

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, bürden 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 Lickliders 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)

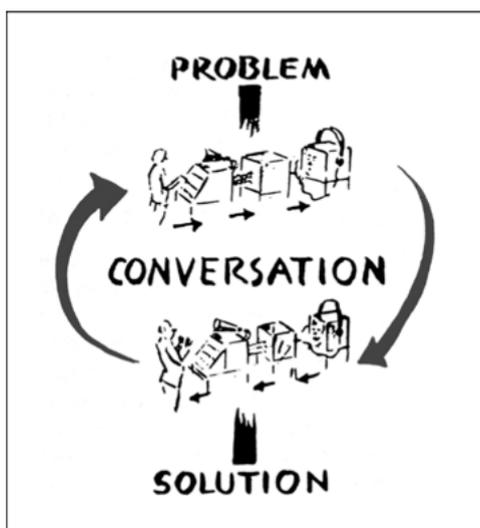
Auch wenn in der Wirklichkeit der 60er Jahre der Dialog zwischen Mensch und Maschine im Wesentlichen im Austausch von Kommandos und Daten bestand, war mit dieser Form von Interaktivität doch eine neue Problemlösetechnik verbunden. Sie kann vielleicht so charakterisiert werden, dass die Trennung zwischen 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)

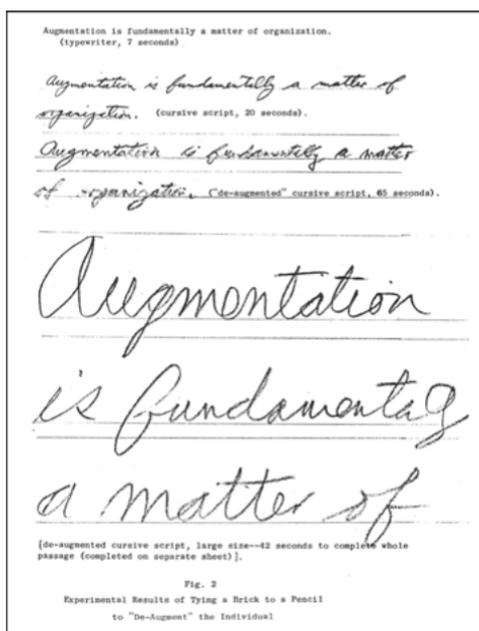
Unter Bezugnahme auf die bekannte Sapir-Whorf-These, dass unsere Sprache bestimmt, wie wir was denken können, formuliert Engelbart seine auf die symbolische Manipulation erweiterte »Neo-Whorf-Hypothese«: »Both the language used by a culture, and the capability for effective intellectual activity are 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««, 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 elaborierter 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*nterface *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 »flatter« 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,

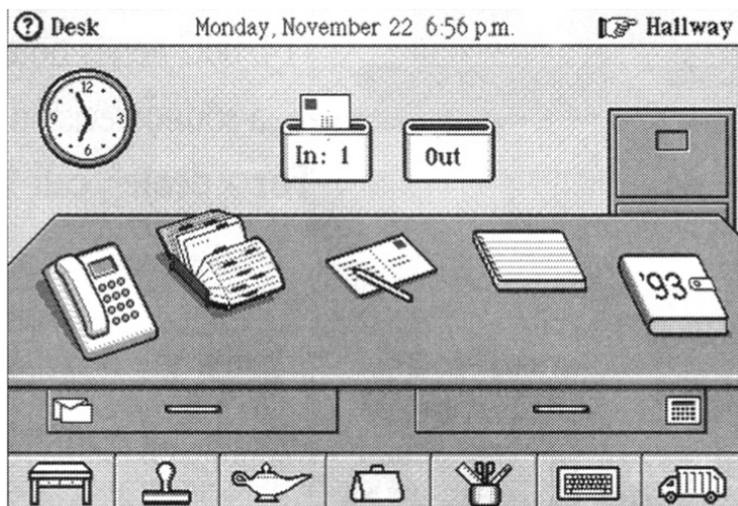
und der Versuch, diese Formen der Kontextualisierung auf die Interaktion mit dem Computer zu übertragen, beschäftigt die ganze Geschichte des User Interface Designs mit mehr oder minderem Erfolg. Wie wir gesehen haben, sollten *Modes* in den früheren Entwürfen den Nutzer von einer Kleinlichkeit der Steuerung 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 überfüllte 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 wütigen 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.

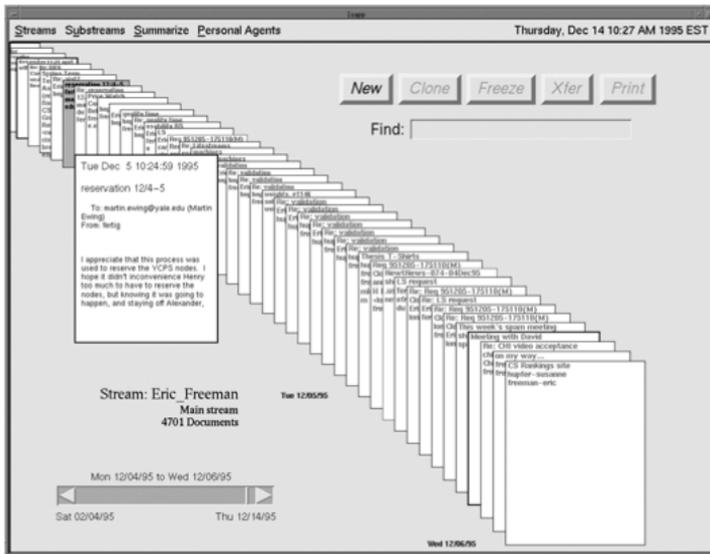
Auf dem Weg dorthin befand sich das radikalere Konzept von *Lifestreams*, das aber unzeitig auf einem Holzweg war. Scott Fertig, Eric Freeman und David Gelernter stellten, im gleichen Jahr wie Gentner und Nielsen ihre Mac-Kritik, die Konzeption eines ›time-ordered stream of documents that functions as 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 verkörpert 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 realistischweise 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 mitlauscht, 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‹ Schreib-tischcomputer 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 Privilegiensystem 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

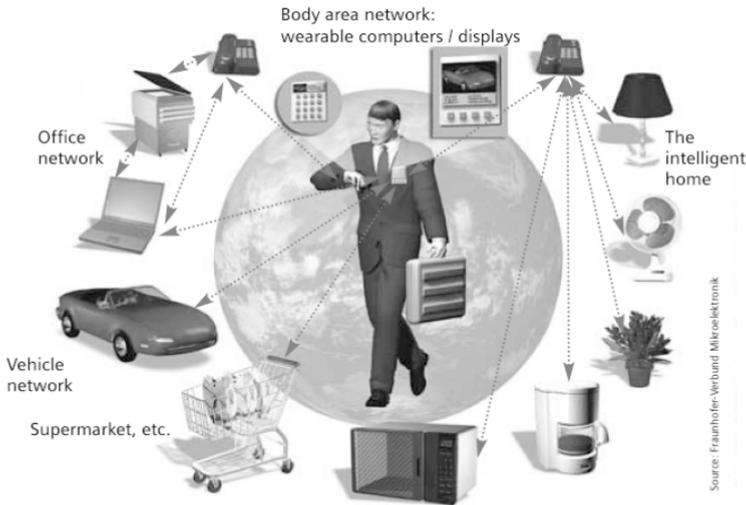
In einem Ausblick werden das Active Badge Konzept dann mit anderen Überwachungs- und Managementfunktionen der Haustechnik integriert und die Menschen problemlos in der automatisiert verwalteten Umwelt positioniert. »The Active Badge extends the concept of an integrated building to take into account the 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. Massenhafter verkaufen ließe sich jedoch die Bewehrung des mobilen Individuums, dem Beruf und Privatleben im Umherschweifen verschwimmt. 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 Hammerzuhändenheit 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)

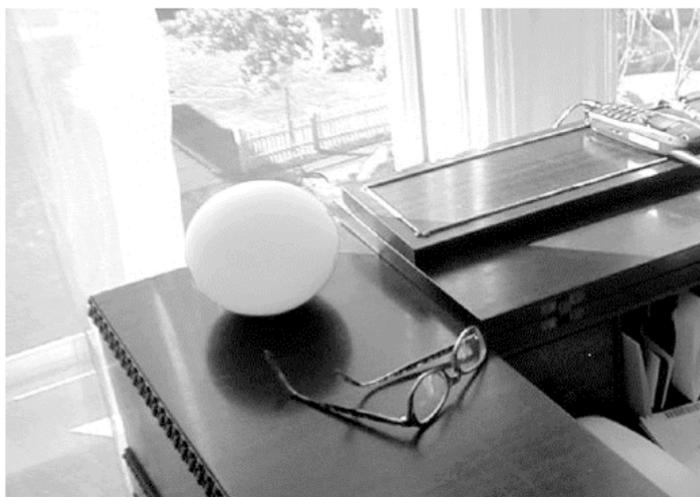
Im Laufe der Zeit hat sich für Weiser als Leitidee das (auf Donald Normans ›affordance‹ Bezug nehmende) Konzept des »*calm computing*« herauskristallisiert, was eher »the desired state of mind of the user« als eine Hardware-Ausstattung meint. Statt sich dem »excitement of interaction« mit einem PC auszusetzen, 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.