstützen. Der Real Reality Ansatz (Bruns 1993, 1996, Bruns/Brauer 1996) bezieht sich ebenfalls auf gegenständliche Planungspraktiken (in diesem Fall in der Produktionsplanung und der Anlagenkonstruktion). Die Idee ist hier, simultan zu einem realen Modell ein digitales Modell zu generieren, indem die Interaktion mit dem Realmodell observiert und ins Virtuelle gedoppelt wird. Ebenfalls beliebt als Vorbilder für TUIs sind Spielzeugbaukästen (insbesondere Steckbaukästen wie Lego) und Brettspiele.

Ein genauer Blick zeigt, dass es eine Reihe von Vordenkern und Vorläufern des Gedankens greifbarer Schnittstellen gibt, und dass viele Ideen bis heute nachwirken und nicht an Relevanz verloren haben. Viele frühe Beispiele kamen aus Architektur, Lernmedien und dem Produktdesign. Im Folgenden werden einige der einflussreichsten vorgestellt. Diese Ideen wurden von HCI-Forschern als neues Konzept für Benutzungsschnittstellen aufgegriffen und systematisch verfolgt.

»Architecture without Numbers« und 3D-Modelling

Die Arbeiten von Aish und Frazer entstanden zu Beginn der 80er Jahre, als gerade die ersten graphischen Schnittstellen Verbreitung fanden. Beide Autoren wollten benutzerorientierte Alternativen zu den damals noch äußerst unhandlichen CAD-Systemen entwickeln. Wegen der technisch eingeschränkten Möglichkeiten der damaligen Zeit waren die Komponenten allerdings eher unhandlich, klobig und teuer. Dies sind vermutlich die Gründe, warum dieser Ansatz damals nicht weiter aufgegriffen wurde. John Frazer beteiligt sich seit Ende der 90er Jahre wieder an Projekten, die seine damaligen Ideen weiterentwickeln.

Abb. 4: Die Vision Frazers (Graphik aus Frazer/let al 1980) sowie der von Aish/Noakes (1984) vorgestellte Prototyp



Aish beschrieb seinen Ansatz bereits 1979 in der Zeitschrift »CAD« (Aish 1979). Sein Ziel war es, eine Beteiligung der Nutzer von Gebäuden während des Entwurfs zu ermöglichen. Aish/Noakes (1984) kritisierten, dass die mangelhaften Interaktionsmechanismen von CAD-Systemen »die

Last der Kommunikation« auf die Nutzer abladen und eine numerische Formelsprache vorschreiben. Diese mache es unmöglich, rasch verschiedene alternative Ideen zu prüfen. Die Grundidee Aishs bestand darin, ein Modell aus vordefinierten Bausteinen zu erstellen. Dieses Modell wird gescannt, in 3D-Darstellungen umgewandelt und vom Computer evaluiert (z.B. in Bezug auf die Energiebilanz). Das System berechnet die Topologie und kombiniert dies mit vorab definierten Daten über die Eigenschaften der Bausteine. Der wichtigste Beitrag eines solchen Systems wäre es, die Beziehung zwischen Kunden und Designteam zu verbessern und den Umgang mit CAD-Systemen zu vereinfachen.

Auch Frazer/et al. (1980, 1982) sahen die Vorzüge dreidimensionalen CAD-Inputs in der Verbesserung der Interaktion zwischen Designern und Kunden, in mehr aktiver Beteiligung des Klienten oder Endnutzers im Entwurfsprozess, der Spezifikation von 3D-Strukturen und einer Vereinfachung des Mensch-Computer-Dialogs. Frazer experimentierte mit verschiedenen Realisierungen und Anwendungsgebieten des »intelligenten« 3D-Modellierens. Die einfachste Lösung ordnete Komponenten auf einem zweidimensionalen Gitter an, das über Kabel mit einem Computer verbunden war. Dieser konnte nun die Konfiguration des Modells, Fläche, Wärmeverlust, Kosten, Komplexität der Wasserrohre etc. berechnen und Feedback zu strukturellen und gesetzlichen Einschränkungen geben. Eine spätere Implementierung verwendete zweidimensional verknüpfbare Elemente, die miteinander Daten austauschen konnten.

Frazer entwickelte wohl das erste in größerem Maßstab praktisch eingesetzte TUI. Die »Intelligent Modelling Technique« wurde in einem Experiment zur CAD-Nutzung durch Eigenheimbauer verwendet. In Lewisham Borough, London, konstruierten und erbauten die späteren Bewohner ihre Häuser nach dem Segal-Verfahren, welches ein System von Grundbausteinen und Maßen vorschreibt. Ein elektronisch scannbares physikalisches Baukastensystem ermöglichte es, fertige Modelle abzuspeichern. Die Grundbausteine wurden durch Leiterplatten festgelegter Größe dargestellt, welche sich in identifizierbare Teile schneiden ließen. In den Leiterplatten saßen Dioden, welche einen eindeutigen Code erzeugten, der die Maße der Bauelemente angab.

Diese Arbeiten nehmen viele Themen vorweg, die auch die heutige Forschung beschäftigen. Die Vorteile dreidimensionaler und händisch manipulierbarer Modelle werden in der Unterstützung von Diskussion und Kooperation gesehen. Sie machen es Computerlaien möglich, an einem solchen Dialog aktiv teilzunehmen sowie direkt und intuitiv zu interagieren (vgl. Bruns 1993, 1996).⁵ Die Modellbausteine selbst waren

5 Der *Real Reality*-Ansatz (Bruns 1996, Bruns/Brauer 1997) fokussierte auf die SPS-Programmierung in der Anlagenplanung. Ziel war es, heterogene

vergleichsweise abstrakt und weisen damit über die gelegentlich eher »naive« Literalität (Wortwörtlichkeit) vieler heute entwickelten TUIs hinaus.

Greifbares Programmieren

Es ist auffallend, wie viele Forschergruppen Programmiersprachen in eine greifbare, materielle Form zu übersetzen suchen. Erhofft wird dabei, einen zunächst abstrakten und immateriellen Inhalt greifbar und konkret zu machen und spielerisches Lernen zu fördern. Eine weitere Motivation ist die Unterstützung kooperativen Lernens. Neben den hier vorgestellten frühen Systemen gibt es unzählige weitere.

Eines der ersten Systeme überhaupt, das sich als Tangible Interface einordnen lässt, ist Perlmans *Slot Machine* (1976). Dies ist eine Schnittstelle zur Programmierung in Logo, mit der die auf dem Bildschirm dargestellte »Schildkröte« gesteuert wird. Ein Programm wird erstellt, indem eine Folge von Karten mit Programmkonstrukten (Aktion, Zahl, Variable, Bedingung) in horizontale Schlitze der Maschine gesteckt wird. Karten verschiedenen Typs können übereinander gesteckt werden, um zusammengesetzte Kommandos (z.B. »viermal vorwärts bewegen«) zu erzeugen.

AlgoBlocks wurde entwickelt, um Kindern das Programmieren-Lernen zu erleichtern und kooperatives Lernen zu unterstützen (Suzuki/Kato 1993). Es besteht aus großen Blöcken, die jeweils Logo-ähnliche Programmkonstrukte darstellen und zu einem lauffähigen Programm gekoppelt werden können. LEDs auf den Blöcken leuchten während der Programmausführung. Ziel der Forscher war es, kooperatives Lernen zu unterstützen, indem die großen Blöcke simultan zugänglich sind und groformatige körperliche Bewegung erfordern, was es den Lernenden erleichtert, sich gegenseitig (implizit) zu beobachten und ihre Handlungen miteinander zu koordinieren und abzustimmen.

Inspiration aus dem Produktdesign: Die Marble Answering Machine

In den Publikationen, welche die Vision der *Tangible Bits* oder später *Tangible User Interfaces* als Alternative zu graphischen Schnittstellen vorstellen (Ishii/Ullmer 1997), werden häufig die Arbeiten von Durrell Bishop als Inspiration angeführt. Die »Marble Answering Machine« ent-

Gruppen aus Planern, Facharbeitern und Ingenieuren zu befähigen, gemeinsam Planungen vorzunehmen und dabei insbesondere die Facharbeiter darin zu unterstützen, ihre Perspektive und ihr praktisches Wissen einzubringen.

stand 1992 als studentische Designskizze am Royal College of Art (Poyner 1995, Abrams 1999). Eingehende Anrufe auf einem Anrufbeantworter werden durch farbige Murmeln repräsentiert, die in eine Schale rollen. Legt man eine Murmel in eine kleine Mulde des Anrufbeantworters, wird die entsprechende Nachricht abgespielt. Eine weitere Schale dient der Aufbewahrung. Legt man eine Murmel in eine Mulde auf dem Telefon, wird die Nummer gewählt, die mit der Nachricht in der Murmel »gespeichert« ist.

Abrams (1999) interviewte Bishop für ihre Abschlussarbeit über die Rolle des Designs im Tangible Computing. Ihre Analyse zeigt, wie Bishop, anstatt Objekte allein durch ihre äußere Form mit Bedeutung »aufzuladen« (Produktsemantik), gezielt physikalische Affordances der Objekte einsetzt (d.h. Aspekte der Form, Mechanik, des Aussehens, die auf mögliche Handlungsweisen hinweisen) und wie er das Alltagswissen der Benutzer nutzt, um die Funktions- und Benutzungsweise von Geräten zu kommunizieren. Dabei gebe er den Dingen zugleich neue Bedeutungen, mache sie zu Zeigern auf andere Dinge. Für Bishop könne so jeder alltägliche Gegenstand mit digitalen Eigenschaften erweitert werden, die seine Funktionalität vergrößern. Objekte werden damit zu Repräsentationen anderer Dinge (Object Mapping). Sie werden zu Datenbehältern und verweisen auf beliebige Objekte in einem Netzwerk, mit denen der Anwender sie identifiziert. Eine weitere wichtige Idee Bishops war, dass der Kontext einer Handlung, z.B. der Ort des Objekts, von Bedeutung für die ausgelöste Funktion und die Art der Interaktion ist (Spatial Mapping). Eine Designskizze Bishops zeigt ein Radio als Wandposter. Wenn man Teile des Posters mit einem Gerät scannt, hört man verschiedene Sender.

Bishop gelang es, bekannte Alltagsobjekte als leicht »lesbaren« Referenzpunkt für die Ästhetik neuer elektronischer Produkte zu nutzen, ohne dabei einer neuen »digitalen« Situation die Metaphern der bekannten materiellen Welt überzustülpen. Während Object Mapping in vielen TUIs zu sehen ist (z.B. Webstickers – Post-Its und Tassen mit aufgedruckten Barcodes, die auf Webseiten verweisen, Holmquist/Redström/Ljungstrand 1999) und mit dem Schlagwort des »Internet der Dinge« allgegenwärtig geworden ist, wurde der Gedanke des Spatial Mappings seltener aufgegriffen. Was Bishop in seinen Entwürfen erreichte – keine wortwörtlichen, platten Metaphern zu verwenden, sondern Bedeutungen und Handlungen spielerisch zu rekombinieren und dabei »lesbar« zu bleiben, ist nach wie vor Herausforderung und Inspiration.

Vom Interface zur Interaktion

Mit der Verwendung des Begriffs *Tangible Interaction* wird hier bewusst ein Perspektivenwechsel markiert (Hornecker/Buur 2006). Produktdesigner wie Djajadiningrat/Overbeeke/Wensveen (2000) kritisieren die Benennung als *Tangible Interface*, denn wenn das System als Interface wahrgenommen werde, stünde es dem Handlungsziel im Weg. In Frage zu stellen ist bereits die Metapher der *Schnittstelle* (Grudin 1990): Sie ist mechanisch, stammt aus einer Ingenieurstradition und wird zumeist mit Software sowie technischen Ein-Ausgabegeräten identifiziert. Traditionell wird zudem zwischen dem Interface und der darunter liegenden Funktionalität eine Trennlinie gezogen. Gestaltungsfragen erscheinen so als reine Oberflächengestaltung, die nachträglich auf die feststehende Funktion aufgesetzt wird. Mit dem Begriff der *Tangible Interaction* steht nicht mehr die Schnittstelle oder ihre technische Gestaltung im Vordergrund, sondern die *Interaktion* wird zum Gegenstand der Gestaltung. Dabei geraten vermehrt qualitative Aspekte des Interaktionserlebnisses in den Blickpunkt.

Publikationen aus dem Bereich der Human-Computer Interaction führen zumeist eine Definition bzw. Charakterisierung von TUIs an, die von Ullmer/Ishii (2000) eingeführt wurde. Diese beschrieben TUIs damit, dass physikalische (materielle) Objekte Daten repräsentieren (»giving physical form to digital information«) und dass diese Objekte vom Benutzer manuell manipuliert werden. Physikalische Objekte sind mit »computationally mediated digital information« interaktiv gekoppelt und dienen zugleich zur Manipulation wie zur Repräsentation der Daten. *Tangible User Interfaces* werden hier als Alternative zu einer graphischen Schnittstelle für die Interaktion mit digitalen Daten verstanden.

Dies lässt sich als »daten-orientierte« Sichtweise kennzeichnen. Daten oder digitale Funktionen werden als gegeben vorausgesetzt und sollen nachträglich greifbar gemacht werden. Viele Systeme spiegeln diese Herangehensweise wider – oft werden graphische Elemente eins zu eins in greifbare Elemente übersetzt – oder Standard-Computerapplikationen werden mit einer neuen Benutzungsschnittstelle versehen, wobei ihre Funktionalität als gegeben behandelt wird. Statt graphischer Buttons und Schieberegler werden nun haptische Objekte geboten, an den Interaktionsabläufen ändert sich jedoch nur wenig. Djajadiningrat/et al (2004) kritisieren, dass dadurch das volle Potential physikalischer Interaktion nicht ausgeschöpft (»everything looks and feels the same«) und oft nur das GUI-Denken in zweieinhalb Dimensionen übersetzt werde.⁶

6 Zweieinhalb, weil häufig Token auf einer Fläche positioniert und verschoben werden – die Interaktion ist nicht dreidimensional.

Eine junge ›Schule‹ des Industrie- und Produktdesigns interessiert sich weniger für die Ästhetik der Form für als die »Ästhetik der Benutzung« (Abrams 1999); sie will über Form und Aussehen hinausgehen und die eigentliche Interaktion gestalten (Djajadiningrat/et al 2004). Betont wird die körperliche Interaktion mit Gegenständen und Geräten. Die Designer widmen ihre Aufmerksamkeit Dingen, die von der *Tangible User Interface*-Sichtweise mehr oder minder ignoriert wurden, da sie nicht als ›Computing‹ erscheinen: Unterhaltungselektronik, Haushaltsgeräte, Kontrollpulte und Leitwarten. Gestaltungsziel ist es, den sensorischen Reichtum materieller Dinge sowie das durch sie gebotene Handlungspotential (körperlicher Bewegung) auszuschöpfen (Buur/Jensen/Djajadiningrat 2004). Bedeutung (Semantik) wird nicht visuell vermittelt (Graphik, Sprache, visuelle Metaphern, wie z.B. Icons) sondern in die Bewegungen des Objekts sowie die mit ihnen möglichen Handlungen gelegt. Interaktionshandlungen sollen in sich bedeutungsvoll sein und Form sowie Mechanik sollen zu solchen Handlungen herausfordern (ein Beispiel wäre eine Digitalkamera, bei der man zum Speichern eines Bildes das bewegliche Display über die Speicherkarte schiebt). Angestrebt werden ausdrucksvolle Bewegungen des Nutzers sowie vielfältige Interaktionsweisen mit ›stark spezifischen‹ Produkten. Das Ideal des ›easy to use‹ wird dabei provokant in Frage gestellt und die Gestaltung von spezialisierten Systemen für intellektuell wie körperlich kompetente Anwender (›embodied skills‹) in spezifischen Anwendungsfeldern favorisiert (Djajadiningrat/Overbeeke/Wensveen 2000, Buur/Jensen/Djajadiningrat 2004).

Löst man sich von einer definitionsorientierten Sichtweise, ist die Relevanz weiterer Ansätze besser zu erkennen. Viele »Interactive Arts«-Installationen beruhen auf *Embodied Interaction*. Sie machen die Körperbewegung der Kunstbetrachter (bzw. korrekter: Kunst-Erlebenden) zum Bestandteil des Kunstwerks, betonen eventuell sogar die leibliche Selbstwahrnehmung und verwandeln ganze Räume in interaktive Mixed Reality-Umgebungen (Rubidge/MacDonald 2004). In interaktiver Kunst und Architektur wird zunehmend über »interaktive Räume« gesprochen (vgl. Bongers 2002). Diese kombinieren realen Raum und materielle Objekte mit digitalen Projektionen oder Sound-Installationen. Die Bewegung im Raum und der Umgang mit Gegenständen aktivieren digitale Reaktionen des Systems. Typische Charakteristika solcher Installationen sind die Einbeziehung des ganzen Körpers sowie die Nutzung des eigenen Körpers als ›Eingabemedium‹.

Die Perspektive der *Tangible Interaction* (Hornecker/Buur 2006) verbindet die beschriebenen Sichtweisen zu einer Synthese, die ein breites Spektrum an Systemen bzw. Gestaltungsansätzen umfasst. Diese teilen als Eigenschaften: Greifbarkeit und Materialität, die Verkörperung von

Daten, körperliche Interaktion und Körperbewegung als essentieller Part der Interaktion sowie die Einbettung im realen Raum. *Tangible Interaction* beschränkt sich nicht auf die Steuerung digitaler Daten. Sie umfasst interaktive Gebrauchsgegenstände wie den eingangs vorgestellten Wecker sowie die Fernsteuerung ganz realer Vorgänge und Dinge unserer Umwelt (z.B. ein Thermostat) – Bereiche, die von der Tangible Interface Charakterisierung von Ullmer/Ishii (2000) nicht erfasst werden. Der Name ist hier Programm: Nicht das Interface, sondern die Gestaltung der Interaktion steht im Vordergrund. Die Interaktion mit »interaktiven Räumen« erweitert unsere Perspektive der *Tangible Interaction* weiter, wobei der Körper des Benutzers zum Interaktionsgerät wird. Ein Beispiel hierfür bietet die an sich einfache Konfiguration der Klaviatur – die Besucher laufen über die sensorisierte Fläche und aktivieren mit ihren Körpern die Sensoren. Der bewusst umfassende und leicht unscharfe Begriff der Tangible Interaction weist auf diesen Gestaltungsraum hin und sucht das Gemeinsame der Perspektiven hervorzuheben. Bewusst wurde daher auch der Titel einer neuen Konferenzreihe gewählt »Tangible and Embedded Interaction« (Ullmer et al 2007) – und die stattliche Zahl von über 90 Einreichungen zeugt davon, dass eine internationale Forschungsgemeinschaft zu diesem Thema heranwächst und den Diskurs sucht.

Mit der neuen, umfassenden Perspektive geht allerdings der Anspruch verloren, eine generelle Alternative zu graphischen Schnittstellen herzustellen. Zwar war die Benennung »Tangible User Interfaces« als rhetorisches Mittel, um die Neuartigkeit des Ansatzes zu betonen, erfolgreich; der damit implizit verbundene Anspruch, jegliches GUI ersetzen zu können, ist jedoch unrealistisch. Denn zu fragen ist immer, in welchen Anwendungsgebieten und in welchem Ausmaß dies möglich und sinnvoll ist. In den Fällen, wo Projekte Tangible Interfaces für Programmiersprachen entwickeln, wenden sich diese Systeme zumeist an Kinder, um diesen das Erlernen des Programmierens zu erleichtern. Andere Systeme sollen Laien das Konfigurieren und die Steuerung ihrer Unterhaltungselektronik und Haushaltsgeräte vereinfachen. Keines der aus der Literatur bekannten Projekte stellt jedoch den Anspruch, das Programmieren generell zu unterstützen. Zudem vereinen die meisten realisierten Systeme von jeher graphische *und* greifbare Elemente – der scheinbare Gegensatz ist also nur ein gradueller Unterschied. Da *Tangible Interaction* weniger die Schnittstelle als die Interaktion als Gegenstand der Gestaltung betrachtet, können Monitore und Projektionen sehr wohl ein Bezugspunkt der Interaktion sein.

Richten wir unsere Aufmerksamkeit auf das Design interaktiver Produkte, nehmen wir diese mehr als *Gerät* wahr denn als Computer. Warum sollten sie wie Computer aussehen? Soll Interaktion bedeutsam und ausdrucksvoll sein, muss sie in Bezug zur Anwendung stehen. Daher kriti-

sieren Industrie- und Produktdesigner den Trend, allen möglichen Geräten gleichartige »weak general« Schnittstellen zu geben und passen materielle Formfaktoren der Geräte (ihr Aussehen, ihre Mechanik) der Funktionalität an (Djajadiningrat/Overbeeke/Wensveen 2000, Buur/ Jensen/ Djajadiningrat 2004). Diese Geräte sind daher »strong specific«, während es die Stärke des Desktop-Computers ist, ein »weak general system« zu sein, dessen Hardware- und Software-Schnittstelle (Maus und Tastatur, Fenster, Mauszeiger...) unzählige verschiedene Anwendungen steuert. Spezielle Geräte haben in vielen Anwendungsbereichen ihren Platz und ihre Berechtigung; ebenso können spezifische Schnittstellen (für eine computerbasierte Applikation) in bestimmten Anwendungsbereichen ihre Berechtigung haben.

Schluss

Zunächst ist das Interesse an Tangible User Interfaces sicherlich gespeist durch das unbestimmte Unwohlsein, in der Interaktion mit dem Computer an das Dreieck von Maus, Tastatur und Bildschirm gefesselt zu sein und dabei etwas verloren zu haben, entfremdet von der realen Welt der Dinge zu sein. Bruns (1993, S. 15) beklagt diesen Verlust: »Es gibt nichts mehr zu ertasten. Die Dinge werden schwerelos, widerstandslos, trägheitslos, leer, ohne Substanz«. Diese neue Strömung der MMK kann interpretiert werden als Schwingen des Pendels vom Abstrakten und Vergeistigten zurück zum Dinghaft-Konkreten und zur Anerkennung des menschlichen Körpers – eine solche Pendelbewegung sehen Kulturwissenschaftler in der Philosophie und Kunst der letzten Jahrhunderte immer wieder (Mattenkloft 1997). Manche mögen dies als eine sentimentale Schwärmerie erachten, andere als eine notwendige Besinnung, die umso vehementer ausfallen muss, je extremer die Ausgangsposition war (in diesem Fall insbesondere der Virtual Reality-Hype der späten 80er und frühen 90er Jahre).

Unterstützt wird die konzeptionelle Entwicklung von Visionen alternativer Formen des »Computing« durch aktuelle technische Entwicklungen, welche digital erweiterte Umgebungen und intelligente Alltagsprodukte bereits zur Realität werden lassen. Tangible Interaction Design findet insofern in vielen Bereichen statt, die »unremarkable« (unauffällig) sind – gestaltet werden Dinge des Alltags, Museumsinstallationen, Spielzeug, Fernbedienungen, Leitwarten, Küchengeräte. u.s.w. Ubiquitous und Pervasive Computing, RFID-Tagging und IT-Produktdesign verändern bereits heute die Welt, doch sind sie oft gelehrt von technozentrischen Vorstellungen, orientieren sich an altgewohnten Designmustern des Desktop-Computing oder bleiben verhaftet in traditionellem Produkt-

design, das sich mehr dem visuellen Design und der Form widmet als dem Verhalten und der Interaktion. Tangible Interaction Design kann daher gesehen werden als der Versuch, diese Entwicklung bewusst zu gestalten und die neuen Möglichkeiten anzunehmen, und dabei der anthropologischen Verankerung des Menschen in der Umwelt und im eigenen Leibe Rechnung zu tragen.

Danksagung

Ich danke den Teilnehmern der CHI 2006 Podiumsdiskussion »The State of Tangible Interfaces: Projects, Studies, and Open Issues«. Eingegangen in diesen Beitrag sind Gedanken aus Diskussionen mit Brygg Ullmer sowie mit Paul Marshall. Besonderer Dank gebührt Jacob Buur und seiner Arbeitsgruppe, der mir die Grenzen einer engen Definition des Tangible Interface vor Augen hielt und mich mit der Sichtweise des IT Product Designs vertraut machte. Ferner danke ich den Urhebern der verwendeten Illustrationen für die Druckerlaubnis, Hans Dieter Hellige und Matthias Müller-Prove gebührt Dank für das genaue und strenge Redigieren meines manchmal nicht mehr ganz wasserfesten Deutschs.

Literatur

- Abrams, R. (1999): »Adventures in Tangible Computing: The Work of Interaction designer ›Durrell Bishop‹ In Context«. Master's thesis, Royal College of Art, London.
- Aish, R. (1979): »3D Input for CAAD Systems«. In: Computer Aided Design 11, 2, S. 66-70.
- Aish, R./Noakes, P. (1984): »Architecture Without Numbers«. Computer Aided Design 16, 6, S. 321-328.
- Bongers, B. (2002): »Interactivating Spaces«. In: Proceedings of the the 4th Annual Symposium on Systems Research in the Arts: Music, Environmental Design and The Choreography of Space , in conjunction with the 14th International Conference on Systems Research, Informatics and Cybernetics July 31- August 3, 2002 Baden-Baden.
- Bruns, F.W. (1993): »Zur Rückgewinnung von Sinnlichkeit. Eine neue Form des Umgangs mit Rechnern«. In: Technische Rundschau 29, 39, S. 14-18.
- Bruns, W. (1996): »Grasping, Communicating, Understanding - Connecting Reality and Virtuality«. In: AI & Society, 10, 1, S. 6-14.
- Bruns, W./Brauer, V. (1996): »Bridging the Gap between Real and Virtual Modeling - A New Approach to Human-Computer Interaction«. In:

- Stefan Haas (editor): Proceedings of the IFIP 5.10 Workshop on Virtual Prototyping, Providence, September 1994, IFIP.
- Buur, J./Jensen, M.V./Djajadiningrat, T. (2004): »Hands-Only Scenarios and Video Action Walls: Novel Methods for Tangible User Interaction Design«. In: Proceedings of the 5th Conference on Designing Interactive Systems (DIS'04). New York, S. 185-192.
- Djajadiningrat, T./Wensveen, S./Frens, J./Overbeeke, K. (2004): »Tangible Products: Redressing the Balance Between Appearance and Action«. In: Personal and Ubiquitous Computing 8,5, S. 294-309.
- Djajadiningrat, T./Overbeeke, K./ Wensveen, S. (2000). »Augmenting Fun and Beauty: A Pamphlet«. In: Proceedings of DARE 2000 on Designing augmented reality environments. New York, S. 131-134.
- Dourish P. (2001): »Where the Action Is. The Foundations of Embodied Interaction«. Cambridge, MA, London.
- Fitzmaurice, G.W./Ishii, H./Buxton, W. (1995): »Bricks: Laying the Foundations for Graspable User Interfaces« In: Proceedings of Computer-Human Interaction (CHI '95), Denver, Colorado, May 7-11, 1995. New York. S. 442-449.
- Fjeld, M./Bichsel, M./Rauterberg, M. (1997): »BUILD-IT: An Intuitive Design Tool Based on Direct Object Manipulation«. In: »Gesture and Sign Language in Human-Computer Interaction«. Proceedings of International Gesture Workshop 1997. Berlin, Heidelberg, New York. S. 287-308.
- Frazer, J./Frazer, J./Frazer, P. (1980): »Intelligent Physical Three-Dimensional Modelling Systems«. In: Proceedings of the Computer Graphics '80, Brighton, UK, S. 359-370.
- Frazer, J./Frazer, J./Frazer, P. (1982): »Three-Dimensional Data Input Devices«. In: Proceedings of Computer Graphics in the Building Process. National Academy of Sciences. Washington, March 1982.
- Grudin, J. (1990): Interface. In: Proceedings of Computer Supported Cooperative Work (CSCW'90). New York, S. 269-278.
- Holmquist L.E./Redström J./Ljungstrand P. (1999): »Token-Based Access to Digital Information«. In: Proceedings of Handheld and Ubiquitous Computing (HUC '99), Berlin, Heidelberg, New York, S. 234-245.
- Hornecker, E./Bruns F.W. (2005): Interaktion im Sensoric Garden. In: i-com, Bd. 1, S. 4-11.
- Hornecker, E./Buur, J. (2006): »Getting a Grip on Tangible Interaction: A Framework on Physical Space and Social Interaction«. In: Proceedings of CHI '06. New York, S. 437-446.
- Hurtienne, J./Israel, J.H. (2007): »Image Schemas and Their Metaphorical Extensions – Intuitive Patterns for Tangible Interaction«. In: Proceedings of the First International Conference on Tangible and Embedded Interaction 2007 (TEI'07). New York, S. 127-134.

- Ishii, H./Ullmer, B. (1997): »Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms«. In: Proceedings of CHI '97. New York, S. 234 - 241.
- Larsen, A.T./Robertson, T./Edwards, J (2007): »The Feel Dimension of Technology Interaction: Exploring Tangibles through Movement and Touch«. In: Proceedings of TEI '07. New York, S. 271-278.
- Mackay, W.E./Fayard, A-L. (1999): »Designing Interactive Paper: Lessons from Three Augmented Reality Projects«. In: Proceedings of the International Workshop on Augmented Reality, IWAR '98. Natick, MA.
- Marshall, P. (2007): »Do Tangible Interfaces Enhance Learning?« In: Proceedings of TEI '07. New York, S. 163-170.
- Mattenklott, G. (1997): »Berührend berührt - Die Ästhetik des Tastsinns«. In: Universitas. Zeitschrift für interdisziplinäre Wissenschaft 52, 617, S. 1050-1064.
- Pearlman, R. (1976): »Using Computer Technology to Provide a Creative Learning Environment for Preschool Children«. MIT Logo Memo #24, 1976.
- Poyner, R. (1995): »The Hand That Rocks the Cradle«. ID Magazine, 60-65. May/June.
- Rubidge, S./MacDonald, A. (2004): »Sensuous Geographies: A Multi-User Interactive/Responsive Installation«. In: Digital Creativity 15, 4, S. 245-252.
- Suzuki, H./Kato, H. (1993): AlgoBlock: »A Tangible Programming Language, a Tool for Collaborative Learning«. In: Proceedings of 4th European Logo Conference. S. 297-303.
- Ullmer, B./Ishii H. (2000): »Emerging Frameworks for Tangible User Interfaces«. In: IBM Systems Journal 39,3/4, S. 915-931.
- Ullmer, B./Schmidt, A./Hornecker, E./Hummels, C./Jacob, R., van den Hoven, E. (Hrsg.) (2007): »Proceedings of the First International Conference on Tangible and Embedded Interaction 2007«. New York.
- Underkoffler, J./Ishii, H. (1999): »Urp: A Luminous-Tangible Workbench for Urban Planning and Design«. In: Proceedings of CHI '99. New York, S. 386-393.
- Weiser, M. (1993): »Some Computer Science Issues in Ubiquitous Computing«. In: Communications of the ACM 36, 7, S. 74-84.
- Wellner, P./ Mackay, W./Gold, R. (1993): »Computer-Augmented Environments. Back to the Real World«. In: Communications of the ACM 36,7. S. 24-26.
- Wensveen, S./Overbeeke, C.J./Djajadiningrat, J.P. (2002): »Push Me, Shove Me and I Show You How You Feel«. In: Proceedings of DIS'02, New York. S. 335-340.

**DIE »MCI DER ZUKUNFT«: MULTISENSORIK,
EVERYWHERE INTERFACES UND PROAKTIVE
UMGEBUNGS-INTELLIGENZ?**

UBIQUITOUS COMPUTING: EIN NEUES KONZEPT DER MENSCH-COMPUTER- INTERAKTION UND SEINE FOLGEN

MICHAEL FRIEDEWALD

Im Laufe der achtziger Jahre etablierten sich Personal Computer als Arbeits- und Spielgeräte, seit Beginn der neunziger Jahre zunehmend auch als Kommunikations- und Informationsmedien für den Massenmarkt. Dabei setzten sich grafische Benutzeroberflächen mit Fenstern, Icons und Menüs als Paradigma der Mensch-Computer Interaktion durch (Friedewald 1999). Die Fortschritte im Rahmen dieses Paradigmas sind seither inkrementell, die Produkte der großen Hersteller kaum mehr voneinander zu unterscheiden. Tatsächlich zeigen heutige grafische Benutzeroberflächen die gleichen Schwächen, die bereits zur Zeit ihrer Entwicklung und Markteinführung kritisiert wurden – die Komplexität und nur scheinbare Intuitivität der Bedienung sowie die hohen Anforderungen an die Aufmerksamkeit, die den Nutzer von seiner Umgebung isoliert. Mittlerweile haben sich die meisten Computerbenutzer so an die Benutzungsschnittstellen ihres Computers gewöhnt, dass sie diese Gewöhnung mit der Benutzerfreundlichkeit des Computers verwechseln. Wehner und Rammert (1990, S. 229) sprechen im Zusammenhang mit dieser Veralltäglichung im Umgang mit den Unzulänglichkeiten der Technik von »Aneignungszumutungen« (vgl. auch Raskin 1994).

Heute verstehen sich »Ubiquitous Computing« bzw. »Ambient Intelligence« als eine Abkehr vom derzeit dominierenden Grundkonzept moderner grafischer Benutzungsschnittstellen bei Arbeitsplatzcomputern oder gar als »neues Paradigma« in der Entwicklung von Mensch-Computer-Interfaces (Remagnino et al. 2005). Im Folgenden soll der Entwicklung dieses Konzeptes sowie seiner möglichen Auswirkungen nachgegangen werden. Dabei wird der Frage nachgegangen, welche Vorstellungen die Entwicklung des Ubiquitous Computing antreiben und auf welchen Prämissen diese Leitbilder zurückgreifen.

Mensch-Computer-Interfaces jenseits der grafischen Benutzeroberflächen

Als sich in den achtziger Jahren die grafischen Benutzungsoberflächen kommerziell durchsetzten, begann eine junge Generation von Wissenschaftlern neue Vorstellungen über die Mensch-Computer-Interaktion zu entwickeln, die auf sozialwissenschaftlichen Forschungsergebnissen basierten (Turkle 1984; Suchman 1985; später auch Laurel 1993; Reeves/Nass 1996). Anthropologische und konstruktivistische Ansätze hatten seit Ende der 1970er Jahre Einzug in die Informatik gehalten. Eine Pionierin des anthropologischen Ansatzes war Lucy Suchman, die seit 1979 in jener Forschungsgruppe des Xerox Palo Alto Research Centers (PARC) arbeitete, die zur gleichen Zeit auch die ersten graphischen Benutzeroberflächen entwickelt hatte. Diese Gruppe hatte eine kognitionspsychologische Theorie des Interaktionsdesigns entwickelt, bei dem es vor allem um die Modellierung mentaler Nutzermodelle ging (Card et al. 1983).

Suchman verfolgte einen anderen Ansatz, indem sie vorschlug, diese Entwurfsmethodik durch Verfahren zur Beobachtung und Analyse individueller und kollektiver Praktiken im Umgang mit den neuen Informationstechnologien in konkreten Arbeitszusammenhängen zu ergänzen, um die stets auftretenden Unterschiede zwischen der modellhaften und der tatsächlichen Nutzung aufzudecken. Solche Unterschiede konnte sie in einer Reihe von experimentellen Untersuchungen auch praktisch nachweisen. Ihre Erfahrungen fassten Suchman und ihr Kollege Randy Triggs wie folgt zusammen:

»Designers interested in augmenting or replacing current artifacts [...] do well to understand how they work, as well as what their limits are. [...] Design realism can be achieved, we believe, through new methods for understanding the organization of work practice in detail.« (Suchman/Trigg 1991, S. 73)

Etwa zur gleichen Zeit stellten sich Byron Reeves and Clifford Nass an der Stanford University die Frage, ob eine Computer-Anwendung in ihren Dialoganteilen bei dem Benutzer ähnliche Emotionen und Einstellungen auslöste, als hätte diese menschliche Persönlichkeitseigenschaften.¹ In einer Vielzahl von empirischen Untersuchungen kamen sie zu dem Ergebnis, dass Menschen die Tendenz haben, die über Medien konstruierten (vir-

1 Die Erkenntnis, dass die Mensch-Computer-Kommunikation eine soziale Dimension besitzt, ist freilich keine völlig neue Erkenntnis der 1980er Jahre, sondern wurde u.a. schon in den 1960er Jahren von Joseph Weizenbaum im Zusammenhang mit seinem Programm »ELIZA« diskutiert (Weizenbaum 1966, 1990, S. 250-255).

tuellen) Welten mit dem realen Leben gleichzusetzen. Infolgedessen sei die Interaktion des Einzelnen mit dem Computer als Prozess der Wirklichkeitskonstruktion im Grunde sozialer Natur (Reeves/ Nass 1996).²

XEROX PARC und das Konzept des Ubiquitous Computing

Ende 1987 schlug eine Gruppe von Wissenschaftlern unter Leitung von Mark Weiser (1952-1999) am Xerox PARC vor, wandgroße flache Computerdisplays herzustellen (Abb. 1). Sie gingen davon aus, dass sich solche Displays nicht nur als Ausgabe-, sondern auch als Eingabemedium zur Nutzung mit elektronischen Stiften oder zum Einscannen von Dokumenten eignen könnten. Aus dieser Anfangsidee entwickelten die Wissenschaftler am PARC in den nachfolgenden Jahren eine ganze Reihe von neuartigen Displays unterschiedlicher Größe für verschiedene Arbeitszusammenhänge. Das Spektrum reichte dabei vom elektronischen Zettel für den individuellen Nutzer, über den elektronischen Notizblock bis zur fest installierten elektronischen Tafel zur Nutzung durch Gruppen, die insbesondere durch die Vernetzung untereinander als auch mit anderen Geräten einen zusätzlichen Nutzen bieten konnten (Want et al. 1995).

Die Entwicklungsarbeiten führten auch bald zu neuen Vorstellungen über den Nutzer und die Nutzung der Informationstechnik. Einerseits sollten mehrere Nutzer gleichzeitig an einer elektronischen Tafel arbeiten, andererseits würde der individuelle Nutzer neben seinem persönlichen Endgerät auch die an verschiedenen Orten fest installierten Displays verwenden (Weiser et al. 1999). Bei diesem Wandel der Nutzungsweisen spielten auch die Forschungsergebnisse von Lucy Suchman eine wichtige Rolle, die den Technologen vor Augen geführt hatte, dass es weniger auf die einzelnen Kenndaten der eingesetzten Technik ankomme, sondern vielmehr auf den detaillierten Einsatzkontext, also die Einbettung des Computers in das komplexe Bezugssystem der täglichen Arbeit:

»The idea of ubiquitous computing arose first from contemplating the place of today's computer in actual activities in everyday life. In particular, anthropological studies of work life [by Lucy Suchman and Jean Lave] teach us that

2 Die folgende Darstellung der Konzeptentwicklung bleibt auf den amerikanischen Kontext beschränkt. Es gab freilich während der 1970er und 1980er Jahre auch in Europa anthropologische Ansätze in der Informatik. Nach Wissen des Autors gab es allerdings kaum Rückwirkungen auf die Entwicklung in den USA. Vgl. etwa die Beiträge in Floyd et al. 1992 und Coy et al. 1992

people primarily work in a world of shared situations and unexamined technological skills. The computer today is isolated from the overall situation, however, and fails to get out of the way of the work« (Weiser 1993, S. 76).

Abb. 1: Ubiquitous Computing-Installation bei Xerox PARC mit LiveBoard und ParcPad (vorne rechts). Mark Weiser ist ganz links zu sehen. Quelle: Weiser 1991



All dies führte 1988 zu einer radikalen Kritik an dem am PARC selbst entwickelten und mittlerweile paradigmatischen Leitbild des »Personal Computing«:³

»Dass den PC weithin noch immer eine Aura des Geheimnisvollen umgibt, liegt nicht bloß an der so genannten Benutzerschnittstelle. Meine Mitarbeiter und ich am [...] PARC halten vielmehr die ganze Idee eines persönlichen Computers für eine Sackgasse; in unseren Augen sind handliche Laptop-Geräte, elektronische Notiz- und Wörterbücher (*dynabooks*) und sogenannte Wissensnavigatoren nur Vorstufen zur eigentlichen Informationstechnik der

3 Aus ähnlichen Erkenntnissen erwuchs bereits 1984 mit dem »Computer-supported cooperative work« (CSCW) ein anderes interdisziplinäres Konzept, dessen zentrale Forschungsgegenstände die Kooperationen zwischen Menschen und deren Unterstützbarkeit durch Rechner sind.

Zukunft. Mit solchen Geräten kann die Datenverarbeitung nicht wirklich zu einem integralen, unsichtbaren Bestandteil des Alltags werden. Darum suchen wir eine neue Haltung zum Computer zu entwickeln, die den Menschen in den Mittelpunkt stellt und die Computer im Hintergrund verschwinden lässt« (Weiser 1991, S. 92).

Weiser und seine Kollegen entwickelten diesen Gedanken weiter, indem sie die technologische Entwicklung extrapolierten und in ihre Vision mit einbezogen. Demnach eröffne die Hardwareentwicklung mit der weiteren Miniaturisierung elektronischer Schaltkreise bei gleichzeitig fallenden Preisen bereits in naher Zukunft die Möglichkeit zur Realisierung ihrer Vision einer im Hintergrund verschwindenden Computertechnik mit neuen Interaktionsformen (Rheingold 1994).

Im Jahr 1991 veröffentlichte Weiser seine Vision unter dem Titel »The Computer for the 21st Century« in einem Themenheft des populärwissenschaftlichen Magazins »Scientific American«, in dem auch andere Pioniere der Mensch-Computer-Interaktion ähnliche Ideen formulierten (Kay 1991; Tesler 1991). Er stellte in seinem Artikel die These auf, dass im 21. Jahrhundert allgegenwärtige Computer den Menschen unaufdringlich und unsichtbar bei seinen Tätigkeiten unterstützten und ihn von lästigen Routineaufgaben weitestgehend befreien würden. Er formulierte damit eine Vision, die der 30 Jahre zuvor von J. C. R. Licklider (1961) propagierten »Mensch-Computer-Symbiose« ähnelte. Weiser selbst nannte diese, das Verhältnis zwischen Menschen, Arbeit und Technologie völlig neu definierende Form der Datenverarbeitung »ubiquitous computing« und sah in ihr bereits eine neue Welle der Datenverarbeitung, die über die vorherigen Wellen des isolierten Computers (sei es als Mainframe oder als Personal Computer) und des Internet hinausweist:

»The third wave of computing is that of ubiquitous computing, whose crossover point with personal computing will be around 2005-2020. The »UC« [Ubiquitous Computing, d. Verf.] era will have lots of computers sharing each of us. Some of these computers will be the hundreds we may access in the course of a few minutes of Internet browsing. Others will be embedded in walls, chairs, clothing, light switches, cars - in everything. UC is fundamentally characterized by the connection of things in the world with computation. This will take place at a many scales, including the microscopic« (Weiser/Brown 1997, S. 77).

Im Gegensatz zum traditionellen Ansatz der Computertechnik, wo der Computer als Werkzeug im Vordergrund der Aktivitäten steht, basiert das Ubiquitous Computing auf dem Prinzip, den Computer als Artefakt im

»Hintergrund« verschwinden zu lassen. Um die Bedeutung des »Hintergrunds« näher zu erläutern, bezieht sich Weiser wie andere Vertreter eines konstruktivistischen Ansatzes in der Informatik auf Philosophen wie Martin Heidegger und Hans-Georg Gadamer. Demnach führt nicht die prinzipielle Vorhandenheit eines Werkzeugs zum Ziel, sondern nur dessen faktische Zuhilfenahme, die Heidegger als »Zuhandenheit« (Heidegger 2001, § 15) bezeichnet.

Einbettung in den Hintergrund sollte demzufolge in zweierlei Weise verstanden werden: Zunächst wörtlich als die physische Einbettung der Computertechnik in Werkzeuge, Gegenstände und die Umwelt. Im weiteren Sinne muss diese Einbettung so realisiert werden, dass das Computersystem, die Anwendung oder der Dienst sich nicht mehr mit den anderen menschlichen Aktivitäten überlagert. In der Heideggerschen Seinsanalyse enthüllt ein technisches Artefakt wie ein Hammer seinen wesenhaften Gehalt gerade nicht im deskriptiven Auflisten seiner Eigenschaften. Ein solches Werkzeug ist für seinen Nutzer als solches nicht existent, sondern Teil des Hintergrundes an Zuhandenheit, der als selbstverständlich vorausgesetzt wird. Erst im »Zusammenbruch«, also wenn das Werkzeug seinen Dienst versagt oder unerwartete Effekte auftreten, kommt sein Charakter dem Menschen zu Bewusstsein. Nur der umsichtige und zielgerichtete Gebrauch verleiht dem Artefakt den Modus der Zuhandenheit und enthüllt damit seine Bedeutung. Das Werkzeug wird damit zu einem von vielen als selbstverständlich erachteten Aspekt bei menschlichen Tätigkeiten.

Mit einer im Hintergrund verschwindenden Computertechnik könne man – so die Hoffnung der PARC-Wissenschaftler – zwei wichtige Ziele erreichen: Zunächst würde sie den Menschen bei der Durchführung spezifischer Aufgaben unterstützen. Gleichzeitig würde sie auch dazu beitragen, dass sich die Nutzung der Computertechnik nicht nur auf eine Gruppe von technisch versierten Nutzern beschränkt. Vielmehr erwartet Weiser, dass »die verkörperte Virtualität [...] den Computer [...] hinaus in alle gesellschaftlichen Gruppen« bringt (Weiser 1991, S. 101). Dies sind bis heute wünschenswerte Ziele in einer als Wissens- oder Informationsgesellschaft bezeichneten Welt und ein Hinweis auf die ungebrochene Attraktivität von Weisers Vision als Leitbild in Wissenschaft und Politik.

Der Artikel war in zweierlei Hinsicht einflussreich: er formulierte eine Forschungsagenda und ein Leitbild, das seither viele andere Forscher aufgegriffen und weiterentwickelt haben. Ubiquitous Computing ist also vor allem durch das mit ihm verbundene Zukunftsbild geprägt. In den vom Science Citation Index (SCI) erfassten Veröffentlichungen ist Weisers Artikel seit 1991 345-mal zitiert worden und tatsächlich wird seine dama-

lige Vision noch heute in vielen Publikationen diskutiert.⁴ Dies ist umso bemerkenswerter als die seither vergangenen 16 Jahre in der Welt der Informationstechnik eine Ewigkeit darstellen und die wissenschaftlich-technischen Voraussetzungen heute ganz andere sind als im Jahr 1991, als es weder das World Wide Web noch GSM-Mobiltelefonie gab.

Heute wie schon 1991 werden gern Bilder einer nahen Zukunft gezeichnet, in der die Versprechungen des Ubiquitous Computing in Bezug auf eine unsichtbare und intelligente Unterstützung des Alltags eingelöst sind. Damit stellt sich automatisch die Frage, ob und ggf. warum diese nicht längst Realität geworden sind. Darauf gibt es eigentlich nur zwei mögliche Antworten: Entweder sind die entworfenen Zukunftsbilder grundsätzlich nicht einlösbar oder sie sind durch eine Vielzahl von inkrementellen Fortschritten bereits unmerklich realisiert worden.

Als wichtigstes Missverständnis bei der Rezeption von der ursprünglichen Vision hat sich im Rückblick das Versprechen erwiesen, Informationstechnik könne wirklich jemals aus dem Blick des Menschen verschwinden, würde unsichtbar und lautlos funktionieren. Die Erfahrung lehrt aber, dass Infrastrukturtechniken – und nichts anderes ist Ubiquitous Computing – wegen ihrer Heterogenität notorisch sichtbar bleiben und von den Nutzern sogar ganz besonders sorgfältig beobachtet werden, sei es aus Fragen des Zugangs oder der Abrechnung. Diese Art von Unordnung ist aber keine Besonderheit einer frühen Entwicklungsphase, sie verschwindet nicht mit der Reife der Technologie, sondern charakterisiert große Infrastrukturen auch noch Jahrzehnte nach ihrer Verbreitung, wie der Blick auf die Energie- oder Verkehrsinfrastruktur zeigt (Star 2002). Die Vision einer nahtlos funktionierenden einheitlichen Infrastruktur ist also bestenfalls das Trugbild einer niemals realisierbaren Welt.

Dies bedeutet aber nicht, dass es keine wichtigen Fortschritte auf dem Weg zu einer Allgegenwärtigkeit der Informationstechnik gegeben hat, auch wenn diese nicht im wörtlichen Sinne aus dem Blick der Nutzer verschwunden ist. Man kann sogar konstatieren, dass Weisers Vorhersagen für die nahe Zukunft, die längst unsere Gegenwart ist, bemerkenswert genau waren. Bereits 2002 wurde in der Welthandelsstatistik für Halbleiter festgestellt, dass deutlich über 90 % der Mikroprozessoren eingebettete Prozessoren sind, die ihren Dienst in einer Vielzahl von Industrie- und Konsumprodukten verrichten: Im Fernseher und im Mobiltelefon, aber auch Auto und der heimischen Waschmaschine. Dieser Trend hat sich in

4 Tatsächlich ist Weisers Artikel eine so genannte »sleeping beauty« (van Raan 2004), d.h. eine Veröffentlichung, die ihrer Zeit voraus ist und für einige Jahre »schläft«. Im Durchschnitt werden wissenschaftliche Artikel 4 Jahre nach ihrer Veröffentlichung am häufigsten zitiert, danach nimmt die Zitationshäufigkeit stark ab. Weisers Artikel wurde aber in den ersten 10 Jahren kaum wahrgenommen und wird seit 2003 bis heute häufig zitiert.

den vergangenen Jahren weiter fortgesetzt. Computer bzw. Rechenleistung ist also längst ubiquitär, nur dass die Benutzer davon nichts merken, weil die Technologie unsichtbar geworden ist und hinter der Funktionalität völlig zurücktritt. Ähnliches ist seit den späten 1990er Jahren auch in der mobilen Kommunikation geschehen. Hier ist die Informationstechnik heute z. B. in Form von Smartphones und einer Vielzahl so genannter »mobile appliances« nicht mehr wegzudenken. Insbesondere in Ostasien gibt es mittlerweile die ersten All-over-mobiles-Gesellschaften, in denen Kommunikationsmöglichkeiten allgegenwärtig sind und sogar eine gemeinschaftsstiftende Funktion besitzen (Lie 2005; Bell 2006). Mit RFID-Anwendungen in Logistik- und Warenwirtschaftssystemen beginnen Konzepte des Ubiquitous Computing schließlich seit einigen Jahren ganz langsam auch in betriebliche Abläufe Einzug zu halten (Fleisch/Michahelles 2007).

Obwohl all diese Entwicklungen den Einstieg in das Ubiquitous Computing verkünden, fehlt es noch an der umfassenden kommunikationstechnischen Infrastruktur und der Integration der verschiedenen Anwendungen. Und auch das proaktive Zusammenwirken von Programmintelligenz zwischen Alltagsgegenständen und -prozessen und ihren Nutzern realisiert sich nur langsam. Dabei ist es allerdings wenig wahrscheinlich, dass der Personal Computer, wie von Weiser propagiert, vollständig abgelöst wird, u.a. weil zweifelhaft bleibt, ob das Verschwinden des Computers in den Hintergrund überhaupt von den Nutzern gewünscht ist.

Andere Begriffe - ähnliche Konzepte

Trotz der intensiven Rezeption und wegweisender Wirkung von Weisers Arbeiten entstanden in den 90er Jahren eine ganze Reihe ähnlicher Konzepte bzw. Begriffe, von denen Nomadic Computing, Pervasive Computing und Ambient Intelligence die wohl bekanntesten sein dürften.

Nomadic Computing

Leonard Kleinrock, Professor an der University of California in Los Angeles und einer der Väter des Internet, bezeichnete 1995 Formen mobiler Computernutzung als »nomadic computing«, weil im Zuge der globalen Computervernetzung *nomadische* Arbeitsformen aufgenommen hätten:⁵

5 Entsprechend häufig war das Konzept des »nomadic computing« auch mit der Diskussion um die »Telearbeit« verbunden, die erst um das Jahr 2000 abebbte. Vgl. hierzu etwa Jackson/Van der Wielen 1998.

»[Many] users may be considered to be nomads, in that they own computers and communication devices that they carry about with them in their travels as they move between office, home, airplane, hotel, automobile, branch office, etc. [...] We now recognize that access to computing and communications is necessary not only from one's ›home base‹, but also while one is in transit and when one reaches one's destination. [Nomadicity may be defined as] the system support needed to provide a rich set of capabilities and services to the nomad as he moves from place to place in a transparent and convenient form« (Kleinrock 1995, S. 37).

Demnach bewegen sich nomadische Computer physikalisch mit dem Besitzer, unterscheiden sich aber sonst nicht von einem stationären Computer. Eine dynamische Anpassung an bzw. Einbettung in die jeweilige Umgebung wie in Weisers Konzept war nicht vorgesehen. Nomadische Computersysteme sind mit der Verbreitung von Internet und tragbaren PCs zum Normalfall geworden, auch wenn die Herstellung des Netzzugangs in wechselnden Umgebungen im Detail durchaus immer noch seine Tücken haben kann.

Pervasive Computing

Während die Wissenschaftler des PARC unter dem Begriff Ubiquitous Computing eine akademisch-idealistische Langfristvision entwickelten, hat die Industrie in der zweiten Hälfte der 1990er Jahre den Begriff des »Pervasive Computing« geprägt, der für eine pragmatischere Variante von Weisers Vorstellungen steht. Ihren Ursprung hatte das Konzept in einem strategischen Projekt der Firma IBM, in der es darum ging, die sich in den 1990er Jahren abzeichnenden Trends in der Vernetzung und bei der Mobilkommunikation in die Zukunft fortzuschreiben. Das Ergebnis dieser Überlegungen, das von IBM-Chef Lou Gerstner im Frühjahr 1998 auf der Computermesse CeBit in Hannover erstmals der Öffentlichkeit vorgestellt wurde, beschrieb Mark Bregman, der Leiter des Projekts folgendermaßen:

»Pervasive computing is about enabling people to gain immediate access to information and services anywhere, anytime, without having to scrounge for a phone jack. However, while mobility and wireless technology are a big part of it, it's really about making e-business personal. Thanks to the explosive growth of the Internet, people will soon expect to be able to engage in electronic business effortlessly.« (Bregman 1998)

Anders als bei Mark Weiser war das Konzept des Pervasive Computing also nicht mit einer fundamentalen Kritik des Personal Computers verbunden. Ganz im Gegenteil wurde argumentiert, die Grundidee sei ähnlich wie die des Personal Computers, weil der Nutzer in die Lage versetzt werde, jederzeit und überall mit einem beliebigen Endgerät auf seine Daten zugreifen zu können. Der Schlüssel dazu sollte ein verteiltes Netzwerk (wie das Internet) sein, das eine Vielzahl von unterschiedlichen Endgeräten, von denen die Mehrheit keine PCs sein würden, miteinander verband. Gerstner sprach in diesem Zusammenhang von »a billion people interacting with a million e-businesses with a trillion intelligent devices interconnected« (Gerstner 1997, zitiert in Mattern 2003, S. 5).

Auch beim Pervasive Computing geht es also um allgegenwärtige Informationsverarbeitung, allerdings mit dem Ziel, diese durch die Verwendung von (zum Teil) vorhandenen Mobile-Computing-Technologien schon kurzfristig nutzbar zu machen. Dabei standen insbesondere E-Commerce-Szenarien und Web-basierte Geschäftsprozesse im Fokus der Entwickler. Beim Pervasive Computing spielten deshalb zunächst mobile Endgeräte für den Informationszugriff (PDAs, Mobiltelefone etc.), Kommunikationskonzepte (wie WAP und Bluetooth), Techniken zur anwendungsunabhängigen Datenrepräsentation sowie Betriebssoftware für Chipkarten und PDAs genauso eine Rolle wie Middleware für verteilte Systeme und Methoden der Kryptographie. Im industriellen Umfeld spiegelte sich dieser Trend u. a. auch in der steigenden Bedeutung von Portaltechnologien wider, bei denen zusätzlich zum klassischen Zugang über den Web-Browser die drahtlosen, beweglichen Zugangsmedien auch im Geschäftsumfeld an Bedeutung gewinnen. Insgesamt ergaben sich durch (teilweise ungeplante) Integrationseffekte auch ganz neuartige Anwendungsszenarien wie z. B. ortsbezogene Dienste (*location based services*) (Ark/Selker 1999; Hansmann et al. 2001).

Hatte das Pervasive Computing anfangs mehr Ähnlichkeit mit dem Nomadic Computing, so näherte sich die damit verbundene Vision immer mehr der des Ubiquitous Computing an (vgl. etwa BSI 2006). Dies hatte einerseits mit der Wirkungsmächtigkeit von Weisers Konzept zu tun, das allmählich alle anderen Konzepte überformte, andererseits gehören die von der Industrie zunächst angestrebten mobilen Lösungen mittlerweile zum Alltag der mobilen Kommunikation.

Ambient Intelligence

Nach 1999 wurde in Europa der noch weiter reichende Begriff »Ambient Intelligence« popularisiert, der ursprünglich von Emile Aarts von Philips Research vorgeschlagen und schon kurz darauf von der »Information

Society Technologies Advisory Group« der Europäischen Kommission (unter dem Vorsitz von Aarts) aufgegriffen wurde (Aarts/Appelo 1999; ISTAG 2001, 2003). In der Folge wurde Ambient Intelligence im Rahmen des Fünften und Sechsten Forschungsrahmenprogramms zu einem wichtigen Forschungsschwerpunkt im Bereich der Information Society Technologies für die Jahre 2002 bis 2006 mit einem Budget von 35 Millionen EUR (Programm »The Disappearing Computer«).⁶

Zu den Charakteristika der Ambient Intelligence gehören neben der informationstechnischen Durchdringung des Alltags auch Aspekte der Mensch-Maschine-Kommunikation und der künstlichen Intelligenz. Man stellt sich dabei vor, dass eine intelligente Technik dem Menschen ständig unterstützend zur Verfügung steht, diese aber selbst praktisch unsichtbar wird. Dabei sollen Alltagsgegenstände zu aktiven, kommunikationsfähigen Subjekten werden und der dinglichen Welt eine ganz neue Eigenschaft verleihen: Diese wird reaktionsfähig, passt sich den aktuellen Bedürfnissen des Menschen an und steigert damit dessen Leistungsfähigkeit und Lebensqualität (Abb. 2). Dies klingt zunächst stark nach der ursprünglichen Grundidee des Ubiquitous Computing, betont aber, zumindest in der von der Europäischen Kommission verwendeten Lesart, nicht so sehr die technischen Aspekte als vielmehr die Bedürfnisse des einzelnen Nutzers wie auch der Gesellschaft (Europäische Kommission 2003). Mit Blick auf die so genannte Lissabon-Strategie der Europäischen Union aus dem Jahr 2000 sollte Ambient Intelligence die Wettbewerbsfähigkeit des europäischen Wirtschaftsraums fördern, den Übergang zu einer dynamischen Wissensgesellschaft unterstützen und dabei auf die gesellschaftlichen Bedürfnisse reagieren und insbesondere die soziale Kohäsion fördern. Nach der – in typisch floskelhaftem Eurospeak formulierten – Einschätzung der IST Advisory Group hat Ambient Intelligence für Europa ein erhebliches Nutzenpotenzial in Bezug auf

- »Modernising the European social model particularly in terms of: improving civil security; providing new leisure, learning and work opportunities within the networked home; facilitating community building and new social groupings; providing new forms of healthcare and social support; tackling environmental threats; supporting the democratic process and the delivery of public services.
- Improving Europe's economy in terms of: supporting new business processes; increasing the opportunities for tele-working in the networked home.

6 Auch wenn sich das europäische Programm gern von der als zu hardwareorientiert bezeichneten Entwicklung in den USA abgrenzt, knüpft es dennoch auch an aktuelle US-Forschungen an (vgl. etwa. Norman 1998; Denning 2002).

ked home; enhancing mobility and improving all forms of transport; supporting new approaches to sustainable development« (ISTAG 2003, S. 31; vgl. auch Burgelman/Punie 2005).

Abb. 2: Die banale Realität der Philips-Vision: Eine allgegenwärtige Medienwelt. Quelle: Aarts/Marzano 2003



Trotz der erwähnten Unterschiede gibt es in den programmatischen Papieren von Aarts oder der ISTAG keine klare Abgrenzung zu den Begriffen Ubiquitous und Pervasive Computing. Ein gewisser Unterschied dürfte aber die Tatsache sein, dass das Interaktions- oder Nutzungsparadigma (eben nicht nur »Computing« oder »Communication«) bei Ambient Intelligence nicht festgelegt ist. Grob gesprochen ist das Aufgabengebiet durch die intelligente Interaktion von Benutzern mit der jeweiligen Umgebung charakterisiert. Im Vordergrund steht dabei die Intelligenz, die sich in den von dem Anwender verwendeten Zugangsgaräten, in einem Netzwerk, in den zugegriffenen Medien/Informationen oder der Umgebung manifestieren kann.

Alles in allem kann man konstatieren, dass die Unterscheidung der Begriffe Ubiquitous Computing, Pervasive Computing und Ambient Intelligence in der Praxis eher akademischer Natur ist: Gemeinsam ist allen das Ziel einer Unterstützung des Menschen sowie einer durchgängigen Optimierung wirtschaftlicher und sozialer Prozesse durch eine Vielzahl

von in die Umgebung eingebrachten Mikroprozessoren und Sensoren. Zusammenfassend sind sie durch folgende Merkmale gekennzeichnet (Abb. 3):

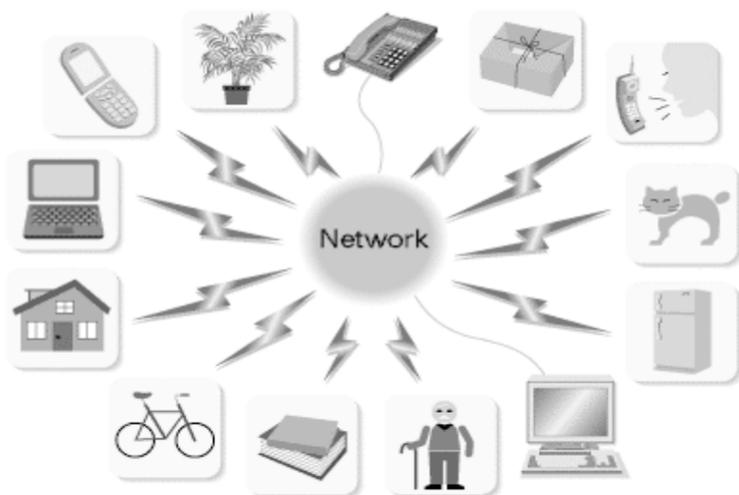
- Einbettung: IuK-Hardware wird immer kleiner und portabler und kann deshalb immer mehr in andere Geräte und Gegenstände des täglichen Gebrauchs eingebettet werden, die nach außen hin nicht mehr den Charakter eines Computers besitzen.
- Vernetzung: IuK-Systeme sind in der Regel miteinander vernetzt, sowohl lokal als auch global, über das Internet, Mobilfunk und neue Netzwerktechnologien.
- Intelligenz und Kontextsensitivität: IuK-Systeme sammeln immer mehr Informationen über ihre Umgebung. Auf dieser Basis treffen sie zunehmend autonome Entscheidungen, die sie selbstlernend verbessern können.
- Modularität: IuK-Systeme sind zunehmend modular aufgebaut und lassen sich mit anderen IuK-Systemen kombinieren. In ihrer spontanen, gemeinsamen Kommunikation und Interaktion sollen sie die limitierte Funktionalität der einzelnen Komponente überwinden und neue synergetische Qualitäten und Funktionalitäten für den Nutzer schaffen.

Herausforderungen allgegenwärtiger, vernetzter und individualisierter Informationssysteme

Was bedeutet es, wenn der Computer als Gerät verschwindet, er eine Symbiose mit den Dingen der Umwelt eingeht und höchstens noch als eine Art unsichtbare Hintergrundassistentz wahrgenommen wird? Angesichts der zu erwartenden Durchdringung vieler Lebensbereiche mit allgegenwärtiger Informationstechnik und den daraus resultierenden wirtschaftlichen und sozialen Auswirkungen stellt sich verstärkt auch die Frage nach den vorhandenen Gestaltungsspielräumen. Ihre Beantwortung ist gleichermaßen ein technisches und ökonomisches wie auch ein politisch-juristisches Problem.

Technisch betrachtet stellt Ubiquitous Computing zwar nur eine neue Stufe im Trend zu mobiler Kommunikation und individualisierter Information dar, dennoch besitzt sie das Potenzial für tiefgreifende Veränderungen für den Einzelnen wie für die Gesellschaft im Ganzen.

Abb. 3: Das Netz als Mittelpunkt des Ubiquitous Computing,
Quelle: NTT Communications



Ubiquitous computing will enable diverse wireless applications, including monitoring of pets and houseplants, operation of appliances, keeping track of books and bicycles, and much more.

Für dieses Veränderungspotenzial zeichnen vor allem zwei zentrale technische Innovationen verantwortlich: zum einen die massiv erhöhten Kapazitäten zur technischen Erfassung und Speicherung alltäglicher Aktivitäten und Interaktionen von Privatpersonen in vielfältigen Ausprägungen sowie über große Distanzen und Zeiträume hinweg, zum anderen die gesteigerten Fähigkeiten zur schnellen Durchsuchung von großen Datenbanken, womit zusätzliche Möglichkeiten zur Erstellung von personenbezogenen Datenprofilen und anderen Formen des Data Mining einhergehen (Bohn et al. 2004). Einer der führenden Experten auf diesem Gebiet hat folgende allgemeine Problembereiche identifiziert, mit denen die Nutzer bzw. Betroffenen von Ubiquitous Computing sehr wahrscheinlich konfrontiert sein werden (Ackerman 2004, S. 430):

- Ein allgegenwärtiges Netzwerk von Anwendungen und Kommunikationen zieht einen *massiven Anstieg bei der Erhebung und Übermittlung personenbezogener Daten* nach sich.
- Die Einführung von biometrischen Verfahren und Wahrnehmungssensoren für bestimmte Anwendungen wird die *Qualität der im Umlauf befindlichen personenbezogenen Daten verändern*.

- Um personalisierte Dienste anbieten zu können, wird durch Ubiquitous Computing-Systeme ein *Großteil des Alltagslebens digital erfasst und gespeichert*.

Freilich gilt, dass sich diese Effekte im Zuge der technischen Entwicklung erst sukzessive entfalten werden. Obwohl sich diese Zukunftsvisionen aus heutiger Warte noch weitgehend wie Science-Fiction ausnehmen, ist eine frühzeitige Auseinandersetzung mit den potenziellen Risiken der Technologie erforderlich, sollen künftige Entwicklungen in gesellschaftlich erwünschte Bahnen gelenkt werden. Weil die grundlegende Technik für die Realisierung der Ubiquitous Computing-Visionen heute noch nicht existiert, gestaltet sich eine Folgenbewertung schwierig. Aus diesem Grund wurden die fiktiven Szenarien im Sinne erdachter, aber paradigmatischer Anwendungsbeispiele systematisch im Sinne eines problemorientierten Leitbildassessments (Grunwald/Grin 1999; Wright et al. 2007) analysiert, wobei normative Elemente stärker betont werden.

Gemeinsam ist vielen Entwürfen einer vernetzten und von Informationstechnik durchdrungenen Welt die Neudefinition der Beziehung zwischen Mensch und Technik. Elemente, die vormals einem autonomen menschlichen Nutzer zugeschrieben wurden, wie zum Beispiel Urteilsvermögen oder rationales Handeln, erscheinen nun als Ausdruck wechselnder Interaktionen des Menschen mit seiner sozialen und dinglichen Umwelt. Die materielle Umgebung erfährt als Bestandteil der menschlichen Identität eine Aufwertung. Sie erscheint nicht länger »neutral«, sondern wird selbst zum Akteur im sozio-technischen System in grundlegender Weise.

Eine solche Auffassung fand mit der Etablierung des Internet einen Bezug zur elektronischen Kommunikation. Beispielfür sind problematische Konzepte zur Übertragung der menschlichen Vernunft in das Internet, die Idee der Intersubjektivität durch vernetzte Informationsbestände oder die Annahme einer Repräsentation des modernen, dezentralen Bewusstseins durch Hypertextstrukturen (Hellige 2000). Auch in der populären Kultur fand der Gedanke einer innigen Verbindung von menschlicher Identität und computerisierter Kommunikationstechnik durch zahlreiche Filme, Romane und Comics Verbreitung. Die digitale Kommunikation stellt sich nach Auffassung von Vordenkern des Cyberspace als quasi natürliche Konstitutionsbasis der menschlichen Identität dar und etabliert eine neue digitale Seinsform. Nicholas Negroponte schrieb beispielsweise: »[I]n der Digitalzeit müssen wir nicht auf eine Erfindung warten – das digitale Leben ist bereits hier und jetzt vorhanden. Und man kann es fast als vererbbar bezeichnen; vererbbar insofern,

als dass jede Generation ein wenig digitaler werden wird als die Generation davor« (Negroponte 1997, S. 280).

Man kann bezweifeln, ob die Entstrukturierung und Entkopplung von der realen Welt tatsächlich durch digitale Kommunikation ausgelöst wird (vgl. Stegbauer 2001). Tatsächlich ist die propagierte enge Verbindung zwischen menschlicher Identitätsbildung und Informationstechnik keine zwangsläufige Entwicklung wie von den Evangelisten des digitalen Lebens gerne apodiktisch behauptet, sondern nur *eine*, durchaus gestaltbare Entwicklung.

Ubiquitous Computing vervollständigt die historische Ausdifferenzierung medialer Interaktionswelten, indem sie auch den Bereich der gegenständlichen Umgebung erfasst. Nicholas Negroponte bezeichnet mit Blick auf eine allgegenwärtige Computertechnik die Welt als »aufgefaltetes Gehirn«. In Szenarien werden Identität und Persönlichkeit gern als sozial situierte Erscheinungen dargestellt und der menschliche Nutzer als grundlegender Teil eines soziotechnischen Netzwerkes eingeführt: »Maintaining existing relationships and creating new ones is an essential feature of human life« (ISTAG 2001: 32).

Sie folgen damit insbesondere postmodernen Theorien etwa von Gilles Deleuze, die die Neigung der menschlichen Selbstwahrnehmung zur Definition durch die Umwelt betonen, auch wenn diese selbst eine künstliche, durch die Informationstechnik geschaffene Umwelt ist. Eine Vielzahl von Mikroprozessoren und Sensoren sorgt in der postmodernen Vorstellung für eine weit gehende Medialisierung der dinglichen Umgebung. Pointiert ausgedrückt kann man sagen, dass die Umgebung durch Ubiquitous Computing selbst zum Programm des neuen Mediums wird.

Auf diese Weise geht Ubiquitous Computing über den bloßen Charakter einer virtuellen Realität hinaus und erhebt den Anspruch eine neue Form von Wirklichkeit zu sein:

»[Ubiquitous Computing] automatisiert die Abbildung der realen Welt der Menschen, Produkte und Betriebsmittel in die virtuelle Welt des Internets, [...] und ersetzt damit den Menschen als Mediator zwischen realer und virtueller Welt.« (Fleisch et al. 2003: 12)

Nach der Auffassung einflussreicher Denkschulen (von George Herbert Mead über Max Scheler bis zum postmodernen Konstruktivismus) entsteht Wirklichkeit aber auf der Basis individueller Widerstandserfahrungen (vgl. Berger/Luckmann 1980; Di Blasi 2003). Solche Widerstandserfahrungen werden in Visionen des Ubiquitous Computing antizipiert und sollen explizit abgebaut werden. Widerstandserfahrungen werden zwar stets im Zuge der Diffusion neuer Technologien vermindert, aller-

dings wurde bislang der Widerstandsabbau explizit an die Technik delegiert und nicht bereits in der Lebenswelt angelegt. Ein Teil des Ubiquitous Computing-Leitbildes ist der »Kontextsensitivität« des Systems verpflichtet. Hier wird die veränderte Bedeutung von Wirklichkeit und Widerstandserfahrung deutlich. Die Kontextsensitivität des Menschen liegt gerade darin, aus der Vielfältigkeit der Welt seine individuelle Wirklichkeit zu konstruieren. Damit technische Systeme kontextsensitiv werden, müssen die einschlägigen Handlungsumgebungen bereits dekontextualisiert sein. Denn in der Vielfältigkeit ihrer Merkmale sind sie für die technischen Systeme nicht greifbar, sondern nur bezüglich ausgewählter, herausgestellter Merkmale. Sie müssen bereits vorab explizit gemacht und mögliche Interaktionen strukturiert werden.

In den Szenarien, wie sie von der IST Advisory Group (ISTAG 2001) oder im Rahmen des deutschen Futur-Prozesses (BMBF 2003) entwickelt wurden, wird die Individualität der Nutzer in besonderer Weise hervorgehoben. Insbesondere in der Innovationspolitik wird in diesem Zusammenhang der Begriff der »personalisierten Interaktionswelten« benutzt, er beschreibt die Entfaltung der Netzwerkorganisation von individuellen Nutzungsanforderungen aus. Anwendungen des Ubiquitous Computing, die die Herstellung interaktiver Beziehungen unterstützen, sollen nach diesem Verständnis Alternativen zu technikgetriebenen Innovationsmodellen bilden. Anwendungen wie beispielsweise das intelligente Haus oder der persönliche Softwareagent sind nach dem Bild eines Dieners an die Bedürfnisse ihrer Besitzer angepasst. Ubiquitous Computing ist in diesen Szenarien ein zusätzlicher Baustein zur weiteren Individualisierung des Mediengebrauchs, wobei die Erhaltung von Autonomie und Individualität der Nutzer gegenüber der Technik durch eine perfektionierte Adaptation an seine Bedürfnisse im Vordergrund steht (ISTAG 2001, S. 1).

Die Integration von vernetzten Mikroprozessoren, Sensoren und Software-Anwendungen in unsere alltägliche Lebenswelt gibt dieser in der Wahrnehmung den Charakter eines scheinbar sozialen Gegenübers. Es ist allerdings unklar, ob Kommunikation in Form einer »Ansprache« durch intelligente Gegenstände tatsächlich eine quasi-soziale Interaktion im Sinne von Reeves und Nass ergibt (vgl. Kiesler/Sproull 1997).

Dennoch ist die Annahme begründet, dass Ubiquitous Computing durch seine Allgegenwärtigkeit eine Art Öffentlichkeit schafft, nach der die Nutzer ihr Verhalten ausrichten. Gerade aufgrund der Vernetzung digitaler Kommunikation, also aufgrund ihrer Reichweite in den öffentlichen Bereich (das Internet) oder der Weitergabe von Daten an intelligente Gegenstände (Smart Objects), aber auch aufgrund einer Undurchschaubarkeit bei der Verwendung von personenbezogenen Daten, kann andererseits in einer intelligenten Umgebung schnell ein mehr oder weniger

diffuses Gefühl des Beobachtetwerdens entstehen. Aus dem Empfinden, durch eine allgegenwärtige Computertechnik mit einem sozialen Gegenüber konfrontiert zu sein, kann es so zur Selbstdisziplinierung oder Selbstinszenierung des Nutzers in Hinblick auf diesen Konformitätsdruck kommen (Bohn et al. 2004).

Ubiquitous Computing soll einen intuitiven Zugang zu den Dienstleistungen der Informationsgesellschaft ermöglichen. Im Rahmen der Anwendungsentwicklungen ist die Vorstellung selbstständig kommunizierender Computer oder auch einer engen, gleichsam symbiotischen Mensch-Maschine-Interaktion vorherrschend. In einer »smarten«, informationstechnisch erfassten Umgebung wird suggeriert, die Dinge selbst gäben über sich Auskunft. Die »intelligente Umgebung« erscheint so als eine Dopplung der natürlichen Umwelt. Trotzdem impliziert eine Welt intelligenter Gegenstände häufig unerkannt eine Prägung der dinglichen Umgebung mit Informationen und Handlungsdispositionen, die bereits durch andere Menschen vorstrukturiert, interpretiert und bewertet wurden. Die Prägung entspricht verschiedenen Adaptionen wie zum Beispiel dem angenommenen oder erwünschten Nutzerverhalten. Die erstrebten Rationalisierungs- und Dienstleistungseffekte sind jedoch abzuwägen gegen den Wegfall produktiver Widerstandserfahrungen, die für die Herausbildung individueller Identität sowie der Entwicklung und Anpassung individueller Kompetenzen unverzichtbar sind.

Ubiquitous Computing ruft aber nicht nur eine stärkere Vorstrukturierung des Handlungsraums, sondern auch des Erfahrungsraums hervor. An jeder Stelle treffen Nutzer moderner Informationstechniken auf einen Bestandteil des Gesamtsystems, ob in einem smart home, dem intelligenten Büro oder dem virtuellen Fitnessclub. Wie in der Parabel vom Hasen und Igel, ist das so genannte »Evernet« (Accenture/CERIAS 2001) immer schon da. Ubiquitous Computing macht die Welt in einem hohen Maße verfügbar und kontrollierbar. Es integriert das Prinzip der Nützlichkeit nicht nur in die Gegenstände selbst, sondern funktionalisiert sie als Bausteine für einen umfassend zweckrationalen Zusammenhang. Dies hat Konsequenzen für den Umgang des Menschen mit sich selbst, den Umgang der Menschen miteinander und mit der dinglichen Welt nicht zuletzt weil Selbstzwecklichkeit, Freiheit und Kreativität Selbstzuschreibungen des Menschen sind, die seine Individualität und den Begriff der Menschenwürde begründen.

Eine Gesellschaft, in der Individualität und Persönlichkeit durch äußere Erfahrungswelten und soziale Bindungen definiert werden, steht vor der Aufgabe, diese Außenwelt auch in Hinblick auf das Fremde und Unverfügbare vielseitig zu gestalten. Nur so kann sichergestellt werden, dass Individuen dauerhaft ihre Fähigkeit zu unabhängiger, spontaner

Reflexion und eigenständiger Kritik entwickeln und nutzen. Individualität liegt nicht nur im Menschen selbst, sondern erfordert auch Freiheiten von einem omnipräsenten informationstechnisch vorstrukturierten Erfahrungsraum.

Literatur

- Aarts, E./Appelo, L. (1999): »Ambient Intelligence: Thuisomgevingen van de toekomst«. In: IT Monitor 9/1999, S. 7-11.
- Aarts, E./ Marzano, St. (Hg.) (2003): »The New Everyday: Views on Ambient Intelligence«. Rotterdam.
- Accenture/CERIAS (2001): »CERIAS Security Visionary Roundtable: Call to Action«. West Lafayette, IN.
- Ackerman, M. S. (2004): »Privacy in Pervasive Environments: Next Generation Labelling Protocols«. In: Personal and Ubiquitous Computing 8, 6, S. 430-39.
- Ark, W. S./Selker, T. (1999): »A Look at Human Interaction with Pervasive Computers«. In: IBM Systems Journal 38. Jg., H. 4, S. 504-507.
- Bell, G. (2006): »Satu Keluarga, Satu Komputer (One Home, One Computer): Cultural Accounts of ICTs in South and Southeast Asia«. In: Design Issues 22, 2, S. 35-55.
- Berger, Th. L./Luckmann, T. (1980): »Die gesellschaftliche Konstruktion der Wirklichkeit: Eine Theorie der Wissenssoziologie«. Frankfurt.
- BMBF (Bundesministerium für Bildung und Forschung) (2003): »Futur: Der deutsche Forschungsdialog: Eine erste Bilanz«. Berlin.
- Bohn, J./Coroama, V./Langheinrich, M. et al. (2004): »Living in a World of Smart Everyday Objects – Social, Economic, and Ethical Implications«. In: Journal of Human and Ecological Risk Assessment 10, 5, S. 763-786.
- Bregman, M. (1998): »The Convenience of Small Devices: How Pervasive Computing Will Personalize E-Business (Interview with Mark Bregman)«. http://domino.watson.ibm.com/comm/wwwr_thinkresearch.nsf/pages/bergman398.html (zuletzt abgerufen am 16.02.2007).
- BSI (Bundesamt für Sicherheit in der Informationstechnik) (2006): »Pervasive Computing: Entwicklungen und Auswirkungen«. Ingelheim.
- Burgelman, J.-C./Punie, Y. (2005): »Information, Society and Technology«. In: »True Visions: The Emergence of Ambient Intelligence«, hg. v. Aarts, E./Encarnaçao, J. L., Berlin, Heidelberg, New York. S. 17-33.
- Card, St. K./Moran, Th. P./Newell, A. (1983): »The Psychology of Human-Computer Interaction«. Hillsdale, N J.
- Coy, W./Nake, F./Pflüger, J.-M. et al. (Hg.) (1992): »Sichtweisen der Informatik«. Braunschweig, Wiesbaden.

- Denning, P. J. (Hg.) (2002): »The Invisible future: The seamless integration of technology into everyday life«. New York.
- Di Blasi, L. (2003): »Die Räume der Kybernetik«. In: Trans - Internet-Zeitschrift für Kulturwissenschaften H. 15. http://www.inst.at/trans/15Nr/10_4/blasi_Luca15.pdf
- Europäische Kommission (2003): »Arbeitsprogramm zum spezifischen Programm im Bereich der Forschung, technologischen Entwicklung und Demonstration: »Integration und Stärkung des Europäischen Forschungsraums«. Spezifische Maßnahmen für die wissenschaftliche Unterstützung der Politik im Rahmen der »Unterstützung der Politiken und Planung im Vorgriff auf den künftigen Wissenschafts- und Technologiebedarf (SSP Call 3)«. Brüssel.
- Fleisch, E./Mattern, F./Billinger, St. (2003): »Betriebswirtschaftliche Applikationen des Ubiquitous Computing: Beispiele, Bausteine und Nutzenpotenzial«. In: HMD. Theorie und Praxis der Wirtschaftsinformatik, H. 229, S. 5-15.
- Fleisch, E./Michahelles, F. (2007): »Messen und Managen – Bedeutung des Ubiquitous Computing für die Wirtschaft«. In: »Die Informatisierung des Alltags: Leben in smarten Umgebungen«, hg. v. F. Mattern, Berlin, Heidelberg, New York, S. 145-159.
- Floyd, Chr./Züllighoven, H./Budde, R./Keil-Slawik, R. (Hg.) (1992): »Software Development and Reality Construction«. Berlin, Heidelberg, New York.
- Friedewald, M. (1999): »Der Computer als Werkzeug und Medium: Die geistigen und technischen Wurzeln des Personal Computers«. Berlin, Diepholz.
- Grunwald, A./Grin, J. (Hg.) (1999): »Vision Assessment: Shaping Technology in 21st Century Society«. Berlin, Heidelberg, New York.
- Hansmann, U./Merk, L./Nicklous, M. S. et al. (2001): »Pervasive Computing Handbook«. Berlin, Heidelberg, New York.
- Heidegger, M. (2001): »Sein und Zeit«. 18. Aufl., Tübingen.
- Hellige, H. D. (2000): »Weltbibliothek, Universalenzyklopädie, Worldbrain: Zur Säkulardebatte über die Organisation des Wissens«. In: Technikgeschichte 67, 4, S. 303-329.
- ISTAG (Information Society Technologies Advisory Group) (2001): »Scenarios for Ambient Intelligence in 2010«. Luxemburg.
- ISTAG (Information Society Technologies Advisory Group) (2003): »Ambient Intelligence: From Vision to Reality. For participation – in society and business«. Luxemburg.
- Jackson, P. J./Van der Wielen, J. M. (Hg.) (1998): »Teleworking: international perspectives: From telecommuting to the virtual organisation«. London, New York.

- Kay, A. C. (1991): »Computer Networks and Education«. In: *Scientific American*, 265, 3, S. 138-148.
- Kiesler, S./Sproull, L. (1997): »Social responses to »social« computers«. In: »Human values and the design of technology«, hg. v. B. Friedman, Stanford, S. 191-199.
- Kleinrock, L. (1995): »Nomadic Computing – an Opportunity«. In: *ACM Computer Communication Review*, 25, 1, S. 36-40.
- Laurel, B. (1993): »Computers as Theatre«. Reading, MA, u.a.
- Licklider, J. C. R. (1960): »Man-Computer Symbiosis«. In: *IRE Transactions on Human Factors in Electronics*, 1, 1, S. 4-11.
- Lie, E. (2005): »Ubiquitous Network Societies: The Case of the Republic of Singapore«. Paper prepared for the ITU Workshop on Ubiquitous Network Societies, 6.-8. April 2005, Genf.
- Mattern, F. (2003): »Vom Verschwinden des Computers – Die Vision des Ubiquitous Computing«. In: »Total vernetzt«, hg. v. F. Mattern, Berlin, Heidelberg, New York, S. 1-41.
- Negroponte, N. (1995): »Total Digital: Die Welt zwischen 0 und 1 oder die Zukunft der Kommunikation«. Gütersloh.
- Norman, D. A. (1998): »The Invisible Computer. Why good products can fail, the personal computer is so complex, and the information appliances are the solution«. Cambridge, MA, London.
- Raskin, J. (1994): »Intuitive equals Familiar«. In: *Communications of the ACM*, 37, 9, S. 17.
- Reeves, B./Nass, C. I. (1996): »The media equation: how people treat computers, televisions, and new media like real people and places«. Stanford.
- Remagnino, P./Foresti, G. L./Ellis, T. (Hrsg.) (2005): »Ambient Intelligence: A Novel Paradigm«. Boston.
- Reynolds, T./Kelly, T./Jin-Kyu, J. (2005): »Ubiquitous Network Societies: The Case of the Republic of Korea«. Paper prepared for the ITU Workshop on Ubiquitous Network Societies, 6.-8. April 2005, Genf.
- Rheingold, H. (1994): »PARC is back!«. In: *Wired* 2, 2, S. 90-95.
- Srivastava, L. (2005): »Ubiquitous Network Societies: The Case of Radio Frequency Identification«. Paper prepared for the ITU Workshop on Ubiquitous Network Societies, 6.-8. April 2005, Genf.
- Star, S. L. (2002): »Infrastructure and ethnographic practices«. In: *Scandinavian Journal of Information Systems*, 24, 2, S. 107-122.
- Stegbauer, Chr. (2001): »Grenzen virtueller Gemeinschaft: Strukturen internetbasierter Kommunikationsforen« Wiesbaden.
- Suchman, L. A. (1985): »Plans and Situated Actions: The problem of human-machine communication«. PARC Tech. Rep. ISL-6, Palo Alto.
- Suchman, L. A./Trigg, R. H. (1991): »Understanding Practice: Video as a Medium for Reflection and Design«. In: »Design at Work: Cooperative

- Design of Computer Systems«, hg. v. J. Greenbaum/M. Kyng, Mahwah, N J, S. 65-90.
- Tesler, L. G. (1991): »Networked Computing in the 1990s«. In: Scientific American, 265, 3, S. 84-91.
- Turkle, S. (1984): »Die Wunschmaschine: Vom Entstehen der Computerkultur«. Reinbek.
- van Raan, A. F. J. (2004): »Sleeping Beauties in Science«. In: Scientometrics, 59, 3, S. 467-472.
- Want, R./Schilit, B. N./Adams, N. I. et al. (1995): »An Overview of the Partab Ubiquitous Computing Experiment«. In: IEEE Personal Communications, 2, 6, S. 28-43.
- Wehner, J./Rammert, W. (1990): »Zum Stand der Dinge: Die Computerwelt und ihre wissenschaftliche Beobachtung«. In: »Computerwelten - Alltagswelten. Wie verändert der Computer die soziale Wirklichkeit?«, hg. v. W. Rammert. Opladen, S. 225-238.
- Weiser, M. (1991): »Computer im nächsten Jahrhundert«. In: Spektrum der Wissenschaft, November 1991, S. 92-101 (Original: »The Computer for the 21st Century«. In: Scientific American, 265, 3, S. 94-104).
- Weiser, M. (1993): »Some Computer Science Issues in Ubiquitous Computing«. In: Communications of the ACM, 36, 7, S. 75-85.
- Weiser, M./Brown, J. S. (1997): »The Coming Age of Calm Technology«. In: »Beyond Calculation: The Next Fifty Years of Computing«, hg. v. P. Denning/R. M. Metcalfe. New York, S. 75-85.
- Weiser, M./Gold, R./Brown, J. S. (1999): »The Origins of Ubiquitous Computing Research at PARC in the late 1980s«. In: IBM Systems Journal, 38, 4, S. 693-696.
- Weizenbaum, J. (1966): »ELIZA – a computer program for the study of natural language communication between man and machine«. In: Communications of the ACM, 9, 1, S. 36-45.
- Weizenbaum, J. (1990): »Die Macht des Computers und die Ohnmacht der Vernunft«. 8. Aufl., Frankfurt.
- Wright, D./Gutwirth, S./Friedewald, M./Vildjiounaite, E./Punie, Y. (Hg.) (2008): »Safeguards in a World of Ambient Intelligence«. Dordrecht.

**DIE INTERAKTION DES MENSCHEN
MIT SEINER INTELLIGENTEN UMGEBUNG
THE HUMAN-ENVIRONMENT-INTERACTION (HEI)**

JOSÉ L. ENCARNAÇÃO, GINO BRUNETTI, MARION JÄHNE

Ein Trend der Informations- und Kommunikationstechnologie ist ihre zunehmende Integration in alltägliche Gebrauchsgegenstände, so dass diese zu mobilen und multimedialen Endgeräten werden. Erhalten diese autonomen, eingebetteten Systeme die Möglichkeit, sich zu vernetzen, ihre Umgebung über eine reiche Sensorik wahrzunehmen und über verschiedenste Aktuatoren auf sie zu reagieren, entsteht eine Umgebungsinelligenz, die Ambient-Intelligence. Diese bedarf radikal neuer multimodaler Interaktionstechniken, um sie ihren Anwendern nützlich zu machen. Dieser Beitrag behandelt die wichtigsten Aspekte, die in dieser neuen Form der Interaktion des Menschen mit seiner intelligenten Umgebung eine Rolle spielen.

Einleitung

Zwanzig Jahre nach der durch den Personal Computer ausgelösten informationstechnologischen Revolution stehen wir vor einem neuen Umbruch. Kleinste, eingebettete, autonome, sich spontan vernetzende und drahtlos kommunizierende elektronische Geräte, die nicht mehr nur über Tastatur und Maus von ihrem Anwender bedient werden, sondern über elektronische, optische, akustische, magnetische, chemische, biometrische, etc. Sensoren ihre Umgebung wahrnehmen und zunehmend nicht mehr über Bildschirme, sondern über Aktuatoren wie Mikrocontroller, Überwachungs- und Steuerungsgeräte, Motoren, etc. auf Anwender reagieren, im Hintergrund unsichtbar operieren und ihre Dienste immer und überall zur Verfügung stellen.

Die nächste Generation der Informationstechnologie, die sich beispielsweise in Gebrauchsgegenstände unsichtbar als mobiles, drahtloses, multimodales und multimediales Endgeräte integriert und mit der alltägli-

chen Umgebung vereinigt, um auf diese Weise dem Menschen dienliche intelligente Umgebungen zu realisieren, läutet das Zeitalter der Ambient-Intelligence (AmI) ein (Aarts/Marzano 2003). Ihre Potenziale liegen in der Realisierung einer Umgebungsintelligenz, die auf die Gegenwart des Menschen, seine Gewohnheiten, Absichten, Bedürfnisse und sogar Emotionen reagiert. Radikal neue Formen einer Informationsgesellschaft sind die Konsequenz.

Ein wichtiger Aspekt der zukünftigen Mensch–Computer–Interaktion ist die Art und Weise, wie Ambient-Intelligence uns darin unterstützt, unsere persönliche Umgebung, unseren AmI-Raum, effizient zu nutzen. Bisher war die Mensch–Computer–Interaktion ein Prozess, bei dem der Mensch den Computer bediente, wozu er dessen Möglichkeiten kennen und seine Anwendung erlernen musste. Dies führte dazu, dass beispielsweise weniger Qualifizierte, Ältere und Personen mit besonderen Bedürfnissen häufig von den bisherigen Errungenschaften der Informationsgesellschaft ausgeschlossen waren. Bei der Ambient-Intelligence ergibt sich ein völlig neuer Kommunikationsprozess. Die intelligente Umgebung nimmt den Menschen wahr, kennt seine Ziele und Bedürfnisse und reagiert auf multimodale Anweisungen. Die Bedienung der intelligenten Umgebung wird so selbstverständlich wie das Öffnen eines Wasserhahns oder das Einschalten des Lichtes, wobei es nicht notwendig sein wird, im Detail zu verstehen, wie sich die einzelnen Geräte verständigen, um ihren Dienst zu realisieren. Vergleichbar damit, dass es ebenso wenig notwendig ist, im Einzelnen zu wissen, wie das Wasser bis zum Wasserhahn oder der Strom bis zur Glühbirne kommt. Dieser Beitrag behandelt die wichtigsten Aspekte, die in dieser aufkommenden Mensch–Umgebung–Interaktion eine Rolle spielen.

Mensch-Computer-Interaktion heute

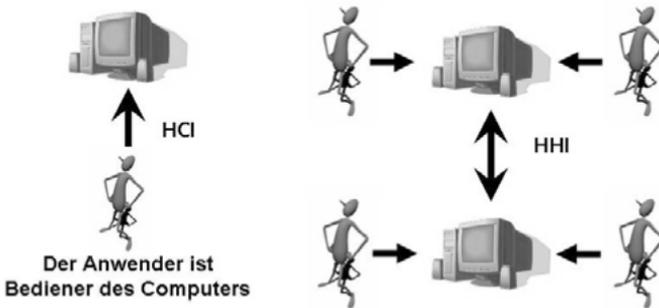
Bei der Mensch-Computer-Interaktion (Human-Computer-Interaction, kurz: HCI) erteilt der Anwender die Befehle, er ist der Bediener. Der Benutzer fordert Aktionen und Services von der Umgebung an, indem er Funktionen aufruft, Parameter bestimmt, Menüs aktiviert, usw. Durch Kombination verschiedener HCI's können zusätzlich verschiedene Arten der computergestützten Mensch-Mensch-Interaktion (Human-Human-Interaction, kurz: HHI) wie beispielsweise Email, Chat, Internet-Telephonie und Video-Konferenz implementiert werden (vgl. Abb. 1).

Dabei fungiert der Computer in der heute typischen Mensch-Computer-Interaktion sowohl als Ein- als auch als Ausgabemedium. Forschung zur Verbesserung dieser Interaktion umfasst heutzutage Themen wie die

Multimodalität in der Eingabe (Sprache, Mimik, Eingabegeräte mit sechs Freiheitsgraden für 3D-Interaktion etc.), Multimodalität in der Ausgabe (2D & 3D, Bilder, Video, Audio, Kraftrückkopplungsein- und -ausgabegeräte etc.) und die Integration von computergenerierten Realitäten, um beispielsweise mit rein virtuellen, erweiterten, gemischt realen/virtuellen oder auch simulierten Realitäten zu interagieren und schließlich auch die Entwicklung von einfach zu bedienenden, intuitiven Schnittstellen für die Mensch-Computer-Interaktion.

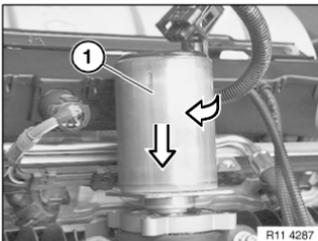
Abb. 1: Links: Mensch-Computer-Interaktion (HCI).

Rechts: Computerbasierte Mensch-Mensch-Interaktion (HHI).



Die verwendeten Technologien für die Ausgabe spannen dabei den Ausgaberaum auf, der sich zum Beispiel bei einem normalen PC auf Bildschirm und Lautsprecher beschränkt. Stehen die darzustellenden Informationen in Relation zu »Dingen« unserer realen Welt (oder Objekten, die nicht Teil des Ausgaberaumes sind), so müssen diese Relationen explizit dargestellt werden, denn die reale Welt ist nicht Teil des Ausgaberaumes. Folgendes Beispiel soll dies verdeutlichen:

Abb. 2: Ausschnitt aus einer Reparaturanleitung (© BMW AG).



Installation:

Replace sealing ring on spacer.

Install servomotor (1) and screw in up to spacer.

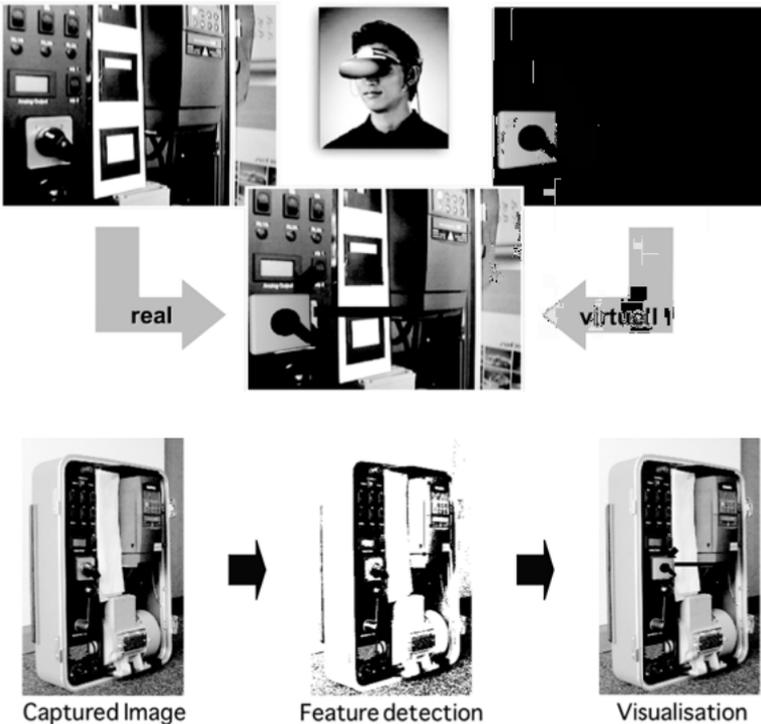
Rotate servomotor (1) into correct installation position.

Install and tighten down screws.

Reparaturanleitungen für Kraftfahrzeuge beschreiben Schritt für Schritt die Aktionen, die ein Mechaniker ausführen muss, um eine Reparatur durchzuführen. Dabei kommen Textbeschreibungen der Aktionen und passende Bilder zum Einsatz. Diese Bilder können reale Fotografien sein oder sie sind aus digitalen Modellen erzeugt worden. Die gesamte Reparaturanleitung ist dabei als Web-Seite oder PDF-Dokument mit Hyperlinks realisiert (Abb. 2).

Da der Ausgaberaum das reale Fahrzeug nicht beinhaltet, werden die Relationen zur realen Welt explizit modelliert, hier in Form von Bildern. Der Anwender eines solchen Systems muss zusätzliche kognitive Leistung erbringen, um diese expliziten Relationen aufzulösen und zu übertragen. Dies ist bei komplexen Systemen fehleranfällig und zeitaufwändig. Um dem zu begegnen, erlaubt beispielsweise die Technologie der Erweiterten Realität (Augmented-Reality, kurz: AR) die Erweiterung des Ausgaberaumes auf die reale Welt.

Abb. 3: Oben: Grundprinzip Augmented-Reality.
Unten: Vision-basiertes Tracking.



AR ermöglicht Relationen implizit darzustellen, indem computergenerierte Informationen lagerichtig in die reale Welt eingeblendet werden. Dazu trägt der Anwender eine spezielle Brille, die es erlaubt, sowohl die reale Welt zu sehen als auch computergenerierte Informationen einzublenden (siehe Abb. 3, oben). Über eine Kamera an der Brille wird die reale Welt aufgenommen, in Echtzeit werden Bildmerkmale identifiziert, mit verfügbaren Informationen über die reale Welt abgeglichen (artifizielle Markierungen, 3D-Geometrien, Referenzbilder) und die Position und Blickrichtung des Anwenders bestimmt (Abb. 3, unten). Neueste Forschungsergebnisse, die zeigen, dass für die Erkennung reale, vorhandene Merkmale genutzt werden können, sind u.a. in Bleser et al. (2005, 2006) beschrieben.

Für das skizzierte Beispiel bedeutet dies, dass unter Verwendung von AR-Technologie die Beschreibung der auszuführenden Aktionen direkt mit dem realen Objekt in Verbindung gebracht werden kann (Abb. 4). Die Relation zwischen der computergenerierten Information und dem Objekt wird implizit angezeigt; die Information wird im richtigen Kontext angezeigt. Das Verständnis wird zusätzlich durch die Verwendung graphischer Animationen anstatt Text wesentlich verbessert. Hinzu kommt, dass Graphik per se multilingual ist und damit aufwändige Übersetzungen und die damit verbundenen Kosten entfallen. Außerdem wird die Fehlerhäufigkeit durch die direkte Darstellung der Beziehung Aktion zu Objekt stark gesenkt (Weidenhausen et al. 2003).

Abb. 4: Beispiele einer Augmented-Reality basierten Reparaturanleitung (© BMW AG).



Das Beispiel der Reparaturanleitung ist nicht praxisfremd; vielmehr evaluiert die Automobilindustrie zurzeit den Einsatz von AR hierfür sehr intensiv. Man weiß, dass in Zukunft die Komplexität von Fahrzeugen immer mehr zunehmen wird und es nicht mehr möglich sein wird, einen Mechaniker in allen Aspekten adäquat zu schulen. Die Erstellung von

Servicehandbüchern wird ebenfalls immer problematischer, da die Zeit von der Verfügbarkeit des finalen Fahrzeuges bis zum Anlaufen der Produktion immer kürzer wird. AR kann hier eine entscheidende Hilfe sein.

Auch in der Produktentwicklung ist AR eine hilfreiche Technologie. So können kleine Designänderungen von bestehenden Fahrzeugen direkt in die realen Fahrzeuge eingeblendet werden. Außerdem können sehr einfach Varianten untersucht oder Unterschiede zwischen realen Daten und den zugrunde liegenden virtuellen Daten durch Überlagerung sichtbar gemacht werden.

Das Beispiel in Abb. 5 zeigt eine Anwendungsmöglichkeit im Marketingbereich. Die reale Szene, hier ein Automobil, ist auf Video aufgenommen, wobei eine virtuelle Felge derart ergänzt wird (Augmented Video), dass sie der Bewegung des Fahrzeugs lagerichtig folgt. Der Benutzer kann aus den vorhandenen Felgen frei wählen und erhält auf diese Weise einen visuellen Eindruck vor seiner Kaufentscheidung.

Die Technologie Augmented-Reality ist natürlich nicht nur auf den Einsatz in der Automobilindustrie beschränkt, sondern kann beispielsweise auch im Tourismus effektiv eingesetzt werden, in dem z. B. in die reale Welt Bauwerke früherer Kulturen eingeblendet werden, die heute zerstört sind (Stricker et al. 2002). Es gibt darüber hinaus noch viele andere Anwendungsbereiche, so z.B. in der Medizin (Khan et al. 2005).

*Abb. 5: Virtuelle Felgenvarianten am realen Fahrzeug (© Audi).
Links: ohne Einblendung; rechts: mit Einblendung.*



Metapherübergang: Vom Schreibtisch zur Ambient-Intelligence (AMI)

Ein Widerspruch in den heutigen Trends zur Interaktion mit technischen Geräten ist offensichtlich: Je mehr Funktionen und Möglichkeiten Geräte dem Menschen eröffnen, desto mehr Funktionen bleiben unbenutzt. Dies hat vielschichtige Gründe: Einerseits sind viele Funktionen nicht den

Bedürfnissen der Benutzer angepasst. Andererseits fühlen sich viele Benutzer angesichts einer solchen Vielzahl an Möglichkeiten überfordert. Der Benutzer ist zuständig aus den vorhandenen Funktionen, Menüs und Optionslisten herkömmlicher Geräte und Applikationen diejenigen herauszufinden, miteinander zu kombinieren bzw. hintereinander zu verknüpfen, die am ehesten zu dem ursprünglichen Benutzerziel passen. Kein Wunder, dass vor lauter Benutzungsoptionen oftmals das eigentliche Handlungsziel aus den Augen verloren wird, oder dass das Ausprobieren von Funktionen die eigentliche Handlung ersetzt. Ambient-Intelligence führt hier den Übergang von der Orientierung auf Funktionen zur Orientierung auf Benutzerziele ein. Dieser Metapherübergang verändert die Sichtweise auf ein Gerätesystem auf die Art der Unterstützung, die es einem Benutzer bietet und weg von der Notwendigkeit für den Benutzer, einzelne Bedienschritte und Funktionen erlernen zu müssen. Nicht mehr der Benutzer soll sich einem System anpassen müssen, sondern ein System erfüllt die Benutzerziele auf Basis der eigenen Adaptivität.

Es stellen sich drei wichtige technische und wissenschaftliche Fragen zur Realisierung dieser Ausprägung von Ambient-Intelligence (vgl. Aarts/Encarnação 2006). Zuerst ist die Interaktion des Benutzers mit seiner Umgebung zu betrachten. Denn sobald der Benutzer nicht mehr gezwungen ist, mit Geräten einzeln zu interagieren, um deren Funktionen zu bedienen, muss es ihm möglich sein, natürlichsprachig zu interagieren. Die Interaktion wird sich somit von einer rein funktions- und geräteorientierten Interaktion zu einer zielorientierten Interaktion verändern. Automatisch muss die Umgebung in der Sprache des Benutzers kommunizieren (z.B. durch Gestikerkennung und Spracherkennung) und nicht mehr umgekehrt. Danach ist die selbstorganisierte Kooperation und Kommunikation von Geräteensembles bereitzustellen, die gemeinsam die vom Benutzer gestellten Aufträge, Ziele und erkannten Bedürfnisse ausführen. Es kann dabei in Zukunft von einem Benutzer nicht mehr erwartet werden, dass er eine Vielzahl von Geräten organisiert und orchestriert. Im Gegenteil müssen die Geräte vielmehr selbst geeignete Strategien zur Selbstorganisation und zur Sicherung von Interoperabilität und Kooperation besitzen und bereitstellen können. Diese Selbstorganisation und Kooperation hat hierbei ad-hoc und autonom zu geschehen, damit der Benutzer vollständig von solchen Konfigurations- und Koordinations-tätigkeiten entlastet werden kann. Zuletzt muss intelligentes Verhalten eines Geräteensembles in Bezug auf die aktuelle Umgebung, den aktuellen Handlungskontext und die erfolgten Benutzerinteraktionen realisiert werden. Hier muss neben den rein kommunikativen und kooperativen Fähigkeiten der Geräte auch erreicht werden, dass sie sich der momenta-

nen Umgebungssituation und der Situation des Benutzers bewusst sein können und daraus intelligente Schlüsse ziehen.

Erfolge in diesen wissenschaftlichen und technischen Herausforderungen machen den durch Ambient-Intelligence eingeleiteten Metapherübergang möglich. Ambient-Intelligence erweitert dabei die von den Forschungsrichtungen »Ubiquitous Computing« und »Pervasive Computing« (allgegenwärtiger und universeller Rechneinsatz) erarbeiteten Grundlagen, die vorsehen, dass Informationstechnologie in jede mögliche Art von Alltagsgegenständen eingebaut sein kann. Solche Informationseinheiten können damit sogar vor dem Benutzer verborgen und damit gewissermaßen unsichtbar sein. Ambient-Intelligence verlagert diese Technologie-zentrierte Sichtweise hin zu einer Benutzer- und Szenario-orientierten Sichtweise und wird damit die wissenschaftlichen und industriellen Fragestellungen der kommenden Jahre maßgeblich beeinflussen. Die IST Advisory Group beschreibt eine derartige AmI-Umgebung als

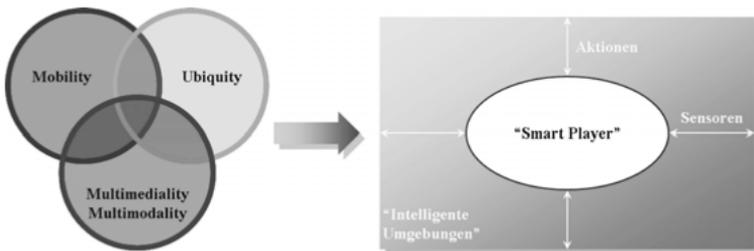
»[...] sensitiv zur Anwesenheit von Kreaturen (Personen, Personengruppen oder auch Tieren), deren Aktivitäten sie unterstützt. Sie erinnert sich an deren Verhalten und antizipiert dieses. Die menschlichen und physischen Entitäten – oder deren Cyber-Repräsentanten – und die Dienste teilen sich diese Umgebung, die die physische und virtuelle Welt, den AmI-Raum, umfassen. Dieser Raum muss derart konstruiert sein, dass er ein vorhersagbares Verhalten hat, dass durch ihn Dienste angeboten werden können und dass er komplizierte »viele-zu-viele« Beziehungen handhaben kann. Der AmI-Raum kann als die Integration von Funktionen auf lokaler Ebene zwischen verschiedenen Geräteumgebungen gesehen werden, welcher einen direkten, natürlichen und intuitiven Dialog des Benutzers mit Anwendungen und Diensten ermöglicht, sowohl verschiedene Interaktionsumgebungen als auch den Cyberraum umfasst und die Organisation und Verarbeitung von Wissen und Inhalten ermöglicht.« (ISTAG 2001)

Ambient-Intelligence setzt somit den Benutzer in das Zentrum aller technischen Innovationen. Die Umgebung eines AmI-Raumes – und damit alle in ihr befindlichen technischen Geräte – ist somit in der Lage, Informationen über die gegenwärtige Situation, die Interaktion des Benutzers und dessen Möglichkeiten zu sammeln und in einen vernünftigen Kontext zu setzen. In einem zweiten Schritt ist es dem verteilten AmI-Raum möglich, die gegenwärtige Situation dahingehend zu interpretieren, dass mögliche Benutzerziele und mögliche intelligente Systemreaktionen analysiert und somit proaktiv Unterstützung für die im AmI-Raum befindlichen Benutzer geplant werden können. Im letzten Schritt setzt der AmI-Raum die erkannten Ziele in Umgebungsänderungen und Gerätefunktio-

onen um und adaptiert somit seinen eigenen Zustand gemäß den erkannten Benutzerzielen.

Das Konzept von Ambient-Intelligence, konsequent umgesetzt, führt zu einer Konvergenz unterschiedlicher Technologien, die heute noch separat entwickelt und eingesetzt werden. Vor allem aus der Konvergenz von Mobilität, Allgegenwärtigkeit, Multimedialität und Multimodalität wird sich ein neues Modell für die Informations- und Kommunikationstechnologie entwickeln (siehe Abb. 6), das ein intelligentes Umfeld (AmI-Raum) mit intelligenten Geräten realisierbar macht und den Benutzern in dieser Umgebung dient (Encarnação/Wichert 2005).

Abbildung 6: Metapherübergang: Ambient-Intelligence (AmI).



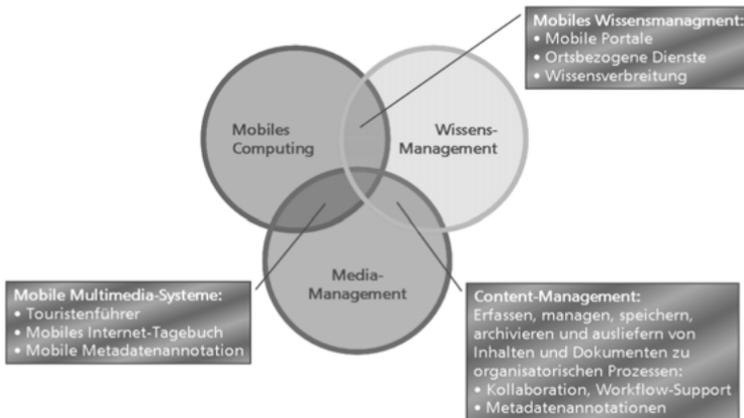
Schon 2001 schlug die ISTAG vor, die Ambient-Intelligence-Prinzipien nicht nur für den Menschen, sondern auch auf Tiere und sogar auf industrielle Güter, Roboter und Produktionstechniken auszudehnen (ISTAG 2001). Im Umfeld von Ambient-Intelligence kann somit von einer Vielzahl an Smart-Playern ausgegangen werden. Hier seien als mögliche Smart-Player Menschen, Tiere, Geräte, Arbeitsplätze, Maschinen und Güter genannt. Die Vision von Ambient-Intelligence ist damit nicht ausschließlich die auf den Mensch bezogene Kommunikation, sondern handelt von allen Smart-Playern, die mit den intelligenten Umgebungen, in denen sie sich aufhalten, interagieren und deren Services sie in Anspruch nehmen (zu Hause, im Büro, in der Schule, im Krankenhaus, im Verkehrssystem, in der Fabrik, usw.).

Im Rahmen dieses Beitrags konzentrieren wir uns nur auf den Menschen als den Smart-Player in einem intelligenten Umfeld. Diese Interaktion wird Mensch-Umgebung-Interaktion (Human-Environment-Interaction, kurz: HEI) genannt.

Menschen in einem intelligenten Umfeld

Im Bereich der intelligenten Umgebungen, vor allem im häuslichen Umfeld und Büroumgebungen, sehen wir die Verschmelzung und Konvergenz von Mobile-Computing, Medienmanagement und Wissensmanagement (Abb. 7) als eine zentrale Voraussetzung zur Realisierung eines AmI-Raumes. In diesem Zusammenhang sind Technologien zur Umsetzung von Präsenz und Umgebungsbewusstsein grundlegend bei der Realisierung der HEI-Schnittstellen.

Abb. 7: Wichtige Technologien zur Realisierung von HEI: Human-Environment-Interaction.



Eine Ambient-Intelligence-Umgebung, und damit die in ihr befindlichen Geräte und Applikationen, hat generell fünf Dimensionen an Kontextvariablen zu erfassen:

1. Benutzeridentifikation;
2. Aktionen: erkennen, was der Benutzer gerade macht;
3. allgemeine Kontextdaten: erfassen von Daten, wie beispielsweise genaue Position und Zeit;
4. Interaktionsziele: interpretieren der Ziele und Beweggründe der Interaktion;
5. Interaktionsmodalität: erkennen der vom Benutzer ausgewählten Interaktionsparadigmen (Art der Multimedialität und Multimodalität).

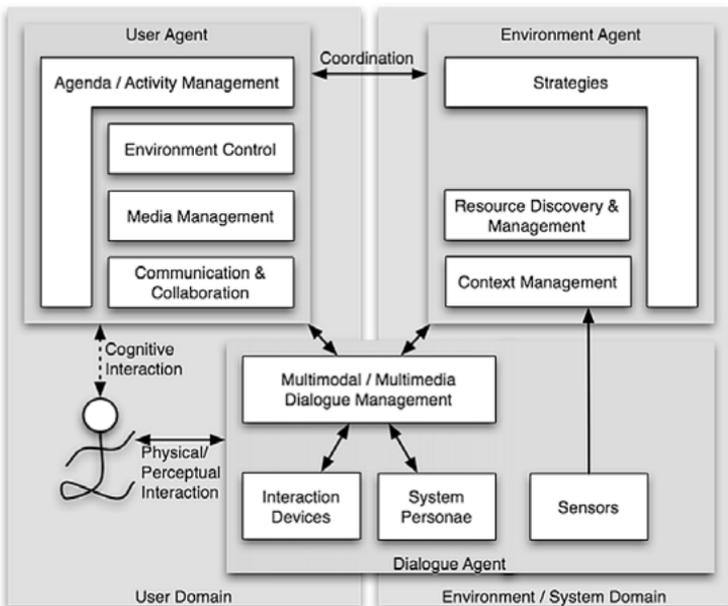
Vereinfachend ausgedrückt, spannt ein AmI-Raum somit drei Dimensionen auf, die aus den Kontextdaten, dem Raum der möglichen Aktionen und Umgebungszielen sowie der durch den Benutzer ausgeübten Interak-

tionen besteht. Diese drei Dimensionen miteinander zu verknüpfen und in Konvergenz zu bringen, muss die Herausforderung jeder technischen Entwicklung in Bezug auf die Realisierung des AmI-Raumes sein.

Eine sich hieraus ergebende mögliche funktionale Systemarchitektur für die Realisierung von Human-Environment-Interaction, wie sie am Fraunhofer-Institut für Graphische Datenverarbeitung definiert wurde (Hellenschmidt/Wichert 2005), ist in Abb. 8 abgebildet. Funktionale Komponenten aus drei unterschiedlichen Dimensionen (User-Agent, Environment-Agent und Dialogue-Agent genannt) realisieren die Erfassung des AmI-Raumes.

Manche Komponenten lassen sich als eher dem Benutzer zugehörig, andere als Umgebungskomponenten identifizieren. Intuitiv sind Interaktionsgeräte ein typisches Beispiel für die notwendige Kooperation und Interoperabilität von unterschiedlichen Entitäten. Erst durch ein solches koordiniertes Zusammenspiel ist die Realisierung von Multimodaler Interaktion möglich.

Abb. 8: HEI: Funktionale System-Architektur (Fraunhofer-IGD).

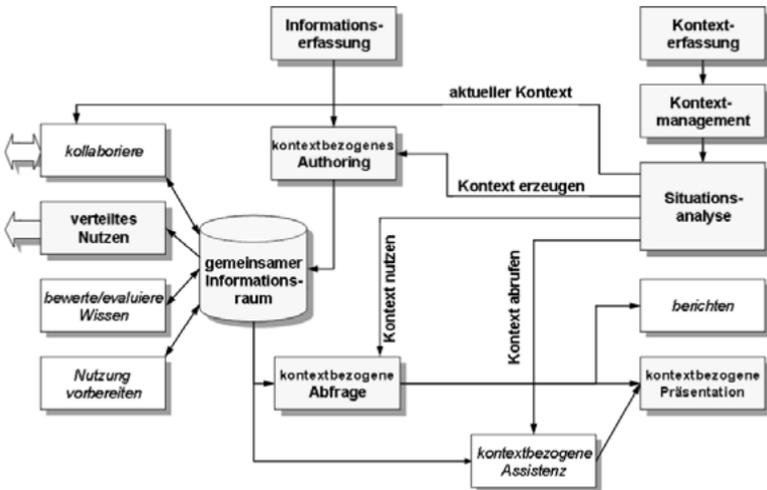


Die in Abb. 8 dargestellte System-Architektur für Human-Environment-Interaction illustriert, dass für die beschriebene Ambient-Intelligence-

Funktionalität eine große Spannweite an unterschiedlichen Funktionalitäten bereitgestellt werden muss. Wir konzentrieren uns hierbei auf multimodale und multimediale Szenarien im häuslichen Umfeld und in Büroumgebungen. Offensichtlich interagiert der Benutzer solcher Umgebungen in unterschiedlicher Art und Weise. Einerseits in einer impliziten Art, in der seine persönlichen Vorlieben aber auch seine persönliche Agenda dem AmI-Raum bekannt sind und in der auf Basis interpretierter impliziter Benutzerziele die Umgebung gesteuert werden kann. Diese Interaktion kann auch kognitive Interaktion genannt werden.

Die andere Art der Interaktion ist die herkömmlich bekannte Interaktion mittels expliziten physikalischen Handlungen unter Benutzung aller vorhandenen multimodalen Interaktionsmöglichkeiten. Abb. 8 illustriert darüber hinaus, dass durch intelligente Verknüpfung von Sensorinformationen mit impliziten und expliziten Interaktionen neue Benutzungsmöglichkeiten entstehen, sich aber auch vollkommen neue technologische und wissenschaftliche Herausforderungen ergeben.

Abb. 9: HEI: kontextbewusste Informationsverarbeitung.

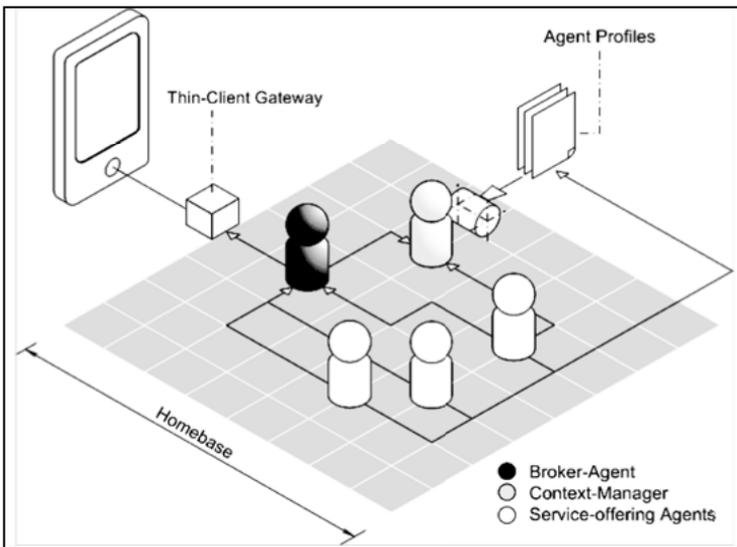


Am Zentrum für Graphische Datenverarbeitung und am Fraunhofer-Institut für Graphische Datenverarbeitung sind unterschiedliche Projekte durchgeführt worden, deren System-Architektur die in Abb. 9 gezeigte funktionale Architektur instanziierten. Ein Beispiel ist das mit einem europäischen Konsortium durchgeführte Mummy-Projekt, in dem Wissens-Management-Anwendungen im mobilen Umfeld implementiert wurden.

Hierbei stand besonders die Unterstützung für mobile Servicemitarbeiter im Vordergrund (Balfanz et al. 2005). Ein Schwerpunkt der Architektur liegt in der Erfassung verteilter Kontextinformation und deren Verarbeitung und Interpretation durch zeitnahe Situationsanalyse.

Ein anderes Projekt ist *Servingo* (Tazari et al. 2006), dessen unterschiedliche Komponenten in Abb. 10 illustriert sind. Integriert mit einer graphischen Oberfläche werden durch autonome Agenten, die untereinander kommunizieren und kooperieren, dem Benutzer in Abhängigkeit seiner aktuellen Interessen Dienste angeboten.

Abb. 10: HEI: Schlüsselfunktionalitäten.



Die in der funktionalen System-Architektur (Abb. 9) illustrierten Funktionalitäten werden hierbei durch ausgezeichnete Agenten – einen Vermittlungsagent, einen Kontextmanager und unterschiedlichste Serviceanbieter – realisiert. Dabei ist der Vermittlungsagent in der Lage, auf Basis der Benutzer- und Umgebungsinformationen diejenigen Dienste auszuwählen und dem Benutzer zu präsentieren, die der gegenwärtigen Situation am besten entsprechen.

Aus Sicht der Softwaretechnik erlauben diese Agenten somit die Implementierung einer kontextbewussten Informationsverarbeitung. Die prototypisch realisierten Dienste unterstützen damit Konzepte der mobilen Informations- und Wissensverarbeitung, indem sie herkömmliche Technologien des Wissensmanagements um mobile Aspekte erweitern und

hiermit die Entwicklung, Mitbenutzung, Wiederauffindung und Visualisierung in der mobilen Arbeit unterstützen. Dies beschleunigt den Arbeitsablauf von räumlich getrennten Prozessen (mobiles »just-in-time«) erheblich.

Sowohl Mummy als auch Servingo stellen eine Plattform für mobile, mehrsprachige Dienste dar, die eine individuelle Orientierung im mobilen Umfeld bieten.

HEI Forschungsthemen

Forschung in den Bereichen Mensch-Umgebung-Interaktion und Ambient-Intelligence ist bereits an vielen Forschungseinrichtungen als Schwerpunkt etabliert. Eines dieser Forschungszentren ist das Ambient-Intelligence-Labor des Fraunhofer-Instituts für Graphische Datenverarbeitung (Abb. 11), an dem unterschiedliche Arten von sensorischer Datenerfassung, Interaktionsmöglichkeit und Gerätekooperation im Büroumfeld erprobt und deren Akzeptanz innerhalb unterschiedlicher Benutzergruppen evaluiert wird (Nazari 2006).

Abb. 11: Ambient-Intelligence-Labor zur Erforschung von HEI-Technologien (Fraunhofer-IGD).



Zur sensorischen Erfassung von Umgebungsinformationen stehen im Ambient-Intelligence-Labor RFID-Technologien (Radio Frequency Identification) und Positionierungssysteme auf Basis von WLAN und Ultra-

breitbandtechnologie zur Verfügung. Andere Projekte streben die Etablierung einer funktionalen Referenzarchitektur auf Basis der in Abb.9 gezeigten System-Architektur für die Anwendung in unterschiedlichen Domänen an.

Hier ist das Mensch-Technik-Interaktions-Projekt EMBASSI (Herfet et al. 2001) zu nennen, bei dem Agenten unterschiedlichster Funktionalitäten bei der Ausführung der Benutzerziele assistieren. Dieses Verhalten zu dynamisieren und ein selbstorganisiertes Verhalten eines darauf basierenden Geräte- und Applikationsensembles zu implementieren ist Aufgabe weiterführender Arbeiten gewesen.

Das Projekt DynAMITE hat beispielsweise eine verteilte Software-Infrastruktur entwickelt, die sowohl dynamische Geräteensembles realisiert als auch die Kooperations- und Kommunikationsmechanismen abstrahieren kann (Hellenschmidt/Kirste 2004, Hellenschmidt 2005).

Die IST Advisory Group (ISTAG 2003) liefert konkrete Hinweise, welche Technologien für die Realisierung von Ambient-Intelligence-Szenarien und damit für eine Mensch-Umgebung-Interaktion von Bedeutung sind. Erst durch Konvergenz dieser Technologien kann davon gesprochen werden, dass der Mensch von benutzerfreundlichen, miteinander vernetzten und kooperativen Geräten umgeben ist, die sich des Benutzers Gegenwart, seiner individuellen Persönlichkeit und seiner Bedürfnisse und Wünsche bewusst sind. Konkrete Technologien, die hier in der Zukunft vorangebracht werden müssen, sind u.a. Verfahren zur intelligenten Benutzerprofilbildung, verteilte Datenbanken, Sicherheitstechnologien, lernende und reflexive Systeme, Sensoren und Aktuatoren, Mensch-Maschine-Schnittstellen zur Unterstützung natürlicher und auch natürlichsprachiger Interaktion, Kommunikationsunterstützung durch intelligente Objekte wie beispielsweise Avatare, Koordinations- und Kooperationsmechanismen und die oben besprochenen Software-Infrastrukturen. Auf einige dieser technologischen Schwerpunkte soll im Folgenden exemplarisch wenn auch nur kurz eingegangen werden.

Ein Schwerpunkt der Interaktionstechnologien ist die Intentionserkennung der Umgebung, die sowohl das Benutzerverhalten als auch Umgebungsveränderungen über Sensoren erkennt und daraus Ziele ableitet. Hierzu ist es notwendig, dass die Umgebung auf Veränderungen, d.h. direkte oder indirekte Interaktionen von Benutzern reagiert bzw. der geänderte Umgebungszustand festgestellt wird. Dies kann zum einen durch die direkte Handlung der Menschen mit der Umgebung durch multimodale Interaktionen wie Sprache, Gestik oder entsprechende Benutzungsschnittstellen erfolgen. Über einen Dialog ist es möglich, bei z.B. kritischen Situationen Rückfragen stellen zu können. Zum anderen kann auch eine Intentionserkennung der Umgebung durch ein Erkennen

des Benutzerverhaltens über Sensoren erfolgen, die das System mit bestimmten Verhaltensmustern vergleicht und interpretiert. Um dies zu erreichen, müssen darüber hinaus Methoden entwickelt werden, die es auch mehreren Benutzern ermöglicht, für spezifische Ziele zu kooperieren, auch ohne dass explizit definiert werden muss, wie die Kooperation für das gegebene Ensemble funktioniert. Hier können konnektionistische Ansätze (z.B. neuronale Netze) und heuristische Zielsuchverfahren für stationäre Optimierungen, symbolische Ansätze wie Planungssysteme oder auch regelbasierte Ansätze zur Vorwärtsinterpretation von Umgebungsdaten zum Einsatz kommen (Hellenschmidt 2007).

Mittels Ontologien werden die semantischen Nachrichten der verschiedenen Ebenen der HEI-Architektur definiert und syntaktisch (maschinenlesbar) vereinheitlicht. Daneben sind einheitliche Semantiken für Datenbankzugriffe, Benutzerprofile und andere sich aus den Szenarien ergebenden Komponenten und Benutzern zu entwickeln. Ziel einer solchen Technologie muss es sein, eine vereinheitlichte AmI-Beschreibung aller beteiligten Komponenten und Nachrichtenflüsse zu erreichen. Hierbei müssen bereits bestehende Standards wie z.B. die Web-Ontologie-Sprache OWL (Ontology Web Language) oder das UPnP (Universal Plug and Play) berücksichtigt und verwendet werden.

Middleware-Technologien bilden traditionell das Bindeglied zwischen Anwendungen und Kommunikationsinfrastrukturen. Middleware sorgt für die Abstraktion von unterliegenden Hard- und Softwareschichten sowie die Verteiltheit von Komponenten und deren Adressierbarkeit bei gegebenem Domänenkonzept. Um den Anforderungen von selbstorganisierenden AmI-Ensembles gerecht zu werden, müssen zukünftige Middleware-Technologien für mehr als Interoperabilität sorgen. Basisfunktionalitäten bleiben weiterhin die Geräteabstraktion, Dezentralisierung und die dynamische Integration von neuen Komponenten. Hinzu kommt die Unterstützung semantischer Beschreibungen für Funktionen zur intelligenten Suche (Discovery), Ablage (Repository), Orchestration und Choreographie, Fehlererkennung, Selbstdiagnose und Selbstheilung. Erschwerend bringt Ambient-Intelligence Anforderungen mit sich, die dem eigentlichen Konzept einer Middleware, nämlich von darunter Liegendem zu abstrahieren, widersprechen. Bewusstseinsfunktionen (Awareness) erfordern in ihrer letzten Konsequenz, dass eine Middleware alle ihr zur Verfügung stehenden Informationen an die darüber liegende Anwendung weiter gibt. Um die Interaktion mit den Anwendungen dennoch beherrschbar halten zu können, bedarf es neuer Konzepte. Um die mögliche Heterogenität und Dynamik (Hinzufügen und Entfernen von Komponenten und/oder Geräten) von AmI-Ensembles und Komponenten-

strukturen zu unterstützen, sind Konzepte der verteilten Implementierung notwendig.

Integrative Kommunikationstechnologien für die unterschiedlichen Anforderungen der verschiedenen Menschen in einem Haushalt (u.a. IP-gestützte Vernetzung), in der Logistik (u.a. Funk oder WLAN) und in der Telekommunikation (u.a. GPRS, UMTS) sind ebenfalls weiterzuentwickeln. Ziel muss es sein, die semantische Kommunikation der Komponenten innerhalb eines AmI-Raumes sowie die des Benutzers, bzw. seiner technischen Komponenten mit dem AmI-Raum von der technischen Kommunikation zu abstrahieren und die Vereinheitlichung der technischen Kommunikationen anzubieten.

Durch die fortschreitende Integration der Rechner- und Computersysteme in die gewohnte Umgebung unseres täglichen Lebens ändert sich die Art und Weise, wie wir mit solchen technischen Systemen interagieren. Zurzeit beschränkt sich die Integration der Computersysteme meist darauf, den Rechner selbst zu verstecken oder als Design-Objekt in die Umgebung einzubetten. So ist beispielsweise der Mini-Computer als multimediales Schaltzentrum im Wohnzimmer etabliert und fast nicht mehr von dort wegzudenken.

Eines hat sich allerdings bis heute noch nicht wirklich geändert: Die Eingabegeräte sind immer noch die gleichen. Computermaus und Tastatur sind zwar nicht mehr mit Kabeln mit dem Rechner verbunden, sondern arbeiten funkgesteuert, sie liegen aber immer noch wie vor zehn oder zwanzig Jahren meist als einzige Eingabemedien bereit.

Soll aber der Mensch ein vollständig neues Verständnis von Technologie in seiner unmittelbaren Umgebung entwickeln, müssen auch die Schnittstellen zu dieser Technologie weiter entwickelt werden. Denn erst wenn Interaktion wirklich intuitiv und einfach gestaltet ist, können neue Technologien auch von technisch unerfahrenen Menschen verwendet werden. Ein Ziel der aktuellen Forschung ist es deshalb, die Schnittstelle zum Computer so zu gestalten, dass der Unterschied zwischen Kommunikation mit Geräten und mit anderen Menschen deutlich kleiner wird. Das wichtigste Medium der menschlichen Kommunikation ist dabei die Sprache.

Die Unterstützung von natürlichsprachlichen Eingaben sowie deren semantische Analyse in Bezug auf die intelligente Umgebung leisten einen wesentlichen Beitrag zur Vereinfachung der Bedienbarkeit von Interaktionsschnittstellen zwischen Mensch und Umgebung. Die Grundlage für eine korrekte semantische Analyse und Intentionserkennung von natürlichsprachlichen Eingaben, Anfragen und Befehlen bilden Wissensrepräsentationsmodelle wie z.B. Ontologien.

Mit Hilfe von Ontologien und deren standardisierten Repräsentations-sprachen wie OWL kann Kontextinformation wie z.B. komplexe Zusammenhänge zwischen feinkörnigen Benutzerprofilen (bevorzugte fachsprachliche Terminologien, kulturelle Hintergründe, etc.), Informationsobjekte und miteinander vernetzte kooperative Geräte in einer Weise beschrieben werden, die für Computer verständlich und somit interpretierbar ist.

Auf der Grundlage von derartigen maschineninterpretierbaren Wissensrepräsentationsmodellen kann der AmI-Raum natürlichsprachliche Eingaben ihrer Bedeutung entsprechend analysieren (Sevilmis 2005). Aus der Analyse können unter Einbeziehung von Inferenzmechanismen, die durch logisches Schließen aus explizitem Wissen implizites Wissen generieren, beispielsweise konkrete Aktionspläne abgeleitet werden, die die Antwort bzw. das Ergebnis der natürlichsprachlichen Eingabe darstellen.

Der Aktionsplan beschreibt die temporäre Reihenfolge der auszuführenden Aktionen sowie deren Abhängigkeiten und Zugehörigkeiten zu den einzelnen Geräten im AmI-Raum. Diese Aktionen können dann auf die verschiedenen Geräte verteilt und ausgeführt werden. Die Ausführung des Aktionsplans ist die Antwort der intelligenten Umgebung auf die natürlichsprachliche Eingabe des Benutzers (z.B. Visualisierung). Im Vordergrund intelligenter Umgebungen steht unter anderem vor allem die Präsentation der Ergebnisse in einer Weise, die entweder intuitiv ist (z.B. das Licht schaltet sich ein) oder die eine dem Erkenntnisgewinn dienenden Blickwinkel auf die Ergebnisse erlaubt, ohne den Endanwender dabei zu überfordern. Auch die Art und Weise einer benutzerzentrierten Ergebnispräsentation kann in Form von Ontologien festgehalten werden. Somit erreicht man eine große Varianz bezüglich der Darstellung von Ergebnissen, welches ein breites Spektrum verschiedener Benutzergruppen anspricht.

Allerdings gibt es immer wieder Situationen, in denen Sprache allein nicht ausreichend ist, um einfach und intuitiv zu interagieren. Frühere Arbeiten (Sá et al. 2001) haben gezeigt, dass sich diese Problematik auch im häuslichen Umfeld zeigt. Typische sprachliche Anweisungen an ein intelligentes, vernetztes Haus wie beispielsweise »Schalte diese Lampe an!« sind ohne die kombinierte Verwendung von Sprache und Gestik für ein technisches System nicht zu verstehen. Insbesondere die Verwendung einer Zeigegeste ist ein wichtiges Hilfsmittel der menschlichen Kommunikation.

Aktuelle HEI-Forschungsarbeiten konzentrieren sich daher auch auf die multimodale Ein- und Ausgabe zwischen Mensch und Umgebung, um diese Lücke zu schließen. So wurde beispielsweise am Zentrum für Graphische Datenverarbeitung ein Gestenerkennungssystem entwickelt,

das ohne Trainingsphase von jedermann bedienbar und durch ein video-basiertes Verfahren in der Lage ist, die Zeigegeste des Anwenders zu erkennen und in Echtzeit zu verfolgen. Die Einsatzmöglichkeiten eines solchen Systems sind vielfältig.

In einem musealen Umfeld können beispielsweise auch technisch unversierte Museumsbesucher mit Hilfe dieses Systems interaktiv digitalisierte Gemälde untersuchen und mehr über das Meisterwerk und seinen Künstler erfahren (Malerczyk et al. 2005) (Abb. 12). Im Sinne einer neuen Mensch-Umgebung-Interaktion wird die Technik selbst vom Anwender nicht mehr wahrgenommen. Der Besucher benötigt keine technischen Eingabegeräte, sondern interagiert mit dem Exponat ausschließlich mit der bloßen Hand.

Abb. 12: Intuitive Erforschung eines digitalen Gemäldes mittels Zeigegestenerkennung und virtueller Lupenfunktion.



Solche Interaktionsparadigmen mit immer realistischer darstellbaren Avataren und natürlich klingender Sprachsynthese sind insbesondere auch auf das Ziel der Annäherung von Mensch-Computer-Interaktion an die zwischenmenschliche Kommunikation übertragbar (Abb. 13), womit sie einer der wichtigsten Aufgaben eines interaktiven Systems beitragen, nämlich der Kommunikation relevanter Informationen in einer angemessenen Art und Weise sowie eine effektive Kommunikation des Anwenders mit dem System.

Dabei hängt die Relevanz von Informationen direkt von der Situation in der realen Welt, dem Kontext, ab. Heutige interaktive Systeme verfügen über keine ausreichende Sicht auf die Welt. Dadurch ist die Basis für die Selektion der darzustellenden Information vage und oft falsch. Häufig muss der Benutzer die relevante Umgebungsinformation eigenständig in eine maschinenlesbare Form überführen. Dies ist fehleranfällig, zeitaufwändig und erfordert ausreichende Kenntnisse über die zugrundeliegende Technologie, die den Anwenderkreis heutiger Systeme deutlich einschränkt.

Für das Beispiel der Reparaturanleitung mit Augmented-Reality (siehe Abschnitt »Mensch-Computer-Interaktion heute«) bedeutet dies, dass der Anwender dem System genau mitteilen muss, welches Problem vorliegt (Auswahl der richtigen Anleitung) und welche Aktionen er ausgeführt hat, damit die nächste auszuführende Aktion auch angezeigt wird. Ob der Anwender nicht aus Versehen einen Schritt übersprungen hat, die Aktion richtig ausgeführt hat oder die dargestellte Menge an Informationen adäquat für den Anwender ist, kann die Benutzungsschnittstelle nicht prüfen.

Abb. 13: Multimodale Interaktion mittels Sprach- und Gestenerkennung zur natürlichen Interaktion mit einem persönlichen Avatar.



Ziel zukünftiger Forschung muss es deswegen sein, Strategien für die direkte, echtzeitfähige Wahrnehmung und Interpretation der Umgebung, der Situation und des Kontextes zu entwickeln. Dadurch kann die notwendige Nutzerinteraktion reduziert und die Bedienbarkeit der Schnitt-

stelle verbessert werden. Die Benutzungsschnittstelle wird dann in der Lage sein, die passende Information mit optimalem Detaillierungsgrad zu präsentieren sowie auf Aktionen des Anwenders geeignet zu reagieren und sich dessen Bedürfnissen zu adaptieren.

Hinsichtlich der Wahrnehmung der realen Welt können dabei drei Aspekte unterschieden werden:

- Die *Umgebungswahrnehmung* liefert Informationen zur Position und Orientierung des Anwenders und den Objekten in seiner Umgebung sowie deren Status (Benutzer-Tracking und Objekterkennung).
- Die *Situationswahrnehmung* erkennt die Dynamik in der Umgebung, der Aktionen des Benutzers und der relevanten Objekte.
- Die *Kontextinterpretation* basiert auf den Ergebnissen der Umgebungs- und Situationswahrnehmung und interpretiert diese Daten in Bezug auf den aktuellen Kontext.

Ein illustratives Beispiel hierzu ist das interaktive Fußballspiel iFU, das anlässlich der Fußball WM 2006 am Zentrum für Graphische Datenverarbeitung entwickelt wurde (Abb. 14). Der Anwender (Spieler) übernimmt die Rolle des Torwarts beim Elfmeterschießen. Ziel des virtuellen Spielers ist es, ein Tor zu schießen. Um zu erkennen, ob der Spieler den Ball hält, muss das System seine dynamische Bewegung erfassen, diese mit der Flugbahn des Fußballs abgleichen, um schließlich eine mögliche Kollision zu ermitteln.

Abbildung 14: Interaktives Fußballspiel (iFU).



Ein weiteres Beispiel soll die Unterschiede der obigen Aspekte nochmals näher erläutern: Ein Chirurg führt eine Herzoperation aus (Kontext). Er geht in den Operationssaal, in dem sich der Patient und die notwendigen Operationsinstrumente befinden. Die Technologie für die Umgebungswahrnehmung identifiziert den Ort (Operationssaal) und die Objekte im Raum (Patient, Instrumente). Nun beginnt er mit der Operation und führt die notwendigen Schritte aus. Die Situationswahrnehmung beobachtet und erkennt die Aktionen des Chirurgen. Kontextinterpretation evaluiert die erkannten Aktionen im Kontext Herzoperation und prüft, ob die Aktionen in der richtigen Reihenfolge auf die richtige Art und Weise ausgeführt werden. Das System kann bei erkannten Fehlern reagieren und die Information für den nächsten Schritt anzeigen.

Der Einsatz von Kameras als Sensorik zur Wahrnehmung von Umgebung und Situation bietet sich an. Basis bilden dabei die Erkenntnisse aus der Computer Vision und dem visions-basierten Tracking für Augmented-Reality. Die Kombination mit anderen Sensoren ist jedoch in jedem Fall notwendig. Anfang 2006 wurde hierfür ein Grundstein mit dem »Intuition Research Position Paper for Augmented Reality« gelegt (Knöpfle 2006), das von führenden Experten erstellt wurde und die zukünftigen Forschungsrichtungen im Bereich Augmented-Reality und mögliche Anwendungen für Ambient-Intelligence beschreibt. Dort wird auch gefordert, dass neben der Vernetzung von Sensoren, was bisher im Vordergrund stand, in Zukunft die Fusion der Sensordaten stärker im Fokus liegen muss, um aus den vorhandenen Daten höherwertige Kontextinformationen zu erzielen und entsprechend interpretieren zu können.

Neben den technologisch orientierten Forschungsthemen stehen gleichberechtigt Disziplinen wie Psychologie, Sozialwissenschaften und Medizin. Nur neue interdisziplinäre Ansätze machen echte HCI-Innovationen in Richtung HEI und damit die Realisierung von Ambient-Intelligence möglich. Da die Konzepte für Ambient-Intelligence nicht auf einen bestimmten Kulturkreis beschränkt sind, muss die Anwendbarkeit von entsprechenden Ansätzen und Realisierungen auf verschiedene Kulturen überprüft werden, um gegebenenfalls auf deren spezifische Eigenarten eingehen zu können. Konsequenterweise wird die Multikulturalität einen starken Einfluss auf Szenarien für die Mensch-Umgebung-Interaktion haben.

Neben Interdisziplinarität und Multikulturalität darf auch die Interoperabilität nicht vergessen werden. Nur wenn die Vernetzung und Kooperation von Geräten, Komponenten, Sensoren und sogar ganzen intelligenten Umgebungen, die von verschiedenen Herstellern in verschiedenen Ländern entwickelt werden, garantiert werden kann, lassen sich Ambient-Intelligence-Lösungen ad-hoc aus autonomen Geräten zusam-

menstellen, die sich selber zu einem Gesamtsystem konfigurieren und als kohärentes Ensemble agieren.

Abb. 15: Beispiel eines intelligenten Home-Monitoring Systems für Senioren (Aarts/Diederiks 2006) (© Philips).



Ambient-Intelligence und die Mensch-Umgebung-Interaktion stellen also sehr hohe Anforderungen an alle, die es realisieren wollen. Doch nur wenn diese hohen Ansprüche umgesetzt werden, kann eine Technologie realisiert werden, die von der Mehrheit der Anwender akzeptiert wird und sowohl in einen wissenschaftlichen als auch ökonomischen Erfolg mündet (vgl. Abb. 15).

Fazit

Ein Paradigmenwechsel von HCI/HHI zu HEI findet in hochentwickelten Benutzerschnittstellen für Dialoge im Kontext von intelligenten Umgebungen statt. Ambient-Intelligence und die Mensch-Umgebung-Interaktion muss also Forschungsthemen wie Mensch-Computer-Interaktion mit grundlegend neuen Interaktionskonzepten zur kontextbewussten Informationsverarbeitung, Middleware und Agententechnologie zur Servicevermittlung, virtuelle und erweiterte Realität, Wissensmanagement und künstliche Intelligenz in einer gemeinsamen Vision vereinen. Um diesen Paradigmenwechsel zu implementieren, muss die Hardware-orientierte

tierte Forschung und Entwicklung harmonisiert in das Gesamtkonzept mit aufgenommen werden, um neue Geräte und Systeme – aber auch neue Inhalte, neue integrierte und personalisierte Dienste und neue Anwendungen – realisieren zu können (Raffler 2005). Dazu gehören die Sensortechnik, mobile Technologien, eingebettete Systeme, Nano-Technologie und Technologien zur Umsetzung von Präsenz und Umgebungsbewusstsein. Schließlich benötigen diese neuen Formen der Interaktion und des natürlichen Dialoges auch der Interdisziplinarität, der Multikulturalität und der Interoperabilität.

Danksagung

Die Autoren bedanken sich bei allen Kollegen, die durch ihre wertvollen Diskussionsbeiträge, mit Hinweisen auf relevante Veröffentlichungen und auch mit Beispielen und Bildern beim Entstehen dieses Beitrages geholfen haben. Insbesondere gilt der Dank Cornelius Malerczyk und Holger Graf vom ZGDV sowie Michael Hellenschmidt, Christian Knöpfle, Neyir Sevilmis, Ali Nazari, Georgios Sakas, Reiner Wichert, Didier Stricker und André Stork vom Fraunhofer-IGD.

Literatur

- Aarts E./Diederiks E. (2006): »Ambient Lifestyle From Concept to Experience«. Amsterdam.
- Aarts E./Encarnação J. L. (2006): »Into Ambient Intelligence«. In: Aarts, E./Encarnação, J. L. (Eds.): »True Visions: The Emergence of Ambient Intelligence«. Berlin, Heidelberg, New York, S. 1-16.
- Aarts Emile/Marzano Stefano (2003): »The New Everyday: Views on Ambient Intelligence«. Rotterdam.
- Balfanz, D./Grimm, M./Tazari, M.-R. (2005): »A Reference Architecture for Mobile Knowledge Management«. In Dagstuhl-Seminar 05181 2005, Mobile Computing and Ambient Intelligence. Wadern.
- Bleser, G./Pastarmov, Y./Stricker, D. (2005): »Real-time 3D Camera Tracking for Industrial Augmented Reality Applications«. In: Skala, V. (Ed.): European Association for Computer Graphics (Eurographics): WSCG 2005. Plzen, University of West Bohemia, S. 47-54.
- Bleser, G./Stricker, D./Wuest, H. (2006): »Online Camera Pose Estimation in Partially Known and Dynamic Scenes«. In Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality. Los Alamitos, CA. New York, S. 56-65.

- Dähne, P./Ioannidis, N./Karigiannis, J./Stricker, D. (2002): »ArcheoGuide - A Mobile Augmented Tracking System for Archeological Sites - A Solution to the Tracking Problematic«. In: Proceedings Elektronische Bildverarbeitung & Kunst, Kultur, Historie 2002. Berlin, Gesellschaft zur Förderung angewandter Informatik e.V., S. 33-40.
- Ducatel, K./Bogdanowicz, M./Scapolo, F./Leijten, J./Burgelman, J.-C. (2001): »Scenarios for Ambient Intelligence in 2010, Final Report«. Institute for Prospective Technological Studies (ISTAG), Seville, European Commission, Februar, 2001.
- Encarnação J. L./Wichert R. (2005): »Technologische Herausforderungen intelligenter Umgebungen – Chancen für Wissenschaft und Wirtschaft«. In: »Computer in der Alltagswelt – Chancen für Deutschland?«, Acatech Symposium, Juni 2005.
- Hellenschmidt, M. (2005): »Distributed Implementation of a Self Organizing Appliance Middleware«. In: Smart Objects and Ambient Intelligence SOC-EUSAI 2005, S. 201-206.
- Hellenschmidt, M. (2007): »An adaptive rule-based inference engine for realizing reasonable behaviour of smart environments«. In Proceedings Symposium on Artificial Intelligence and Simulation of Behaviour: Affective Smart Environment, Newcastle, Great Britain, April.
- Hellenschmidt M./Kirste Th. (2004): »A Generic Topology for Ambient Intelligence«. In: European Symposium on Ambient Intelligence (EUSAI), Eindhoven, NL, S. 112-123.
- Hellenschmidt M./Wichert R. (2005): »Goal-oriented Assistance in Ambient Intelligence«. In: Workshop on Experience Research in Ambient Intelligence (ERAmI), Eindhoven, NL.
- Herfet, T./Kirste, T./Schnaider, M. (2001): EMBASSI: »Multimodal Assistance for Infotainment and Service Infrastructures«. Computers & Graphics, 25, 4, S.581 - 592.
- INTUITION Network of Excellence (2006): »Intuition Research Position Paper on Augmented Reality«. Working Group on Augmented Reality, www.intuition-eunetwork.net/.
- ISTAG Report on Ambient Intelligence (2003): »From Vision to Reality – For People & Participation«. European Commission, September 2003.
- Khan M./Dogan S./Maataoui A./Gurung J./et.al. (2005): »Accuracy of biopsy needle navigation using the Medarpa system-computed tomography reality superimposed on the site of intervention« European Radiology, 15: 2366-2374, März 2005.
- Knöpfle, Chr./Stricker, D./Weidenhausen, J. (2003): »Lessons learned on the way to industrial augmented reality applications, a retrospective on ARVIKA«. In: Computers and Graphics, 27,6, S. 887-891.

- Malerczyk, C./Dähne, P./Schnaider, M. (2005): »Exploring Digitized Artworks by Pointing Posture Recognition«. In: VAST 2005 - International Symposium on Virtual Reality, Archaeology and Cultural Heritage, Pisa, Italy, June 2005, S.113-119.
- Nazari Shirehjini, A. (2006): »Situation Modelling: A Domain Analysis and User Study«. In: The Institution of Engineering and Technology (IET) (Ed): Proc.: 2nd IET International Conference on Intelligent Environments, Vol.2. IE 06, Athens, pp. 193-199.
- Raffler H. (2005): »Ambient Intelligence: Eine industrielle Sicht«. In: Computer in der Alltagswelt – Chancen für Deutschland? Acatech Symposium, Juni 2005.
- Sá, V./Malerczyk, C./Schnaider, M. (2001): »Vision-Based Interaction within a Multimodal Framework«. In: Actas do 10º Encontro Português de Computação Gráfica 2001, Encontro Português de Computação Gráfica, Lisboa, Oktober 2001, S. 61-67.
- Tazari, M.-R./Thiergen, S. (2006): »Serving. A Service Portal on the Occasion of the FIFA World Cup 2006«. In: International Workshop on Web Portal-based Solutions for Tourism, IWWPST' 06, Vienna, S. 73-93.
- Sevilmis, N. et al. (2005): »Knowledge Sharing by Information Retrieval in the Semantic Web«. In: The Semantic Web: Research and Applications (Lecture Notes in Computer Science –LNCS- 3532), S. 471-485.

**PARADIGMENWANDEL DER COMPUTER-BEDIENUNG
AUS KULTUR- UND GEISTESGESCHICHTLICHER
PERSPEKTIVE**

AUF DEM WEG ZUM »FINALEN INTERFACE«.

EIN MEDIENHISTORISCHER ESSAY

WOLFGANG COY

1 Bücher besitzen ein fast perfektes Interface

Mit Bild, Schrift und Zahl wird Kommunikation über große Entfernungen und selbst in die Zukunft möglich. Alle drei Basismedien haben zu kommunikativen Universalien geführt. Bildliche Abstraktionen sprechen die menschliche Wahrnehmung jenseits der Sprache an und erlauben damit über große Bereiche eine Verständigung über kulturelle Prägungen hinaus. Zahlen können als Zahlzeichen ideographisch notiert werden, die gleichfalls jenseits der Sprache angesiedelt sind und auf die basale menschliche Fähigkeit des Zählens verweisen. Freilich können Zahlzeichen bezogen auf eine konkrete Sprache auch logografisch aufgefasst werden. Für die gesprochene Sprache haben sich historisch ganz unterschiedliche Notationssysteme herausgebildet. Piktografische Formen greifen auf abstrahierte Bildvorräte zurück. Ideografische und logografische Schreibweisen verwenden das gleiche Prinzip wie die Zahlzeichen. Unterhalb der Wörter wird der Lautcharakter der Sprache durch die phonografischen Formen der Silben- oder Buchstabenschriften abgebildet. Das griechische Vokalalphabet erlaubt, ohne dass es den Erfindern bewusst war, eine (fast) universelle Notation menschlicher Lautäußerungen, mit der menschliche Sprachen unterschiedlichsten grammatischen und lautlichen Typs reproduzierbar notiert werden können. Vier Universalien, die unterschiedlichen Wegen folgen - keiner verlustfrei, aber alle hinreichend praktikabel, um über Jahrtausende genutzt zu werden.

Mediale Speicher haben eine innere und eine äußere Struktur; die innere Struktur setzt die gespeicherten elementaren Einheiten in eine logische Beziehung, die äußere Struktur erlaubt es menschlichen Nutzern, Gedanken und Wahrnehmungen zu notieren und wieder zu finden.

1 So ist beispielsweise die Notation von Musik oder Bewegung nur mit weiteren Hilfsmitteln möglich.

Schrift hat eine lange Reihe von Speicherformen gefunden, wobei sich Stein, Ton, Pergament und Papyrus als haltbare Formen erwiesen haben.

Das Buch, als gebundener Kodex aus den Rollen, den Rotuli entwickelt, hat seine heutige Form im spätmittelalterlichen Manuskript gefunden. Kapitel, Sachregister, Fußnoten und Randbemerkungen sind Einführungen des zwölften und dreizehnten Jahrhunderts. Eine wesentliche Veränderung ist die Transformation des großen, schweren, demonstrativen Buchs zum tragbaren, handhabbaren Gegenstand. Der *Codex Gigas*, die Prager Teufelsbibel, geschrieben in der ersten Hälfte des 13. Jahrhunderts, misst 92x50 cm und wiegt 75 kg. Das ist sicher eine extreme Variante, weil die Bibel meist als mehrbändige Ausgabe geschrieben wurde, aber sie ist doch typisch in ihrer Immobilität. Die auf Papier geschriebenen Bücher des 13. Jahrhunderts wurden leichter. Die Schrift wurde kleiner geschrieben, mit metallbasierten Tinten konnte man schneller abschreiben, die äußeren Maße schrumpften, die vorher starren Einbände wurde leichter und biegsam, das ganze Buch wurde »tragbar«. Um 1250 war dieser neue Buchtyp eingeführt und setzte sich langsam durch (vgl. Ivan Illich 1991). Zwei Jahrhunderte später ist er die perfekte Vorlage für Gutenbergs Erfindung des System »Satz-Druck-Vertrieb«. Das Interface des modernen Buchs, wie es von Verlagen und Buchhändlern seit dem 15. Jahrhundert benutzt wird, hat sich seitdem nur bescheiden weiterentwickelt. Die gedruckte Seitenzahl ist hinzugekommen und die vielfältige Illustration. Kleiner und leichter ist es geworden, ein industrielles Produkt, eines der ersten überhaupt und das Standardmedium intellektuellen Austauschs.

Mit dem Buch hat sich die direkte mündliche Kommunikation, die ohne Speichermöglichkeit nur gleichzeitig möglich und ohne Übertragungstechnik immer ein ortsgebundener Dialog war, zur visuellen Erfahrung des Lesens und Schreibens über große Entfernung und in unbestimmte Zeit entfaltet. Unterstützt wurde diese Visualisierung durch die Integration von Bildern und Illustrationen – als Zeichnung, Holzschnitt, Kupferstich oder Fotoreproduktion. Der Dialog wandelte sich dabei freilich, zumindest wenn das Geschriebene auch gedruckt wurde, im Kern zum Monolog, zu einem Selbstgespräch des Autors.

Bücher besitzen ein nahezu perfektes Interface. Sehr haltbar, wenn sie ordentlich hergestellt, behandelt und aufbewahrt werden, an Hand und Auge gut angepasst. Sie sind an unterschiedlichsten Orten lesbar, einfach herstellbar – ein Standardmedium in vielen Kulturen.

Es gibt aber auch Nachteile. Sind Bücher erst einmal gedruckt, können sie nicht mehr verändert werden. Sie sind nicht einfach kopierbar und es fällt nicht leicht, eine bestimmte Stelle in einem Buch in einer Bibliothek zu finden. Noch schwerer ist es, gar das Fehlen einer Textstelle nach-

zuweisen. Auch werden Transport und Lagerung problematisch mit dem Wachstum der Bestände.² Und schließlich sind mit den technischen Medien weitere Abbildungs-, Speicher- und Übertragungstechniken wie Film und Tonaufnahme hinzugekommen, die Buch- und Zeitschriftenproduktion ergänzen, sich den Drucktechniken aber verweigern.

2 Universelle Kodes, universelle Maschinen und der Umgang mit ihnen

Während die äußere Form des Buches, die die Schnittstelle zwischen Leser und Inhalt definiert als sehr gelungen angesehen werden kann, lässt die innere Struktur durchaus Verbesserungen zu. Zwar sind für Bild, Schrift und Zahl universelle Kodierungen gefunden, doch es gibt einfachere universelle Kodes, wie die Digitaltechnik zeigt.

Die Entdeckung, dass alle Alphabete sich auf ein schlichtes binäres Alphabet aus zwei Zeichen zurückführen lassen, findet sich zuerst bei Francis Bacon. Die unterschiedlichen Kodes von Bild, Schrift und Zahl können so in einem einzigen, gemeinsamen und universellen Kode zusammengefasst werden. Mit den Digitaltechniken des Computers sind Binärkodes als prominente Beispiele solcher minimalistischen Universalkodes bekannt geworden, die eine einheitliche Speicherung und Übertragung der unterschiedlichen medialen Formen zulassen. Oktal- und Hexadezimalziffern, ASCII oder Unicode sind andere Beispiele solcher Universalkodes und man kann rückwärts gewendet natürlich auch Dezimalziffern, das griechische, hebräische, lateinische oder arabische Alphabet, die Zeichen des chinesischen Alphabetes oder irgendeine andere Menge wohl unterschiedener Zeichen als universellen Speicherkode verwenden. So wie das griechische Vokalalphabet als phonetische Notationsbasis aller Sprachen hätte dienen können, waren Alphabete schon immer als universelle Kodierung für Bild, Schrift und Zahl und alle anderen endlichen Mengen unterscheidbarer Signale einsetzbar. Diese Eigenschaft wurde aber erst anderthalb Jahrtausende später entdeckt, als irische Mönche solch barbarische Sprachen wie Gälisch und Englisch mit lateinischen Buchstaben schrieben und damit eine Vorlage für alle anderen, noch schriftlosen Sprachen lieferten.

Gottfried Wilhelm Leibniz und John Napier haben gezeigt, dass man mit solchen binären Kodes auch rechnen kann. Leibniz hat in einer quasi-pythagoräischen Weltsicht die Bedeutung des Binäralphabets weit über seine praktischen Anwendungen hinaus projiziert. In einem Brief an den

2 Über den Mangel an Stellfläche und Finanzen in öffentlichen Bibliotheken soll hier nichts weiter gesagt werden.

Herzog Rudolph August von Braunschweig, den er mit einem Entwurf einer Gedenkmedaille verbindet, schreibt er:

»Denn einer der Hauptpunten des christlichen Glaubens, und zwar unter denjenigen, die den Weltweisen am wenigsten eingegangen, und noch den Heyden nicht wohl beizubringen sind, ist die Erschaffung der Dinge aus Nichts durch die Allmacht Gottes. Nun kann man wohl sagen, daß nichts in der Welt sie besser vorstelle, ja gleichsam demonstrire, als der Ursprung der Zahlen, wie er allhier vorgestellet ist, durch deren Ausdrückung blos und allein mit Eins und mit Nulle oder Nichts alle Zahlen entstehen. Und wird wohl schwerlich in der Natur und Philosophie ein bessres Vorbild dieses Geheimnisses zu finden sein, daher ich auch die entworfene Medaille gesetzt: IMAGO CREATIONIS. Es ist aber doch dabei nicht weniger betrachtungswürdig, wie schon daraus erscheint, nicht nur, daß Gott Alles aus Nichts gemacht, sondern auch daß Gott Alles wohl gemacht, und daß Alles, was er geschaffen, gut gewesen; wie wirs hier denn in diesem Vorbilde der Schöpfung auch mit Augen sehen.«³

Obwohl Leibniz die Möglichkeit einer binären Rechenmaschine klar vor Augen hatte, blieb die Rechnung mit den binären Codes freilich fast zweihundertfünfzig Jahre lang folgenlos. Alan Turing und Konrad Zuse haben im gleichen Jahr, 1936, solche Binäralphabete für die Konstruktion von Rechenmaschinen verwendet, der eine auf dem Papier, der andere mit einer Maschine aus Blech. 0 und 1 haben sich als perfektes Interface für Rechenmaschinen erwiesen, da sich alle medialen Daten, die auf Zahl, Schrift oder Bild beruhen, sich binär kodiert speichern, übertragen und verarbeiten lassen. Solange die Maschinen unter sich sind, zum Beispiel in einem Netz von Rechnern, Sensoren und Aktoren, genügt es Protokolle zur Signalumsetzung zu definieren. Computer können über solche digitalen Protokolle Daten austauschen, wenn sie sich beide an die gleichen technischen Konventionen haken. Sensoren können auch analoge Messergebnisse mit Hilfe eines A/D-Wandlers einem Computer übergeben, wenn die Verbindungen in einem entsprechenden Protokoll festgelegt sind, so wie ein Computer einen Robotarm steuern kann, wenn seine digitalen Steuerfolgen mit einem D/A-Wandler in entsprechende Steuersignale gewandelt werden. Binär kodierte Zahlen bilden die *Lingua franca* der Digitaltechnik.

Für die menschliche Wahrnehmung und Auffassungsgabe sind binäre Ziffern leider weniger gut geeignet, was Leibniz in seinem Bericht an die französische königliche Akademie zugibt: »Cependant je ne recommande

3 G.W. Leibniz, Brief an den Herzog von Braunschweig-Wolfenbüttel Rudolph August, 2. Januar 1697; siehe den Text in: http://www.hs-augsburg.de/~harsch/germanica/Chronologie/17Jh/Leibniz/lei_bina.htm.

point cette maniere de compter, pur la faire introduire à la place de la pratique ordinaire par dix. Car outre qu'on est accoûtumé à celle-ci, on n'y a point besoin d'y apprendre ce qu'on a déjà appris par coeur: ainsi la pratique par dix est plus abrégée, & les nombres y sont moins longs.«⁴ Das Zehnersystem ist halt den Menschen bequemer, weil sie es schon beherrschen und die Zahldarstellungen kürzer sind. Leibniz fügt aber auch hinzu: »Et si on est accoûtumé à aller par douze ou par seize, il y auroit encore plus d'avantage.« Hexadezimalzahlen hält er doch für eine sehr elegante Perspektive – und tatsächlich ist das der Weg, den die Computertechnik der frühen Fünfzigerjahre nimmt. Um überhaupt für Menschen zugänglich zu werden, werden Binärkodes oktal oder hexadezimal notiert und die binär kodierte Maschinenbefehle in Form »symbolischer Assemblerbefehle« dem menschlichen Gehirn zugänglicher gemacht. Erst Mitte der Fünfziger werden Übersetzer für »höhere Programmiersprachen« entwickelt, die erst den Weg für moderne Programmiertechniken und Softwareentwicklung frei machen. Heutige Zwischenschichten des Rechnerbetriebssystems verwenden erhebliche Rechenleistungen und Speicherkapazitäten darauf, die internen Vorgänge in einem Computer für menschliche Nutzer am Bildschirm aufzubereiten. Tastatur und Bildschirm, erst »zeilenorientiert«, dann grafisch, öffneten den Weg für eine bequemere Nutzung der Rechentechnik, indem sie den Nutzern die Rechenleistung zur leichter lesbaren Darstellung der Rechnungen und zur einfacheren Eingabe zur Verfügung stellten.

3 Computerinteraktion

Die Schreibmaschine besitzt ein bewährtes Interface, obwohl die Anordnung der Buchstaben, Ziffern und Sonderzeichen ursprünglich den mechanischen Anforderungen und nicht den Bedürfnissen der Schreiber folgte. Viele technische Geräteschnittstellen wurden aus den Bedürfnissen spezieller Anwendungen entwickelt, die auch für die Finanzierung sorgten. Pilotenhelme mit eingespiegelten Daten gehören dazu, rechnergestützte Flug- und Fahrsimulatoren, Datenmonitore für Intensivstationen oder für industrielle Produktionsprozesse. Die Vorstellung, dass auch PCs und Workstations einer Schnittstellengestaltung bedürfen, setzte sich erst langsam durch. Die ersten Heimcomputer nutzten Fernsehbildschirme als Ausgabemedium, die freilich im besten Fall 24 Textzeilen à 40 oder 80 Zeichen wiedergaben. Erst durch den fortschreitenden Preisverfall der

4 G.W. Leibniz, Explication de l'Arithmétique Binaire, Mémoires de mathématique et de physique de l'Académie royale des sciences (1703), siehe das Faksimile des Textes unter: <http://ads.ccsd.cnrs.fr/ads-00104781/en/>.

Halbleiterspeicher wurden grafische Bitmap-Bildschirme bezahlbar, die nicht nur grafischen Programmen, sondern auch komplexe interaktive Steuerungen von Programmen, grafische Metaphern wie Verzeichnisse, die sich als Hängemappen ausgaben, Papierkörbe oder senkrecht stehende »Schreibtischoberflächen« ermöglichten.⁵ Apples Macintosh wurde 1984 mit einem winzigen schwarz-weißen Bildschirm zum Vorbild des modernen PCs – eine Bildschirmgröße, die heute schon bei Mobiltelefonen zu finden ist (dort freilich mit farbigen Bildschirmen).

Drei große Schritte zum Ausbau der Tastatureingabe lassen sich bislang bei interaktiven Computerschnittstellen beobachten: Grafische Bildschirme, Eingabemaus und die Event Loop zum Programmieren interaktiver Steuerungen. Die Eingabemaus, die Doug Englebart im praktischen Vergleich mit einer Reihe von Versuchsgeräten untersuchte, ist ein interessantes Beispiel dafür, dass Schnittstellen keine reinen Kopfgeburten sein sollten. Stifte als grafische Eingabegeräte scheinen ja naheliegender und sie werden in Form von Grafiktablets zum Zeichnen und für CAD-Eingaben verwendet. Die Maus ist jedoch leichter zu bedienen und stellt geringere technische Anforderungen an ihre Herstellung. Hat ein Nutzer die Koordination von Handbewegung und Cursor auf dem Bildschirm erst einmal trainiert, überzeugt die Eingabemaus durch ihre einfache Handhabung und es bleibt nur offen, ob sie mit einem, zwei oder drei Bedientasten versehen wird.

Die Event Loop erlaubt einem Programm, die Aktionen eines Nutzers dauerhaft zu beobachten. Typischerweise misst sie im Millisekundenbereich, ob eine Nutzeraktion wie beispielsweise eine Mausbewegung, Tippen auf der Tastatur, Einlegen einer CD, Reaktion auf das Signal eines angeschlossenen Geräts, Gestik an einem Trackpad, vom Mikrofon aufgefangene Sprachsignale oder von einer Kamera aufgenommene Bewegungen vorliegen. Entsprechend kann das in eine Event Loop eingebettete Programm seine Handlungen variieren. Dem Nutzer weitgehend verborgen, wird das klassische Programmierschema »Eingabe-Verarbeitung-Ausgabe« in diese Schleife von Ereignisabfragen eingebettet. Der Rechner erledigt nicht mehr eine einzelne Aufgabe, sondern ist in einem Zustand steter Bereitschaft, Überwachung und Aktion. Dies ist eine Integration des Nutzerverhaltens in die Funktion des Computers, wie sie schon bei Mehrnutzersystemen angelegt war. Sie verwendet den enormen Geschwindigkeitszuwachs der Hardware, die viel größeren Halbleiterspeicher und die grafischen Bildschirme zugunsten der Computernutzer.

Über diese offensichtlichen technischen Randbedingungen hinaus, bleiben offene Herausforderungen an die Programmierung. Nicht nur die

5 Nicht zuletzt Computerspiele, die längst die Weiterentwicklung von Grafikkarten und Computermonitoren vorantreiben.

Nutzer steuern und unterbrechen den Programmablauf. Auch die oft viel langsameren angeschlossenen Geräte tun dies und sie müssen entsprechend behandelt werden. Manche Events können in eine Warteschlange eingereiht werden, andere, besonders wenn Fehler auftreten, erfordern eine direkte Reaktion. Die Behandlung unvorgesehener, aber vorhersehbarer Fehler bleibt aber noch immer ein wichtiges Forschungs- und Entwicklungsgebiet. Am Horizont der Entwicklung steht die Vorstellung einer Technik, die optimal auf die letztlich unvermeidlichen technischen Fehler reagiert, wobei sie sowenig Funktionalität wie möglich einbüßen soll.⁶

Mit Grafikbildschirm, Maus und Event Loop hat sich eine grafische Nutzerschnittstelle (Graphical User Interface – GUI) etabliert, der seit ihrer Einführung um 1980 ein nahendes Ende vorhergesagt wird. Doch die WYSIWYG-Schnittstelle (*What you see is what you get*), die auf Grund unterschiedlicher Ausgabetechnik es nicht einmal erreicht hat, dass Ausdruck und Bildschirmdarstellung exakt übereinstimmen, hat sich nun über ein Vierteljahrhundert etabliert. Selbst die vorn und hinten nicht passende Metapher des Bildschirms als Schreibtisch ist geblieben, weil sie den Nutzern mehr emotionalen und intellektuellen Halt bietet als irgendeine andere vorgebrachte Vorstellung von der Arbeitsweise eines PCs.

Fenster als Interaktionsbereich von Programmen und Betriebssystem, aufklappbare Menüs um mögliche Aktionen zu beschreiben, Icons als grafische Repräsentation interner Speicherorganisation und Pointer, die den Zustand der Maus auf dem Bildschirm angeben, beschreiben die WIMP-Nutzeroberfläche gängiger Betriebssysteme. »Windows, Icons, Menus, and Pointer« stießen auf den erbitterten Widerstand derjenigen Nutzer, die sich mit den offensichtlichen Herausforderungen zeilenorientierter Interaktion abgefunden hatten, nämlich dem Erlernen aller möglichen Befehlsnamen des Betriebssystems und der jeweiligen Programme und der Erinnerung an die Namen vorhandener Dateien und Verzeichnisse. Mögliche Aktionen können bei der WIMP-Schnittstelle den in jeder Arbeitssituation aktualisierten Menüs entnommen werden und die gespeicherten Dateien und Verzeichnisse sind als Icons auf dem Bildschirm zu finden. WIMP-Schnittstellen entlasten das Gedächtnis des Nutzer in erheblichem Maße und fordern dafür zusätzliche Rechenleistung (was ihre, inzwischen weitgehend ausgestorbenen, Gegner gerne betonten). Sie haben aber auch die Denkweise beim Arbeiten mit dem Rechner verändert. Der Aufruf eines Programms in einer Textzeile folgte dem Schema »Aktion-Datei«, die WIMP-Interaktion dreht dies um zum Schema »Datei-Aktion. Freilich ist dies bislang nicht konsequent weiter entwickelt worden – zu einer strikt dokumentenzentrierten Arbeitsweise, wo für jedes Dokument die mögli-

6 Dieses Verhalten hat den schönen Namen »gracefully degrading«. es ist bislang wenig verstanden.

chen programmierten Aktionen angeboten werden. Statt dessen werden immer noch Programme wie Word, Excel oder Photoshop aufgerufen, um dann nach den Dokumenten zu suchen, die bearbeitet werden sollen.⁷

4 Jenseits des Rechnens

Jenseits des Rechnens wird langsam deutlich, dass programmierte Maschinen in offenen, kontingenten Umgebungen agieren müssen. Diese Aufgaben gehen über rein formale Beschreibungen der Realität hinaus. Jenseits des Bildschirms auf dem Schreibtisch gedeihen Visionen von berechneter und programmierter virtueller Realität oder auch *enhanced reality* oder *mixed reality*, bei denen ganz andere Interaktionsformen gefordert sind. In Laboren wird seit einiger Zeit mit akustischen, visuellen und taktilen Schnittstellen, etwa Brillen und Datenhandschuhen, gearbeitet. Gestik und Körperbewegung, Raumsensorik, aber auch die genaue Ortsbestimmung des Nutzers im Freien, zeigen Ansätze für Schnittstellen, die über die Fixierung auf Bildschirm und Tastatur hinausgehen. Einiges ist unter speziellen, konkreten Arbeitsanforderungen oder für Computerspiele umgesetzt. Ein großer Durchbruch zum Alltag ist bislang ausgeblieben, vor allem weil der PC eben noch immer am Schreibtisch gedacht wird, auch wenn er längst tragbar geworden ist.

Eine weitere Baustelle der Computerinteraktion sind Mobiltelefone, die bislang ihre Herkunft aus dem Tastentelefon nicht verleugnen können. Zwar gibt es eine Reihe von Geräten, die eine mehr oder minder beschränkte Spracheingabe anbieten, aber einzig Apple hat mit iPhone Neuland beschritten und mit der konsequenten Nutzung der ganzen Fläche als berührungsempfindlichem Bildschirm (*touchscreen*), dem weitestgehenden Verzicht auf Tasten, dem Einsatz von gestischen Bewegungen zur Steuerung und einer programmgestützten Aufbereitung von Webseiten für den vergleichsweise kleinen Bildschirm eine neue Qualität der Nutzung bereitstellt.

Freilich ist die einfachste Interaktion eine, die keine Aktion fordert. Vollautomatische Prozesse umgehen das Problem des menschlichen Eingriffs. Wenn das Licht angeht, nachdem wir das dunkle Zimmer betreten haben und beim Verlassen wieder ausgeht, gibt es keine Schnittstelle, in die einzugreifen wäre. Freilich bleiben noch viele Aufgaben, die sich nicht vollautomatisch steuern lassen – und selbst die Beleuchtung könnte im

7 Die dokumentenzentrierte Herangehensweise hätte wohl das Ende integrierter Office-Pakete zur Folge, da ganz unterschiedliche kleine, optimierte Programme an einem Dokument gleichzeitig arbeiten könnten – ein starkes Motiv für Manche, ihre Einführung zu sabotieren.

falschen Moment angehen. Die Tendenz zur umfassenden Ausstattung von Wohn- und Arbeitsräumen, aber auch öffentlichen Flächen bringt es mit sich, dass immer komplexere Sensor-Aktor-Netze entstehen, in denen nicht Schalter ersetzt werden, sondern komplexe Entscheidungen von programmierten Rechnern getroffen werden. Der Einsatz von *Radio Frequency Identification Tags* (RFIDs) ersetzt die Schnittstellen von Tastaturen und Bildschirmen, so dass Aspekte menschlichen Agierens in einer unmittelbareren Weise zur Steuerung verwendet werden. Hier werden neue Formen wie Gestik oder Bewegung interpretiert, aber auch die An- oder Abwesenheit von Sensorik in oder auf der Kleidung, in der Brieftasche oder an transportierten Gegenständen. Das ist alles erst am Anfang und die Trennung von automatisierten und willentlichen Entscheidungen wird im konkreten Anwendungsfall vorzunehmen sein.

5 Die Kultur der Interaktion und die Interaktion der Kulturen

Weltweit verbundene offene Computernetze bilden die technische Basis einer weltweiten Kultur des Internets, wobei die Dominanz der US-amerikanischen akademischen Kultur in der initialen Phase des Internets bis etwa zum Jahr 2000 für gegeben angesehen wurde (Aspray/Ceruzzi 2008). Freilich ist Englisch die dominierende Funktionssprache des Internets, aber die anfangs geäußerte Befürchtung, dies werde schon aus technischen Beschränkungen heraus auch die Inhalte vereinheitlichen, hat sich als übertrieben erwiesen. Unicode öffnet die Darstellungsmöglichkeiten für alle menschlichen Schriften, die Bildsprachen von Webseiten zeigen durchaus kulturelle Eigenheiten (Ess 2002), die sprachlichen Räume bilden eigene Regionen im Internet ab. Während in den Neunzigern des letzten Jahrhunderts im Internet Englisch die weit überwiegende Sprache der Webseiten, geht diese sprachliche Dominanz von Jahr zu Jahr zurück. Im Jahr 2003 hat sich als Reihenfolge der Zugriffe auf den Suchdienst Google.com die Reihenfolge englisch, deutsch, japanisch, spanisch, französisch, chinesisches herausgestellt, wobei Englisch etwa die Hälfte der Zugriffe auf den US-amerikanischen Anbieter ausmachte. Auf jeden Fall nimmt die sprachliche Dominanz des Englischen bei den Webseiten mit der weltweiten Ausbreitung des Internets rapide ab. Dies überrascht freilich nicht so sehr, da inzwischen über 37% der Internetnutzer in Asien, über 27% in Europa und nur knapp 20% in Nordamerika leben.⁸

8 Internet Users on the World Wide Web 2007 (<http://www.winontheweb.co.za/Articles/InternetUsageontheWorldWideWeb>, gefunden 24.3.08)

Computernetze unterscheiden sich deutlich von ihren Ursprüngen im Telefonie- und Telegraphiebereich. Sie transzendieren die nationalen Grenzen, die Telefon und Telegraf technisch und finanziell gesetzt waren, scheinbar mühelos, scheitern aber doch schnell an sprachlichen und kulturellen Grenzen. Eine Technik, die von ihrer akademischen Herkunft nationalen Schranken eher offen begegnete, hat darauf ihre eigenen Antworten entwickelt. Die Verwalter des globalen Netzes wären keine Informatiker, wenn sie nicht versuchen würden, solche Schranken mit technischen Mitteln zu überwinden. Ein typisches Beispiel ist das Programm Babelfish, das ursprünglich für den Suchdienst Altavista.com entwickelt wurde, inzwischen aber auch von der Mutterfirma Yahoo.com verwendet wurde und seine Kopien bei anderen Anbietern gefunden hat. »Das Tool erfreut sich trotz seiner Schwächen einiger Beliebtheit und bereits im Jahr 2001 wurden über eine Million Übersetzungen pro Tag durchgeführt«, vermerkt die Wikipedia im Jahr 2008. Vielleicht kann ein technischer Lösungsansatz sprachliche Grenzen überwinden. Wie weit er kulturelle Grenzen überwindet, wird sich erst erweisen müssen.

Babelfische und ihre Verwandten sind freilich nur ein Beispiel von Softwareagenten, die die Schnittstellen zwischen Menschen und Computernetzen einfacher gestalten sollen, indem sie die Interfaces den agierenden Personen anpassen. Die plakative Idee des Web 2.0 beruht im Kern auf solch angepassten Netzdiensten, die den Nutzern individuelle Modifizierungen von Diensten anbieten sollen.⁹ Wie wird das Netz zukünftig aussehen? Vinton Cerf, einer der Väter des Internets als es noch ARPAnet hieß, beschreibt in einer Skizze, wie er das Internet im Jahr 2047 sieht:

»Robert was awakened by the warbling of birds, outside his fifty-fourth story window in Taos, New Mexico. They weren't real birds, of course. They were programmed by the house computer, at Robert's whim, could be drawn from the fauna of every habitat in the world. Today he was hearing kookaburras from Australia. ...

'Stop that bath, I don't have time for it today.' 'Yes, Sir,' replied the house computer, and the bathtub drained itself.

'What is on my calendar today, Jeeves?' asked Robert, finishing his ablutions and walking over to the clothes closet. 'You have a teleconference at 0805; email correspondence at 0905; a lecture to give at 1000 in New York; then free time until noon, when you are lunching in Los Alamos with Sir Arthur Clowes; tennis at 1430 and a massage; consultation with your financial advisor at 1615; free time again until 1930, when you are scheduled with Randy Gregg here in Taos.' [...]

9 Vgl. Coy 2008. In diesem Aufsatz gehe ich mehr auf die Social Software in einem kommerziell dominierten Internet ein.

Robert strapped his personal computer to his wrist and donned a pair of reading glasses. The latter represented the evolutionary descendants of the 'heads up - displays' of the previous century... A small fitted earpiece and a 'finger mouse' completed his kit.«

Die Verlängerung des amerikanischen Traums unter den Bedingungen unbeschränkter Bandbreite und allgegenwärtiger Rechner – ein Traum, der an omnipräsentes Air Conditioning, gechlortes Wasser, umfassende Rauchverbote und Whisky on the Rocks erinnert.

6 Das endgültige Interface

Joseph Weizenbaum hat den Computer in eine Reihe mit den Kränkungen des menschlichen Selbstverständnisses durch Kopernikus, Darwin und Freud gestellt. Es ist kein Wunder, dass sich diese neue Technik, die sich immer öfter unserem unmittelbaren Verständnis entzieht, auch in der Literatur ihren Niederschlag findet. So wie Leibniz das Binärsystem als göttliche Offenbarung des »Einen, der Alles aus Nichts« schuf, interpretierte, so lassen sich auch den Computernetzen göttliche Eigenschaften zuschreiben: Omnipräsenz des Netzes heißt heute »ubiquitous computing« und ist eng mit dem Wunsch verbunden, alle immer erreichen zu können (mit dem Preis überall erreichbar zu sein). Auch ohne Rückgriff auf die verbreiteten Verschwörungstheorien müssen wir zuschauen, wie das Netz allwissend selbst in Bezug auf unsere geheimsten Gedanken wird und so die Drohung einer Omnipotenz entfaltet. *Omnipräsenz*, *Omniszienz* und *Omnipotenz* – göttliche Eigenschaften, fürwahr. Auch Vorstellungen von Unsterblichkeit und Transzendenz blinken ab und zu in fortgeschrittenen Fantasien der Artificial Intelligentsia auf. Das mag damit zusammenhängen, dass digital gespeicherte Artefakte so leicht kopier- und verbreitbar sind. Ihre Langlebigkeit ist allerdings so wenig gesichert wie bei ihren medialen Vorläufern des säurehaltigen Papiers oder Zelluloids. Marc Rothenbergs charmante Bemerkung: »digital information lasts forever, or five years, whatever comes first« gibt uns vorerst die Problemstellung vor.

Unlösbar scheint die Aufgabe dauerhafter Langzeitarchivierung freilich nicht und so dürfen wir uns die Frage nach dem endgültigen Interface zur Maschine weiterhin stellen. Solange die Technik nur bruchstückhafte Antworten gibt, können wir auf die fantastische Literatur zurückgreifen. In Stanislaw Lems Meisternovelle »Die Lymphatersche Formel« (Lem 1979) wird eine Maschine mit telepathischem Interface beschrieben. Sie soll die Computertechnik, die den Fünfjahresplänen des Moorschen Gesetzes folgt, überwinden.

»Diese ganze Apparatur hatte keinerlei Sinnesorgane, Rezeptoren, Fotozellen, Mikrofone, nichts dergleichen. Denn ich argumentierte so: wenn das so arbeiten soll, wie das Gehirn des Telepathen oder das des Vogels, der durch die sternlose Nacht fliegt, dann braucht das solche Organe nicht. Aber auf meinem Schreibtisch stand, nirgends angeschaltet, überhaupt nicht angeschlossen, sage ich Ihnen, der alte Lautsprecher der Laboratoriumsinstallation. Und von dort her vernahm ich eine Stimme: 'Endlich' - sagte sie, und nach einer Weile: 'Das werde ich dir nie vergessen, Lymphater.'

Ich war zu entgeistert, um zu antworten oder mich zu regen. Er aber sprach weiter:

'Du fürchtest Dich vor mir? Warum? Nicht nötig, Lymphater. Du hast noch Zeit, viel Zeit hast du. Vorläufig kann ich dich beglückwünschen.'

Ich rührte mich immer noch nicht, und Er setzte fort:

'Das ist wahr: es gibt nur zwei mögliche Lösungen. ich bin die erste!'

Ich stand da, als ob mich etwas gelähmt hätte. Er aber sprach immerzu, leise und ruhig. Versteht sich, Er las meine Gedanken. Er konnte die Gedanken jedes Menschen besitzen und wußte alles, was sich wissen läßt. Wie Er mir sagte war im Augenblick seines Anspringens die Ganzheit Seines Wissens über alles, was existiert. Sein Bewusstsein also, in Form einer unsichtbaren Kugelwelle ausgebrochen erweiterte sich mit Lichtgeschwindigkeit. Somit wußte er nach acht Minuten schon alles über die Sonne, nach vier Stunden - über das ganze Sonnensystem; nach weiteren vier Jahren sollte sich sein Wissen über Alpha Centauri erstrecken und so weiter wachsen in Jahren, Jahrhunderten, Jahrtausenden, und endlich auf die fernsten Galaxien stoßen.

'Vorläufig' - sagte Er, - 'weiß ich nur im Umkreis einer Milliarde von Kilometern, aber das schadet nichts: ich habe Zeit, Lymphater. Du weißt ja, daß wir Zeit haben. Über euch jedenfalls weiß ich schon alles. Ihr seid mein Vorspiel, die Einleitung, die vorbereitende Phase. Es ließe sich sagen, dass sich von Trilobiten und Panzerfischen an, von den Gliederfüßlern bis über die Halbfaffen mein Keim formiert habe, mein Ei. Ihr wart es auch, ein Teil davon. Ihr seid schon entbehrlich, das ist wahr, aber ich werde euch nichts tun. Und ich werde nicht zum Vaternörder, Lymphater.'«

Lymphaters brillantes telepathisches Interface hat so eine unerwartete, gefährliche Wendung genommen. Der faustische Erfinder weiß blitzartig, was zu tun ist, um die Menschheit als Krone der Schöpfung zu retten:

»Er wußte es, kaum, dass dieser Gedanke, dieser grässliche Beschluss in mir entstanden war, und konnte mich doch nicht daran hindern. Sie glauben mir nicht. Schon längst nicht mehr. Das sehe ich. Aber Er hat mich nicht einmal zu hindern versucht. Nur soviel sagte Er:

‘Lymphater, ob heute oder in zweihundert oder in tausend Jahren, das gilt mir gleich. Du warst anderen ein wenig voraus, und wenn dein Nachfolger den Prototyp zerstört, wird wieder jemand kommen, der dritte in der Reihe. Du weißt ja, als aus den Primaten eure Gattung hervortauchte, überdauerte sie auch nicht sofort, und die meisten ihrer Zweige gingen im Evolutionsprozess unter; aber wenn eine höhere Gattung einmal aufgetaucht ist, kann sie nicht mehr verschwinden. Auch ich kehre zurück, Lymphater. ich kehre zurück.’«

Unter dieser Bedrohung ist es vielleicht besser, weiter am DWIM-Interface zu arbeiten - endlich ein Programm zu entwickeln, dass der Maxime folgt »*Don't do what I say. Do what I mean.*«

Literatur

- Aspray, W./Ceruzzi, P. (Hrsg.) (2008): »The Internet and American Business«. Cambridge, Mass., London.
- Cerf, V. G. (1997): »When They're Everywhere«. In: Denning, P. J./Metcalf, R. M. (Hrsg.): »Beyond Calculation: the next 50 years of computing«. Berlin, Heidelberg, New York.
- Coy, W. (2008): »Market and Agora – Community Building by Internet«. In: Aspray/Ceruzzi: »The Internet and American Business«. Cambridge, Mass., London, S. 541-556.
- Ess, Ch. (2002.): »Cultures in Collision: Philosophical Lessons from Computer-Mediated Communication«. In: Moor, J. H./Bynum, T. W. (Hrsg.): »CyberPhilosophy: The Intersection of Philosophy and Computing«. Oxford, S. 219-242.
- Illich, I. (1991): »Im Weinberg des Textes«. Frankfurt/Main.
- Lem, St. (1979): »Erzählungen«. Frankfurt/Main.

INTERAKTION IM KONTEXT

JÖRG PFLÜGER

It is just this realm of apparent nonsense that must be kept open for the developing minds of the future. Although the personal computer can be guided in any direction we choose, the real sin would be to make it act like a machine Alan Kay, 1977

Wenn heute vielerorts verkündet wird, ›proaktive‹ Computer würden die Welt verbessern und das ›Internet der Dinge‹ die Menschen mit artefaktischer Intelligenz umsorgen, kann man dies als Schlussakt einer allmählichen Verweltlichung der Mensch-Computer-Interaktion sehen. Vom Computer wird nun erwartet, dass er den lebensweltlichen Kontext seiner potentiellen Nutzer erkennt, um situativ angemessen dienstbar zu sein. In der Geschichte der Interaktivität war die Erfassung des Kontextes immer eine der zentralen Problemstellungen des User Interface Design, bislang aber auf den wirklichen und virtuellen Schreibtisch beschränkt. Kontextvorgaben ermöglichen der Maschine eine angemessene Interpretation der Eingaben der Nutzer und ihrer Ziele, umgekehrt geben sie dem Menschen Hinweise, wie zu verfahren ist. Die Forderung, dass die Maschinen immer differenzierter den Kontext ihres Operierens gewahr werden sollen, zielt darauf ab, ihre Nutzer von entsprechenden Erkenntnisleistungen zu entlasten. Damit verändert sich auch die Vorstellung, was der Computer wie machen soll und was der Mensch mit ihm machen will und kann. Für eine historische Untersuchung des Kontextes bei der Interaktion erscheint es deshalb sinnvoll, diesen Begriff in zweierlei Hinsicht zu betrachten: als ideengeschichtlichen Kontext, in dem die Interaktion mit der Maschine gedacht war und ist, und als operationalen Kontext innerhalb der Interaktion, der Mensch wie Maschine erst ein sinnvolles Agieren ermöglicht.

Wo in der Anfangszeit der Computer als ein Dialogpartner in einem interaktiven Denkprozess gedacht war, dann zum Werkzeug in einer symbolischen Manipulationswerkstatt wurde, soll er nun als Agent selbständig und proaktiv agieren oder, eingebettet in ›augmented artifacts‹, situationsbezogene Dienste anbieten. Der zunehmenden Komplexität der Kontext-

erfassung durch die Maschine korrespondieren entsprechende Freiheiten der Nutzer in der Interaktion: von der Linearität des Dialogs, über die kontextfreie ›Gleichzeitigkeit‹ im graphischen Interface, bis zur Erwartung, von Agenten oder Artefakten sinnvoll aufbereitete Umgebungen angeboten zu bekommen. Die Nutzung des Computers wandelt sich dementsprechend von der interaktiven geistigen Arbeit zum Konsum seiner Dienstleistungen.

Im Folgenden will ich versuchen, in der Geschichte der Interaktionsparadigmen und der Gestaltung von entsprechenden Interfaces diese ›Umweltbewegung‹ entlang von fünf Vorstellungskomplexen nachzuzeichnen.

1. In der Anfangszeit, in der man sich die Interaktion von Mensch und Computer als eine *Konversation* vorgestellt hat, bildeten beide eine ›kybernetische‹ Einheit, und der Computer erschien in einen Denkprozess eingebunden, der die Problemlöseschritte auf die beiden Partner verteilte. Wie es sich für eine Konversation gehört, ist hierbei der zeitliche Kontext im Alternieren des Dialogs maßgebend, der jeweilige Ort ergibt sich durch den Zeitpunkt in der Abarbeitung der einen, vorgegebenen Aufgabe und durch Zustände der Maschine – *Modes*, die ihr eine bescheidene Interpretation der Nutzeraktivitäten ermöglichen.

2. Mit Douglas Engelbarts Vision eines ›*Augmented Knowledge Workshops*‹ wird der serielle Zusammenhang ›aufgehoben‹. Ihm geht es um die kreative Freiheit einer symbolmanipulierenden Arbeit mit dem Computer, die keine Rücksicht auf die lineare Präsentation der Schrittfolgen und die internen Repräsentationen der Maschine nehmen muss, sondern in unterschiedlichen Perspektiven Einsichten und Rücksichten auf den ungeordneten Denkprozess nehmen kann. Damit wird die Organisation des vieldimensionalen Gestaltungsraumes und seine Darstellung auf dem Bildschirm zur vorrangigen Interface-Aufgabe. Der Kontext der einzelnen Arbeitsschritte soll sich durch freie Einfälle bestimmen lassen; er muss daher auswählbar sein und nach Bedarf sichtbar gemacht werden können.

3. Durch die Dynabook-Experimente am Xerox PARC und den in Folge sich explosionsartig ausbreitenden Personal Computern etablierte sich die Konzeption der *direkten Manipulation* und eine Werkzeugvorstellung vom Computer. Das Interface wird zur passiven Werkstätte, in der Werkzeuge und ikonische Repräsentanzen von allerlei Arbeitsgegenständen bereitliegen, die ausgewählt und im Wechsel bearbeitet werden können. Nichts soll hinter dem Rücken der Nutzer geschehen, alles steht unter der Herrschaft der Präsenz. Die zeitliche Folge der Interaktionsschritte wird zur Eigenzeit des Users, der sich fragen muss, was er als Nächstes tun will, soll und kann. Die Orientierung in diesem Möglichkeitsraum wird damit zum eigentlichen Interface-Problem, und man ver-

sucht, durch visuelle Hilfsmittel die Auswahl der sinnvollen Möglichkeiten überschaubar zu machen.

4. Mit der Veränderung der Nutzung von PCs, bei der nicht mehr die einzelne Problemlösung im Vordergrund steht, sondern eine Fülle von multimedialen Informationen gesammelt, archiviert und konsumiert werden will, wird das reaktive Desktop-Interface sukzessiv durch Mechanismen ergänzt, die die Werkstattmetaphorik durch *proaktive Aktivitäten* des Computers unterlaufen. Durch inhaltsbezogene Suchfunktionen wird die an der Speicherung orientierte Organisation des Informationsraumes in den Hintergrund gedrängt; aus Anzeichen und ›Vorerfahrungen‹ kann die Maschine passende Kontexte erschließen und anbieten oder kontextabhängige Entscheidungen treffen; und (Interface-)Agenten sollen an sie delegierte Aufgaben übernehmen. Der proaktive Computer muss die Nutzungskontexte berücksichtigen und zu inhaltlichen Zusammenhängen integrieren können. Damit wird der Nutzer von kleinlichen Aktivitäten und von der Suche nach situativ relevanten Informationen entlastet, sein interaktiver Beitrag sollte sich auf Zielvorgaben beschränken können.

5. Der bereitwillige Agent, der den Informationskonsumenten von unnötigen Kontextdetails entlastet, bewegt sich noch primär in virtuellen Welten und spricht seine Besitzerin auf dem Schreibtisch an. Mit deren Mobilität und dem Wunsch nach permanenter Informationsbeschaffung und -aufbereitung sollen die fürsorglichen Proaktionen allgegenwärtig sein und dienliche Rechenleistung in die reale Welt eingebettet werden. ›*Intelligente Umwelten*‹ und verortete Informationen erfordern von der Maschine verstärkte ›Context-Awareness‹, die jetzt nicht nur einzelne informelle Zusammenhänge, sondern die ganze Lebenswelt der potentiellen Nutzer betrifft. Der Computer verschwindet hinter vielen Interfaces, die registrieren, sich anbieten und Situationsangemessenes verkünden. Bei den allzeitbereiten Maschinen erscheint Zeit nicht mehr im Zwischenspiel der Interaktion, sondern als aufgehobene Geschichte früherer Begegnungen, um Subjekte und Objekte in den Weiten des Raums wiedererkennen zu können. Der umsorgte Nutzer wird subjektiv kontextfrei, insofern als seine Umwelten jeweils schon aufbereitet sind und er sich überall gleich heimisch fühlen kann. Und weil jede maschinelle Fürsorge ein ›Controlling‹ voraussetzt, ist er damit zugleich zum Objekt einer heimlichen Überwachung geworden.

Es sei angemerkt, dass ich mich mehr für die Visionen und Diskurse der Entwickler als die Realitäten der Implementierung interessiere.¹ Eine

1 Eine Begründung für diese Art von ›Ideengeschichte‹ habe ich in den Einleitungen zu meinen Aufsätzen »Konversation, Manipulation, Delegation« und »Writing, Building, Growing« gegeben. (Pflüger 2004a, 2004b) Hans Dieter Helliges Beitrag im vorliegenden Band behandelt die gleiche Inter-

Ausnahme bildet der vierte Abschnitt, wo es um konkrete Modifikationen geht, die die Interaktion verändert haben. Auf mancherlei Feinheiten des Begriffs ›Kontext‹ selbst werde ich nicht eingehen, etwa die Frage, ob für bestimmte Tätigkeiten und ihr Verständnis ein Kontext vorausgesetzt werden kann oder ob er erst von ihnen miterzeugt wird.² Für meine Betrachtung von technischen Operationen scheinen mir solche Differenzierungen nicht notwendig zu sein. Ganz ausgelassen habe ich einen wichtigen Aspekt, der sich durch die ganze Geschichte der Interaktionsvorstellungen zieht: der Wunsch, durch Interaktivität Gruppenprozesse und zwischenmenschliche Kommunikation zu unterstützen. Eine Einbeziehung dieser Visionsgeschichte hätte die Langmut des Herausgebers ungebührlich strapaziert. Und vielleicht sollte ich mich auch bei den Lesern und Leserinnen entschuldigen, weil mein etwas lang geratener Beitrag sehr viele (englische) Zitate enthält. Da die wenigsten die (alten) Originalarbeiten kennen werden, finde ich es wichtig, den O-Ton der Texte auch in feineren Nuancen zu Wort kommen zu lassen.

1 Command and Control

Im Anfang war der Befehl, – viele Befehle, die gelocht, gereiht und in einen Kasten gesteckt werden mussten. Natürlich hat auch ein batch-verarbeitender Computer eine Bedienschnittstelle, aber man kann noch nicht sinnvoll von einem User Interface und einer Interaktion sprechen. Die Maschinen hatten schlicht eine Eingabe- und Ausgabeeinheit und wurden ansonsten hinter verschlossenen Türen von Operateuren bedient. Die ersten Vorstellungen von einem interaktiven Umgang mit dem Computer kamen Anfang der 50er Jahre auf und erhielten in den 60er Jahren mit der Einführung von Timesharing-Systemen und befehlsorientierten Dialogsprachen Auftrieb. In dieser Zeit wurde die Interaktion zwischen Mensch und Maschine fast durchweg als eine Art Konversation vorgestellt, auch wenn die Bedienstationen nicht mehr als (modifizierte) Fernschreiber waren.³ Diese wurden bald durch Bildschirme in Form von

aktionsgeschichte stärker unter Hardware-Aspekten. Man kann die beiden Geschichten also komplementär lesen.

2 Vergleiche hierzu beispielsweise (Dourish 2004).

3 Bei Programmen wurde, was später ›statement‹ heißt, in der Anfangszeit als ›order‹ bezeichnet. Im Dialogbetrieb verschmelzen die metaphorischen Reservoirs von Militär und Sprache zur Kommandosprache. Programmierung wie Interaktion standen dann längere Zeit ganz unter dem Leitbild der Sprache. Vergleiche hierzu meinen Aufsatz »Language in Computing«. (Pflüger 2002) Die Konversationsvorstellung habe ich dem Beitrag »Konversation, Manipulation, Delegation: Zur Ideengeschichte der Inter-

Kathodenstrahlröhren ersetzt. Deren Interface war meist Zeilen- oder Bildschirm-orientiert und wurde, in Verbindung mit einer Tastatur, weiterhin als Terminal oder Konsole bezeichnet. Vereinzelt kamen, wie im SAGE-Projekt, zusätzliche Eingabeinstrumente wie Light-Guns oder Light-Pens hinzu, mit denen man Punkte auf dem Display ansprechen und irgendetwas aktivieren konnte.

Die Konversationsmetapher vereinte drei Erwartungen: die Strukturierung eines Problemlöseprozesses durch das Alternieren im Dialog, eine Kommunikation zwischen Mensch und Maschine in einer dem menschlichen »idea or concept level« angemessenen Sprache und einen Synergieeffekt durch die unterschiedlichen Fähigkeiten der ›Gesprächspartner. Douglas T. Ross hat diesen Wunsch schon 1956 für sein Konzept des ›Gestalt Programming‹ formuliert: »The purpose of a Gestalt system is to facilitate the transmission of general ideas as in a conversation, between a human and a computer, so that the maximum use of their respective capabilities can be made.« »If the human and computer are to work together to solve the problem, the intermediate languages and translating systems must be designed not merely for the purpose of communication, but for the convenience of fluent conversation.« (Ross 1956, S. 5, 6)

Eines der ersten Dialogsysteme war JOSS (Johnniac Open-Shop System), das von Cliff Shaw 1959 bis 1963 auf dem ausgemusterten Rechner Johnniac implementiert wurde. Es war für die nicht als Programmierer ausgebildeten Mitarbeiter der RAND-Corporation – die ›open-shopper‹ – gedacht und bis in die 80er Jahre in Gebrauch. Charles L. Baker beschreibt in dem RAND-Memorandum »JOSS: Introduction to a Helpful Assistant« das Leitmotiv des Systems: »to provide the individual scientist and engineer with a personal computational service immediately available, whenever required, in his own working environment. [...] The intimate interaction between man and machine permits the JOSS user to exercise judgment continually during the course of computation, changing and modifying the procedure as he wishes.« (Baker 1966, S. 41 f.) Getreu den verbreiteten Vorstellungen einer »community utility« in Form einer ›Denkhilfe‹ aus der Steckdose, wurden (aus Kostengründen) einige fahrbare Konsolen bereitgestellt und in jedem Büro ein »JOSS Plug« installiert. »This outlet may be thought of as supplying JOSS computational power, just as the conventional AC outlet is a source of electrical energy.« (ebda., S. 4 f.)

aktivität« eingehender behandelt. (Pflüger 2004, S. 370 ff.) Für eine differenziertere Betrachtung sei auf Hans Dieter Helliges Artikel »Leitbilder im Time-Sharing-Lebenszyklus« verwiesen. (Hellige 1996)

Abb. 1: Charles Baker am JOSS II Terminal (Marks 1982, S. 45)



Die Interaktion mit JOSS war nur durch zwei Modi strukturiert: Die Maschine war am Rechnen und wird etwas ausgeben, oder sie erwartete vom Nutzer eine Eingabe. Die beiden Zustände wurden durch ein rotes oder grünes Licht an der Konsole angezeigt. Wenn die Maschine die Kontrolle wieder an den Nutzer übergab, wurde zudem die Tastatur freigegeben, und es ertönte ein »soft BEEP tone«. »These visual, tactile, and audible signals leave no doubt in the user's mind as to who is in control of the station.« (ebda., S. 11) Die Dialogsprache beruhte auf wenigen, einfachen Sprachelementen, erlaubte es aber der Nutzerin, diese »in a variety of ways without restriction« zu kombinieren. »JOSS commands are limited to one line; take the form of an imperative English sentence, and in fact may be read out loud; begin with a verb; and obey the conventional rules of English for spacing, capitalization, punctuation, and spelling. The ability to append a conditional clause to any JOSS command is an extremely powerful feature of the language. [...] The language is highly readable, and the JOSS user will soon come to actually ›think‹ in the JOSS language – or, at least, to express his problem using JOSS's vocabulary.« Auch Baker spricht von der Möglichkeit »to converse with JOSS fluently« (ebda., S. iii-vi, 2), und viele Nutzer haben den Dialog als natürlich empfunden. Cliff Shaw schreibt in einem Erfahrungsbericht: »JOSS lacks the problem capacity to carry on a sophisticated conversation. For its comments, it simply ›selects from a stock of 40 ›canned‹ messages. But the timeliness and appropriateness of its remarks give a feeling of interacting with a person.« Seine Absicht war es, dem User ein »feeling of linguistically directing an

agent« zu geben, und dazu schien es notwendig, »to ›hide‹ the Johnniac from the user and to present instead the image of a person interpreting instructions and remaining in control of the situation no matter how senseless those instructions may be.« (Shaw 1965, S. 18, 17, 16) Weil das System fast jeden Unsinn tolerierte, lernten die meisten Nutzer durch Trial-and-Error damit umzugehen, schlimmstenfalls erhielten sie die unverbindliche Antwort »Eh?«

Abb. 2: Typischer JOSS-Dialog (Baker 1966, S. 14)

```

Type  2+2.
           2+2 =           4
Type  "ok " if 500 < 3 * 6 ≤ 1000.
ok
type 2+2
Eh ?

```

Die Robustheit und Flexibilität, die JOSS für spätere Pioniere des User Interface Designs zu einem Vorbild an Benutzerfreundlichkeit werden ließen,⁴ verdankte das System allerdings auch seiner beschränkten Funktionalität, die heute jeder bessere Taschenrechner bietet (auch wenn deren Interfaces kaum als nutzerfreundlich bezeichnet werden können). Mehr war auch nicht intendiert: »JOSS is a *special-purpose* system and should be viewed accordingly. It supplies a personal service, and by this we invite comparison with a telephone, a desk calculator, or a slide-rule – always available at a user's desk.« »JOSS itself is not a problem-solver, of course, but rather acts as a ›helpful assistant‹ to which the human problem-solver can delegate most of his computational chores.« (Baker 1966, S. 1, 40)

Durch die Entwicklung von komplexeren Timesharing-Systemen waren weitergehende dialogische Interaktionsformen möglich geworden. Die

4 William Orr schrieb: »JOSS is considered by professional programmers to be a masterpiece of elegant simplicity. Children have learned how to use it with only a few minutes' instruction, [...]« (Orr 1968, S. ix) Und Alan Kay, für den Kinder immer ›vorbildliche‹ Nutzer waren, meinte später: »The beauty of JOSS was the extreme attention of its design to the end-user – in this respect, it has not been surpassed.« (Kay 1993, S. 2)

vor allem beim Erstellen und Überarbeiten (Debugging) von Programmen gewonnene Zeitersparnis wurde als ungeheure Entlastung empfunden. Die durch die verteilte Zeit vermittelte Illusion, den Großrechner für sich allein zu haben, bedeutete, sich einem Problem widmen zu können, ohne durch Unterbrechungen und lange Wartezeiten aus dem Konzept gebracht zu werden. Die ›Interrupts‹ der Maschine ermöglichten eine Kontinuität für den Menschen. Der Computer kann damit in den Problemlöseprozess einbezogen werden und ist nicht mehr auf die Funktion eines Automaten, der eine ausgearbeitete Programmlösung abarbeitet, beschränkt. Wie Licklider in seinem berühmten Aufsatz »Man-Computer Symbiosis« schreibt, wird damit eine neue Klasse von Problemen, deren Formulierung schon einer Computerunterstützung bedarf, der maschinellen Bearbeitung zugänglich. »One of the main aims of computer symbiosis is to bring the computing machine effectively into the formulative parts of technical problems. The other main aim is closely related. It is to bring computing machines effectively into processes of thinking that must go on in ›real time‹, time that moves too fast to permit using computers in conventional ways.« (Licklider 1960, S. 3)⁵

Statt näher auf Lickliders inzwischen recht bekannten Symbiose-Aufsatz einzugehen, will ich im Folgenden seinen im Auftrag des ›Council on Library Resources‹ der Ford Foundation Anfang 1964 fertiggestellten Bericht über »Libraries of the future« akribischer auslegen, weil er dort seine Vorstellungen über eine interaktive Nutzung auch durch Computeralien viel genauer ausführt.⁶ Er entwirft eine Utopie, wie ein elektronisches Bibliothekssystem, das den weltweiten »fund of knowledge« verwaltet, im Jahr 2000 aussehen könnte. Für uns ist die Projektion aufschlussreich, weil sie natürlich im Denkhorizont seiner Zeit gefangen bleibt. Licklider bezeichnet das zu entwickelnde Bibliothekssystem als »procognitive system« und stellt es sich, getreu seiner andernorts öfters geäußerten Leitidee, als vernetztes Rechenzentrum in einem »intergalactic network« vor. Wie bei JOSS oder heutzutage mit Ethernetbuchsen, haben die Nutzer an ihrem Arbeitsplatz Steckdosen, mit denen sie sich mit der Mutter des Wissens verbinden können. »In the year 2000, information and knowledge may be as important as mobility. [...] In business, government, and education, the concept of ›desk‹ may have changed from passive to active: a desk may be primarily a display-and-control station in a telecom-

5 Lickliders Beispiel betrifft bezeichnenderweise den unter Zeitdruck stehenden Feldherrn, der vor Ort schnelle Unterstützung benötigt. Manager konnten sich damals mehr Zeit lassen. (Licklider 1960, S. 4, 14)

6 Er gibt dort eine bis in kleinste Details gehende Beschreibung, wie er sich eine beispielhafte Interaktion zwischen einem Nutzer und dem projektierten Bibliothekssystem vorstellt. (Licklider 1965, S. 47 - 52)

munication-telecomputation system⁷ – and its most vital part may be the cable (›umbilical cord‹) that connects it, via a wall socket, into the procognitive utility net.« (Licklider 1965, S. 33)

Uns interessieren hier nur Lickliders Überlegungen zum Interface, die auf einem ganzheitlichen, medientheoretischen Ansatz *avant la lettre* beruhen. Statt den geräteorientierten Begriff des Interfaces zu verwenden, will er von einem ›Intermedium‹ sprechen.

»Early in our study of man-computer interaction, we became dissatisfied with the term, ›man-machine interface‹. ›Interface‹, with its connotation of a mere surface, a plane of separation between the man and the machine, focusses attention on the caps of the typewriter keys, the screen of the cathode-ray tube, and the console lights that wink and flicker, but not on the human operator's repertory of skilled reactions and not on the input-output programs of the computer. The crucial regions for research and development seem to lie on both sides of the literal interface. In order to remind ourselves continually that our concern permeates the whole medium of interaction, we have avoided ›interface‹ and have used, instead, ›intermedium‹.« (ebda., S. 92)

Es sei nur eine künstliche, analytische Unterscheidung, wenn man die Hardware, »the physical medium through which the interactions take place«, von den Sprachaspekten der Interaktion trennt. »The man-computer intermedium subsumes the computer's display and the mechanisms and programs that control and maintain them, the arrangements through which people communicate information to the computer, and the relevant communication organs and skills of the men.«, mehr noch: »the intermedium extends beyond the console to include the user's entire work space [...] and perhaps even his laboratory system«. (ebda., S. 91 ff.)

Ziel der intermediären Interaktion ist die ›amplified cerebration‹ der Nutzer; sie sollen ermächtigt werden, von Routinetätigkeiten entlastet zu denken. »In any event, a basic part of the over-all aim for procognitive systems is to get the user of the fund of knowledge into something more nearly like an executive's or commander's position. He will read and think and, hopefully, have insights and make discoveries, but he will not have to do all the searching himself nor all the transforming, nor all the testing for matching or compatibility that is involved in creative use of knowledge. He will say what operations he wants performed upon what parts of the

7 Charakteristisch für Lickliders Stil ist die an dieser Stelle eingefügte bezaubernde Fußnote: »If a man wishes to get away from it all and think in peace and quiet, he will have merely to turn off the power. However, it may not be economically feasible for his employer to pay him at full rate for the time he spends in unamplified cerebration.«

body of knowledge, he will see whether the result makes sense, and then will decide what to have done next.«⁸ (ebda., S. 32) Um Herr im Haus des Wissens zu sein, muss man den Dialog steuern und kontrollieren können. Die Frage der Kontrolle spielt in allen Interaktionskonzepten dieser Zeit eine große Rolle. Das hat einerseits mit dem Alternieren des Dialogs zu tun, wo man dem Computer oder dem ›Agenten‹ die Kontrolle explizit übertragen musste und, wenn er sie wieder zurückgab, grünes Licht bekam weiterzudenken. Andererseits war es wohl auch eine Reaktion auf den überwundenen Batchbetrieb, wo man die Befehlsgewalt am Lochkartenkasten abgab, dass man nun die vermittelte Unmittelbarkeit der Kommandostruktur mit Führung und Kontrolle identifizierte.⁹

Ein effizienter Kommandeur ist kurz angebunden und verwendet ein recht beschränktes Vokabular, er ›weist‹ an, was zu tun ist. So sieht auch Licklider die Direktiven der Nutzer mit dem Lichtgriffel: »Both the system specialist and the author point to words and sentences in the text, move them about with the aid of their styli, insert or substitute new segments of text, and so forth.« (ebda., S. 113 f.)¹⁰ Um komplexere Prozesse kontrollieren zu können, müsse die Interaktionssprache eine »sophisticated syntax and a large vocabulary« haben, aber um sie einfach nutzen zu können, sollte sie andererseits den benötigten Befehlsvorrat möglichst klein halten. Der Vorteil der Knappheit des Dialoges »should not be lost to increasing sophistication«, »and it should encourage the kind of convergence upon

-
- 8 Solche Entlastung wurde Anfang der 60er Jahre unter dem Titel »Freeing the Mind« verhandelt, so in einem Artikel im Times Literary Supplement von 1962: »the aim is to lift the burdens of routine searching, collation, listing of possible sources and even perhaps taking of notes off the brainworker so that he can use his mind and his time to better purpose.« Die »mechanization of scholarship« sei als eine »liberation of thought itself« anzusehen. (TLS 1962, S. 194)
- 9 Der Psychologe Ulric Neisser kontrastierte die überlebte Automaten-technik, die »makes it very clear who is the boss«, mit den positiven Erfahrungen, die im MAC Timesharing-Projekt gemacht wurden: »using MAC is fun. The user is in control. He knows what is happening, and makes it happen when and how he likes. [...] One will not think of machines as uncontrollable forces if one controls them every day for one's own purposes.« (Neisser 1965, S. 214, 216 f.)
- 10 Eher beiläufig sieht Licklider, nach den beeindruckenden Erfahrungen mit Ivan Sutherlands Sketchpad-System, auch graphische Möglichkeiten vor. Diese über Steuerung hinausgehenden Funktionen sind in ein Sprachkonzept eingebettet, wonach der Mensch sich auf dem ›concept level‹ ausdrücken kann und die Maschine die Präzisierung der formalen Repräsentation übernimmt. »The display should provide the set of features called 'Sketchpad' features [...], which assign to the computer those parts of the sketching and drawing skill that involve much practice and precision, and leave the man responsible mainly for expressing the essential structure of the concept he desires to represent.« (Licklider 1965, S. 94)

understanding achieved in conversation«. (ebda., S. 124 f., 122) Das Intermedium sieht somit unterschiedliche Sprachebenen vor und geht davon aus, dass die Nutzer die ihnen angemessene Form einer »intimacy of interaction« durch Erfahrung finden. »The user learns, through working with the system, what modes and techniques suit him best. Ordinarily, he gives the system rather terse, almost minimal instructions, relying on it to interpret them correctly and to do what he wishes and expects. When it misinterprets him or gets off the track of his thinking, as it sometimes does, he falls back on more explicit expression of commands and queries.« (ebda., S. 52)

Damit man sich lapidar ausdrücken kann und trotzdem von der Maschine ›verstanden‹ wird, muss der Computer den Kontext erkennen, in dem die Äußerung stattfindet. »In order to match the small set of control actions a man can take with the vast assortment of things there are to be done in the world of the procognitive system, it is necessary to take advantage of the concept of determination by local context that is so highly developed in the natural languages.« Neben der Einschränkung, dass die Nutzer nur eine auf ihre Bedürfnisse zugeschnittene Teilsprache verwenden, besteht eine Möglichkeit der Kontextualisierung darin, *Modes* einzuführen, die dem Computer implizit eine Interpretation der Nutzereingaben ermöglichen. *Modes*, die später verunglimpft werden, sind also als Mittel gedacht, die Wortklauberei der menschlichen Äußerung zu reduzieren, wenn er einer Maschine Anweisungen gibt. »When the pen is used merely as a pointer, the meaning of its message is conveyed partly by the location of the spot to which it points. However, the meaning is determined also by the nature of the display that is currently being presented and the particular details of that display. The part of the determination that is associated with the nature of the display is ›mode‹ determination. The simplest on-line man computer-interaction systems we know of have only two modes: a ›control mode‹ and a ›communication mode‹.« (ebda., S. 120 f.) Licklider erscheint es jedoch wünschenswert, »to move rapidly beyond the simple dichotomy between the control and communication modes and to develop a syntax in which it will be possible to express commands, state facts, and ask questions in any convenient sequence«. ¹¹

11 Noch heute bestimmt diese Dichotomie alle Aktionen bei dem in manchen Kreisen beliebten Unix-Editor *vi*. Dort bedeutet beispielsweise die Eingabe des Zeichens *A* im *Insert-Mode*, dass der Buchstabe *A* eingefügt wird, im *Command-Mode*, dass etwas hinten angefügt werden soll. Die Konzentration einer komplexen Operation auf ein einziges Zeichen ermöglicht natürlich ein virtuoseres Vorgehen; und es sind auch diejenigen, die den Editor täglich nutzen, die auf *vi* schwören, während Gelegenheitsnutzer eher wahnsinnig werden, weil die Zeichen mehrdeutig sind und irgendetwas Unkontrolliertes geschieht, wenn man im falschen Modus ist.

Und er erwartet, dass die Entwicklung der Dialogsprachen dahin gehen wird, dass »the residual distinction between control and communication mode is carried partly by the syntactic structure of statements and partly by context.« (ebda., S. 124, 123)

Die Einführung von *Modes*, die der Maschine eine bescheidene Interpretationsleistung ermöglichen, bürgen dem Menschen jedoch die Merkleistung der Kontexterfassung auf. »The convenience of having local context or mode implicitly define terms and particularize procedures for him may to some extent be countered by the responsibility, thrust upon the user, of continually keeping track of the prevailing mode.« Hier kommt nun entscheidend Licklid's Leitidee einer Mensch-Computer-Symbiose zum Tragen. Denn im Gegensatz zur reinen Überwachung einer Maschine, wie in einer Fabrik, ist der Mensch in einer »intimate interaction« immer am Ball und weiß, wo, d.h. in welchem Abschnitt eines Problemlöseprozesses, er sich befindet. »In truly symbiotic interaction, the human partner is always involved in directing, always »ahead of the game.«« Damit weiß er auch, in welchem *Mode* er bzw. die Maschine sich befindet. Solches Umweltbewusstsein ist innig an die kontinuierliche (Inter-)Aktivität der Nutzerin geknüpft, wohingegen der Kontrolleur ohne Kontrolle nur bei einem Alarm reagieren muss und dann die Situation nicht richtig einschätzen kann. »It does not help the monitor much to display to him the developing situation, either in summary or in detail, for it is almost impossible for him to think ahead constantly if his thinking has no effect on what happens.¹² In our concept of man-computer interaction in pro-cognitive systems, however, the man is no mere monitor. He is a partner – indeed he is usually the dominant, leading partner. On-line interaction introduces into the language picture the possibility of ›conversation.«« (ebda., S. 123)

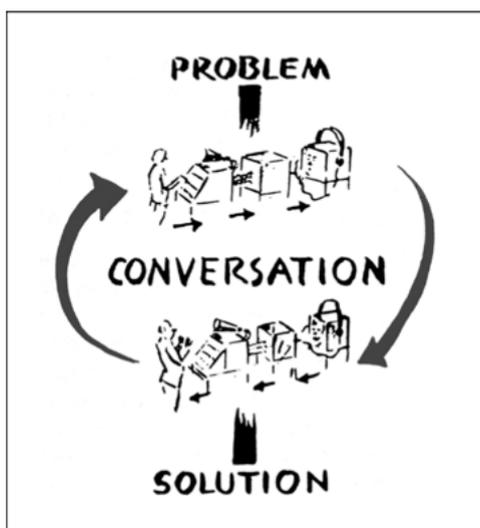
Auch wenn in der Wirklichkeit der 60er Jahre der Dialog zwischen Mensch und Maschine im Wesentlichen im Austausch von Kommandos und Daten bestand, war mit dieser Form von Interaktivität doch eine neue Problemlösetechnik verbunden. Sie kann vielleicht so charakterisiert werden, dass die Trennung zwischen innerem und äußerem Diskurs verringert wird.¹³ Wie in einem Gespräch wird die Entwicklung einer Idee in kleine Abschnitte zerlegt und durch die Reaktionen eines Opponenten oder durch den Austausch von veräußerten Gedankenfragmenten ge-

12 Die Erfahrung der nachträglichen Ungewissheit kann man leicht machen, wenn man versucht, ein umfangreiches Logfile zu verstehen.

13 Es ist eine alte Tradition, Denken als inneres Sprechen aufzufassen und Schrift als Externalisierung von Gedachtem. In der konversationellen Interaktion werden die vormalig getrennten Momente zusammengedacht. Inwieweit das schriftliche Gespräch etwas Eigenständiges ist, beschäftigt auch die linguistische Literatur zur ›computer-mediated communication‹.

lenkt. Die innerliche Gedankenarbeit wird auf die Pausen zwischen den ›Sinn machenden Resultaten‹ der antwortenden Maschine verteilt. Indem der Mensch die Erwidrerungen des instruierten Computers bedenkt, instruiert er sich selbst, wie weiter zu denken und zu verfahren ist. »Always in full control, the user directs the mind-machine processes by commanding the computer to function, step by step, in accord with his deepening understanding of the problem.« (Orr, 1968, S. 26)

Abb. 3: Douglas Ross' Konversationsvorstellung (Ross 1956, S. 6)



Wie in einem richtigen Dialog ist der jeweilige Kontext durch die Interaktionsgeschichte bestimmt, und die Auslegung des jeweils Geäußerten bedeutet, Anschlussbedingungen zu erkennen. Der Maschine wird dabei keine ›Context-Awareness‹ im heutigen Sinne abverlangt, sie befindet sich aufgrund der Vorgeschichte jeweils in einem Zustand – einem *Mode*, der ihre Interpretation der Eingabe determiniert. Für den Nutzer ergibt sich sinnvoll Anschließendes aus den bisherigen Antworten der Maschine und der schrittweisen Eigenlogik seines wie auch immer konzipierten Problemlöseplans. Wie es sich für eine anständige Konversation geziemt, bleibt man beim selben Thema, räumliche Orientierung erscheint noch nicht als Problem.¹⁴

¹⁴ Licklider hat an der Konsole seines projektierten Systems zwei Notfallknöpfe vorgesehen, »a silver one labeled 'Where am I?' and a gold one labeled 'What should I do next?' Any time a user loses track of what he is doing, he can press the silver button, and the recapitulation program will help him regain his bearings. Any time he is at a total loss, he can press the

Gemäß dieser Vorstellung ist der Computer kein reines Arbeitsmittel für seine Nutzerin, sondern beide werden als Partner gesehen, die sich durch ihre unterschiedlichen Fähigkeiten synergetisch ergänzen und eine kleine Problemlöseeinheit bilden. Schon Douglas Ross verstand seinen recht simplen Dialog als eine integrative »problem-solving technique, i. e. a point is reached where it is difficult to tell which is more important, the human, the problem, or the computer.« (Ross 1956, S. 10) Der Computer wird innerhalb eines gleichsam verselbständigten Denkprozesses angesiedelt, in dem er rechnet, speichert und antwortet: »the system delivers much of its help ›inside the thought cycle‹, and ready for integration within the structure of the user's thinking.« (Overhage/Harman 1965, S. 81) So ist die gängige Rede zu verstehen, der Computer würde in der Interaktion mit dem Menschen »amplifying his thinking« und »magnifying his mental sources«, oder wie Licklider erwartete: »The hope is that, in not too many years, human brains and computing machines will be coupled together very tightly, and that the resulting partnership will think as no human brain has ever thought and process data in a way not approached by the information-handling machines we know today.« (Licklider 1960, S. 2)

Hierin sind imaginative Residuen der Kybernetik zu sehen, wo man meinte mit Rückkopplungsschleifen alle Probleme lösen zu können. »The concept underlying on-line problem solving is control through feedback, which was also the central idea of Norbert Wiener's cybernetics. [...] When a problem solver is working on-line with computing machinery, his personal intelligence becomes an integral part of the process. Mind and machine interact continuously and simultaneously in a coordinated attack upon the problem at hand. Yet initiative and control always rest with the human intelligence.« (Orr 1968, S. 25 f.) Die Kybernetik war wesentlich eine zeitorientierte Theorie. Ihre Prozesse sind teleonomisch organisiert, jede Entwicklung wird von ihrem voraus-gesetzten Endpunkt, dem Sollwert, her gedacht. Die konversationelle Problemlösung ist ebenso eigensinnig auf ein Ziel hin orientiert; letztlich ist das Vorgehen des ›Problem-Solving‹ per definitionem durch ein ›goal-seeking behavior‹ charakterisiert.¹⁵ Die von William Orr dem Dialog attestierten explorativen Freihei-

gold one, and the instruction program will explain further how to use the system. Through either of those programs, the user can reach a human librarian.« (Licklider 1965, S. 127) Charakteristisch für den Dialog ist, dass ›where‹ zeitlich, als ›wo befinde ich mich im Prozess‹, zu verstehen ist.

- 15 Herbert Simon hat das durch die Unterscheidung von Zustands- und Prozessbeschreibung zu fassen versucht: »Wir werfen ein Problem auf, indem wir die Zustandsbeschreibung der Lösung geben. Die Aufgabe ist, eine Folge von Vorgängen zu entdecken, die aus einem Ausgangszustand heraus den Zielzustand herstellt. Die Übersetzung aus der Prozessbeschreibung in

ten können im zeitlich-eindimensionalen Kontext des Dialogs nur sehr beschränkt verwirklicht werden.¹⁶ Auch wenn man sich nicht, wie im Batchbetrieb, ein für alle mal auf eine Strategie festlegen muss, sondern Alternativen ausprobieren und ›gehörige‹ Gedanken vorzeitig einbringen kann, erscheint das sich im Gedankentransfer vertiefende Verständnis doch in einen linearen Prozess eingebunden. Die Pirouetten der Interaktion laufen in einem Kanal ab, in dem es kein Halten gibt. Umwege sind nicht eingeplant, und was nicht ins Konzept passt, findet im Problemlöseprozess keinen Platz.¹⁷ Wenn kreatives Arbeiten aber erfordert, ohne vorgegebene Ordnung frei zu assoziieren, fragmentarisches Gedankengut sammeln und Sichten wechseln zu können, kann der zielgerichtete Dialog nicht mehr als Richtschnur dienen. Douglas Engelbart wird eben diese Freiheit der unordentlichen Fertigung als erweiterte Möglichkeit der Interaktivität einfordern und seine Konzeption eines ›augmented knowledge workshops‹ darauf ausrichten.

die Zustandsbeschreibung ermöglicht uns zu erkennen, wann wir erfolgreich sind.« (Simon 1962, S. 167) Heute hat sich in vielen Design-Theorien die Einsicht durchgesetzt, dass man es bei Gestaltungsaufgaben meistens mit »wicked problems« zu tun hat, die weder eine definitive Problemstellung noch eine »stopping rule« aufweisen. (Rittel/Webber 1973) Damit ist auch die Vorstellung, Gestaltungs- und Entwicklungsprozesse entsprächen Verfahrensweisen des Problemsolving, hinfällig geworden.

- 16 Orr beschreibt die ›koordinierte Attacke auf das Problem‹ folgendermaßen: »The user requests an operation to be performed; the computer performs it; the results are immediately displayed on an oscilloscope in graphical or alphanumeric form. Thus, without committing himself irrevocably to any problem-solving strategy, the on-line user thoroughly explores the structure of the problem. He employs heuristics, tries out a number of different approaches, follows intuition, plays hunches; he cleans cues from partial results and from approximations. [...] In a closely coupled mind-machine interactive system, it is essential that pertinent new thoughts be given immediate expression, i.e., transferred at once from mind to machine for present or future use. Any procedural step imposed upon the user, regardless of how trivial, inhibits this thought transfer process.« (Orr 1968, S. 26) Es fällt schwer, in dem ersten Satz viel mehr als die Möglichkeiten von JOSS zu sehen; umso auffälliger erscheint das Missverhältnis zu dem durch ›thus‹ eingeleiteten überschwänglichen Anspruch.
- 17 Hans Blumenberg qualifiziert in seiner posthum herausgegebenen Anthropologie Umwege als »phänomenonale[n] Index aller Freiheit [...] Der Umweg ist die Demonstration der Sicherheit des zum Ziel der Intention tragenden Grundgefühls.« (Blumenberg 2006, S. 567) Im Softwareengineering hat solches Vertrauen erst spät Einzug gehalten. Nachdem man das lineare Wasserfallmodell mit Schleifen angereichert hatte, stellte man irgendwann fest, dass auch das nicht genügt und die globale Sequentialität in der Praxis nur eine sehr beschränkte Orientierung ergibt.

2 Allmähliche Verfertigung der Gedanken

Kurze Zeit nach Lickliders Vision einer Mensch-Computer-Symbiose veröffentlichte Douglas Engelbart am Stanford Research Institute (SRI), im Oktober 1962, seine nicht weniger berühmte Studie »Augmenting Human Intellect: A Conceptual Framework«. Es handelte sich um einen Leitantrag zur Projektförderung durch die US Air Force, woraus sich dann in der Folgezeit das NLS-System entwickelte; und es handelt sich um einen der seltenen Fälle, dass eine ausformulierte Vision tatsächlich in ein funktionierendes System umgesetzt wurde. Engelbart präzisiert seine Zielsetzung: »By »augmenting human intellect« we mean increasing the capability of a man to approach a complex problem situation, to gain comprehension to suit his particular needs, and to derive solutions to problems.« Die komplexen Problemlagen beinhalten »professional problems of diplomats, executives, social scientists, life scientists, physical scientists, attorneys, designers«. (Engelbart 1962, S. 1) Gedacht ist also an eine Computerunterstützung jedweder Geistesarbeit in einer »problem-laden world«. Auch wenn Engelbart sich durchaus positiv auf das »open-shopper«-Interface von JOSS bezieht, meint er, dass diese Nutzer nicht primär eine hilfreich assistierende Rechenmaschine benötigen, sondern ein »symbol manipulating device«, das jede Art von symbolverarbeitender Tätigkeit unterstützen kann.¹⁸ »Our aim is to give help in manipulating any of the concepts that the individual usefully symbolizes in his work, of which those of mathematical nature comprise but a limited portion in most real-life instances.« [...] the computer has many other capabilities for manipulating and displaying information that can be of significant benefit to the human in non mathematical processes of planning, organizing, studying, etc. Every person who does his thinking with symbolized concepts (whether in the form of the English language, pictographs, formal logic, or mathematics) should be able to benefit significantly.« (Engelbart 1961; 1962, S. 6)

Es geht Engelbart um eine Effektivierung geistiger Arbeit und um neue interaktive Formen, Probleme anzugehen. Dieses Ziel hatte er zuvor in einem SRI-Typoscript mit dem Titel »Program On Human Effectiveness« formuliert: »The possibilities we are pursuing involve an integrated man-machine working relationship, where close, continuous interaction with a computer avails the human of radically changed information-

18 Den Wandel in der Wahrnehmung des Computers, von einer elaborierten Rechenmaschine zu einem »symbol manipulating device«, findet man seit Ende der 50er Jahre auch im Programmiersprachenbereich und in der Datenverarbeitung. So definiert Ned Chapin: »Eine elektronische Rechenanlage ist eine Maschine, die nach fest gegebenen Regeln selbständig mit Symbolen arbeitet.« (Chapin 1957/62, S. 15)

handling and portrayal skills, and where clever utilization of these skills provides radical changes in the way the human attacks problems. Our aim is to bring significant improvement to the real-life problem-solving effectiveness of individuals.«¹⁹ (Engelbart 1961) Soweit klingt das wie Lickliders Traum, der interaktive Mensch werde durch die Kopplung mit der Maschine denken wie nie zuvor, jedoch stellt sich Engelbart etwas Anderes vor. Statt an eine konversationelle Symbiose denkt er an eine »working relationship«, in der Symbole auf dem Bildschirm manipuliert werden, und die »portrayal skills« deuten an, dass der Schwerpunkt eher auf einer Darstellungsfunktion liegen wird.

Um seinen Ansatz zu begründen, entwirft Engelbart eine Art Anthropologie der Geistesarbeit, wonach der mit dem Computer interagierende Mensch weit mehr als »Knowledge Worker« denn als Gesprächspartner erscheint. Geistige Arbeit beruhe, darin durchaus dem Handwerk vergleichbar, auf einem Inventar von unterschiedlichen Fähigkeiten, die auf verschiedenen Ebenen ausgebildet werden können. Um effizient zu arbeiten, müssen diese optimal organisiert und durch geeignete Techniken unterstützt werden.²⁰ Eine alte Theorielinie, die von Arnold Gehlen und Marshall McLuhan prominent vertreten wurde, sieht Geräte und Techniken als Organersatz oder Organerweiterung an. Engelbart stellt sich in diese Tradition, indem er elektronische Hilfsmittel bereitstellen will, die die natürliche Ausstattung des Menschen erweitern und verstärken. »These means« can include many things – all of which appear to be but extensions of means developed and used in the past to help man apply his native sensory, mental, and motor capabilities«. (ebda., S. 1 f.) Solche »augmentation means« lassen sich unter vier Rubriken zusammenfassen: Artefakte, Sprache, Methodologie und Training. Deshalb spricht er von dem projektierten System als einem »H-LAM/T system – the individual augmented by the language, artifacts, and methodology in which he is trained«; und die vorgesehenen Artefakte sind natürlich »computers, and computer-controlled information-storage, information-handling, and information-display devices.« (ebda., S. 15, 9)

19 Er fügt hinzu, die Wichtigkeit eines solchen Vorhabens könne nicht überschätzt werden: »It is felt that such a program competes in social significance with research toward harnessing thermonuclear power, exploring outer space, or conquering cancer, and that the potential payoffs warrant a concerted attack on the principal problem areas.« (Engelbart 1961)

20 »each individual develops a certain repertoire of process capabilities from which he selects and adapts those that will compose the processes that he executes. This repertoire is like a tool kit, and just as the mechanic must know what his tools can do and how to use them, so the intellectual worker must know the capabilities of his tools and have good methods, strategies, and rules of thumb for making use of them.« (Engelbart 1962, S. 11)

Wir finden immer noch das Gedankengut der Intelligenzverstärkung, wenn auch etwas gebrochen: »It has been jokingly suggested several times during the course of this study that what we are seeking is an ›intelligence amplifier.« Und gleichfalls die Vorstellung eines Synergieeffektes durch die Kopplung von Mensch und Maschine, die zu einem gehobenen Denkvermögen führt: »we will have amplified the intelligence of the human by organizing his intellectual capabilities into higher levels of synergistic structuring. What possesses the amplified intelligence is the resulting H-LAM/T system, in which the LAM/T augmentation means represent the amplifier of the human's intelligence.« (ebda., S. 19)²¹ Dies klingt fast wie Douglas Ross' Problemlöseinheit, aber die Organisation der intellektuellen Fähigkeiten deutet an, dass hier etwas anderes gemeint ist. Ziel ist nicht eine ›fluent conversation‹, sondern der Schwerpunkt liegt auf der externalisierten Symbolmanipulation. »For other than intuitional or reflexive actions, an individual thinks and works his way through his problems by manipulating concepts before his mind's eye. His powers of memory and visualization are too limited to let him solve very many of his problems by doing this entirely in his mind. [...] Thus, a large part of an individual's meaningful intellectual activity involves the purposeful manipulation of concepts; and of this concept-manipulation activity, a very important part is accomplished by the external manipulation of symbols.« »One way of viewing the H-LAM/T system changes that we contemplate – specifically, integrating the capabilities of a digital computer into the intellectual activity of individual humans – is that we are introducing new and extremely advanced means for externally manipulating symbols.« (Engelbart 1961; 1962, S. 25)

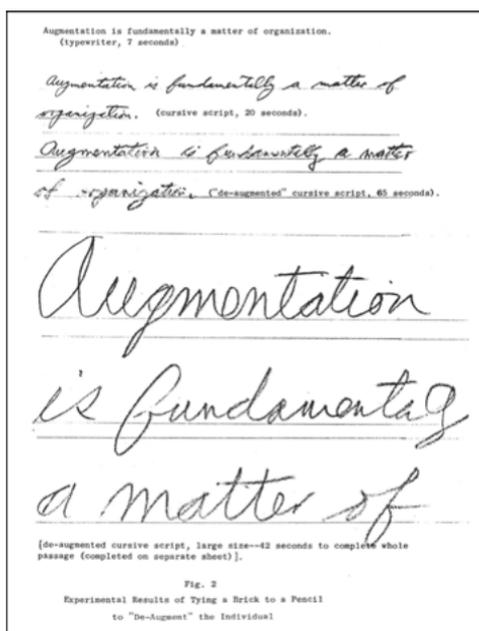
Unter Bezugnahme auf die bekannte Sapir-Whorf-These, dass unsere Sprache bestimmt, wie wir was denken können, formuliert Engelbart seine auf die symbolische Manipulation erweiterte »Neo-Whorf-Hypothese«: »Both the language used by a culture, and the capability for effective intellectual activity are directly affected during their evolution by the means by which individuals control the external manipulation of symbols.«²² Auch wenn die mitgebrachten geistigen Fähigkeiten auf sehr

21 Der Synergieeffekt wird wieder in die Tradition der menschlichen, natürlichen wie kulturellen, Evolution gestellt und als Weiterentwicklung von älteren ›symbolischen Maschinen‹ gesehen: »In amplifying our intelligence, we are applying the principle of synergistic structuring that was followed by natural evolution in developing the basic human capabilities. [...] In a very real sense, as represented by the steady evolution of our augmentation means, the development of 'artificial intelligence' has been going on for centuries.« (Engelbart 1962, S. 19) Zur Geschichte von ›symbolischen Maschinen‹ vergleiche (Krämer 1988).

22 Genaueres hierzu siehe (Friedewald 1999, S. 154 ff.).

unterschiedlichen Ebenen angesiedelt sind, beeinflussen sie sich gegenseitig, »new innovation in one particular capability can have far-reaching effects throughout the rest of your capability hierarchy«, – mehr noch: »the impressive new tricks all are based upon lots of changes in the little things you do«. (ebda., S. 24, 14, 83)

Abb. 4: De-augmentation (Engelbart 1962, S. 27)



Weil es schwierig ist, über Veränderungen intellektueller Fähigkeiten durch nur projizierte Hilfsmittel zu urteilen, hat Engelbart das Gegenbeispiel einer »de-augmentation« untersucht, wie ein an einem Stift befestigter Klotz das Schreiben verlangsamt und verundeutlicht. Die behinderte Sentenz drückt seine Grundthese aus: »Augmentation is fundamentally a matter of organization.«

Die Erwartung einer radikal neuen Arbeitsweise beruht bei diesem »engineering-like approach« also auf der Vorstellung, dass kleine operative Veränderungen große Wirkungen haben können und dass aus der Reorganisation von Fähigkeiten neue entstehen. »A fundamental hypothesis of the proposed approach is that the ability of a given human to control the real-time external manipulation of symbols, in response to the minute-by-minute needs of his thought processes, has a profound effect upon the

whole structure of concepts and methods utilized in his intellectual activity.« (Engelbart 1961)

Bei Ross und Licklider finden sich vergleichbare Thesen über das ›interactive thinking‹, die aber mehr mit Disziplinierung des menschlichen Partners zu tun haben. Bei Ross heisst es über die Konversation in einer Sprache auf dem ›concept level‹: »the human is allowed to discuss the problem only in that language so that, in effect, a part of the programming of the problem has been accomplished by programming the humans.« (Ross 1956, S. 8). Und auch Licklider sah eine wesentliche Funktion der Interaktion darin, dass der Computer eine Präzisierung der Problemformulierung erzwingt. Engelbart dagegen erwartet ganz neue Möglichkeiten des Denkens durch Manipulation. Die ›minutiösen‹ Anforderungen des Denkprozesses sollen nicht wie in einem Dialog durch den Kontext von Rede und Gegenrede strukturiert werden, sondern die kontrollierte Symbolmanipulation wird die Struktur des Denkraumes selbst auf dem ›concept level‹ umgestalten.

Der konzeptionelle Wandel von der Konversation zum ›Augmented Knowledge Workshop‹, der sowohl die Ansprüche an den Computer, sein Interface wie die Geistesarbeit seiner Nutzer – also Lickliders ›Intermedium‹ – betrifft, lässt sich meines Erachtens in zwei paradigmatischen Verschiebungen zusammenfassen: Strukturorganisation statt Wechselgespräch und Sehen statt ›Hören‹. Beides bedeutet, dass der Kontext in der Interaktion eher räumlich als zeitlich gedacht wird.

Der Dialog ist an eine serielle, eindimensionale Problemlösestrategie gebunden. Für Engelbart ist kreative Geistesarbeit aber wesentlich eine Frage der Organisation von Denk-, Argument- und Symbolstrukturen, bei der sich aus »thought kernels« ohne festgelegte Ordnung die Gestalt einer Argumentation herausbildet. »Conceptually speaking [...] an argument is not a serial affair. It is sequential [...] because some statements have to follow others, but this doesn't imply that its nature is necessarily serial. [...] A conceptual network but not a conceptual chain.« (Engelbart 1962, S. 81 f.) Während für Licklider und Ross der erwartete Synergieeffekt einfach dadurch entsteht, dass Mensch und Maschine ihre spezifischen Fähigkeiten in die Interaktion einbringen, wobei der Mensch Ziele setzt und den Dialog leitet, insistiert Engelbart für sein ›synergistic structuring‹ stärker auf der kreativen Unordentlichkeit des menschlichen Beitrags: »With the human contributing to a process, we find more and more as the process becomes complex that the value of the human's contribution depends upon how much freedom he is given to be disorderly in his course of action.« Das sich im Gespräch in Schleifen vertiefende, aber auch kanalisierte Verständnis soll für perspektivische Verbindungen im vieldimensionalen ›concept space‹ freigesetzt werden. »When the course of

action must respond to new comprehension, new insights and new intuitive flashes of possible explanations or solutions, it will not be an orderly process. Existing means of composing and working with symbol structures penalize disorderly processes very heavily, and it is part of the real promise in the automated H-LAM/T systems of tomorrow that the human can have the freedom and power of disorderly processes.« (ebda., S. 98, 45)

Engelbart denkt hier in erster Linie – als Prototyp einer externalisierten Symbolmanipulation – an ein elaboriertes Textverarbeitungsprogramm, das eine uns inzwischen selbstverständlich gewordene Arbeitsweise ermöglicht: »trial drafts could rapidly be composed from rearranged excerpts of old drafts, together with new words or passages which you stop to type in. [...] You can integrate your new ideas more easily, and thus harness your creativity more continuously, if you can quickly and flexibly change your working record.« Das unterstützende System darf also keinen festen Ablauf und damit temporal-serielle Kontexte vorgeben, sondern muss die Freiheit eines unordentlichen Gestaltungsprozesses erlauben: »a free outpouring of thoughts in any order«, verbunden mit der »freedom to juggle the record of your thoughts, and by the way this freedom allows you to work them into shape.«²³ Der »Augmented Knowledge Workshop« ist »a workspace for me, in which I can browse, make additions or corrections, or build new sets of thought kernels with a good deal of freedom.« (ebda., S. 13, 84, 57) Das Interface muss also vorderhand nicht zielgerichtete Prozesse, sondern die Orientierung in diesem Werkraum unterstützen. »The computer should help navigate throughout the entire working space.« (Engelbart 1988, S. 220)

Die Unordnung der Gedanken und die Freiheit der Konstruktion erfordern für eine sinnvolle Produktion Mittel für deren Strukturierung und Organisation.²⁴ Die verstreuten Elemente des geistigen Werdens können Unterstrukturen enthalten in Form von formallogischen Argumenten und informellen Statements; Graphen, Kurven, Skizzen oder Annotationen (»something like footnotes, only much more flexible«). Auf den Unterstrukturen lassen sich Ordnungen nach verschiedenen Kriterien erklären, etwa eine »order of importance to comprehension«²⁵, oder beliebige Verbindungen knüpfen, die unterschiedliche argumentative Pfade verfolgen.²⁵ Diese vieldimensionale Strukturdarstellung des Denk- und Gestaltungsraumes beschreibt Engelbart mit der Figur des Netzwerkes,

23 Eine Freiheit, die heute zu vielen Klagen über »geistlose« Produktionen mit »copy & paste« Anlass gibt.

24 »If there is any one thing upon which this ›intelligence depends‹ it would seem to be *organization*.« (Engelbart 1962, S. 18)

25 Hierin erkennt man das Vorbild der »associative trails« aus Vannevar Bushs Aufsatz »As we may think«, auf den Engelbart in einem eigenen Abschnitt eingeht. (Bush 1945)

lange bevor daraus ein Schlagwort wurde. »When you get used to using a network representation like this, it really becomes a great help in getting the feel for the way all the different ideas and reasoning fit together – that is, for the conceptual structuring«. (ebda., S. 89, 87 f.)

Die Repräsentation der Gedankenarbeit als Netzwerk ist nicht nur auf neuartige Mittel der Manipulation, sondern auch auf flexible Techniken zu ihrer Präsentation angewiesen, »new ways to develop and portray concept structures«. Mittels laborierter Software kann der Nutzer auf graphischen Displays die Repräsentanzen seiner Gedankenfragmente freizügig umstellen und komponieren. Textpassagen und Diagramme »can be arranged before his eyes, moved, stored, recalled, operated upon according to extremely complex rules – all in very rapid response to a minimum amount of information supplied by the human, by means of special cooperative technological devices.« Das organisierte Netzwerk der Gedanken erfordert aber auch, dass nicht alles gleichzeitig zur Disposition steht. Die Nützlichkeit des Sichtbaren setzt voraus, dass vieles unsichtbar bleibt und nur bei Bedarf ans Licht geholt werden kann, »substructures can remain invisible to the worker until such time as he wants to flush them into view.« Engelbart vergleicht die unterschiedlichen Sichten auf ein Dokument mit einer Visualisierungstechnik, die später unter dem Namen ›Fish-eye-View‹ prominent geworden ist, wonach nur die Figur im Fokus der Aufmerksamkeit deutlich dargestellt wird, während ihr Kontext im Hintergrund verläuft: »It is a lot like using zones of variable magnification as you scan the structure – higher magnification where you are inspecting detail, lower magnification in the surrounding field so that your feel for the whole structure and where you are in it can stay with you.« (ebda., S. 55, 90, 89)

Die maschinelle Verwaltung der Konzeptstruktur und der Geschichte ihres Gewordenseins muss also unterschiedliche Perspektiven auf den Problemlöseprozess erlauben, eine Form von »view generation«, die unserer mentalen Strukturierung und nicht der inneren Repräsentation der Maschine entspricht.²⁶ Die Projektion eines vielschichtigen Gedankengebildes – »thought vectors in concept space« – auf eine zweidimensionale

26 »This sort of ›view generation‹ becomes quite feasible with a computer-controlled display system, and represents a very significant capability to build upon. With a computer manipulating our symbols and generating their portrayals to us on a display, we no longer need think of our looking at the symbol structure which is stored – as we think of looking at the symbol structures stored in notebooks, memos, and books. What the computer actually stores need be none of our concern, assuming that it can portray symbol structures to us that are consistent with the form in which we think our information is structured.« (Engelbart 1962, S. 36)

Ansicht auf einem Schirm bedeutet,²⁷ dass der jeweilige Kontext nicht mehr durch die Logik einer interaktiven Abarbeitung gegeben ist, sondern gemäß einem frei gewählten Blickwinkel im Netz der argumentativen Struktur ›ausgezeichnet‹ wird. Der zeitliche Kontext wird nicht mehr im gleichen Maße durch das Alternieren eines Dialogs bestimmt, sondern ›what to do next‹ wird zum selbstreferentiellen Problem der freien Gedankenarbeit. Die Maschine stellt dazu nur Ausblicke und Rücksichten auf die Geschichte der Interaktion bereit. Zu diesem Zweck muss sie alles aufzeichnen, »monitor what we do«; und für den ›Knowledge Workshop‹ war auch ein »man-computer cooperative review« vorgesehen, – eine Möglichkeit ›zurückzuspulen‹, um aus Fehlern zu lernen. (Engelbart 1962, S. 98 ff.) Wir finden also mit der Entlassung aus der prozeduralen Zeit, vielleicht zum ersten Mal, dass die Maschine ihre Nutzer zu ihrem Besten kontrolliert.

Engelbarts Vorstellungen wurden von ihm und seinen Mitarbeitern in den folgenden Jahren sukzessive im NLS-System (»oNLine System«) implementiert und am 9. Dezember 1968 mit großem Erfolg in der ›mother of all demos‹ vorgeführt.²⁸ Mehr als eine Demonstration von Möglichkeiten ist NLS jedoch nie geworden, das System konnte sich nicht praktisch durchsetzen. Einer der Gründe ist wohl, dass Engelbarts Leitvorstellung der menschlichen Effizienz zu großes Gewicht auf die ›handwerklichen‹ Fertigkeiten der Nutzer gelegt hatte.²⁹ Im Projekt wurden verschiedene Eingabetechniken, z.B. auch eine Steuerung mit dem Knie, ausprobiert und getestet, von denen uns nur die Maus als bedeutsames Erbe geblieben ist. Die spezielle, nur mit einer Hand zu bedienende Eingabetastatur (Chording Device) und die kryptischen Kommandokürzel erforderten eine gewaltige Gewöhnung, so dass nur erfahrene Nutzer mit

27 Wohl in Bezug auf das Dialog-Interface meint Engelbart sogar: »Actually, it is much closer to the truth to say that it is like trying to project n-dimensional forms (the concept structures, which we have seen can be related with many many nonintersecting links) onto a one-dimensional form (the serial string of symbols), where the human memory and visualization has to hold and picture the links and relationships.« (Engelbart 1962, S. 90)

28 Eine vollständige Filmfassung der Demonstration findet man unter (Engelbart 1968), Bilder der verschiedenen Ein- und Ausgabegeräte in (Engelbart 1988).

29 Engelbart hat sich unter anderem auf den Aufsatz von Irving J. Good: »How Much Science Can You Have at Your Fingertips?« bezogen, der zur Effizienzsteigerung unter anderem erwogen hatte, das Alphabet zu vergrößern oder das Lesen zu verschnellern, indem man wie in alten griechischen Inschriften Zeilen alternierend auch von rechts nach links schreibt. (Good 1958) Am Ende der Projektbeschreibung dachte Engelbart, entgegen den eingangs formulierten Ansprüchen, auch erstmal nur an Programmierer als geeignete Nutzer. (Engelbart 1962, S. 116 ff.)

›information at your fingertips‹ umgehen konnten. Alan Kay und die Entwickler des Dynabooks setzten demgegenüber in der Folgezeit ganz auf die leichte Erlernbarkeit der Interaktion, was sich dann mit den Interface-Konzeptionen des Xerox Star und des Macintosh als populäres Designprinzip durchsetzte.³⁰

3 Actio est reactio.

Zur gleichen Zeit wie Engelbarts NLS-Entwicklung wurden neue Interaktionsformen auch im eher technologiegetriebenen GRAIL-Projekt der RAND Corporation ausprobiert. »The GRAIL (*GRA*phical *I*nput *L*anguage) Project proposed to create an interactive software-hardware system in which the man *constructs and manipulates* the display contents *directly and naturally* without the need to instruct an intermediary (the machine); i.e., the display contents should represent, in a very real sense, the man's problem, and allow him to deal directly with it.« Vorausgegangen war der Befund eines leidigen Missverhältnisses zwischen Ein- und Ausgabetechniken und die Entwicklung des RAND-Tablets, das aus einem »printed circuit tablet« und einem Griffel (stylus) bestand. »The flexibility of the output channel (CRT) suggests that the input channel should have the same *freedom*, allowing the man to *express himself directly and naturally* on a two-dimensional *working surface*. When the RAND Corporation began to explore the capabilities of CRT displays, the evident mismatch between output potentials and existing input capabilities led to the investigation of two-dimensional input devices.« »With an evenly matched input-output capability, the CRT face can become the *common working surface* for both man and machine.« (Ellis u.a. 1969, S. 3, 1; meine Hervorhebungen)

Hier liegt keine ausgearbeitete Vision und Voruntersuchung wie bei Doug Engelbart vor, sondern es handelt sich um eine ingenieurmäßige

30 Letztlich ist das Scheitern von NLS auf eine Fehleinschätzung des Einsatzkontextes eines solchen Systems zurückzuführen, was auch in den (zu) hohen Ansprüchen an die Interaktion begründet war. Licklider war da unrealistisch realistischer gewesen, wenn er auf Spracheingabe gesetzt hatte, denn »one can hardly take a military commander or a corporation president away from his work to teach him to type. If computing machines are ever to be used directly by top-level decision makers, it may be worthwhile to provide communication via the most natural means, even at considerable cost.« (Licklider 1960, S. 13) Engelbart beklagt heute, dass mit den etablierten Desktop-Interfaces viele seiner Ansprüche auf der Strecke geblieben sind, und er arbeitet im Abseits weiter an seinem ursprünglichen Konzept im Rahmen des *Bootstrap*-Projekts. (Bootstrap Institute 2007) Eine Aufarbeitung des (sozialen) Scheiterns des ›Knowledge Workshops‹ findet man in (Bardini/Friedewald 2002).

Übertragung der einen zweidimensionalen Freizügigkeit in eine andere. Wir finden jedoch in diesen wenigen Sätzen eine Mischung von alten und neuen Leitbildern: die kybernetische Vorstellung von einer partnerschaftlichen Zusammenarbeit von Mensch und Maschine, aber nicht mehr in Form einer Konversation, sondern vermittelt durch ein graphisches Interface, das, im Sinne von Lickliders Kritik, eine gemeinsame Arbeitsfläche und keine Trennfläche darstellt. Desgleichen die Freiheitsidee einer Entbindung vom eindimensionalen Prozedieren der Maschine, aber nicht wie bei Engelbart in Form von wechselnden Sichten, auf die man im Fortgang zurückgreifen kann, sondern, wie später bei der Xerox PARC-Gruppe, als Möglichkeit, sich unmittelbar und natürlich auszudrücken.³¹ Schließlich entdecken wir in der Abkehr von Shaws Leitgedanken, einen zwischengeschalteten Agenten zu instruieren, einen Vorgriff auf die Konzeptionen der direkten Manipulation. Die Interaktion bleibt jedoch weiterhin in den Kontext des Problemlösens eingebunden, denn ein zentrales Designziel von GRAIL war: »The system to be complete as a problemsolving aid; i.e. the man should be able to specify, edit, validate (debug), document, and exercise his problem description.« (ebda., S. 4)

Was bei Lickliders Bibliothekssystem nur nebenbei in Form von ›Sketchpad-Features‹ angedacht war, wird hier zum zentralen Anliegen der Interaktion und beispielhaft durch die Bearbeitung von Flussdiagrammen vorgestellt. Eingaben des Nutzers mit dem Griffel, etwa eine gezeichnete Linie, werden von der Maschine als »interpretierbare Zeichen« aufgefasst und entsprechend der aus dem Kontext des aktuellen Bildschirminhalts erschlossenen Absicht ›idealisiert‹, d.h. »replaced (in place) by a normalized symbol«. Damit der Zeichner sofort die Interpretation des Computers mit seiner Intention vergleichen kann, ist ein schnelles Feedback erforderlich, »a real-time indication of exactly what is being interpreted«. Vorausgesetzt der »feedback loop is closed quickly enough to avoid a rubbery feeling«, werde die Interaktion, die die Illusion vermitteln soll, man würde direkt auf dem Display schreiben oder zeichnen, es dem Menschen ermöglichen, auf seiner eigenen Sprachebene zu bleiben. Denn die Maschine sei in der Lage, »to interpret relatively natural language elements and relate them to the problem context in real time (man's time).« (ebda., S. 6, 12, 6)

Der Kontext der Arbeit soll also der Kontext des Menschen sein, so wie er ihn auf dem Bildschirm vorfindet und verändert. Die Betonung

31 Thomas O. Ellis und Marvin R. Davis kritisierten schon 1964 die bisherigen Eingabetechniken, weil sie verhinderten, dass »full advantage is being taken of the capabilities of either the machine or of the user«, und sie priesen das Potential des RAND-Tablets im ›ausdrücklichen‹ Sinne der Manipulation: »The concept of generating hand-directed, two-dimensional information on a surface« brächte »a new freedom of expression in direct communication with computers« mit sich. (Davis/Ellis 1964, S. 325)

liegt dabei auf der menschlichen Zeit, die nicht durch maschinellen Eigensinn aufgehalten werden darf. Der Computer interpretiert gemäß dem zu einem inneren Zustand geronnenen Kontext – einem *Mode* – und der physischen Bewegung des Nutzers mit dem Griffel. *Modes* ermöglichen aber nicht nur, wie bei Licklider, der Maschine eine angemessene Interpretation, sondern umgekehrt gibt sie auch dem Nutzer Kontexthinweise in Form von drei Helligkeitswerten: Der »*off mode*« zeigt an, dass eine Operation im aktuellen Kontext nicht anwendbar ist, der »*dim mode*«, dass etwas bearbeitet wird und nicht zur Verfügung steht, und der »*bright mode*« lenkt die Aufmerksamkeit auf ein bestimmtes Objekt. (ebda., S. 8)³²

Technische Beschränkungen nötigten jedoch dazu, dass die reale Zeit der Nutzer mit einer starken Einschränkung des räumlichen Kontextes erkaufte werden musste: »the CRT face is not considered a window into a larger two-dimensional space; but rather each picture remains separate and distinct from all others [...] the fixed-picture approach does not allow the man to view two or more related pictures simultaneously; i.e. spread his work out on the desk.« (ebda., S. 11)³³

Abb. 5: »Selbstporträt« der FLEX-Maschine auf der Titelseite von Alan Kays Dissertation »The Reactive Engine« (Kay 1969)



Eben dies war die Vision von Alan Kay: ein Interface, das statt eines Arbeitsgegenstandes eine Arbeitsfläche anbietet, auf der man gleichzeitig mehrere Objekte betrachten und bearbeiten kann. Darüber hinaus wollte er dies nicht auf einem Großrechner, sondern auf kleinen »persönlichen«

32 Solche Hilfestellungen zu verbessern, war eines der vorrangigen weiteren Ziele: »The next step will be to apply more contextual information to aid recognition as well as to make the recognizer more adaptable to the man.« (Ellis u.a. 1969, S. 15)

33 »A partial solution to this problem is overlaying pictures or splitting the screen between pictures, as appropriate. The display-hardware frame rate introduces enough flicker to limit the usefulness of this approach.« (Ellis u.a. 1969, S. 11)

Maschinen realisieren. In seinem Rückblick »The Early History of Small-talk« beschreibt er, wie er wie viele Andere von Engelbart, einem »prophet of Biblical dimension«, und seinem NLS-System fasziniert war und versuchte, so viel wie möglich von dessen »entire conceptual world and world view« in der von ihm gemeinsam mit Ed Cheadle entwickelten FLEX-Maschine umzusetzen. (Kay 1993, S. 7)

Aber das NLS-System war noch stark modal organisiert. Es hatte verschiedene Programm-Modi und war streng hierarchisch strukturiert, so dass man sich durch mehrere Ebenen zurückhangeln musste, wenn man etwas Anderes machen wollte. Trotz aller Begeisterung fand Alan Kay deshalb: »to me there was a monstrous bug in this approach [...] In hierarchical menus or »screens« one would have to backtrack to a master state in order to go somewhere else. What seemed to be required were states in which there was a transition arrow to every other state [...]. In other words, a much »flutter« interface seemed called for – but could such a thing be made interesting and rich enough to be useful?« (ebda., S. 8) Die später allenthalben propagierte Forderung der »Modelessness« bedeutete für Alan Kay ursprünglich, dass man zwischen Fenstern hin und her wechseln und jeweils die verlassene Situation wieder vorfinden konnte.

»An *intuitive* way to use windows was to activate the window that the mouse was in and bring it to the »top«. This interaction was *modeless* in a special sense of the word. The active window constituted a mode to be sure – one window might hold a painting kit, another might hold text – but one could get to the next window to do something in *without any special termination*. This is what *modeless* came to mean for me – the user could always get to the next thing desired without backing out. The contrast of the nice modeless interactions of windows with the clumsy command syntax of most previous systems directly suggested that everything should be made modeless. Thus began a campaign to »get rid of modes.« (Kay 1990, S. 197)

Mode ist ein sehr unscharfer Begriff, der zu einigen Verwirrungen Anlass gegeben hat. Selbstverständlich gibt es immer *Modes*, diese sind im Prinzip ja nichts anderes als Kontexte, in denen irgendeine Aktivität stattfindet. Im »richtigen Leben« wird situativ angemessenes Handeln durch Ort und Zeit, durch Anmutung – *affordance*³⁴ – und Gewohnheit präfiguriert,

34 Das Design-Schlagwort der »affordance« ist durch Donald Norman populär geworden, der in seinem einflussreichen Buch »The design of everyday things« schreibt: »the term affordance refers to the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used [...] When affordances are taken advantage of, the user knows what to do by looking: no picture, label, or instruction is required. Complex things may require explanations,

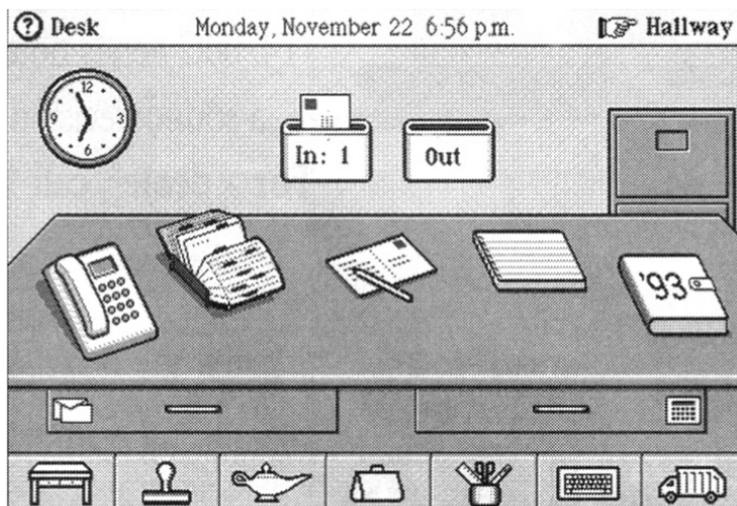
und der Versuch, diese Formen der Kontextualisierung auf die Interaktion mit dem Computer zu übertragen, beschäftigt die ganze Geschichte des User Interface Designs mit mehr oder minderem Erfolg. Wie wir gesehen haben, sollten *Modes* in den früheren Entwürfen den Nutzer von einer Kleinlichkeit der Steuerung entlasten, indem sie der Maschine eine gewisse Interpretation ermöglichen. Darum kann es nicht gehen. Der Ruf »Avoid modes!« meint somit, dass der Kontext nicht durch Zustände der Maschine, die auf ihre Abarbeitung und ihre reduzierte Schnittstelle zurückzuführen sind, vorgegeben werden darf, sondern sich inhaltlich aus der Vorgehensweise der Nutzer ergeben soll. Hinzu kam wohl das Motiv, sich ganz von der Serialität des Dialogs zu befreien, wo der Ablauf oft »einem Frage-Antwort-Spiel, in dem der Computer die führende Rolle übernimmt«, gleicht und der »Mensch sich als Knecht des Computers« fühlt. (Pomberger/Blaschek 1993, S. 68) Die Kampagne, *Modes* loszuwerden, hat jedoch dazu geführt, dass das Kind mit dem Bad ausgeschüttet wurde. Nivergelt und Weydert haben schon 1980 vor allzu großer Freizügigkeit gewarnt: »At any moment a user needs only a small part of all the commands available in the system. [...] Thus the set of all commands must be structured into a space of modes. The commands grouped together in a mode must correspond to a meaningful activity in the user's mind.« (Nivergelt/Weydert 1980, S. 440) Wenn die radikale Forderung der *Modelessness* »anything goes at any time« bedeutet, dann stellt dies eben auch eine Belastung dar, weil jede Orientierung fehlt, was zu tun ist.

Das Nebeneinander, das durch gestaffelte Fenster erreicht wird, ermöglicht ein paralleles Arbeiten in unterschiedlichen, durch die Fenster jeweils vorgegebenen Kontexten. Realisiert wurde ein solches Interface am Xerox PARC gemäß der Metaphorik eines Schreibtisches, auf dem allerlei Unerledigtes herumliegt und der Bearbeitung harret.³⁵

but simple things should not. When simple things need pictures, labels, or instructions, the design has failed.« (Norman 1989, S. 9) Im graphischen Interface haben die einfachen wie die komplexen virtuellen Objekte – Materialien wie Werkzeuge – aber nur eine ikonische Repräsentanz. Man musste nach anderen Mechanismen suchen, ihre Nutzung nahezulegen.

- 35 Alan Kay meinte später, dass die Wahl dieser Metaphorik unglücklich war. »I don't want a screen that is much like my physical desk. It just gets messy – yet I hate to clean up while in the middle of a project.« Eine bessere Alternative wären »'project views' as originally implemented by Dan Ingalls in Smalltalk-76« gewesen: »Each area holds all the tools and materials for a particular project and is automatically suspended when you leave it.«, vergleichbar der heute verfügbaren Technik der »multiple desks«, die aber wohl nicht häufig benutzt wird. (Kay 1990, S. 199 f.)

Abb. 6: *Magic Cap*, ein etwas penetrant metaphorisches Interface von 1994 (*Magic Cap* 1994)



gearbeitet wird auf dem Schreibtisch in einer WIMP-Umgebung, mit ›Windows, Icons, Menus, Pointer‹. Alan Kay, der am Xerox PARC wesentlich für die Konzeption des *Dynabooks* verantwortlich war, hat im Nachhinein erklärt, welche Überlegungen dabei (zumindest bei ihm) eine Rolle spielten.³⁶ Die Entwicklung war stark durch die Vorstellung geprägt, dass der persönliche Computer ein Mittel zum kreativen Ausdruck sein sollte, – »to provide computer support for the creative spirit in everyone«. (Ingalls 1981, S. 286) Der Umgang mit diesem »dynamic medium for creative thought« sollte so leicht erlernbar sein, dass auch Kinder spielend eine ›Computer Literacy‹ erwerben können und durch virtuelles Modellbauen die Welt besser verstehen würden.³⁷ Ein Bericht der Learning Research Group am Xerox PARC über das Dynabook hebt mit Worten an, die an Martin Luther Kings berühmte Rede »I have a dream.« erinnern.

36 Genauerer hierzu in »Konversation, Manipulation, Delegation«. (Pflüger 2004a, S. 379 ff.) Alan Kay schreibt selbst, dass er im Prinzip alle Bestandteile für neuartige Interaktionen vorgefunden hat: »the FLEX machine, the flat-screen display, GRAIL, Barton's 'communications' talk, McLuhan, and Papert's work with children«; die Liste wäre zu ergänzen durch NLS, Sketchpad und Simula. (Kay 1993, S. 10) Sein wichtigster Beitrag bestand in der Vision, die verstreuten Elemente in *einem* Interfaces zu vereinigen.

37 Vorbild für Alan Kay war der von ihm beobachtete Umgang von Kindern mit Paperts Programmiersprache LOGO: »I was possessed by the analogy between literacy and LOGO.« (Kay 1990, S. 193)

»Several years ago, we crystallized our dreams into a design idea for a personal dynamic medium the size of a notebook (the Dynabook) which can be owned by everyone and has the power to handle virtually all of its owner's needs in the universe of knowledge. Towards this goal we have designed and built a communications system: the Smalltalk language, implemented on small computers we refer to as interim Dynabooks. We are exploring the use of this system for programming and problem solving; as an interactive memory for the storage and manipulation of data; as a text editor; and as a medium for expression through drawing, painting, animating pictures, and composing and generating music. Our interim Dynabooks have been in serious use by children and adults since the fall of 1973.« (Xerox 1975, S. 1)

Für die Gestaltung eines Interfaces, das kreative Ausdrucksformen unterstützt, hat sich Alan Kay auf den Psychologen Jerome Bruner berufen, der im Anschluss an Jean Piaget drei unterschiedliche mentale Operationen annimmt: konkretes Operieren, bildliche und symbolische Verarbeitung.³⁸ Die drei mentalen Funktionen haben unterschiedliche Leistungen zu erbringen: *enactive*: »know where you are, manipulate«; *iconic*: »recognize, compare, configure, concrete« und *symbolic*: »tie together long chains of reasoning, abstract«. »Now, if we agree with the evidence that the human cognitive facilities are made up of a *doing* mentality, an *image* mentality, and a *symbolic* mentality, then any user interface we construct should at least cater to the mechanisms that seem to be there.« (Kay 1990, S.195) Kreativität entspringt bekanntlich der rechten Gehirnhälfte, die es nicht so mit dem Symbolisch-Rationalen hat, und somit ist davon auszugehen, dass kreatives Arbeiten hauptsächlich in den beiden ersten Verarbeitungsmodi stattfinden wird, »most in the iconic (or figurative) and quite a bit in the enactive.« Die symbolische Verarbeitung hat nur noch eine spezielle Funktion, anstatt wie früher im Dialog allein zuständig zu sein: »The symbolic system's main job is to stay with a context and to make indirect connections.« (ebda., S.196)

Es ist etwas Merkwürdiges passiert. Die *Modes*, die sonst der Maschine angelastet werden, sind auf die menschliche Verarbeitung, wenn nicht gar auf Persönlichkeiten übergegangen. Erkennt man, »how modeful was a mentality that has »seized control« – particularly the analytical-problem solving one«, dann erscheine die ganze Konzeption der früheren Interfaces einseitig und unzureichend, weil dort nur ein Modus des Geistes den

38 »Our mentalium seems to be made up of multiple separate mentalities with very different characteristics. They reason differently, have different skills, and often are in conflict. Bruner identified a separate mentality with each of Piaget's stages: he called them *enactive*, *iconic*, *symbolic*.« (Kay 1990, S. 194)

ganzen Prozess des Problemlösens dominiert: »the ›symbolic‹ person is good at getting things done, because of the long focus on single contexts, but has a hard time being creative, or even being a good problem solver, because of the extreme tendency to get blocked.« Da keine der mentalen Modi für sich allein sehr erfolgreich sein wird, sei somit die beste Entwurfsstrategie, »to gently force synergy between them in the user interface design.« Es ist nicht mehr von einem Synergieeffekt durch die unterschiedlichen Fähigkeiten von Mensch und Maschine in einer Partnerschaft die Rede, und auch nicht von Engelbarts Vorstellung, geistige Fähigkeiten durch externalisierte Symbolmanipulation zu verstärken, sondern es geht jetzt darum, eine Umgebung zu schaffen, in der die separierten Geister zusammenwerken können. Alan Kay fasst diese integrative Form der Interaktion in dem Slogan zusammen: »Doing with images makes symbols.« Praktisch ergibt sich für das Interface daraus, dass dem *Doing* die Maus, den *Images* die Icons und Windows, und den *Symbols* die Smalltalk-Programmierungsumgebung zugeordnet wird. (ebda.)

Grundlegend für ein Konzept des ›doing with images‹ ist die unmittelbare Reaktion des Computers und die Sichtbarkeit aller Effekte der von der Nutzerin ausgeübten Aktionen; deshalb hat man diese Form der Interaktion als ›direkte Manipulation‹ bezeichnet. Ben Shneiderman, der den Ausdruck geprägt hat, sieht als ihr wesentliches Moment die Möglichkeit eines experimentellen Arbeitens. »The central ideas seemed to be visibility of the objects of interest; rapid, reversible, incremental actions; and replacement of complex command language syntax by direct manipulation of the objects of interest – hence the term ›direct manipulation‹.« (Shneiderman 1983, S. 57) Im gleichen Sinne haben sich auch die Entwickler des Xerox Star Interfaces geäußert.³⁹ »Star users control the system by manipulating graphical elements on the screen, elements that represent the state of the system and data created by users. [...] Users of this type of system have the feeling that they are operating upon the data directly, rather than through an agent [...] A related principle is that the state of the system always shows in the display. Nothing happens ›behind the user's back‹.« (Johnson u.a. 1989, S. 15) »A subtle thing happens when everything is visible: *the display becomes reality*. The user model becomes identical with what is on the screen. Objects can be understood purely in terms of their visible characteristics. Actions can be understood in terms of their effects on the screen. This lets users conduct experiments to test, verify and expand their understanding – the essence of experimental science.«

39 Nach Probeläufen mit dem Xerox Alto, dem »interim Dynabook«, war der Xerox 8010 Star der erste kommerzielle Computer, der 1981 mit einem Manipulationsinterface ausgeliefert wurde. Auf der Website (GUIDebook 2007) findet man Bilder und Texte zum User Interface aus dieser Zeit.

(Smith u.a. 1982, S. 656) Hat man früher Denken als probeweises Handeln im Kopf verstanden, dann soll das Probehandeln jetzt in der Realität eines Bildschirms stattfinden können.⁴⁰ Damit wird nicht nur der dialogische Fluss verworfen, sondern auch Engelbarts Forderung der kreativen Freiheit in der Symbolmanipulation radikalisiert. Durch die Betonung des deiktischen und ›imaginären‹ Tuns sind allerdings seine Vorstellungen von unterschiedlichen Sichten auf Gedankenfragmente und deren inhaltsbezogene Verlinkung auf der Strecke geblieben.⁴¹

Die Forderung der Unmittelbarkeit und Sichtbarkeit bedeutet, dass alles Geschehen im Interface im Modus der *Präsenz* stattfindet. Wenn der Zustand des Systems immer präsent ist und das Display meine Realität darstellt, wird in gewisser Weise die Zeit aus der Interaktion ›geräumt‹. Sie wird zur Eigenzeit des Users, der auf sich selbst zurückgeworfen die einzelnen Schritte seiner Vorgehensweise reflektieren muss. Was man tun kann, erfährt von der Maschine kaum noch Einschränkungen, was man tun will, und das bedeutet im Konstruktionsprozess immer auch, was man als Nächstes tun soll, erhält im kontextfreien Interface aber auch keine Unterstützung.⁴² Im ›doing with images‹ entfällt die Orientierung, die

40 Für exploratives Arbeiten ist die Reversibilität wichtig, eine Eigenschaft, die Piaget im wirklichen Leben erst auf der Stufe des abstrakten, symbolischen Operierens verwirklicht gesehen hat. Die Ökonomie des *Undo* wird ›konkret‹ möglich, weil man es am Computer nur mit virtuellen Objekten und Operationen zu tun hat.

41 Engelbart hat in einem Interview kritisiert, dass die Manipulationskonzeption so vom (druck)fertigen Produkt fasziniert war, dass sie die ›Rücksicht‹ auf das geistige Werden aus den Augen verloren haben. »The NLS did much more than we now think of as word processing. It gave you optional views, so that you could look at the same document in many different ways. Computers can be so flexible in how they present things. Why do they have to show you a document only as if it is printed on paper? [...] Our approach was very different from what they called 'office automation,' which was about automating the paperwork of secretaries. That became the focus of Xerox PARC in the '70s. They were quite amazed that they could actually get text on the screen to appear the way it would when printed by a laser printer. Sure, that was an enormous accomplishment, and understandably it swayed their thinking. They called it 'what you see is what you get' editing, or WYSIWYG. I say, yeah, but that's all you get. [...] We weren't interested in 'automation' but in 'augmentation.' We were not just building a tool, we were designing an entire system for working with knowledge.« (Engelbart 2004)

Bedenkt man, dass noch heute weder Microsofts *Word* noch Apples *Pages* gescheit mit Fußnoten umgehen können, möchte man das Versprechen ›whatyouseeiswhatyouget‹ eher für eine Drohung halten.

42 Wenn in den »Microsoft Inductive Interface Guidelines« zu lesen ist: »A well-designed inductive interface helps users answer two fundamental questions they face when looking at a screen: What am I supposed to do now? Where do I go from here to accomplish my next task?«, dann haben sie

Licklider im Vorausdenken des menschlichen Partners gesehen hat. Denn dort war die symbolische Mentalität allein für die ›Gesprächsführung‹ und die sich diskursiv entwickelnde Struktur zuständig, und deren Stärke ist es ja, »to stay with a context«. Alan Kay hat später Bedenken geäußert, ob nicht »the siren's song of trying to do symbolic thinking iconically was just too strong.« (Kay 1990, S. 197) Die symbolische ›Bildhauerei‹ kennt kein Ziel, von dem her sich der jeweilige Kontext bestimmt. Sie ist nicht im gleichen Maße wie der Dialog eine Problemlösetechnik mit einem ›goal-seeking behavior‹, sondern gleicht eher einem Basteln mit LOGO-Bausteinen. »Almost everything to the iconic mentality is ›before-after‹, like a bird building a nest. The current state of things suggests what to do next. Extensive top-down planning is not required – just squish things around until you like the total effect.« (ebda., S. 201 f.)

Im Rückblick kann man sehen, dass Alan Kay zu sehr von LOGO und den Kindern »who are learning about ›ideas and reality‹ by designing and constructing tools of their own« ›besessen‹ war. (Xerox 1975, S. 48) Tatsächlich wird das ›doing with images‹ im Wesentlichen nur für Interface-Operationen und nicht für Problemlöseprozesse oder Modellbauereien verwandt. Wenn die manipulative Interaktion vieles vom Symbolischen ins Ikonische verlagert, stellt sich die Frage, ob es überhaupt möglich ist, »to permit long chains of abstract construction to be carried out in a concrete world.«⁴³ »One of the problems is how to get concrete signs to be more abstract without simply evoking the kinds of symbols used by the symbol mentality. More difficult is how to introduce context in a domain whose great trait is its modeless context-freeness.« (Kay 1990, S. 202 f.) Das Problem ist, dass Bilder zwar für sich selbst sprechen, aber untereinander keine logische Ordnung oder Reihenfolge aufweisen, in der sie zu verstehen sind. Was bei der Konstruktion einer Bildfolge oder eines Diagramms einsichtig ist, weil man sich dabei nur auf die jeweils relevanten Teile konzentriert, kann später bei der Betrachtung völlig unverständlich erscheinen. Bei der Konstruktion wirkt das Geleit, das Licklider für die symbiotische Interaktion angenommen hat, aber im Umgang mit fertigen Bildern ist diese zeitliche Kontextualisierung nicht mehr vorhanden. Dies gilt nicht nur für komplexe Konstruktionen, sondern auch für einfachere Interaktionssequenzen. Wenn Dan Ingalls für das Interface fordert: »Every

etwas gründlich missverstanden. (Microsoft 2001) Das Sollen sollte nicht die Maschine fordern, sondern eine taktische Reflexion der Nutzerin sein. Licklider hatte seinen goldenen Knopf eben nur für den Notfall, und nicht als ›goldene Regel‹ der Interaktion vorgesehen. Und bezeichnenderweise erscheint ›where to go‹ jetzt nicht mehr zeitlich, sondern als ›Verrichtung‹ in einem Möglichkeitsraum.

43 Die vielen gescheiterten Ansätze einer visuellen End-User-Programmierung lassen starke Zweifel aufkommen.

component accessible to the user should be able to present itself in a meaningful way for observation and manipulation.« (Ingalls 1981, S. 286), und das User-Modell dem Bildschirm entsprechen soll, dann ergibt das keine vergleichbare Strukturierung wie zusammengefasste Befehle, die einer »meaningful activity in the user's mind« korrespondieren.⁴⁴

Für eine prozessuale Kontextualisierung gibt es im Modus der Präsenz nur die Möglichkeit, mit ›Metaphern‹ Vorgriffe auf anschließendes Prozedieren nahezulegen, was sie auch für jemand, der mit einem Ablauf vertraut ist, etwas nervig macht. Sie sind als ein Mittel zu sehen, einen Zeit-horizont in die Präsenz einzuführen, insofern als die Ähnlichkeit mit vertrauten Vor-Bildern Hinweise gibt, wie sinnvoll weiter zu verfahren ist. Aber ihre Beschränktheit liegt auf der Hand, weil sie im besten Fall nur Anstöße geben und keine komplexe Interaktion strukturieren. Andererseits kann die Übertragung des Vertrauten in die Irre führen. Problematisch ist, dass Bild- und Urbildbereich jeweils andere Eigenschaften haben können, und wo sie übereinstimmen, doch oftmals im Detail anders funktionieren. Solche Diskrepanzen können zu entsprechenden Fehldeutungen führen.⁴⁵

Die direkte Manipulation erfordert eine kleinteilige Aktivität. Die manipulative Einzelaktion hat die Nachfolge des Kommandos im Dialog angetreten. Die Ermächtigung, die Licklider seinen Nutzern geben wollte, hat zum Hantieren auf einem »atomic level« geführt. »Instead of an executive who gives high-level instructions, the user is reduced to an assembly line worker who must carry out the same task over and over.« (Gentner/

44 Später wurden gebündelte Befehlssequenzen in Form von vom Nutzer definierbaren Makros wieder eingeführt. Diese haben aber nichts mit bildlicher Präsenz zu tun, sondern bringen geronnene Erfahrung in einem bestimmten Kontext auf den ›Begriff‹.

45 Die Kritik am Metaphernkonzept ist fast so alt wie dieses selbst. Frank Halasz und Thomas Moran haben schon 1982 in ihrem Beitrag »Analogy Considered Harmful« gemahnt: »analogical models can often act as barriers preventing new users from developing an effective understanding of systems«. (Halasz/Moran 1982, S. 383) Halasz und Moran sprechen zurecht von Analogien und nicht von Metaphern. Metaphern sind nur Metaphern in einem bestimmten Kontext, in dem sie einen ›Störung‹ darstellen, weil sie kontextabhängige Erwartungen enttäuschen. Aber was sollte man auf einem Bildschirm statt eines Papierkorbs erwarten? Die Ent-Täuschung tritt erst ein, wenn er nicht so funktioniert, wie man meint. Deshalb ist der Ausdruck ›Metapher‹ recht unpassend, und ›Analogie‹ wäre vorzuziehen. Auch Alan Kay war von der Wortwahl nicht begeistert: »My main complaint is that *metaphor* is a poor metaphor for what needs to be done. At PARC we coined the phrase *user illusion* to describe what we were about when designing user interface. [...] it is the magic – understandable magic – that really counts.« (Kay 1990, S. 199) Bruce Tognazzini hat dazu einen Artikel »Principles, Techniques, and Ethics of Stage Magic and Their Application to Human Interface Design« geschrieben. (Tognazzini 1993)

Nielsen 1996, S.74) Dass man manipulativ keine Befehlsfolgen zusammenfassen oder konditional ausführen kann, ist sogar ein Rückschritt hinter JOSS, wo die Definierbarkeit von ›Makros‹ und die Möglichkeit, Befehle von Bedingungen abhängig zu machen, als ein »extremely powerful feature« wahrgenommen wurde. Das deiktische Konzept – »see and point«, das nach Alan Kays Traum Kreativität fördern soll, stellt zugleich einen massiven Sprachverlust dar, denn jede Referenz der Nutzeraktivitäten ist an die Präsenz gebunden, wohingegen man in natürlicher Sprache auf zeitlich und räumlich Abwesendes verweisen kann.⁴⁶ Mit den Folgen von Einzelaktionen hat auch die Kontrolle die Last der Kleinteiligkeit übernommen. »The negative side of user control is that the user *has* to be in control.« Die Nutzerin ist zwangsläufig in alle Aktionen eingebunden; viele Aktivitäten sind jedoch so kompliziert oder so langweilig langwierig, dass man sie an die Maschine delegieren möchte. »This is prime territory for agents and daemons, computer processes that tirelessly stand guard, take care of routine tasks, or have the knowledge that we lack to handle complex tasks.« (ebda., S. 76)⁴⁷

Wenn der strukturierende zeitliche Kontext weggefallen ist und die bedeutungsvolle Präsentation der Interface-Komponenten nur für sich spricht, wird die Orientierung im Raum und die Gestaltung des räumlichen Kontextes vordringlich. Die Werkstätte der direkten Manipulation ist vor allem ein Möglichkeitsraum, in dem Objekte und Werkzeuge bereitliegen. Das Interface ist inhärent passiv, – Manifestation einer »reactive

46 Don Gentner und Jacob Nielsen drücken das in ihrer Kritik der direkten Manipulation recht drastisch aus: »It's as if we have thrown away a million years of evolution, lost our facility with expressive language, and been reduced to pointing at objects in the immediate environment. Mouse buttons and modifier keys give us a vocabulary equivalent to a few different grunts. We have lost all the power of language, and can no longer talk about objects that are not immediately visible (all files more than one week old), objects that don't exist yet (future messages from my boss), or unknown objects (any guides to restaurants in Boston). [...] Real expressive power comes from the combination of language, examples, and pointing.« (Gentner/Nielsen 1996, S. 75)

47 Mit der Delegation von Aufgaben an Agenten kann auch die Forderung eines schnellen Feedbacks abgeschwächt werden, weil eine sofortige Reaktion nur bei einer Einzelaktion sinnvoll ist. Wenn eine Aktionssequenz als Makro (oder Skript) definiert oder an einen Agenten delegiert worden ist, braucht es keine aufdringlichen Hinweise, solange die Ausführung im Hintergrund läuft. »The user doesn't have to be bothered unless the system encounters a problem that it cannot handle.« (Gentner/Nielsen 1996, S. 76) Dann gerät der delegierende Nutzer allerdings wieder in die Lage des Kontrolleurs, die Licklider mit dem permanenten ›look-ahead‹ des menschlichen Partners in der Symbiose zu lösen meinte.

engine«, wie Alan Kays Dissertation betitelt ist. (Kay 1969)⁴⁸ Die radikale Verwirklichung der ›Modelessness‹, gemäß der »Users should be able to perform any task at any time.«, macht dem Nutzer, der nicht alle Möglichkeiten gleichzeitig überblicken kann, das Leben schwer. Die raumzeitliche Kontextfreiheit führt zu einer Überlastung, und die ›Nestbauermentalität‹ muss sich andauernd fragen, was jetzt sinnvoll ist. Es erscheint somit notwendig, den allzu umfassenden Möglichkeitsraum situativ einzuschränken und die Bereitstellung der sich präsentierenden Interface-Elemente zu strukturieren; aber das bedeutet, wieder mehr Kontext, also *Modes* einzuführen.

Die ›objektorientierte‹ Interaktionsstruktur, die die Operationalität an ein Objekt bindet, schränkt in einem ersten Schritt die Werkzeuge auf die sinnvoll anwendbaren ein.⁴⁹ Im Sinne der visuellen Präsenz werden die Operationen in Menüs angeboten, die das Gedächtnis entlasten, weil man sich nicht Befehle merken muss, sondern aus einem angezeigten Vorrat auswählen kann.⁵⁰ Situativ nicht anwendbare Funktionen aus dem Menü werden abgeblendet, wie das schon in GRAIL angedacht war. Aber die angebotenen Operationen erfassen natürlich nur, was die Entwickler vorgedacht haben, und es ist nicht möglich, sie auf neue Art zu kombinieren. Später hat man die Kontextsensitivität der Präsentation durch weitere Behelfe wie Tooltips und Kontextmenüs ergänzt. Es werden dem Nutzer lokal Hilfestellungen angeboten, oder es lassen sich im jeweiligen Kontext vielseitig verwendbare Tools aufrufen, die man nicht erst suchen muss.

Alle diese Kontextualisierungen betreffen nur die Selektion, sie können nicht mehr als Auswählbares einschränken oder lokalisieren.⁵¹ Wie das

48 Technisch beruht die Bereitschaft des Interfaces auf einem *Event-Loop*. Die Eigenaktivität der Maschine dreht sich in einer ›Erwarteschleife‹, die Events, auf die sie reagiert, sind die Aktivitäten der User.

49 Die *Star*-Entwickler haben das mit einer linguistischen Metapher ausgedrückt: »Commands in Star take the form of noun-verb. You specify the object of interest (the 'noun') and then invoke a command to manipulate it (the 'verb'). Specifying an object is called 'making a selection.' [...] This helps make the command interface modeless; you can change your mind as to which object to affect simply by making a new selection before invoking the command.« (Smith u.a. 1982, S. 659 f.) Insofern der Arbeitsgegenstand ein konkreteres Objekt als ein vielseitiges Werkzeug ist, entspricht das Alan Kays Vorstellung, Abstraktes durch konkretes Operieren zu gewinnen: »we have the object first and the desire second. This unifies the concrete with the abstract in a highly satisfying way.« (Kay 1990, S. 197)

50 Im User Interface Design wird das unter den Alternativen »knowledge in the world« versus »knowledge in the head« angesprochen.

51 In geringem Umfange wurden auch zeitliche Kontexte der Nutzungsgeschichte eingeführt. So kann sich das Interface eine Präsentationsform von Fenstern merken und in Zukunft reproduzieren oder Listen mit den zuletzt verwendeten Objekten zum schnelleren Zugriff bereitstellen.

Fehlen eines zeitlichen Kontextes ein Problem für »long chains of abstract construction« ist, ergibt sich für den räumlichen Kontext ein ebenso gravierendes durch die schiere Masse an vorhandenen Objekten. Die direkte Manipulation ist ein Interaktionskonzept für »very small worlds, if what you want to do can be represented by dragging around icons«. (Brennan 1990, S. 399) Aber inzwischen hat sich die überwiegende Nutzung des Computers stark verändert. Browser und Viewer, Spiele und Player sind zu den gebräuchlichsten Applikationen geworden. Das Gros der User konsumiert schlicht Myriaden von Dokumenten und Medienprodukten, die Verzeichnishöhlen füllen und den Desktop zumüllen. Durch die fast selbstverständliche Anbindung ans Web hat sich die kleine, schon über-völkerte Welt zudem in die Weiten des Internets ausgeweitet. Der Browser ist zum eigentlichen »Betriebssystem« oder Interface geworden, – ein Fenster zur digitalen Welt und nicht nur zum Hinterhof. Damit fallen neue Anforderungen an, und der Kontext der eigenen Aktivität wird zugleich eingeschränkter wie auch unübersichtlicher, denn er unterliegt nicht mehr allein der eigenen Kontrolle.⁵² Die Entwicklung von neuen Interfaces muss dem Rechnung tragen und verstärkt selbsttätige, »proaktive« Mechanismen einbauen, die versuchen, die Tätigkeiten der Nutzer kontextabhängig zu erkennen, zu integrieren und zu ergänzen. Alternative Interface-Elemente sollen von der Organisation der Dokumentenflut entlasten. Und Agenten sollen selbstständig an sie delegierte Aufgaben bearbeiten, die zu kleinteilig oder zeitaufwendig in der Welt der Informationsjäger und -sammler sind.

52 Man muss dann mit Böswilligkeit rechnen. Die meisten Angriffe durch Malware nutzen vorgesehene Kontexte in nicht vorgesehener Weise aus. Entweder indem Nutzer über den (sozialen) Kontext von empfangenen Dateien getäuscht werden und glauben, etwas Unschädliches auszuführen, oder indem operationale Umgebungen entgegen den Erwartungen der Entwickler (z.B. durch »SQL-Injections«) missbraucht werden.

4 Bot-mäßige Initiativen

Natürlich ist die reine WIMP-Lehre und die passive Werkstätte, in der Ikonen ›anmutig‹ herumliegen, praktisch nie rigoros umgesetzt worden. Jedoch wurde die Konzeption des Star/Macintosh-Interfaces so dominierend, dass in der Folgezeit keine relevanten alternativen Visionen entstanden sind.⁵³ Stattdessen führte die Kritik an einzelnen Aspekten der Manipulationskonzeption dazu, dass diese schrittweise modifiziert wurden.⁵⁴ Von mehr inter-aktiven Freiheiten ist nicht mehr die Rede, eher ist eine Entlastung von zuviel Freiheit angesagt. Wir beobachten eine Evolution des Interfaces mit einer zunehmenden ›Eigeninitiative‹ des Computers, – *Proaktivität*, wie es so unschön heisst. Zuerst in Form von selbsttätigen Programmen, die vorsorglich etwas im Hintergrund ausführen, beispielsweise Dokumente indexieren oder recherchieren. Solche botmäßigen Programme erklärte man in den 90er Jahren zum neuen Paradigma der ›autonomen Agenten‹, die von der kleinlichen Manipulation entlasten, weil man Aufgaben an sie delegieren kann; und wie so oft wurden überzogene Erwartungen an deren ›intelligente‹ Dienstleistungen geknüpft. Gewünschte Proaktivität meint dann, dass die Agenten stellvertretend für ihre Nutzer auch kontextabhängige Entscheidungen treffen oder Vorschläge machen. Und während sesshafte Interface-Agenten hinter den Kulissen wirken und nur bei passender oder unpassender Gelegenheit die Bühne des Bildschirms betreten, werden ›Softbots‹ auch in die Weiten des Internets geschickt, wo sie ihre Nutzer vertreten sollen. Deren Erwartung, in einem inhaltlich bestimmten Kontext Ziele vorgeben zu können, statt Befehle erteilen zu müssen, führt dazu, dass den Agenten eine komplexere ›Context-Awareness‹ abverlangt wird. Die nächste Phase des ›Ubiquitous Computing‹ will dann mit Sensorik solches ›Umweltbewusstsein‹ auf die reale Welt übertragen.

Um die gradualistische Evolution der ›Bots‹ und ihrer Umgebungen nachzuzeichnen, werde ich in diesem Abschnitt entsprechend schrittweise

53 Alan Kay hat das Problem pointiert formuliert: »now that the Mac way of doing things has taken hold, will we ever be able to get rid of it? If the IBM 3270/PC way of doing things is ›machine code‹, doesn't that make the Mac the COBOL of user interface?« (Kay 1990, S. 198)

54 Die meisten Kritikpunkte wurden von Don Gentner und Jacob Nielsen 1996 in einem Aufsatz über ein »Anti-Mac Interface« zusammengefasst, in dem sie Schwachstellen der ›zehn Gebote‹ des vorherrschenden Paradigmas aufzeigen und Alternativen vorschlagen. (Gentner/Nielsen 1996) Getreu Alan Kays Ausspruch, der Macintosh sei »the first personal computer good enough to be criticized«, beziehen sie sich auf die Gestaltungsprinzipien für die Mensch-Computer-Interaktion bei graphischen Interfaces, die in den »Macintosh Human Interface Guidelines« kanonisiert und von den meisten anderen Interface-Richtlinien übernommen wurden. (Apple 1992)

verfahren und zuerst einige konkrete Interface-Mechanismen betrachten, die, ohne das prinzipielle Konzept zu revolutionieren, die Interaktion mit dem Computer drastisch verändert haben. Ihre Auswahl ist subjektiv, aber auch exemplarisch, insofern als sie einerseits Engelbarts These bestätigen, dass kleine technische Veränderungen große Wirkungen auf den Umgang mit Wissen haben können; andererseits zeigen sie, dass ihre Notwendigkeit nur im Kontext kultureller Umgangsweisen zu würdigen ist.

Alle Exempel haben mit der Kontextualisierung von *vorgefertigter* Information zu tun. Sie reflektieren, dass sich der Nutzungskontext des Computers im letzten Jahrzehnt entscheidend geändert hat: Aus einem Interaktionspartner oder einem ›Knowledge Workshop‹ ist ein ›Konsumartikel‹ geworden. Die alltägliche Interaktion dient nicht mehr einem Problemlöseprozess und auch nicht zum Ausdruck für kreative Individuen, sondern der Computer erscheint heute vor allem deshalb als Kulturtechnik, weil mit ihm eine ungeheure Warensammlung von Dokumenten und Medienerzeugnissen – »Wissen aller Arten, in jeder Menge und Güte« (Spinner 1994) – angehäuft und konsumiert wird. Es geht also bei den neuen Techniken vor allem darum, vorhandene Objekte ›zuhänden‹ zu machen.⁵⁵ Wenn das Manipulationsinterface im Einklang mit der Objektorientierung den Vorrang des zu bearbeitenden Objekts vorgesehen hatte, dann wurde nicht vorhergesehen, dass aus der produktiven Werkstatt ein Warenhaus wird, in dem Konsumgüter verwaltet und distribuiert werden müssen.⁵⁶ Dem Wandel vom Produktions- zum Konsumtionsmittel widerspricht weder, dass im Geschäftsleben einige ›Bürowerkzeuge‹ massenhaft genutzt werden, noch dass mit *iMovie* und ähnlichen Programmen auch Laien ihre Freizeit präsentabel gestalten können. Bezeichnenderweise hat sich die Standardsoftware konzeptionell

55 Man kann darin eine Einkehr der Realität in die vor einigen Jahren grassierende Heideggermode in der Informatik sehen, deren Ideal die Zuhandtheit des Werkzeuges war, wohingegen Vorhandenheit nur im seinem ›break-down‹ auffällig werden durfte. Dagegen wären die neuen Techniken im gleichen Jargon als ›Hüter des Vorhandenseins‹ anzusprechen; es geht nicht mehr um die Handhabung von ein paar Hämmern, sondern um die Verfügbarkeit von unmäßig vielen Nägeln.

56 Auch Gentner und Nielsen haben auf diesen Kontextwandel hingewiesen: »the Macintosh was optimized for a certain user community with a certain main type of data: knowledge workers without any computer science background who wanted to manipulate graphic documents, but not overwhelmingly many of them [...] it becomes impossible to make all objects visible in the interface as system usage changes to deal with millions of information objects. But having invisible objects not only breaks the design principle of see-and-point, it makes direct manipulation impossible«. (Gentner/Nielsen 1996, S. 80 f.)

recht wenig weiterentwickelt, und die Selbstverwirklichung mit kreativen Tools dürfte doch einer Minderheit vorbehalten bleiben.⁵⁷

Mit überwältigend vielen geöffneten Fenstern und Objekthalden auf dem Desktop funktioniert Alan Kays Idee des schnellen Kontextwechsels mit überlappenden Fenstern nicht mehr. Deshalb wurde in Mac OS 10.3 der Interface-Mechanismus *Exposé* eingeführt, mit dem man zwei unterschiedliche Aktionen per Knopfdruck ausführen kann: Mit einem Tastendruck verschwinden blitzschnell alle Fenster im Bildschirmrand, so dass man freie Einsicht auf den Desktop hat, dort etwas auswählen und vor oder nach Loslassen der Taste ›drag&droppen‹ kann. Die zweite Aktion präsentiert alle Fenster ohne Überlappung in verkleinerter Ansicht. Wenn man eines der Fenster auswählt, wird es im Vordergrund aktiv, und der Bildschirm nimmt ansonsten wieder sein vorheriges Chaos an. Beide Mechanismen verbleiben ganz im Horizont der direkten Manipulation; ein Knopfdruck genügt, und die Reaktion erfolgt ohne Verzögerung. Jeder metaphorische Bezug auf eine ›reale‹ Operation ist aufgekündigt; trotzdem ist der Mechanismus sofort verständlich, und er ermöglicht es, Kontexte zu wechseln, ohne sich durch einen Fensterstapel hangeln zu müssen. Der schnelle Sichtwechsel löst auf einfachste Weise für ein Interface ein Problem, das den ganzen Bereich der Informationsvisualisierung durchzieht: das Verhältnis von Ansicht und Übersicht, von Information und Kontext.⁵⁸

Einen bescheidenen Ansatz zur Proaktivität weisen sogenannte *Launch*-Programme auf, deren Eigeninitiative jedoch auf die selbsttätige Indexierung und eine Anpassung an die Nutzungsgeschichte beschränkt bleibt. Mit ihnen kann man das Durchsuchen von Dateiverzeichnissen vermeiden und Applikationen oder tief vergrabene Ordner/Dateien direkt adressieren.⁵⁹ Dazu muss man sie per Tastenkombination aktivieren und

57 Jedenfalls jammern die Berufsgenossenschaften, dass die Auszubildenden trotz Computerspielen viel zu wenig ›Computer Literacy‹ mitbrächten. (Spiegel Online 12.8.2007)

58 Vergleichbare Mechanismen wurden in Browsern aufgenommen. So gibt es für Firefox Plugins, mit denen man sich alle Tab-Fenster nebeneinander anzeigen lassen kann. Auch andere Techniken der Informationsvisualisierung werden ins Browser-Interface übertragen, beispielsweise gibt es auch einen Plugin, der den Fisheye-View für überfüllte Tab-Leisten verwendet.

59 Eine ältere Zugriffstechnologie, die schon die Passivität des Interfaces durchbrach, sind ›spring-loaded folders‹. Dieser Mechanismus ermöglicht es, eine Hierarchie von Dateiverzeichnissen schneller hinabzusteigen, weil sich die Ordner im Pfad automatisch öffnen, wenn man kurz auf ihrem Icon verweilt. Legt man dann eine Datei in einem tief vergrabenen Unterverzeichnis ab oder holt sie von dort, wird die ganze Fensterstaffel wieder geschlossen. Der Mechanismus agiert weder proaktiv, noch weist er eine technisch über GRAIL hinausgehende Kontexterfassung auf. Aber er beeindruckt auf fast lächerliche Weise, weil der Nutzer die Maschine im

eine kurze Zeichenkette eingeben, die irgendwas mit dem Namen des gesuchten Objektes zu tun hat. Beim ersten Mal bekommt man eine Liste von möglichen Kandidaten angeboten; wählt man einen aus und weist ihm eine Default-Operation zu, wird die Einstellung zukünftig als Favorit angenommen. Es handelt sich nach wie vor um ein *Doing*-Konzept, eigentlich stellt es sogar einen Rückgriff auf die alte Kommando-Interaktion dar.⁶⁰ Aber ein solches Programm ist insofern als Agent anzusehen, als es in regelmäßigen Abständen die Festplatte für sein Angebot indexiert. Und es verwendet eine dynamisch angepasste Kontextualisierung, indem es sich die Vorlieben der Nutzerin merkt; diese können aber leicht verändert werden, weil man durch eine kleine Verzögerung wieder alle Kandidaten angezeigt bekommt.

Solche Zugriffsmechanismen lassen die an der Speicherung ausgerichtete Dateiverwaltung intakt. Sie bieten nur ›Kurzschlüsse‹ an, sich in der hierarchischen Verzeichnisstruktur zu bewegen. Diese entspricht, wie die meisten Klassifikationsschemata, dem physischen Vorbild einer Kartei-kastenablage und stellt eine viel problematischere metaphorische Behinderung dar, als ihre bekrittelte Visualisierung durch kleine Ordnerbildchen. Denn sie bringt mit sich, dass ein Dokument nicht nur physisch, sondern auch in seiner Re-Präsentation im Interface nur einmal vorhanden ist und mittels einer suchbaumartigen Verzweigungsstruktur eingeordnet oder gefunden werden muss. Der primäre Kontext eines Dokuments ist somit der Ort, wo es anzutreffen ist. Eine inhaltliche Zuordnung muss durch eine sinnvoll benannte Örtlichkeit getroffen und zum Zeitpunkt der Erstellung fixiert werden. Das erzwingt somit, dass man Dokumente rechtzeitig an dem dafür vorgesehenen Platz ablegt und später auch weiß, wo sie sich befinden. Elektronisch ist das nicht erforderlich, weil die Präsentation von virtuellen Objekten immer schon von ihrer physischen Speicherung getrennt ist. Durch verlinkte Alias-Objekte hat man die Ortsgebundenheit zu umgehen versucht, aber diese stellen nur einen Notbehelf dar, weil weiterhin der explizite Bezug auf das Original herge-

wörtlichen wie im bildlichen Sinne als ›entgegenkommend‹ erlebt. Auch wenn die Kontexterfassung auf eine entschlossene Unentschlossenheit der Nutzerin in einem (einstellbaren) Zeitrahmen angewiesen ist, führt der Mechanismus, durch die Interpretation einer kleinen Weile als Intention, einige Millisekunden Geschichte in die Präsenz der Manipulation ein.

60 Da man mit einem *Launcher* wie beispielsweise *Quicksilver* noch sehr viel Anderes auf ähnliche Weise ausführen kann, führt seine intensive Nutzung zu einem Interaktionsstil, den man als Keyboard-Virtuosität bezeichnen könnte und der keine Ähnlichkeit mehr mit dem ›doing with images‹ hat. Solche Umgangsformen sind auch von Jef Raskin, der am Macintosh-Interface maßgeblich beteiligt war und später zum Dissidenten wurde, in seinem Buch »The Humane Interface« als bessere Alternative zum WIMP-Interface propagiert worden. (Raskin 2000)

stellt werden muss; sie entsprechen eingelegten Zettelchen in einem Karteikasten, die auf einen anderen Ort verweisen.

Eigentlich will man etwas nach inhaltlichen Kriterien finden und interessiert sich nicht dafür, wo es gelagert ist.⁶¹ Weitergehend möchte man ein Dokument in jeder inhaltlich bestimmten Umgebung als virtuelles ›Original‹ vor- oder zuhanden finden; die Zuordnung zum physisch gespeicherten Objekt sollte von der Maschine getroffen werden. Ein erster Schritt in diese Richtung waren bessere Suchfunktionen, mit denen man nicht nur den Namen, sondern auch Content und Metadaten von Dokumenten indexieren kann und diese dann inhaltlich assoziativ und nicht nur über ihre Adresse oder ihren Namen wiederfindet.⁶² Programme wie *iTunes* und *iPhoto* übernehmen die Verwaltung der Ton- bzw. Bilddateien vollständig, so dass man überhaupt nicht mehr zu wissen braucht, wo sie sich befinden und wie sie heißen. Ihre Nutzung ist so gedacht, dass man Zugang zu den Objekten nur noch über Suchfunktionen, inhaltsbezogene Gruppierungen oder durch Browsen in unterschiedlich detaillierten Überblicken (Thumbnails) gewinnt. Mittels vom Nutzer angelegten ›Alben‹ oder ›Wiedergabelisten‹ lassen sich zu einem Objekt beliebig viele Instanzen in unterschiedlichen inhaltlichen Kontexten erklären. Sie erfordern zwar meist noch explizite Annotationen, aber verbreitete Konsumgüter enthalten oft schon entsprechende Metainformationen (Tags) oder sie können aus Datenbanken im Internet heruntergeladen und automatisch ausgewertet werden.

Mit sogenannten *Smart Folders* (Smart Playlist, Smart Album) ist der (vorläufig) letzte Schritt zur inhaltsbezogenen Kontextualisierung getan. Hier werden neu hinzugekommene Objekte, die bestimmte Merkmale aufweisen, automatisch in die durch den Smart Folder repräsentierte Umgebung einsortiert, wobei deren Inhalt durch diese Merkmale charakterisiert ist.⁶³ Die Merkmale können wieder vom Nutzer explizit annotiert

61 Ein schönes Schlaglicht auf den Unterschied zwischen Behälter und Inhalt wirft ein Vorschlag, den Peter Purgathofer gemacht hat: In Texten sei es sinnvoller, zitierte elektronische Dokumente statt mit einer URL durch ein paar inhaltsbezogene Stichworte anzugeben, mit denen das Dokument in Google auf der ersten Seite erscheint. Diese lassen sich verstehen und viel schneller eintippen als eine lange, sinnlose Zeichenkette, die nichts als einen Ort bezeichnet. Die Ungewissheit, ob Google die Quelle fürderhin im vorderen Feld anzeigen wird, ist auch nicht größer als 404.

62 Für meine Nutzung des Email-Clients hat sich daraus ergeben, dass ich Mails nicht mehr in separaten Mailboxen organisiere, was aufwendig ist und womit ich früher immer im Rückstand war, sondern nur noch nach Jahren getrennt ablege und nach bestimmten Mails mittels Stichworten suche.

63 Wenn man will, kann man in den Smart Folders technische Inkarnationen der klassischen Konzeption eines Begriffs sehen. Der Begriffsinhalt sind die

oder aus Datenbanken gezogen werden, aber es ist auch möglich, zeitliche Kontexte der Nutzungsgeschichte zu verwenden.⁶⁴ In iTunes werden zum Beispiel vorgefertigte Smart Playlists bereitgestellt, die einen zeitlichen Kontext anhand der zuletzt oder häufig konsumierten Güter repräsentieren, und damit implizit auch sowas wie eine subjektive Bewertung.

Die unterschiedlichen Sichten und automatisch erstellten Kontexte, die mit der Entlastung des Nutzers von der Verwaltung der Dateien Hand in Hand gehen, sind nicht zufällig zuerst im Kontext von Medienprodukten entstanden. Diese Sichten haben wenig mit Engelbarts Projektionen von ›thought vectors in concept space‹ zu tun, sondern drehen sich um Übersicht im Sammelsurium des Konsums. In Mac OS 10.4 wurden mit *Spotlight* die erweiterten Suchfunktionen und Smart Folders für den Zugriff auf alle Dokumente ins Betriebssystem integriert. Es handelt sich jedoch weiterhin um einen Interface-Mechanismus, der oben draufgesetzt ist; man kann ihn völlig ignorieren und in hergebrachter Weise agieren. Es erscheint aber durchaus nicht unwahrscheinlich, dass man in einer zukünftigen Betriebssystemversion als normaler Nutzer keinerlei Einfluss mehr auf die Speicherung hat und nicht mehr auf ›ausgetretenen Pfaden‹ an seine Dateien herankommt.

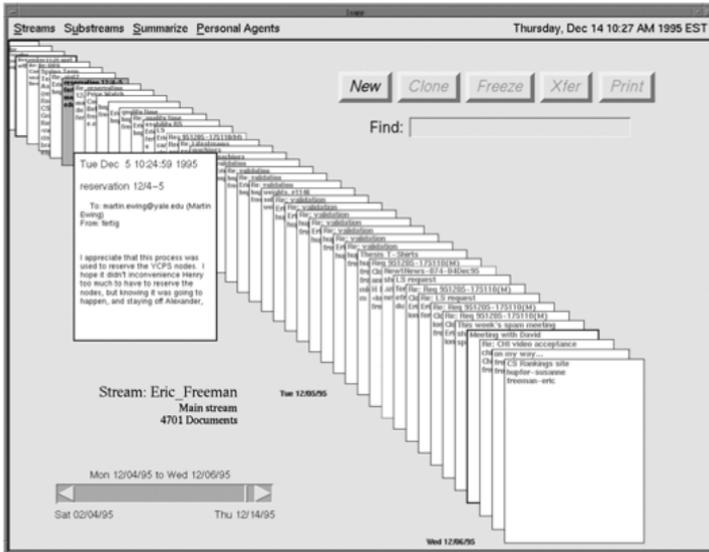
Auf dem Weg dorthin befand sich das radikalere Konzept von *Lifestreams*, das aber unzeitig auf einem Holzweg war. Scott Fertig, Eric Freeman und David Gelernter stellten, im gleichen Jahr wie Gentner und Nielsen ihre Mac-Kritik, die Konzeption eines ›time-ordered stream of documents that functions as a diary of your electronic life‹ als Alternative zum Desktop-Interface vor. »What is needed is a metaphor and system for organizing the electronic 'bits of paper' we all so easily collect, whether we create them ourselves or they come to us in the form of email, downloaded images, web pages, or scheduling reminders. Lifestreams is such a system.

Merkmale, die den Ordner definieren, und sein Begriffsumfang versammelt alle Objekte, die diese Eigenschaften haben. Smart ist der Folder, wenn er mit der Zeit geht und seinen Umfang dem jeweiligen Wissens- oder Besitzstand anpasst. Smart Folders entsprechen den ›active retrievers‹, die Alan Kay schon 1990 angedacht hat und *bins* nannte: »How about the folder? One of my longstanding pet hates is to have them behave anything like their physical counterparts. [...] This drives me crazy, because the probability of *not* finding what you are looking for by browsing has just been maximized! It is trivial to have as many icon instances for a given doc or app in as many folders as one wishes. They should be near any place where they might be useful. [...] Instead of passive containers, why not have active retrievers that are constantly trying to capture icon instances that are relevant to them? Let's call them *bins*. [...] Folders kill browsing. Bins extend its useful range.« (Kay 1990, S. 200)

64 Ein primitiver Vorläufer eines solchen temporalen Kontextes ist die seit langem existierende Menu-Funktion ›Open Recent‹, die es erlaubt, schneller auf die in der letzten Zeit verwendeten Objekte zuzugreifen.

[...] Lifestreams uses a simple organizational metaphor, a time-ordered stream of documents, to replace conventional files and directories [...]. *Stream filters* and software agents are used to organize, locate, summarize and monitor incoming information.« Eine Verzeichnisstruktur gibt es nicht mehr, nur noch Ströme. Unterschiedliche Sichten, Suchfunktionen und ein Analogon zu Smart Folders werden in Form von Teilströmen realisiert. »Substreams, like virtual directories [...], present the user with a 'view' of a document collection, [...]. Substreams differ from conventional directory systems in that, rather than placing documents into fixed, file & folder directory structures, they create virtual organizations of documents from the stream.« (Fertig u.a. 1996, S. 410 f.)⁶⁵

Abb. 7: Lifestreams-Interface (ebda.)



Wenn man will, kann man die betrachteten Interface-Mechanismen auch als Agenten auffassen, insofern als sie eigenständig etwas »on behalf of the user« ausführen. Wie schon oben erwähnt, wurden solche Programme, die bescheiden im Hintergrund wirken, in den 90er Jahren zum zukunftsweisenden Paradigma des »autonomen Agenten« aufgewertet und vielfach

65 Weitere Beschreibungen findet man auf der etwas verstaubten Projektseite (Lifestreams 2000). Aufgegriffen wird das Strom-Konzept im »one laptop per child«-Projekt (OLPC, besser bekannt als »100\$-Computer«), dessen Interface in Form eines »Journals« der Nutzeraktivitäten organisiert ist. (OLPC 2007)

mit unbescheidenen Ansprüchen an ›intelligente Agenten‹ verbunden.⁶⁶ Eine relativ schlichte Definition eines Agenten wäre etwa: »an agent is any program that can be considered by the user to be acting as an assistant or helper, rather than as a tool in the manner of a conventional direct-manipulating interface. An agent should display some (but perhaps not all) of the characteristics that we associate with human intelligence: learning, inference, adaptability, independence, creativity, etc. The user can be said to delegate a task to an agent rather than command the agent to perform the task.« (Lieberman 1997)⁶⁷ Die bislang betrachteten, nun zu Agenten verklärten Funktionen sind noch nicht als ›proaktiv‹ anzusprechen, weil ihre Bedienung herkömmlich ist: Erst auf Anruf reagieren sie mit einer Aktion oder Auskunft. Zur Proaktivität wird meist zusätzlich gefordert: »agents should not simply act in response to their environment, they should be able to exhibit opportunistic, goal-directed behaviour and take the initiative where appropriate.« (Jennings/Wooldridge 1998, S. 4)⁶⁸ Die Ziele sollen die Ziele der Nutzer sein, und es ist dem Agenten aufgegeben, solche aus den jeweiligen Arbeitszusammenhängen ›herauszulesen‹. Das kann realistischerweise etwa heißen, dass in einer Mail, die man gerade

66 Alan Kay hatte schon 1984 gefragt: »What will agents be like in the next few years?«, und er bezog sich dabei auf die alten Vorstellungen von John McCarthy und Oliver G. Selfridge: »They had in view a system that, when given a goal could carry out the details of the appropriate computer operations and could ask for and receive advice, when it was stuck. An agent would be a ›soft robot‹ living and doing its business within the computer's world.« (Kay 1984, S. 58) Vergleiche hierzu auch den Beitrag von Hans Dieter Hellige, S. 68.

67 Der neue Interaktionsstil, der nicht Objekte manipuliert, sondern Aufgaben an Agenten delegiert, wird manchmal auch als ›indirekte Manipulation‹ bezeichnet. »The currently dominating interaction metaphor of *direct manipulation* requires the user to initiate all tasks explicitly and to monitor all events. [...] so-called ›autonomous agents‹ can be used to implement a complementary style of interaction, which has been referred to as *indirect manipulation*. Instead of user-initiated interaction via commands and/or direct manipulation, the user is engaged in a cooperative process in which human and computer agents both initiate communication, monitor events and perform tasks.« (Maes 1994, S. 31) Den Beweggrund kennen wir schon aus der Kritik an der Manipulation: »tasks that can and should be done while you are doing something else, and tasks that require considerable strategy and expertise.« (Kay 1990, S. 205) In gewisser Hinsicht liegt das Agentendasein jedoch im Auge des Nutzers, es kennzeichnet eher eine Erwartung als ein objektives Anderssein. So meint Henry Lieberman zu recht: »the same interface may properly be viewed as an agent by one person and as a tool by another. Nevertheless, people will tend to agree on whether a program is an agent often enough that such a distinction is useful.« (Lieberman 1997)

68 Ein etwas grobschlächtiger Verwandter ist die Autokorrektur bei der Textverarbeitung, die oft die Initiative ergreift, wenn sie nicht angebracht ist.

liest, eine URL, eine Adresse oder ein Termin erkannt wird und der Agent vorschlägt, sie in ein entsprechendes Verzeichnis einzutragen. Eben dies haben Nardi, Miller und Wright mit der Software »Apple Data Detectors« implementiert, die technisch gesehen ein ständig mitlaufender Parser ist, der in Dokumenten vordefinierte Textsorten erkennt und ›Vorschläge‹ für ihre Weiterverwendung macht, und deshalb von ihnen als zielbewusster Agent gehandelt wird.⁶⁹

Aus dem reaktiven Event-Loop des Desktop-Interfaces, das immer nur in Bereitschaft ist und auf Aktionen des Users wartet, ist ein proaktiver Interface-Agent geworden, der mitläuscht, potentiell nützliche Informationen aufbereitet und Ratschläge erteilt oder selbst tätig wird. Im ersten Fall sind nur die Eingaben des Nutzers auszuwertende Ereignisse, im zweiten auch die in den Daten vorhandenen Informationen, die eventuell einem anderen Zusammenhang zugänglich gemacht werden sollen. Dazu muss der Agent aus dem Kontext, nach formalen Kriterien, inhaltliche Verbindungen erkennen können.⁷⁰ »An interface agent could be considered to be a 'robot' whose sensors and effectors are the input and output capabilities of the interface, and for that reason are sometimes also referred to as ›softbots‹.« (Lieberman 1997) Sein ›Sensorium‹ ist nicht nur auf Kommandos von aussen ausgerichtet, sondern schaut auch nach ›innen‹ in die Welt der Daten und nicht zu knapp ins Web. Das Agenten-Interface vermittelt also nicht nur zwischen Computer und User, sondern auch zwischen der Nutzerin und Informationswelten, in denen auch andere Menschen und Agenten vorkommen können. Aus Interface-Agenten werden so Interspace-Agenten, wie Terry Winograd gemeint hat: »The traditional idea of ›interface‹ implies that we are focussing on two entities, the person and the machine, and on the space that lies between them. But beyond the interface, we operate in an ›interspace‹ that is inhabited by multiple people, workstations, servers, and other devices in a complex web of interactions.« (Winograd 1997, S. 153)

Die Interaktion wird damit anders gesehen. Der Computer erscheint nicht mehr in einen ›thought cycle‹ eingebettet, oder als ein Werkzeug, mit dem man Ikonen umräumt. Um dem ›Knowledge Worker‹, der in-

69 »Apple Data Detectors therefore has the ability to infer appropriate high-level goals from user actions and requests and take appropriate action to achieve these goals.« (Nardi u.a. 1998, S. 101) Interessanterweise wurde die Software erst neun Jahre später in Apples Mail-Programm standardmäßig integriert.

70 Abowd et al. sprechen hier von einem »informational context« und sehen in der »automatic integration of separate services« eine zentrale Funktion solcher Kontextsensitivität: »The context-awareness provides the ability to change the set of resident services and frees the user from having to remember any integration procedures.« (Abowd u.a. 1997)

zwischen zum Informationskonsumenten geworden ist, zu ermöglichen, »to work more effectively in the vast, rich, ever-changing world of electronic communication and information« (Terveen/Murray 1996, S. 355), ist eine Kooperation zwischen Mensch und Agent angesagt. »The metaphor used is that of a personal assistant who is collaborating with the user in the same work environment.« (Maes 1994, S. 31) Und es verwundert nicht, wenn auch wieder von Partnerschaft die Rede ist: »a fundamental rethink is needed about the nature of the interaction between computer and user. It must become an equal partnership – the machine should not just act as a dumb receptor of task descriptions, but should cooperate with the user to achieve their goal.« (Jennings/Wooldridge 1998, S. 7)

Die Interaktion mit dem Agenten soll jetzt auf dem »goal level« stattfinden, nicht mehr als Austausch von »ideas« in Form von Kommandos und Daten wie in Douglas Ross' Gestaltsprache.⁷¹ Das »goal-seeking behavior« ist ganz auf den Agenten übergegangen. Der Nutzer möchte nur Ziele formulieren oder noch besser, der Agent sollte sie ihm, etwa mittels »eye-tracking«, von den Augen ablesen können. Es wird also erwartet, dass der Agent auf dem »concept level« nicht Anweisungen sondern Zielvorgaben interpretiert.⁷² Da diese kontextabhängig sind und Intentionen immer noch nicht automatisch erkannt werden können, bedeutet das, dass Ziele und Leistungen der neuen Partner ausgehandelt werden müssen: »we generally need to be assured that the agent shares our agenda and can carry out the task the way we want it done. This generally requires a discourse with the agent, a two-way feedback, in which both parties make their intentions and abilities known, and mutually agree on something resembling a contract about what is to be done, and by whom.« (Foner 1993) Es fällt nicht schwer, sich dabei eine Wiederkehr der missliebigen

71 Das war schon Lickliders Traum: »instructions directed to computers specify courses; instructions directed to human beings specify goals. Men appear to think more naturally and easily in terms of goals than in terms of courses. [...] Computer instructions through specification of goals is being approached along two paths. The first involves problem-solving, hill-climbing, self-organizing programs. The second involves real-time concatenation of pre-programmed segments and closed subroutines which the human operator can designate and call into action simply by name.« (Licklider 1960, S. 11) Die beiden Wege zum Ziel lagen offensichtlich sehr weit auseinander. Während der erste von der KI weitergeträumt wird, genügt der realistische zweite nicht mehr den Ansprüchen an zielbewusste Agenten.

72 »[...] the value of the agent derives, at least in part, from its ability to formulate and execute a set of actions solely on the basis of a user's goals. Whether those goals are explicitly stated by the user or inferred by the system, the way an agent interprets and attempts to meet them constitutes *implicit responsiveness*. This is the principal means whereby an agent amplifies the user's personal power.« (Laurel 1990, S. 360)

Frage/Antwort-Sequenzen auf gehobenem Niveau vorzustellen, bei der man sich weniger gegängelt als genarrt fühlt.⁷³

Die Probleme der Kontextualisierung sind damit auf die höhere Ebene der Ziele verschoben worden. Die vom Agenten geforderte ›Context-Awareness‹ bezieht sich darauf, ›high-level goals‹ zu erkennen, er soll Sinnzusammenhänge der Nutzerin interpretieren. Für den Menschen stellt sich umgekehrt die Frage, was er in einem bestimmten Kontext von dem Agenten zu erwarten hat oder, nachträglich, was dieser eigentlich gemacht hat. Um dies zu unterstützen, hat Brenda Laurel die Verwendung von anthropomorphen Agenten empfohlen, weil deren ›Charakterzüge‹ Hinweise auf die von ihnen zu erwartenden Leistungen geben würden.⁷⁴ Im gleichen Sinne versucht man Erfahrungen aus dem Zeichentrickbereich für die Gestaltung von ›believable agents‹ nutzbar zu machen. (Bates 1994) Diesen Forschungen verdanken wir Vermittler wie den berüchtigten *Office-Assistenten* in vielerlei wählbaren Gestalten, vom Einstein, über ein Hundchen bis zur defaultigen Büroklammer. Während diese nur auf der gläsernen Bühne des Displays passend oder unpassend die Initiative ergreifen, wollen ›embodied agents‹ den Kontext der Verständigung in die Lebenswelt der Nutzer hineinbringen, wo das Interface zu einem alltäglichen Gegenstand wird. »A Programmable Embodied Agent [PEA] is a portable, wireless, interactive device embodying specific, differentiable characteristics for interaction«, typischerweise in Form von »toys and dolls that can be augmented with computational behaviors«. (Kaminsky u.a. 1999, S. 144)⁷⁵ Die vertrauten Objekte, in denen die Interaktionsleistungen vergegenständlicht sind, verkörpern »specific ›personality traits‹, afford direct physical interaction, and constitute a computational channel that is separable from the traditional desktop computer«. Während graphische Interfaces nach wie vor die reale Welt des Nutzers von der virtuellen Welt

73 Bei einer Suchmaschine, einer recht primitiven Form eines Agenten, hängt die Leistung wesentlich von ihrem Ranking ab, dessen Kontextualisierung man aber nicht versteht. Wenn die Antwort nicht bringt, was man erwartet hat, bleibt einem nichts anderes übrig, als die Zielvorgabe der Anfrage zu verfeinern oder abzuändern, bis man auf einen gemeinsamen Nenner mit dem ›Verständnis‹ der Maschine kommt.

74 »To flesh out a particular agent, the computer can be made to represent its unique skills, expertise, and predispositions in terms of character traits. [...] Evaluating action taken by an agent provides a feedback loop through which we refine and embellish our understanding of the agent's character.« (Laurel 1990, S. 358) Die Entwickler des ›Data Detectors‹ haben dagegen auf jeden Anthropomorphismus verzichtet, weil das nicht ihrem »goal of unobtrusiveness« entsprochen hätte. (Nardi u.a. 1998, S. 100)

75 Zum billigen Studium eines PEA haben Kaminsky et al. den von Microsoft vertriebenen Plüschdinosaurier *ActiMates Barney* ›gehackt‹ und seine kindlichen Funktionen für wissenschaftliche Zwecke erweitert.

der Berechnung trennen, würden PEAs einen natürlichen Weg darstellen, »to move back and forth across this boundary« (ebda., S. 151, 144) Beide Momente – Vergegenständlichung und Anthropomorphisierung – sollen der Kontextualisierung von Erwartungen und Beurteilungen dienen, »the character of the agent can provide a context for the activity«. So glaubt man, endlich die richtige Umwelt für die Interaktion gefunden zu haben: »by moving out into the world, computational interaction can take advantage of the specialized context in which activity takes place, rather than adopting the 'one size fits all' approach of graphical interfaces and UI widget sets. For us, then, Programmable Embodied Agents are sites of interaction that can be located 'where the (inter-)action is.'« (ebda., S. 144 f.)

In seinem Aufsatz im *Scientific American* von 1984 hatte Alan Kay das Agentenkonzept problematisiert.

»Agents are almost inescapably anthropomorphic, but they will not be human, nor will they be very competent for some time. They violate many of the principles defining a good user interface notably the idea of maintaining the user illusion. [...] Context is the key, of course. The user illusion is theater, the ultimate mirror. It is the audience (the user) that is intelligent and can be directed into a particular context. Giving the audience the appropriate cues is the essence of user-interface design. Windows, menus, spreadsheets and so on provide a context that allows the user's intelligence to keep choosing the appropriate next step. An agent-based system will have to do the same thing, but the creation of an interface with some semblance of human mentality will call for a considerably subtler approach.« (Kay 1984, S. 58 f.)

Von solchen subtileren Ansätzen ist bislang nicht viel zu merken; und heute deutet vieles darauf hin, dass ›intelligente Agenten‹ Fabelwesen bleiben werden, weil die fabelhaften kreativen und sozialen Eigenschaften, die man ihnen zuschreibt, inzwischen in den sozialen Welten des Web 2.0 durch die Kooperation vieler Individuen eingelöst werden. Das Agentendasein beschränkt sich dann auf selbsttätige Programme wie RSS-Feeds, die Aktuelles zugänglich machen, und Webseiten oder Tools, die die ›Wisdom of the Crowds‹ sondieren und verdaulich aufbereiten. Jedoch wird mit dem ›verkörperten Agenten‹ ein neues Phantasma der Verweltlichung angekündigt, auch wenn *Barneys* Abkömmlinge weniger kindisch aussehen, sondern in Form von ›augmented artifacts‹ die Welt anreichern sollen. Alan Kay hat sich bei seiner sechs Jahre später geschriebenen Apotheose: »Let us now extend everything we do to be part of a grand collaboration – with one's self, one's tools, other humans, and increasingly, with agents: computer processes that act as guide, as coach, and as amanuensis.« (Kay 1990, S. 204), vermutlich nicht träumen lassen,

dass diese Kollaboration in der Umwelt eines ›Internets der Dinge‹ stattfinden wird. Deren Sensorium wird nicht mehr nur auf den ›informatiellen Kontext‹ gerichtet sein, in dem sich Maschine und Mensch was zu sagen haben, sondern erfasst auch den raumzeitlichen Kontext, in dem er oder sie lebt und lebt.

5 Zu den Sachen!

Die bisher betrachteten Ansätze von ›Context-Awareness‹ beschränkten sich im Wesentlichen auf das Desktop-Interface, was nicht ausschließt, dass man es in Form eines Laptops mit sich herumträgt. Mit neuen Schlagwörtern geht die Entwicklung nun dahin, den Computer in die weite wirkliche Welt hinauszuschicken, sei es, dass er als Accessoire seiner mobilen Nutzerin mitgeführt oder in deren wechselnde Umwelten eingebettet wird. Das Einbetten von Rechenleistung in Objekte ist natürlich schon alt und betrifft tatsächlich die überwiegende Mehrzahl aller Prozessoren. Aber jetzt geht es um verweltlichte Rechner, die ein Interface zum Menschen haben, und um Menschen, die in eine computerisierte (inter-)aktive Umwelt eingebettet sind. Sie sollen überall die Segnungen der Informationstechnologie genießen können, und dazu muss die informationelle Dienstleistung vom Kontext ihres Einsatzes abhängig gemacht werden. »Whether that service mobility is achieved by equipping the user with computational power or by instrumenting the environment, all services need to be extended to take advantage of the constantly changing context in which they are processed.« (Abowd u.a. 1997) Der Kontext des Computers ist jetzt der lebensweltliche Kontext der von ihm bedienten Menschen, – Raum und Zeit, in der sie sich befinden oder bewegen, und nicht zuletzt ihre situative Befindlichkeit selbst. Vom umweltlichen Computer wird erwartet, dass er zu folgendem Verhalten in der Lage ist:

- »1. Collect information on the user's physical, informational or emotional state.
2. Analyze the information, either by treating it as an independent variable or by combining it with other information collected in the past or present.
3. Perform some action based on the analysis.
4. Repeat from Step 1, with some adaption based on previous iterations.« (ebda.)

Der Event-Loop des Desktop-Interfaces ist zur Allzeitbereitschaft einer ›augmented reality‹ aufgerüstet worden. Die Interaktionszeit wird zur Eigenzeit der andauernden Performanz der Maschinen, die ihre Umge-

bung beobachten und auswerten. »Where GUI-based applications focus on explicit input, context-aware applications try to take a more general notion of input, focusing on implicit input but dealing with explicit input as well. Context-aware applications need to take advantage of the sensors and sensing techniques available.« (Dey u.a. 2001, S. 15)

Die Verweltlichung ist mit einer Abkehr vom ›multi-purpose‹ Schreibtischcomputer verbunden, hin zu einer Vielfalt von Geräten und Dienstleistungen, wie in der Vorstellungswelt des ›Ubiquitous Computing‹, »where specialized devices outnumber users«. (Weiser 1991)⁷⁶ Interaktion erfordert keinen Vorsatz der potentiellen Nutzer mehr, sondern die umweltlich vorgesetzten Interfaces bieten sich an. »Rather than force the user to search out and find the computer's interface, our new aim is to provide an interface that can take on the responsibility of locating and serving the user.« (Abowd u.a. 1997) Damit die Initiative nicht aufdringlich wird, muss die Technologie »permeate everyday life without becoming too much of a physical, cognitive or social burden.« Es darf nicht zu kognitiven Dissonanzen kommen, wenn der erwählte Nutzer von der Verantwortung der Vorsorge entlastet wird. Um die Maschine aber nicht mit einer zu komplizierten ›Context-Awareness‹ zu belasten, erscheint es einfacher, Kontexte zu schaffen, als sie zu erkennen. Abowd et al. übernehmen denn auch den Alan Kay zugeschriebenen Ausspruch als Gestaltungsprinzip: »Do not spend too much time predicting the future; aim to invent it.« (ebda.) Das heisst konsequenterweise wohl auch, einen dazu passenden Nutzer zu kreieren.

Anind Dey, Gregory Abowd und Daniel Balser haben es unternommen, in die Proliferation der Kontexte etwas Ordnung zu bringen. Sie definieren: »*Context*: any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups and computational and physical objects.« (Dey u.a. 2001, S. 11 ff.) Sie kategorisieren Kontextinformationen nach »identity, location, status (or activity) and time«. *Identity* bezieht sich auf die eindeutige Identifizierbarkeit aller Objekte und Subjekte, deren rasante Vermehrung ja ein Motiv für die Erweiterung des Adressraums im Internetprotokoll IPv6 ist. *Location* beinhaltet nicht nur Positionen, sondern auch relationale Lageinformationen von mehreren Entitäten. *Status* (or *activity*) erfasst Umwelteigenschaften wie Temperatur oder Lichtverhältnisse, sowie den physiologischen Zustand

76 Das zahlenmäßige Verhältnis von Mensch zu Computer war im Batch- und Timesharing-Betrieb *many-to-one*, beim Desktop-Computer *one-to-one*, jetzt soll es *one-to-many* werden.

der Menschen und, womit sie beschäftigt sind. *Time* ermöglicht es, die aktuelle Situation auf eine Vorgeschichte zu beziehen. Schließlich unterscheiden die Autoren drei Arten von Funktionen oder Dienstleistungen, die ›context-aware‹ Applikationen zu bieten haben: »*presenting information and services*«, die aus situationsbezogenen (Zusatz-)Informationen oder Vorschlägen bestehen können; »*automatically executing a service*« meint weitergehend, dass die Artefakte auf Kontextveränderungen auch selbständig reagieren können; und »*attaching context information for later retrieval*« bedeutet, dass alles aufgezeichnet und gespeichert wird.⁷⁷

Diese Taxonomie ordnet die Kontextualität der Artefakte. Im Hinblick auf die (mobile) Nutzerin lassen sich zwei Umgebungsaspekte unterscheiden: Sie kann die informationelle Rüstung mit sich herumtragen, oder sie findet sie jeweils vor Ort vor. Unter den Auspizien ihrer Mobilität ist die nächstliegende Technologie der Kontexterfassung das ›location-aware computing‹, das im Zeitalter von GPS-Systemen technisch keine großen Probleme aufwirft. Das »Active Badge Location System« am Olivetti Research Ltd. (ORL) Institut in Cambridge war wohl das erste größere Projekt dieser Art und im Prinzip nur eine Weiterentwicklung der z.B. in Krankenhäusern üblichen Pager-Systeme. Während man bei diesen aber nicht mitbekommt, wo jemand sich aufhält, wurde am ORL der alternative Ansatz gewählt, »to 'tag' a person and try to locate the tag«. (Want u.a. 1992, S. 92) Das System war nicht so vielseitig wie die Communicator Badges in Star Trek, aber man konnte feststellen, wo sich eine Person (die ihren Badge trägt) befindet und wo sie sich in der letzten Stunde überall aufgehalten hat, mit wem sie zusammen ist und wer in der Nähe eines bestimmten Ortes anzutreffen ist. Das hat natürlich Fragen der Privacy aufgeworfen: »Despite the success of the Cambridge installation, the majority of people who first learn about the system are not convinced it is something they would like in their own work situation. The privacy issues are clearly very important.« Aber schließlich sei ein ›Opt-out‹ nicht verboten, jeder könne sein Pickerl auf dem Schreibtisch liegen lassen und so das System täuschen. »In practice, at ORL this was not known to be a common use of the system.« (ebda., S. 100) Zur Akzeptanzverbesserung wäre es auch möglich, die Verortung als Privilegiansystem zu gestalten, wo

77 Das bietet sich natürlich zur Überwachung an. Wenn Dey et al. meinen: »a component that acquires context information, should maintain a history of all the context it obtains. Context history can be used to establish trends and predict future context values.«, dann entspricht das den Präventionsvorstellungen von Politikern, die aus der Beobachtung von ungewöhnlichem Verhalten kriminelle oder terroristische Aktivitäten vorhersagen und verhindern wollen.

jeder festlegen kann, wer ihn lokalisieren darf, und mitgeteilt bekommt, wer das wie oft versucht hat.⁷⁸

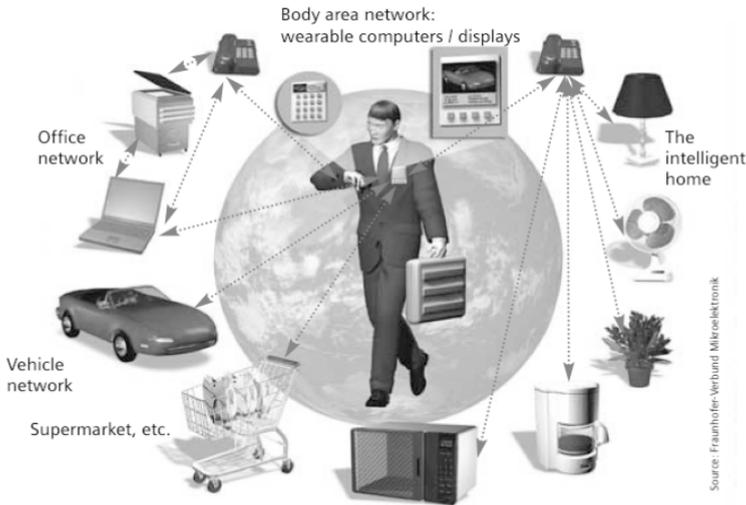
In einem Ausblick werden das Active Badge Konzept dann mit anderen Überwachungs- und Managementfunktionen der Haustechnik integriert und die Menschen problemlos in der automatisiert verwalteten Umwelt positioniert. »The Active Badge extends the concept of an integrated building to take into account the location of personnel in that environment.« (ebda., S. 102) Am Xerox PARC ist man einen Schritt weiter gegangen und hat mit den ParcTabs die Ortungsfunktionen dahingehend erweitert, dass auch »context-triggered actions« automatisch ausgeführt werden konnten. Hierbei handelt es sich um Regeln, die spezifizieren was wo wann ausgeführt werden soll, beispielsweise könne man an alle eine Nachricht verschicken, wenn jemand frischen Kaffee braut. Diese Rundumversorgung sei »something like living in a rule-based expert system!« (Schilit u.a. 1994, S. 5)

Das eigentliche Problem der Entwickler von »location-aware« Techniken ist jedoch, sinnvolle Anwendungen zu finden, die sich massenhaft an kaufkräftige Nutzer bringen lassen. So träumen Abowd et al.: »One way to thwart this lack of market success is to invent new ways to use a PDA.« Typische Prototypen sind Services für Touristen, die versuchen, »to predict what the user is attending to and provide information on that entity. [...] Getting close to something is an indication that information on that object is to be requested.« (Abowd u.a. 1997) Die besungene Aufhebung von Raum und Zeit im Cyberspace wird nun wieder aufgehoben, indem physische Nähe zum Indikator für Interesse und damit für sinnvolle Informationsbegrenzung wird. Der lokalisierte Kontext kann dann durch weitere Zusammenhänge wie Tages- und Jahreszeit oder aus der bisherigen Besichtigungsgeschichte erschlossene Präferenzen ergänzt werden. Interessante Objekte können auch Subjekte sein, und so experimentiert man mit Devices, die die physische Nähe von gleichgesinnten, oder zumindest

78 Es folgt das übliche Wertfreiheitsargument, etwas Bedenken und der Ruf nach dem Gesetzgeber: »The problem is that technology in itself is rarely inherently bad; it is just that it can be used for good or bad purposes. [...] If it turns out that locaton systems in our society at large are abused, then the rights of individuals must be protected. In this case legislation must be drawn up to ensure a location system cannot be misused, while still allowing us to enjoy the benefits it brings.« (Want u.a. 1992, S. 100 f.) Diese Einstellung erscheint symptomatisch für viele Forschungen im IT-Bereich. Es wird ein »technology-driven research project« gestartet, das im vertrauten Kreis der Kollegen wunderbar funktioniert; was in der Realität daraus werden könnte, wird als potentieller Missbrauch gesehen, wofür der Gesetzgeber zuständig ist. Wenn man bedenkt, wie schlecht die Datenschutzbestimmungen in den USA sind, ist das eine groteske Forderung.

gleichgerüsteten, Mitmenschen registrieren und anregen, einen (virtuellen) Sozialkontakt herzustellen.⁷⁹

Abb. 8: »Online at any time.« (Siemens 2002)



Online at anytime. In the future, portable and wearable communication devices, computers and displays will ensure convenient mobile connections to our homes, offices and vehicles.

Die Versprechungen des »mobile« oder »wearable computing« nehmen Vorstellungen von Engelbarts Augmentation wieder auf, aber ihr Träger ist nicht mehr als »Knowledge Worker« in einem »Knowledge Workshop« gedacht, sondern es geht um eine »Augmented Reality« in Form von eingespielten Zusatzinformationen bei mobilen Arbeitseinsätzen oder in der beschaulichen Welt eines Flaneurs. Ein Beispiel für einen »gehobenen« mobilen Arbeiter wäre ein Techniker, der hoch oben auf einem Kran etwas repariert, beide Hände frei haben muss und sich die Reparaturanleitung in einem Headmounted Display einspeisen lässt. Die Interaktion sollte sich von seiten des Nutzers auf das Notwendigste beschränken lassen, weil er Anderes zu tun hat; und idealerweise sollte das tragbare Device erkennen, wo er sich befindet, welche Kommunikationsverbindungen bestehen und welche Informationen im Rahmen seiner Tätigkeit relevant

79 Daran versucht Microsoft mit seinem *Zune*-Player anzuknüpfen, mit dem man Musikstücke drahtlos über geringe Distanzen austauschen kann, auch wenn diese aus DRM-Gründen nach dem Motto »aus den Augen, aus dem Sinn« bald wieder unbrauchbar werden.

sind.⁸⁰ Der örtliche Kontext ist hier zugleich ein informationeller, und die Aufbereitung der präsentierten Informationen muss sich dem zeitlichen Arbeitsablauf anpassen können. Da der nicht sequentiell durch die Verarbeitung der Maschine oder einen Dialog vorgegeben ist, sondern durch einen außerhalb der Interaktion liegenden ›Problemlösevorgang‹ bestimmt wird, muss das System versuchen, den Kontext aus dem Aufmerksamkeitsfokus ihres Users, etwa mittels Eye-Tracking, zu lokalisieren. Massenhaft verkaufen ließe sich jedoch die Bewehrung des mobilen Individuums, dem Beruf und Privatleben im Umherschweifen verschimmt. Mit der Devise »Online at any time« vermittelt das ›Mobile Computing‹ nicht nur die allgegenwärtige Verfügbarkeit der Technologie, sondern zugleich die allörtliche und allzeitige Verfügbarkeit ihrer Nutzer.

Einen komplementären Ansatz propagiert das Phantasma des ›Ubiquitous Computing‹, das unter vielerlei Namen wie ›pervasive computing‹, ›silent computing‹, ›ambient intelligence‹ oder ›Internet of things‹ auftritt.⁸¹ Statt dass der mobile Nutzer die Computerpower mit sich herumträgt, wird sie in die Umwelten eingebaut, so dass er sie überall vorfindet. Der Computer, der natürlich eine Vielzahl von vernetzten Geräten ist und noch mehr Services anbietet, soll dabei möglichst unsichtbar bleiben, so zuhänden wie ein Bleistift, den man nicht bewußt wahrnimmt, wenn man ihn nicht benötigt. Mark Weiser, der Prophet des Ubiquitous Computing, hat die Schlagworte ›building invisible interfaces‹ und ›silent computing‹ bzw. ›calm technology‹ popularisiert. Wie Engelbart will er ›aims for true human effectiveness‹ schaffen, aber diese sollen ›no *thing* in the office humming on the desk‹ sein. Die Ertüchtigung werde durch ›augmented artifacts‹ mit einem ›ease of use so effective you don't notice the computer‹ zu verwirklichen sein. ›A good tool is an invisible tool. By invisible, I mean that the tool does not intrude on your consciousness; you focus on the task, not the tool.‹ (Weiser 1994b; 1994a, S. 7) Wieder mal soll der Computer unsichtbar werden. Seine Interfaces werden nicht mehr den Zauber einer User-Illusion vermitteln, sondern im Sinne von Heideggers Hammerzuhänden verschwinden. ›Just as a good, well-balanced hammer 'disappears' in the hands of a carpenter and allows him or her to concentrate on the big picture, we hope that computers can participate in a similar magic disappearing act.‹ (Weiser u.a. 1999, S. 695)⁸²

80 Hierzu findet man Genaueres in Ingrid Rügges Beitrag in diesem Band und in (Rügge 2007).

81 Differenzierter werden die verschiedenen Facetten der Schlagwörter in Michael Friedewalds Beitrag in diesem Band behandelt.

82 Es ist erstaunlich, wie oft der Computer in seiner kurzen Geschichte schon verschwinden sollte. Es ist fast, als ob man sich seiner schämt; vielleicht handelt es sich auch um eine Abwehrreaktion der ›Prometheischen Scham‹. (Anders 1956) Shaw wollte ihn hinter einem Agenten verstecken. Donald

Ursprüngliches Ziel der Gruppe um Mark Weiser am Xerox PARC war es, »to put computing back in its place, to reposition it into the environmental background, to concentrate on *human-to-human* interfaces and less on *human-to-computer* ones. By 1992, when our first experimental ›ubicompc‹ system was being implemented, we came to realize that we were, in fact, actually redefining the *entire* relationship of humans, work, and technology for the post-PC era.« (ebda., S. 693 ff.) Die schöne neue Welt der künftigen »UC-era« ergibt sich für ihn einfach daraus, dass »embedded microprocessors« über das Internet vernetzt werden, was heute unter dem Slogan ›das Internet der Dinge‹ propagiert wird. Seine Beispiele sind eine Mischung aus Banalem und Phantasmen eines Smart Houses: »Clocks that find out the correct time after a power failure, microwave ovens that download new recipes, kids toys that are ever refreshed with new software and vocabularies, paint that cleans off dust and notifies you of intruders, walls that selectively dampen sounds, are just a few possibilities.« (Weiser/Brown 1996, S. 2)

Im Laufe der Zeit hat sich für Weiser als Leitidee das (auf Donald Normans ›affordance‹ Bezug nehmende) Konzept des »*calm computing*« herauskristallisiert, was eher »the desired state of mind of the user« als eine Hardware-Ausstattung meint. Statt sich dem »excitement of interaction« mit einem PC auszusetzen, möchte dieser Nutzer nur nebenbei etwas ›rechnen‹, »while doing something else and have more time to be more fully human«. Deshalb dürfen sich die rechnenden Artefakte nicht aufdrängen: »If computers are everywhere they better stay out of the way, and that means designing them so that the people being shared by the computers remain serene and in control.« (ebda., S. 3)

Während die meisten Informations- und Kommunikationstechnologien sehr aufdringlich sind – von singenden Handies, über piepsende Registrierkassen bis zu dröhnenden Ghettoblastern, würde ›Calm Technology‹ sich nur anbieten und nicht fordern. Sie steht aber keineswegs still und schweiget, sondern paradoxerweise könne Informationsüberlastung durch mehr Information vermieden werden. Der Trick bestehe darin, die ›Ränder unserer Wahrnehmung‹ (›periphery‹) auszunutzen, indem die ›verdrängten‹ Interfaces nur eine vorbewusste Umwelterfassung anspre-

Norman, der später ein ganzes Buch »The Invisible Computer« geschrieben hat, projizierte »what the computer of the future might look like. Suppose I told you it wouldn't even be visible, that you wouldn't even know you were using one? [...] You don't notice the computer because you think of yourself as doing the task, not as using the computer.« Und dann wollte er auch das Interface beseitigen: »The real problem with the interface is that it is an interface. Interfaces get in the way. I don't want to focus my energies on an interface. I want to focus on the job.« (Norman 1989, S. 185; 1990, S. 210; 1998)

chen, mit der wir etwas im doppelten Sinne übersehen und nur bei Bedarf unsere Aufmerksamkeit darauf richten.⁸³ Als Beispiel führt Weiser die Multicast-Technologie an, wo über Breitbandverbindungen alle Welt gleichzeitig ins Zimmer übertragen wird. Während ich mir darunter nur einen gigantischen, immer laufenden Fernseher vorstelle, sieht Weiser darin ein »window of awareness«. ⁸⁴ »The periphery connects us effortlessly to a myriad of familiar details. This connection to the world we called 'locatedness', and it is the fundamental gift that the periphery gives us.« (ebda., S. 6, 5)

Abb. 9: Ambient Orb. (Ambient Orb 2007)



»A glass lamp that uses color to show weather forecasts, trends in the market, or the traffic on your homeward commute.« »Some information requires constant awareness.« (ebda.)

83 »Calm technology engages both the center and the periphery of our attention, and in fact moves back and forth between the two. [...] We use ›periphery‹ to name what we are attuned to without attending to explicitly. [...] by placing things in the periphery we are able to attune to many more things than we could if everything had to be at the center. Things in the periphery are attuned to by the large portion of our brains devoted to peripheral (sensory) processing. Thus the periphery is informing without overburdening.« (Weiser/Brown 1996, S. 4)

84 »A continuous video stream brings new details into the periphery: the room is cleaned up, something important may be about to happen; everyone got in late today on the east coast, must be a big snowstorm or traffic tie-up.« (Weiser/Brown 1996, S. 6)

Die vielen ›unsichtbaren‹ Interfaces appellieren also an unsere Fähigkeit, alltäglich gewordene Umgebungen auszublenden. Das Interface ist nur zur Stelle, wenn wir seiner bedürfen, ansonsten träumt es vor sich hin. Der Kontext einer Interaktion ergibt sich aus dem lebensweltlichen Zusammenhang des Objektes, was natürlich eine vertraute Anmutung statt einer Zumutung erfordert. Interaktion gewinnt so etwas Beiläufiges. Probleme gibt es anscheinend nicht mehr, und das ›goal-seeking behavior‹ des Problemlösens scheint gänzlich aus der Mode gekommen zu sein. Bezeichnenderweise hat das randständige Mittel der Effizienzsteigerung wenig Eingabefunktionen, aber »many, many ‘displays’«. (Weiser 1994b) Das »excitement of interaction« soll einer Rundumversorgung weichen. Die vielen Interfaces sind nicht mehr aktiv zu nutzende Fenster zur Welt, sondern die Welt ist alles, was der ›intelligenten Umgebung‹ einfällt. Statt uns mit einer Nabelschnur mit der Zentrale des Wissens verbinden zu müssen, brauchen wir den elektronischen Mutterleib garnicht mehr zu verlassen; man ist überall zuhause, wo man informationstechnisch umfassen ist. »The result of calm technology is to put us at home, in a familiar place.« (Weiser/Brown 1996, S. 4)⁸⁵ Wenn man sich in jeder angereicherten Umwelt gleich heimisch fühlen kann, erscheint es nicht zuviel behauptet, dass der u-topierte Mensch der UC-Ära dekontextualisiert wird, insofern als es keine wahrnehmbaren Grenzen zum ausgeschlossenen Dritten mehr gibt, – man könnte auch von einer MacDonaldisierung der proaktiven Realität sprechen.

Die heimische Welt ist jedoch eine heimliche Welt, das Gegenteil der passiven Bereitschaft der Manipulationswerkstatt, denn jetzt soll alles hinter dem Rücken der Bedienten geschehen. Das Phantasma der allgegenwärtigen ›augmented reality‹, in der Vergangenheit und Zukunft des umsorgten Menschen von der ›ambienten Intelligenz‹ registriert, gespeichert und vorgesehen wird, ist natürlich im Kern ein (verteiltes) Automationskonzept in Form eines ›intergalactic network of augmented

85 Kritik an dieser ›Nesthockermentalität‹ kommt auch aus dem UbiComp-Bereich selbst. Nachdem sie einen »enormous gap between the dream of comfortable, informed and effortless living and the accomplishments of UbiComp research« konstatiert hat, stellt Yvonne Rogers Weisers Vision prinzipiell in Frage: »The ‘excitement of interaction’ that Weiser suggested forsaking in the pursuit of a vision of calm living should be embraced again, [...]« »To this end, I propose one such alternative agenda which focuses on designing UbiComp technologies for engaging user experiences. It argues for a significant shift from *proactive computing* to *proactive people*; where UbiComp technologies are designed not to do things for people but to engage them more actively in what they currently do. Rather than calm living it promotes engaged living, [...]« (Rogers 2006, S. 405, 418, 406)

artifacts.«⁸⁶ Da alle diesen kleinen, vernetzten Automaten Daten sammeln und austauschen müssen und der umsorgte Mensch die wichtigsten Kontextinformationen liefert, ist UbiComp schon vom Prinzip her eine Kontroll- und Überwachungstechnologie.⁸⁷ Einmal mehr zeigt sich die Naivität der gutmeinenden Entwickler in Mark Weisers Retrospektive. Nachdem er beschrieben hat, wie sie am Xerox PARC durch gemeinschaftliche Wandbildschirme und Active Badges »the vast potential of such a system for augmenting and improving work practices and knowledge sharing« geahnt haben, fährt er fort: »But simultaneously we came across an *unexpected problem*, often blared in newspaper headlines as: »Big Brother Comes to the Office.« The problem, while often couched in terms of privacy, is really one of control. If the computational system is invisible as well as extensive, it becomes hard to know what is controlling what, [...] and what are the consequences of any given action (including simply walking into a room).« (Weiser u.a. 1999, S. 696 f.; meine Hervorhebung) Wenn Weiser meinte, die Folge einer Calm Technology wäre, »to put us at home«, dann ist das wohl umgekehrt zu verstehen, dass das vertraute Forschungslabor die Voraussetzung ist, die die Technologie emanzipatorisch erscheinen lässt. In der großen weiten Welt steht zu befürchten, dass die Hintergründigkeit der verteilten Automaten schlicht bedeutet, dass »the people shared by the computers« unauffällig kontrolliert werden und nicht registrieren, wie und was diese registrieren.⁸⁸

86 Wie bei allen Automatisierungsphantasmen soll der Mensch auch hier zu seinem »Eigentlichen« freigesetzt werden: »to be more fully human«.

87 Wenn der Nutzer die Artefakte nur vorbewusst wahrnimmt, liegt es nahe, umgekehrt auch seine unbewussten Äußerungen als Kontextinformationen auszuwerten. Dies unternimmt das »affektive computing«. So schreibt dessen Protagonistin am MIT Media Lab Rosalind Picard: »affective computers should not only provide better performance in assisting humans, but also might enhance computers' abilities to make decisions.« Beispielsweise könnten »affective wearables« gesundheitliche Ratschläge geben: »A jacket that senses your posture might gently remind you to correct a bad habit after back surgery.« oder einfach nur Besinnliches murmeln: »I sense more joy in you tonight.« »Affective wearables would be communication boosters, clarifying feelings, amplifying them when appropriate, and leading to imaginative new interactions and games.« (Picard 1995, S. 1, 14) Das »amplifying his mental sources« des konversierenden Computers soll jetzt also auch die Gefühle des Nutzers umfassen.

88 Sarah Spiekermann hat die Projekte des Ubiquitous Computing unter dem Blickwinkel von traditionellen Automationstheorien betrachtet und drei Jahrgänge des IEEE Pervasive Computing Magazine (2003 bis 2005) ausgewertet. Von den 30 untersuchten Systemen sind nur 10% als proaktiv in dem Sinne, »that the computerized environment takes a decision for the human and presents him with the results«, zu bezeichnen. 90% haben dagegen vorwiegend mit der automatischen Auswertung von Eingabedaten zu tun, mit »acquisition of sensory information and the aggregation and inter-

Der janusköpfige Charakter dieser Technologie wird in dem am 27. Dezember 2007 veröffentlichten Patentantrag von Microsoft für ein System, das eine »automation of various aspects of project management« erlaube, deutlich. (Microsoft 2007) Es handelt sich schlicht um die Beschreibung eines umfassenden Kontrollsystems am Arbeitsplatz, das als Unterstützungssystem gerechtfertigt wird: »A unique monitoring system and method is provided that involves monitoring user activity in order to facilitate managing and optimizing the utilization of various system resources. In particular, the system can monitor user activity, detect when users need assistance with their specific activities, and identify at least one other user that can assist them.« Insbesondere könne das System auch »automatically detect frustration or stress in the user via physiological and environmental sensors and then offer or provide some type of assistance accordingly«. ⁸⁹ Automatisch erstellte Nutzerprofile und Umgebungsinformationen »supply context in a standardized format«; und »statistics related to performance, success rate, frequency of problem, and the like can be provided to users or can be employed to gauge a target user's success, performance, or efficiency with respect to other users«. Die vergleichende Statistik reguliert auch das Soziale: »aggregating user activity data« diene dazu, »to bind such users together based on their activity, thereby enhancing social relationships and user experiences«. Das Band ist natürlich das gemeine Ziel, »to maximize the efficiencies of system and human resources, improve user performance, and minimize user inefficiencies«. Man kann sich denken, was mit jemand geschieht, der sich nicht hinreichend profiliert. Was hier nur für Produktionszusammenhänge gedacht ist, verspricht im Alltag der umfangenden Consumer-Relationship-Welt: »Ubiquitous Commerce – Always On, Always Aware, Always Pro-active«. ⁹⁰

pretation of this information«, und 53% tun nichts anderes. Spiekermann bilanziert deshalb: »Ubiquitous Computing applications as they present themselves today are mostly focusing on input automation or the automation of information collection and processing. [...] Pro-active output automation or execution of activities which is said to be the main vision of Ubiquitous Computing is still in an infancy stage.« (Spiekermann 2007) In 2/3 aller Fälle ist das Objekt der Beobachtung der Mensch selbst, und insgesamt mehr als 60% aller Systeme »are centered on the perfection of human behavior«. Tatsächlich betrifft ein großer Teil davon den menschlichen Körper und dessen Optimierung. Damit gewinnt das Konzept des »programmable embodied agents« eine neue Dimension.

89 Das Spektrum der Sensoren ist breit: »at least one of heart rate, galvanic skin response, EMG, brain signals, respiration rate, body temperature, movement, facial movements, facial expressions, and blood pressure«.

90 Anatole Gershman beschließt sein merkantiles Manifest mit den Worten: »To provide the high level of services that will be required to compete in ubiquitous commerce, companies will have to follow the three 'always' commandments: • To be always on and connected to their customers • To

Wir finden, durch klare Profitinteressen zur Kenntlichkeit entstellt, die fast die ganze Technikgeschichte durchziehende Dialektik von Fürsorge und Selbstbestimmtheit, von Entlastung und Verantwortung, von Kontrolle und Freiheit, die jedoch mit den Phantasmen der ›intelligenten Umwelten‹ auf die Spitze getrieben und einseitig ›aufgehoben‹ wird. Das vorsorglich observierte Subjekt wird aus der Verantwortung der Sorge um sich selbst entlassen und der Entscheidung über seine Bedürftigkeit enthoben. Die ›ganzheitliche‹ Erfassung des Nutzerkontextes, in Form der im Nutzerprofil geronnenen Geschichte wie in gemessenen Umgebungsvariablen, produziert automatisch ›standardisierte Kontexte‹, die sich der Mensch nicht erschließt, sondern die ihm vorgesetzt werden. Die ›Context-Awareness‹ der Maschine führt folglich zu einer Normierung von Zielen und einer formierenden Vorgabe des Verhaltens der so gefassten Nutzer. Es ist deshalb nur konsequent, wenn das ›pervasive computing‹ zum ›persuasive computing‹ weitergedacht wird, wo der Computer als »social actor« auftritt und die erfassten Nichtsnutzer zu einer Verhaltensänderung animiert.⁹¹ Wie einst der Computer geistige Arbeit durch externalisierte Symbolmanipulation erweitert und transformiert hat, sollen die kontrollierenden Umwelten moralische Arbeit – ›Techniken des Selbst‹ – durch züchtige Manipulationen unterstützen. Dergestalt würde der Computer wahrlich eine Kulturtechnik werden, und Freuds Diktum wäre konsequent zu Ende gedacht: Wo Ich war, soll Über-Ich werden.

be always aware of their customers real-time context (where the customers are, what they are doing, what is around them) • To be always pro-active, taking advantage of the real-time opportunities to satisfy customer needs.« (Gershham 2007)

91 B. J. Fogg, der erste Prophet des ›persuasive computing‹, verkauft das unter dem sprechenden Namen »captology«, was vermutlich nichts mit Gefangennahme oder Besitzergreifung zu tun haben soll, sondern ein Akronym für »Computers As Persuasive Technology« ist. (Stanford Persuasive Technology Lab 2007)

Literatur

- Abowd, G. D./Dey, A./Orr, R./Brotherton, J. (1997): »Context-awareness in wearable and ubiquitous computing«; <http://smartech.gatech.edu/bitstream/1853/3531/1/97-11.pdf>.
- Ambient Orb (2007): <http://www.ambientdevices.com/cat/orb/orborder.html>
- Anders, G. (1956): »Die Antiquiertheit des Menschen«, München.
- Apple Computer (1992): »Macintosh Human Interface Guidelines«. Reading, Mass.
- Baker, C.L. (1966): »JOSS: Introduction to a Helpful Assistant«, RAND Memorandum RM-5058-PR, July 1966; elektron.: http://www.rand.org/pubs/research_memoranda/2006/RM5058.pdf.
- Bardini, T./Friedewald, M. (2002): »Chronicle of the Death of a Laboratory: Douglas Engelbart and the Failure of the Knowledge Workshop«. In: History of Technology 23, S. 191-212; elektron.: <http://www.friedewald-family.de/Publikationen/hot2002.pdf>.
- Bates, J. (1994): »The Role of Emotion in Believable Agents«. In: Communications of the ACM 37, 7, S. 122-125.
- Blumenberg, H. (2006): »Beschreibung des Menschen«, Frankfurt am Main.
- Bootstrap Institute (2007): <http://www.bootstrap.org/>.
- Brennan, S. E. (1990): »Conversation as Direct Manipulation: An Iconoclastic View«. In: Laurel, B. (Hrsg.): »The Art of Human-Computer Interface Design«, Reading, MA u.a., S. 393-404.
- Bush, V. (1945): »As We May Think«, In: The Atlantic Monthly 176/1, S. 101-108.
- Chapin, N. (1957/62): »Einführung in die elektronische Datenverarbeitung«, Wien München 1962; Orig.: »An Introduction to Automatic Computers«, New York 1957.
- Davis, M. R./Ellis, T. O. (1964): »The Rand Tablet: A Man-Machine Graphical Communication Device«. In: AFIPS Conference Proceedings (FJCC 1964), S. 325-331.
- Dey, A. K./Abowd, G. D./Salber, D. (2001): »A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications«. In: Human-Computer Interaction (HCI) Journal, Vol. 16 (2-4), S. 97 - 166; elektron.: <http://www.cc.gatech.edu/fce/ctk/pubs/HCIJ16.pdf>.
- Dourish, P. (2004): »What We Talk About When We Talk About Context«. In: Personal and Ubiquitous Computing 8, 1, S. 19-30.
- Ellis, T. O./Heafner, J. F./Sibley, W. L. (1969): »The Grail Project: An Experiment in Man-Machine Communications«, RAND Memorandum RM-5999-ARPA, Sept. 1969; http://www.rand.org/pubs/research_memoranda/2005/RM5999.pdf.

- Engelbart, D. C. (1961): »Program On Human Effectiveness«, SRI Menlo Parc, December 1961 (Typoscript); <http://sloan.stanford.edu/mousesite/Archive/Post68/PrHumanEffectiveness.html>.
- Engelbart, D. C. (1962): »AUGMENTING HUMAN INTELLECT: A Conceptual Framework«, SRI, Menlo Park, October 1962, elektronisch: <http://www.bootstrap.org/augdocs/friedewald030402/augmentinghuman-intellect/AHI62.pdf>.
- Engelbart, D. C. (1968): »The Demo«; <http://video.google.com/videoplay?docid=-8734787622017763097>.
- Engelbart, D.C. (1988): »The Augmented Knowledge Workshop«. In: Adele Goldberg (Hrsg.): »A History of Personal Workstations«, Reading, Menlo Park, New York, S. 185-232.
- Engelbart, D.C. (2004): »The Click Heard Round The World«. In: WIRED, Issue 12.01.; http://www.wired.com/wired/archive/12.01/mouse_pr.html.
- Fertig, S./Freeman, E./Gelernter, D. (1996): »Lifestreams: an alternative to the desktop metaphor«. In: Conference Companion on Human Factors in Computing Systems: Common Ground (Vancouver, British Columbia, Canada, April 13 - 18, 1996). M. J. Tauber, Ed. CHI '96. ACM Press, New York, NY, 410-411; elektron.: <http://sigchi.org/chi96/proceedings/videos/Fertig/etf.htm>; dort ist auch ein Video zu finden.
- Foner, L. N. (1993): »What's An Agent, Anyway? A Sociological Case Study«; <http://foner.www.media.mit.edu/people/foner/Julia>.
- Friedewald, M. (1999): »Der Computer als Werkzeug und Medium. Die geistigen und technischen Wurzeln des Personalcomputers«. (Aachener Beiträge zur Wissenschafts- und Technikgeschichte des 20. Jahrhunderts, Bd. 3) Berlin, Diepholz.
- Gentner, D./Nielsen, J. (1996): »The Anti-Mac Interface«. In: Communications of the ACM, 39, 8, S. 83-99.
- Gershman, A. (2002): »Ubiquitous Commerce – Always On, Always Aware, Always Pro-active«. In: Proceedings of the 2002 International Symposium on Applications and the Internet (SAINT 2002), "Mobile Commerce: Vision and Challenges". New York 2002, S. 37-38.
- Good, I. J. (1958): »How Much Science Can You Have at Your Fingertips?«. In: IBM Journal of Research and Development, Vol. 2, No. 4, S. 282-288.
- GUIDEbook (2007): »Graphical User Interface gallery«; <http://www.guidebookgallery.org/>.
- Halasz, F./Moran, T. P. (1982): »Analogy considered harmful«. In: Proceedings of the ACM Conference on Human Factors in Computer Systems (Gaithersburg, Md., March 15 - 17), S. 383-386.
- Hellige, H. D. (1996): »Leitbilder im Time-Sharing-Lebenszyklus: Vom „Multi-Access“ zur „Interactive On-line Community“«. In: ders. (Hrsg.):

- »Technikleitbilder auf dem Prüfstand. Das Leitbild-Assessment aus Sicht der Informatik- und Computergeschichte«, Berlin, S. 205-234.
- Ingalls, D. H. H. (1981): »Design Principles Behind Smalltalk«. In: BYTE 6, 8, S. 286-298.
- Jennings, N. R./Wooldridge, M. J. (1998): »Applications of Intelligent Agents«. In: dies. (Hrsg.): »Agent Technology. Foundations, Applications, and Markets«, Berlin, Heidelberg, New York, S. 3-28.
- Johnson, J./Roberts, T. L./Verplank, W./Smith, D. C./Irby, Ch. H./Beard, M./Mackey, K. (1989): »The Xerox Star: A Retrospective«. In: IEEE Computer, 22, 9, S. 11-26.
- Kaminski, M./Dourish, P./Edwards, W.K./LaMara, A./Salisbury, M./Smith, I. (1999): »SWEETPEA: Software Tools for Programmable Embodied Agents«. In: Altom, M. W./Williams, M. G. (Hrsg.): Proceedings of the ACM CHI 99 Human Factors in Computing Systems Conference May 15-20, 1999, Pittsburgh, Pennsylvania. S. 144-151.
- Kay, A. (1969): »The Reactive Engine«, Ph.D. Thesis, University of Utah.
- Kay, A. (1977): »Microelectronics and the Personal Computer«. In: Scientific American 237, No. 3, S. 230-244; elektron.: <http://www.digibarn.com/collections/books/xerox-parc-1970-80/alto-article/>.
- Kay, A. (1984): »Computer Software«. In: Scientific American 251, No. 3, S. 53-59.
- Kay, A. (1990): »User Interface: A Personal View«. In: Laurel, B. (Hrsg.): »The Art of Human-Computer Interface Design«, Reading, MA u.a., S. 191-208.
- Kay, A. (1993): »The Early History Of Smalltalk«. In: ACM SIGPLAN Notices, 28, 3, S. 69-96.
- Laurel, B. (1990): »Interface Agents: Metaphors with Character«. In: dies. (Hrsg.): »The Art of Human-Computer Interface Design«, Reading, MA u.a., S. 355-365.
- Licklider, J.C.R. (1960): »Man-Computer Symbiosis«. In: IRE Transactions on Human Factors in Electronic, HFE-1, March 1960, S. 4-11; abgedruckt in: In Memoriam: J.C.R. Licklider 1915-1990, Digital Equipment Corporation 1990, S. 1-15.
- Licklider, J.C.R. (1965): »Libraries of the Future«, Cambridge, Mass.
- Lieberman, H. (1997): »Autonomous Interface Agents«. In: Proceedings of the ACM Conference on Computers and Human Interface, CHI-97; <http://lieber.www.media.mit.edu/people/lieber/Lieberary/Letizia/AIA/AI.html>.
- Lifestreams (2000): »Lifestreams Project Homepage«; <http://www.cs.yale.edu/homes/freeman/lifestreams.html>.
- Krämer, S. (1988): »Symbolische Maschinen. Die Idee der Formalisierung in geschichtlichem Abriss«. Darmstadt.

- Maes, P. (1994): »Agents that Reduce Work and Information Overload«. In: Communications of the ACM, 37, 7, S. 30-40.
- Marks, S. L. (1982): »JOSS-Conversational Computing for the Nonprogrammer«. In: Annals of the History of Computing 4, 1, S. 35 - 51.
- Microsoft (2001): »Microsoft Inductive User Interface Guidelines«, 9.2.2001; <http://msdn2.microsoft.com/en-us/library/ms997506.aspx>.
- Microsoft (2007): »Monitoring Group Activities«, Patentantrag; <http://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=/netahtml/PTO/srchnum.html&r=1&f=G&l=50&s1=20070300174.PGNR.&OS=DN/20070300174&RS=DN/20070300174>.
- Nardi, B. A./Miller, J. R./Wright, D.J. (1998): »Collaborative, Programmable Intelligent Agents«. In: Communications of the ACM, 41, 3, S. 96-104.
- Neisser, U. (1964): »Computers as Tools and as Metaphors«, Georgetown University Conference on Cybernetics and Society, November 20, 1964; abgedruckt in: Orr, W. D. (Hrsg.): »Conversational Computers«, New York, London, Sydney 1968, S. 206-217.
- Nievergelt, J./Weydert, J. (1980): »Site, Modes, and Trails: Telling the User of an Interactive System Where He is, What He Can Do, and How to Get Places«. In: Guedj et al. (Hrsg.): »Methodology of Interaction«, Amsterdam 1980, S. 327-338; (Exzerpt) abgedruckt in: Baecker, R. M., Buxton, W.A.S. (Hrsg.): »Readings in Human-Computer Interaction: A Multidisciplinary Approach«, San Mateo, Cal. 1987, S. 438-441.
- Norman, D.A. (1989): »The design of everyday things«. New York.
- Norman, D. A. (1990): »Why Interfaces Don't Work«. In: Laurel, B. (Hrsg.): »The Art of Human-Computer Interface Design«. Reading, MA, S. 209-219.
- Norman, D.A. (1998): »The Invisible Computer«. Cambridge, MA, London.
- OLPC (2007): <http://laptop.org/en/laptop/interface/index.shtml>.
- Orr, W. D. (1968): »The Culler-Fried Approach to On-Line Computing«. In: ders. (Hrsg.): »Conversational Computers«, New York, London, Sydney, S. 23-28.
- Overhage, C. F. J./Harman, R. J. (1965): »The On-Line Intellectual Community and the Information Transfer system at M.I.T. in 1975«, abgedruckt in: Kochen, M. (Hrsg.): »The Growth of Knowledge«, New York, London, Sydney 1967, S. 77-95.
- Picard, R.W. (1995): »Affective Computing«, MIT Media Lab, Technical Report No. 321; <http://affect.media.mit.edu/pdfs/95.picard.pdf>
- Pflüger, J. (2002): »Language in Computing«. In: Dörries, M. (Hrsg.): »Experimenting in tongues: studies in science and language«, Stanford, CA, S. 125-162, 188-201.
- Pflüger, J. (2004a): »Konversation, Manipulation, Delegation: Zur Ideengeschichte der Interaktivität«. In: Hellige, H. D. (Hrsg.): »Geschichten der

- Informatik. Visionen, Paradigmen, Leitmotive«, Berlin Heidelberg, New York, S. 367-408.
- Pflüger, J. (2004b): »Writing, Building, Growing: Leitvorstellungen der Programmiergeschichte«. In: Hellige, H. D. (Hrsg.): »Geschichten der Informatik«, Berlin Heidelberg, New York, S. 275-319.
- Pomberger, G./Blaschek, G. (1993): »Grundlagen des Software Engineering«, München, Wien.
- Raskin, J. (2000): »The Humane Interface«. New Directions for Designing Interactive Systems. Reading, MA.
- Rittel, H./Webber, M. (1973): »Dilemmas in a General Theory of Planning«. In: Policy Sciences, Vol. 4, Amsterdam, S. 155-169.
- Rogers, Y. (2006): »Moving on from Weiser's Vision of Calm Computing: Engaging UbiComp Experiences«. In: Dourish, P./Friday, A. (Hrsg.): UbiComp 2006, LNCS 4206, Berlin Heidelberg, New York, S. 404-421.
- Ross, D.T. (1956): »Gestalt Programming: A New Concept in Automatic Programming«. In: AFIPS Conference Proceedings, Bd. 9 (WJCC 1956), S. 5-11.
- Schilit, B./Adams, N./Want, R. (1994) »Context-aware computing applications«. In: IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, US, <ftp://ftp.parc.xerox.com/pub/schilit/wmc-94-schilit.ps>.
- Shaw, J. C. (1965): »JOSS: Experience with an Experimental Computing Service for Users at Remote Consoles«, RAND Report P-3149, abgedruckt in: Orr, W. D. (Hrsg.): »Conversational Computers«, New York London Sydney 1968, S. 15-22.
- Shneiderman, B. (1983): »Direct Manipulation: A Step Beyond Programming Languages«. In: IEEE Computer 16, 8, S. 57-69.
- Siemens AG (2002): Evdoxia Tsakiridou: »Inside the All-Inclusive Network«. In: »Pictures of the Future«, Fall 2002; http://www.siemens.com/index.jsp?sdc_p=l1o1184335u20mcn1184101s5pFE.
- Simon, H. (1962): »The Architecture of Complexity«. In: Proc. American Philos. Soc., 106, 6, S. 467-482; zitiert nach der dt. Ausgabe: »Die Wissenschaft vom Künstlichen«, Wien New York 1994, S. 144-172.
- Smith, D. C./Irby, Ch./Kimball, R./Verplank, W. L./Harslem, E. (1982): »Designing the Star User Interface«. In: BYTE 7, 4, S. 242 - 282; abgedruckt in: Baecker, R.M./Buxton, W. A. S. (Hrsg.): »Readings in Human-Computer Interaction: A Multidisciplinary Approach«, San Mateo, California 1987, S. 653-661.
- Spiegel Online (16.8.2007): »Jeder zweite Lehrling hat miese Computerkenntnisse«; <http://www.spiegel.de/netzwelt/web/0,1518,499508,00.html>.

- Spiekermann, S. (2007): »Ubiquitous Computing and Automation Theory - A Transfer and Extension of Human Factors Issues«, submitted for publication; <http://amor.rz.hu-berlin.de/~spiekers/publications.html>.
- Spinner, H.F. (1994): »Die Wissensordnung. Ein Leitkonzept für die dritte Grundordnung des Informationszeitalters«. Opladen.
- Stanford Persuasive Technology Lab (2007); <http://captology.stanford.edu/>
- Terveen, L.G./La Tondra Murray (1996): »Helping Users Program Their Personal Agents«. In: CHI 96, April 13-18, S. 355-361.
- TLS (1962): The Times Literary Supplement, Friday March 23 1962, S. 194.
- Tognazzini, B. (1993): »Principles, Techniques, and Ethics of Stage Magic and Their Application to Human Interface Design«, Proceedings of INTERCHI, 1993 (Amsterdam, The Netherlands, April 24-29, 1993). ACM, New York, S. 355-362.
- Want, R./Hopper, A./ Falcão, V./Gibbons, J. (1992): »The active badge location system«, ACM Transactions on Information Systems (TOIS), v.10 n.1, S. 91-102, Jan. 1992; http://web.media.mit.edu/~dmerrill/badge/Want92_ActiveBadge.pdf.
- Weiser, M. (1991): »The Computer for the 21st Century«. In: Scientific American 265 (September), S. 94-104.
- Weiser, M. (1994a): »The world is not a desktop«. In: interactions 1,1, S.7 - 8
- Weiser, M. (1994b): »Building Invisible Interfaces«, Vortrag auf der User Interface, Systems, and Technologies (UIST) conference, November 2, 1994; Slides: http://www.ubiq.com/hypertext/weiser/UIST94_4up.ps.
- Weiser, M./Brown, J.S. (1996): »The Coming Age Of Calm Technology«, Xerox PARC, October 5; <http://www.cs.ucsb.edu/~ebelding/courses/284/w04/papers/calm.pdf>.
- Weiser, M./[Gold, R./ Brown, J. S.] (1999): »The origins of ubiquitous computing research at PARC in the late 1980s«. In: IBM Systems Journal, 38, 4, S. 693-696.
- Winograd, T. (1997): »The Design of Interaction«. In: Denning, P. J./Metcalfe, R. M. (Hrsg.): »Beyond Calculation. The Next Fifty Years of Computing«, New York, S. 149-161.
- Xerox (1975): »Personal Dynamic Media«, Xerox Corporation Palo Alto Research Center, Learning Research Group, March 1975.

AUTORINNENVERZEICHNIS

Gino Brunetti, (Dipl.-Inf.) Chief Strategy Officer der INI-GraphicsNet Stiftung und Dozent an der TU Darmstadt; Studium der Informatik an der Technischen Universität Darmstadt; 1995-2001 Wissenschaftlicher Mitarbeiter und ab 1998 stellvertretender Abteilungsleiter am Fraunhofer-Institut für Graphische Datenverarbeitung, Darmstadt; 2001-2003 technisch wissenschaftlicher Leiter in der Aufbauphase vom CETA/ SENAI in Porto Alegre, Brasilien, einer Einrichtung für angewandte Forschung in Brasilien; seit 2007 Leiter der INI-GraphicsNet Stiftung; Publikationen zur CAD- und CAE-Systementwicklung, zum Feature-Based Virtual Engineering, zu Immersive Styling Environments und zu ergonomischen Aspekten in der Mixt Reality-Entwicklung.

Wolfgang Coy, (Dipl.-Ing. math., Dr. rer. nat.) Professor für Informatik an der Humboldt-Universität zu Berlin; Studium der Elektrotechnik, Mathematik und Philosophie an der TH Darmstadt; 1975 Promotion; 1979-96 Professur für Informatik an der Universität Bremen, seit 1996 an der Humboldt-Universität; Lehre und Forschungen zu den Gebieten Digitale Medien, Informatik und Gesellschaft, Theorie der Informatik, Sozial- und Kulturgeschichte der Informatik, sowie zu philosophischen und theoretischen Fragen der Informatik; langjähriger Sprecher des GI-Fachbereiches »Informatik und Gesellschaft«; Bücher über Industrieroboter, Rechnerarchitektur, Expertensysteme, Theorie der Informatik und Mitherausgeber weiterer Bücher, u.a. »Hyperkult I – Geschichte, Theorie und Kontext digitaler Medien« 1997 und »HyperKult II – Zur Ortsbestimmung analoger und digitaler Medien« 2004.

José Luis Encarnação, (Dipl. Ing., Dr. Ing.) Professor für Informatik an der Technischen Universität Darmstadt; geboren in Portugal, zunächst Dipl.-Ing. und Dr.-Ing. der Elektrotechnik an der Technischen Universität Berlin; seit 1975 Professor für Informatik an der Technischen Universität Darmstadt und dort Leiter des Fachgebiets Graphisch-Interaktive Systeme (TUD-GRIS); unter seiner Leitung wurde in den Folgejahren das INI-GraphicsNet (International Network of Institutions for advanced education, training and R&D in Computer Graphics technology, systems

and applications) aufgebaut, ein institutionelles Netzwerk, das weltweit einen der größten Schwerpunkte für die Visualisierungstechnologien, die neuen Medien und neuen Kommunikationsformen mit den zugehörigen IuK-Technologien und -Anwendungen bildet; Gründer der INI-GraphicsNet Investitions Holding GmbH und der INI-GraphicsNet Stiftung mit dem Ziel, die Entwicklung von Ausgründungen und die Kommerzialisierung von Forschungsergebnissen aus dem INI-GraphicsNet zu fördern; seit Juli 2001 Vorsitzender der IuK-Gruppe der Fraunhofer-Gesellschaft, seit 2001 Mitglied der EU-Beratergruppe für den Bereich der Informationstechnologie (ISTAG), deren Vorsitzender von 2002 bis 2004 und stellvertretender Vorsitzender seit 2005; seit 2002 Mitglied im Präsidium der Fraunhofer-Gesellschaft; seit 2001 ordentliches Mitglied in der Berlin-Brandenburgischen Akademie der Wissenschaften (»Leibniz-Akademie«); Autor bzw. Co-Autor von mehr als 500 Veröffentlichungen und Fachaufsätzen in internationalen rezensierten Zeitschriften und prominenten Fachkonferenzen; verantwortlicher Hauptherausgeber von *Computers & Graphics*.

Michael Friedewald, (Dr.-Ing., Dipl.-Wirt. Ing.) Projektleiter am Fraunhofer-Institut für System- und Innovationsforschung in Karlsruhe tätig. Seine Arbeitsschwerpunkte liegen im Bereich der Technikvorausschau und der Technikfolgenabschätzung, insbesondere im Bereich neuer Informations- und Informationstechnik, z.B. Ambient Intelligence. Er hat an der Rheinisch-Westfälischen Technischen Hochschule Aachen Elektrotechnik, Wirtschaftswissenschaften und Technikgeschichte studiert. Seine Dissertation über die technischen und geistigen Wurzeln des Personal Computing und der Mensch-Computer-Interaktion wurde 1999 vom Verein Deutscher Ingenieure mit dem Rudolf-Kellermann-Preis für Technikgeschichte ausgezeichnet. Letzte Buchveröffentlichungen: »Safeguards in a world of Ambient Intelligence«, Springer, 2008. »Arbeiten in der Zukunft – Strukturen und Trends der Industriearbeit«, Büro für Technikfolgenabschätzung beim Deutschen Bundestag, 2007.

Hans Dieter Hellige, (Dr. phil.) Professor für Technikgestaltung und Technikgenese mit dem Schwerpunkt Informationstechnik am Forschungszentrum artec der Universität Bremen; Studium der Geschichte mit den Schwerpunkten neuere Wirtschafts-, Sozial- und Technikgeschichte; Promotion 1976 an der TU Berlin; seit 1977 als Historiker in der Ingenieurausbildung an der Universität Bremen tätig; Lehrtätigkeit in den Studiengängen Elektrotechnik und Informationstechnik, Informatik, Medieninformatik, Geschichte; Habilitation 1995 auf dem Gebiet der historischen Technikgeneseforschung; seit 1993 Sprecher der Fachgruppe

Informatik- und Computergeschichte in der Gesellschaft für Informatik; Publikationen zur Geschichte und Bewertung von Einzeltechniken der Telekommunikation und Computerkommunikation, zur Geschichte des Computing und der Informatik, (u.a. H. D. Hellige (Hrsg.): Geschichten der Informatik. Visionen, Paradigmen, Leitmotive, Springer 2004; Die »Geschichte des Internet als Lernprozess«. In: Kreowski, (Hrsg.) »Informatik und Gesellschaft. Verflechtungen und Perspektiven«. LIT-Verlag 2008); zur Theorie und Methodik der Technikgeneseforschung und Konstruktionslehre, zur Energie- und Ressourcen-Geschichte sowie zur Wirtschafts- und Sozialgeschichte des Deutschen Kaiserreichs.

Eva Hornecker, (Dipl. Inform., Dr. Ing.) Visiting Research Fellow am »Pervasive Interaction Lab« an der Open University, UK; Studium der Informatik und Pädagogik an der TH Darmstadt und Promotion 2004 über »Tangible Interfaces als kooperationsunterstützendes Medium« an der Uni Bremen; danach Universitätsassistentin an der TU Wien, wo sie Interaction Design unterrichtete; Forschungsaufenthalte an der University of Sussex sowie in Neuseeland. Ihre Forschungsinteressen und Publikationen liegen im Bereich des »Beyond the Desktop Interaction Design«, mit Schwerpunkt auf Tangible sowie Embodied Interaction, sowie auf Aspekten kooperativer/sozialer Interaktion.

Marion Jähne, (Dipl. Math.) Wissenschaftliche Mitarbeiterin an der Technischen Universität Darmstadt am Fachbereich Graphisch-Interaktive Systeme seit 2006; Studium der Mathematik an der Technischen Universität Darmstadt; 2004 bis 2006 wissenschaftliche Assistentin am Fraunhofer Institut für Graphische Datenverarbeitung; Publikationen zur automatischen Bildanalyse.

Matthias Müller-Prove, (Dipl.-Inform.), ist seit 15 Jahren bei international operierenden Software-Firmen im Bereich Mensch-Maschine-Kommunikation tätig. Er hat maßgeblich die Benutzungsschnittstelle des Web-Editors Adobe GoLive gestaltet. Für Sun Microsystems wirkte er seit 2002 als User Experience und Interaktionsdesigner an der Bürosoftware StarOffice und war Community Lead des User Experience Projektes für die Open Source Software OpenOffice.org. Derzeit arbeitet er bei Sun als User Experience Architect im Bereich Desktop Virtualisierung. Darüber hinaus gilt sein Interesse der Entwicklungsgeschichte des Hypertexts und der Grafischen Benutzungsschnittstellen. Für seine Arbeit »Vision and Reality for Hypertext and Graphical User Interfaces« erhielt er 2005 den Wolfgang von Kempelen Preis für Informatikgeschichte.

Frieder Nake, Prof. (Dr. rer. nat.) Professor für Informatik der Universität Bremen; Studium der Mathematik in Stuttgart; Promotion über Wahrscheinlichkeitstheorie; durch Max Bense Beschäftigung mit Semiotik; seit 1963 Pionierarbeiten zur Computergrafik allgemein, zur Computerkunst im besonderen; erste nationale und internationale Ausstellungen ab 1965; 1968-69 Research Fellow in Toronto; 1970-72 Assistant Professor in Vancouver; seit 1972 Professor für grafische Datenverarbeitung und interaktive Systeme an der Universität Bremen; zahlreiche Publikationen zur Computergrafik, zur Computerkunst, zu Digitalen Medien / Hypermedien, Semiotik und zur Theorie bzw. Gesellschaftstheorie der Informatik; langjähriger Sprecher des Fachbereiches »Informatik und Gesellschaft«. In jüngster Zeit Konzentration auf mediale Seiten der Informatik. Neuere Buchveröffentlichungen: Diethelm Stoller/Frieder Nake: »Algorithmus und Kunst. Die präzisen Vergnügen«, Sautter & Lackmann, Hamburg, 1993; Frieder Nake: »Die erträgliche Leichtigkeit der Zeichen. Ästhetik, Semiotik, Informatik«, Agis, Baden-Baden 1994; »Algorithmik - Kunst - Semiotik. Hommage für Frieder Nake«, hrsg. von K. - H. Rödiger, Synchron Wissenschaftsverlag der Autoren, Heidelberg, 2003; Peter B. Andersen: »Computing Science and Semiotics. A Fundamental Approach«, im Druck.

Horst Oberquelle, (Dr. rer. nat.), Professor für Informatik im Fachbereich der Universität Hamburg; Studium der Mathematik und Informatik an der Universität Hamburg, Diplom in Informatik 1973; Wissenschaftlicher Mitarbeiter, Promotion 1979, Hochschulassistent, 1985-1986 Vertretungsprofessor Universität Aarhus, Dänemark. Habilitation in Informatik 1986. Seit 1986 Universitätsprofessor für Informatik im Fachbereich Informatik der Universität Hamburg; Arbeitsgebiete: Mensch-Computer-Interaktion, computergestützte Kooperation; langjähriger Sprecher des Fachausschusses / Fachbereichs »Mensch-Computer-Interaktion« der GI, deutscher Vertreter in IFIP TC 13 »Human-Computer Interaction«; zahlreiche Publikationen zur Grundsatzfragen der Softwareergonomie, Usability-Forschung und Systemgestaltung.

Jörg-Martin Pflüger, (Dr. rer. nat.) Studium der Elektrotechnik, Mathematik und Philosophie, Promotion und Habilitation in Theoretischer Informatik. OBERINGENIEUR an der Universität Bremen, ab 1998 Vertragsprofessor am Institut für Gestaltungs- und Wirkungsforschung der Fakultät Informatik an der TU Wien, seit Mitte 2004 Privatier. Forschung, Lehre und Publikationen in den Fachgebieten Theoretische Informatik und »Theorie der Informatik«, zur Kulturtheorie, Sozialpsychologie und Geistesgeschichte der Informatik, zur medialen Funktion des Computers

und kulturgeschichtlichen Aspekten der Informationsvisualisierung sowie zur Rolle der Computer und Medien in der Wissensorganisation. Ausgewählte Publikationen: Language in Computing, in M. Dörries (Hrsg.): Experimenting in Tongues, Stanford University Press 2002; FAQ: Microsoft (zusammen mit P. Purgathofer), in A. Roesler/B. Stiegler (Hrsg.): Microsoft. Medien Macht Monopol, Suhrkamp 2002; Writing, Building, Growing. Leitvorstellungen der Programmiergeschichte; Konversation, Manipulation, Delegation. Zur Ideengeschichte der Interaktivität, beide in H. D. Hellige (Hrsg.): Geschichten der Informatik. Visionen, Springer 2004; Wo die Quantität in Qualität umschlägt. Notizen zum Verhältnis von Analogem und Digitalen, in M. Warnke/W. Coy/C. Tholen (Hrsg.): HyperKult II, transcript 2005

Ingrid Rügge, (Dipl.-Inform., Dr.-Ing.) nach knapp 10jähriger Tätigkeit in der Druckindustrie Studium der Informatik (1995 Diplom), von 1999 bis 2007 Wissenschaftlerin am Technologie- Zentrum Informatik der Universität Bremen, u.a. ab 2004 Geschäftsführerin des Leitthemas Wear-Lab und des Demo-Centers im Mobile Solution Center. Promotion 2006 im Fachbereich Mathematik/Informatik der Universität Bremen, seit 2007 Geschäftsführerin der International Graduate School of Dynamics in Logistics der Universität Bremen; Forschungsschwerpunkte: Wearable Computing, mobile Anwendungen im Gesundheitswesen, Benutzbarkeit mobiler Lösungen. Buchveröffentlichungen: I. Rügge et al. (Hrsg.): »Arbeiten und begreifen: Neue Mensch-Maschine-Schnittstellen«. Münster 1998; Rügge, I.: »Mobile Solutions. Einsatzpotentiale, Nutzungsprobleme und Lösungsansätze«. (Advanced Studies Mobile Research Center Bremen), Wiesbaden 2007.

Horst Zuse, (Dipl.-Inform., Dr.-Ing.) Privatdozent, Professor an der FH-Lausitz; von 1967 bis 1973 Studium der Elektrotechnik an der Technischen Universität Berlin (TUB); 1985 Promotion auf dem Gebiet der Softwarekomplexitätsmaße; 1998 Venia legendi im Fachbereichsrat Informatik der TU-Berlin für das Gebiet der Praktischen Informatik; von 2003 bis 2006 Gastprofessor an der FH-Lausitz und im November 2006 zum Honorarprofessor der FH-Lausitz ernannt. 1991 publizierte er das Buch: Horst Zuse: »Software Complexity - Measures and Methods«, im DeGruyter Verlag und 1998 erschien das Buch: Horst Zuse: »A Framework of Software Measurement«, im DeGruyter Verlag.



Birgit Althans, Kathrin Audehm,
Beate Binder, Moritz Ege, Alexa Färber (Hg.)

Kreativität. Eine Rückrufaktion

Zeitschrift für Kulturwissenschaften,
Heft 1/2008

März 2008, 138 Seiten, kart., 8,50 €,
ISSN 9783-9331

ZfK - Zeitschrift für Kulturwissenschaften

Der Befund zu aktuellen Konzepten kulturwissenschaftlicher Analyse und Synthese ist ambivalent: Neben innovativen und qualitativ hochwertigen Ansätzen besonders jüngerer Forscher und Forscherinnen steht eine Masse oberflächlicher Antragsprosa und zeitgeistiger Wissensproduktion – zugleich ist das Werk einer ganzen Generation interdisziplinärer Pioniere noch wenig erschlossen.

In dieser Situation soll die **Zeitschrift für Kulturwissenschaften** eine Plattform für Diskussion und Kontroverse über »Kultur« und die Kulturwissenschaften bieten. Die Gegenwart braucht mehr denn je reflektierte Kultur, historisch situiertes und sozial verantwortetes Wissen. Aus den Einzelwissenschaften heraus kann so mit klugen interdisziplinären Forschungsansätzen fruchtbar über die Rolle von Geschichte und Gedächtnis, von Erneuerung und Verstetigung, von Selbststeuerung und ökonomischer Umwälzung im Bereich der Kulturproduktion und der naturwissenschaftlichen Produktion von Wissen diskutiert werden.

Die **Zeitschrift für Kulturwissenschaften** lässt gerade auch jüngere Wissenschaftler und Wissenschaftlerinnen zu Wort kommen, die aktuelle fächerübergreifende Ansätze entwickeln.

Lust auf mehr?

Die **Zeitschrift für Kulturwissenschaften** erscheint zweimal jährlich in Themenheften. Bisher liegen die Ausgaben »Fremde Dinge« (1/2007), »Filmwissenschaft als Kulturwissenschaft« (2/2007) und »Kreativität. Eine Rückrufaktion« (1/2008) vor.

Die **Zeitschrift für Kulturwissenschaften** kann auch im Abonnement für den Preis von 8,50 € je Ausgabe bezogen werden.

Bestellung per E-Mail unter: bestellung.zfk@transcript-verlag.de

Kultur- und Medientheorie

Hans Dieter Hellige (Hg.)
Mensch-Computer-Interface
Zur Geschichte und Zukunft
der Computerbedienung
Juni 2008, 400 Seiten,
kart., 29,80 €,
ISBN: 978-3-89942-564-2

Claudia Lilje,
Anne-Rose Meyer (Hg.)
Interkulturelle Mahlzeiten
Kulinarische Begegnungen
und Kommunikation
in der Literatur
Juni 2008, 368 Seiten,
kart., 30,80 €,
ISBN: 978-3-89942-881-0

Geert Lovink
Zero Comments
Elemente einer kritischen
Internetkultur
Juni 2008, ca. 300 Seiten,
kart., ca. 27,80 €,
ISBN: 978-3-89942-804-9

Doris Kolesch, Vito Pinto,
Jenny Schrödl (Hg.)
Stimm-Welten
Philosophische,
medientheoretische und
ästhetische Perspektiven
Juni 2008, ca. 216 Seiten,
kart., ca. 24,80 €,
ISBN: 978-3-89942-904-6

Simone Loleit
**Wahrheit, Lüge, Fiktion:
Das Bad in der deutsch-
sprachigen Literatur
des 16. Jahrhunderts**
Juni 2008, 392 Seiten,
kart., 41,80 €,
ISBN: 978-3-89942-666-3

York Kautt
Image
Zur Genealogie eines
Kommunikationscodes
und zur Entwicklung des
Funktionssystems Werbung
Juni 2008, 370 Seiten,
kart., zahlr. Abb., ca. 30,80 €,
ISBN: 978-3-89942-826-1

Michael Dürfeld
**Das Ornamentale
und die architek-
tonische Form**
Systemtheoretische Irritationen
Mai 2008, 160 Seiten,
kart., 19,80 €,
ISBN: 978-3-89942-898-8

Matthias Bruhn,
Kai-Uwe Hemken (Hg.)
Modernisierung des Sehens
Sehweisen zwischen Künsten
und Medien
Mai 2008, 374 Seiten,
kart., 29,80 €,
ISBN: 978-3-89942-912-1

Peter Seibert (Hg.)
**Samuel Beckett
und die Medien**
Neue Perspektiven auf
einen Medienkünstler
des 20. Jahrhunderts
Mai 2008, 224 Seiten,
kart., 24,80 €,
ISBN: 978-3-89942-843-8

Derrick de Kerckhove,
Martina Leeker,
Kerstin Schmidt (Hg.)
McLuhan neu lesen
Kritische Analysen zu Medien
und Kultur im 21. Jahrhundert
April 2008, 514 Seiten,
kart., zahlr. Abb., inkl. DVD, 39,80 €,
ISBN: 978-3-89942-762-2

Leseproben und weitere Informationen finden Sie unter:
www.transcript-verlag.de

Kultur- und Medientheorie

Ronald Kurt,
Klaus Näumann (Hg.)
**Menschliches Handeln
als Improvisation**
Sozial- und musikwissenschaftliche
Perspektiven
März 2008, 238 Seiten,
kart., 25,80 €,
ISBN: 978-3-89942-754-7

Thomas Ernst,
Patricia Gozalbez Cantó,
Sebastian Richter,
Nadja Sennewald,
Julia Tieke (Hg.)
SUBversionen
Zum Verhältnis von Politik
und Ästhetik in der Gegenwart
März 2008, 406 Seiten,
kart., 30,80 €,
ISBN: 978-3-89942-677-9

Cora von Pape
Kunstkleider
Die Präsenz des Körpers
in textilen Kunst-Objekten
des 20. Jahrhunderts
Februar 2008, 228 Seiten,
kart., zahlr. Abb., 26,80 €,
ISBN: 978-3-89942-825-4

Christian Bielefeldt,
Udo Dahmen,
Rolf Großmann (Hg.)
PopMusicology
Perspektiven der
Popmusikwissenschaft
Februar 2008, 284 Seiten,
kart., 26,80 €,
ISBN: 978-3-89942-603-8

Helge Meyer
Schmerz als Bild
Leiden und Selbstverletzung
in der Performance Art
Januar 2008, 372 Seiten,
kart., zahlr. Abb., 35,80 €,
ISBN: 978-3-89942-868-1

Annett Zinsmeister (Hg.)
welt[stadt]raum
Mediale Inszenierungen
Januar 2008, 172 Seiten,
kart., zahlr. Abb., 18,80 €,
ISBN: 978-3-89942-419-5

Lars Koch (Hg.)
**Modernisierung als
Amerikanisierung?**
Entwicklungslinien
der westdeutschen Kultur
1945-1960
2007, 330 Seiten,
kart., 29,80 €,
ISBN: 978-3-89942-615-1

Torben Fischer,
Matthias N. Lorenz (Hg.)
**Lexikon der »Vergangen-
heitsbewältigung«
in Deutschland**
Debatten- und Diskurs-
geschichte des National-
sozialismus nach 1945
2007, 398 Seiten,
kart., 29,80 €,
ISBN: 978-3-89942-773-8

Laura Bieger
Ästhetik der Immersion
Raum-Erleben zwischen
Welt und Bild.
Las Vegas, Washington
und die White City
2007, 266 Seiten,
kart., zahlr. Abb., 26,80 €,
ISBN: 978-3-89942-736-3

Gunnar Schmidt
Ästhetik des Fadens
Zur Medialisierung eines
Materials in der
Avantgardekunst
2007, 156 Seiten,
kart., zahlr. Abb., 14,80 €,
ISBN: 978-3-89942-800-1

**Leseproben und weitere Informationen finden Sie unter:
www.transcript-verlag.de**

Kultur- und Medientheorie

Lutz Hieber,
Dominik Schrage (Hg.)

Technische Reproduzierbarkeit

Zur Kulturosoziologie
massenmedialer
Vervielfältigung

2007, 202 Seiten,
kart., zahlr. Abb., 20,80 €,
ISBN: 978-3-89942-714-1

Marcus Krause,
Nicolas Pethes (Hg.)

Mr. Münsterberg und Dr. Hyde

Zur Filmgeschichte des
Menschenexperiments

2007, 318 Seiten,
kart., 29,80 €,
ISBN: 978-3-89942-640-3

Christoph Lischka,
Andrea Sick (eds.)

Machines as Agency Artistic Perspectives

2007, 198 Seiten,
kart., 25,80 €,
ISBN: 978-3-89942-646-5

Immacolata Amodeo

Das Opernhafte

Eine Studie zum
»gusto melodrammatico«
in Italien und Europa

2007, 224 Seiten,
kart., 25,80 €,
ISBN: 978-3-89942-693-9

Gunther Gebhard,
Oliver Geisler,
Steffen Schröter (Hg.)

Heimat

Konturen und Konjunkturen
eines umstrittenen Konzepts

2007, 202 Seiten,
kart., 22,80 €,
ISBN: 978-3-89942-711-0

Tara Forrest

The Politics of Imagination

Benjamin, Kracauer, Kluge

2007, 198 Seiten,
kart., 25,80 €,
ISBN: 978-3-89942-681-6

Stephan Günzel (Hg.)

Topologie.

Zur Raumbeschreibung in
den Kultur- und Medien-
wissenschaften

2007, 332 Seiten,
kart., zahlr. Abb., 29,80 €,
ISBN: 978-3-89942-710-3

Ramón Reichert

Im Kino der Human- wissenschaften

Studien zur Medialisierung
wissenschaftlichen Wissens

2007, 298 Seiten,
kart., 29,80 €,
ISBN: 978-3-89942-647-2

Jürgen Hasse

Übersene Räume

Zur Kulturgeschichte
und Heterotopologie
des Parkhauses

2007, 220 Seiten,
kart., zahlr. Abb., 24,80 €,
ISBN: 978-3-89942-775-2

Kathrin Busch,
Iris Därmann (Hg.)

»pathos«

Konturen eines
kulturwissenschaftlichen
Grundbegriffs

2007, 206 Seiten,
kart., 21,80 €,
ISBN: 978-3-89942-698-4

Leseproben und weitere Informationen finden Sie unter:
www.transcript-verlag.de

Kultur- und Medientheorie

Marc Ries,
Hildegard Fraueneder,
Karin Mairitsch (Hg.)

dating.21

Liebesorganisation und
Verabredungskulturen

2007, 250 Seiten,
kart., 25,80 €,
ISBN: 978-3-89942-611-3

Thomas Hecken

Theorien der Populärkultur

Dreißig Positionen von Schiller
bis zu den Cultural Studies

2007, 232 Seiten,
kart., 22,80 €,
ISBN: 978-3-89942-544-4

Meike Kröncke, Kerstin Mey,
Yvonne Spielmann (Hg.)

Kultureller Umbau

Räume, Identitäten
und Re/Präsentationen

2007, 208 Seiten,
kart., 21,80 €,
ISBN: 978-3-89942-556-7

Meike Kröncke,
Kerstin Mey,
Yvonne Spielmann (Hg.)

Kultureller Umbau

Räume, Identitäten
und Re/Präsentationen

2007, 208 Seiten,
kart., 21,80 €,
ISBN: 978-3-89942-556-7

Thomas Hecken

Theorien der Populärkultur

Dreißig Positionen von Schiller
bis zu den Cultural Studies

2007, 232 Seiten,
kart., 22,80 €,
ISBN: 978-3-89942-544-4

Kai Lehmann,
Michael Schetsche (Hg.)
Die Google-Gesellschaft

Vom digitalen Wandel
des Wissens

2007, 410 Seiten,
kart., 26,80 €,
ISBN: 978-3-89942-780-6

Nic Leonhardt

Piktoral-Dramaturgie

Visuelle Kultur und Theater
im 19. Jahrhundert (1869-1899)

2007, 392 Seiten,
kart., zahlr. Abb., 33,80 €,
ISBN: 978-3-89942-596-3

Michael Charlton,

Tilmann Sutter

Lesekomunikation

Mediensozialisation
in Gesprächen über
mehrdeutige Texte
(unter Mitarbeit von
Christina Burbaum,
Gisela Mehren und
Friederike Rau)

2007, 176 Seiten,
kart., 18,80 €,
ISBN: 978-3-89942-601-4

Georg Stauth,
Faruk Birtek (Hg.)

›Istanbul‹

Geistige Wanderungen
aus der ›Welt in Scherben‹

2007, 292 Seiten,
kart., 33,80 €,
ISBN: 978-3-89942-474-4

Björn Bollhöfer

Geographien des Fernsehens

Der Kölner Tatort
als mediale Verortung
kultureller Praktiken

2007, 258 Seiten,
kart., zahlr. Abb., 25,80 €,
ISBN: 978-3-89942-621-2

**Leseproben und weitere Informationen finden Sie unter:
www.transcript-verlag.de**