KJELD SCHMIDT

VON NIEDERER HERKUNFT

Die praktischen Wurzeln des interaktiven Computing

Der digitale elektronische Rechner wurde als Technologie zur Automatisierung der umfangreichen Rechenarbeit entwickelt, die seit der Französischen Revolution kooperativ – in einer hochentwickelten Form der Arbeitsteilung («menschlicher Computer») – erledigt worden war. Der daraus resultierende digitale elektronische Computer war kaum mehr als eine große automatische Rechenmaschine. An seiner Entwicklung waren Wissenschaftler und Techniker beteiligt, denen die neue Rechenmaschine als Ausrüstung der eigenen Arbeit (Krypotologie und Waffenkonstruktion) diente. In technologischer Hinsicht hat sich diese Entwicklung als Sackgasse herausgestellt, von der bis auf die rechnerische Stapelverarbeitung – etwa bei Lohn- und Gehaltsabrechnungen, Steuerberechnungen und bei bestimmten wissenschaftlichen Berechnungen – nur wenig relevant geblieben ist.

Was man heute normalerweise unter «Computing» versteht, nämlich «Personal Computing», entwickelte sich ursprünglich als Technologie zur Vereinfachung großangelegter kooperativer Arbeit (zuerst im Bereich der Luftverteidigung, später der Flugsicherung und der Flugreservierung), deren Koordination zu komplex war, um mit herkömmlichen manuellen und mechanischen Mitteln bewältigt zu werden. Später verzweigte sich diese Technologie als «Interactive Computing» in alle Richtungen. Es entstanden daraus interaktive Mensch-Maschine-Systeme, etwa Laptops, Smartphones, aber auch «eingebettete» Datenverarbeitungssysteme zur Steuerung von Maschinenanlagen wie Monitorarbeitsplätze, Automotoren und Waschmaschinen, wo die Computerkomponente mit mechanischen oder anderen Einheiten in der Umgebung «interagiert».

Wichtige Paradigmen für Anwendungen des interaktiven Computing weisen in ihrer Entwicklung bemerkenswerte Parallelen auf: Sie gehen auf Praktiker zurück, entstanden als praktische Techniken zum Eigengebrauch oder für die Verwendung durch Kollegen und wurden erst später verallgemeinert.[1]

1 Eine ausführlichere Darstellung und Begründung der hier in medienhistorischer Hinsicht entwickelten Perspektive wurde 2011 unter dem Titel *Cooperative Work and Coordinative Practices. Contributions to the Conceptual Foundations of Computer-Supported Cooperative Work (CSCW)* veröffentlicht. Die Forschung wurde im Rahmen des Projekts «Computational Artifacts» von der Velux Foundation unterstützt.

Die praktischen Ursprünge der Computertechnologien

Anders als die Kernenergie oder die GPS-Navigation entstammen Computertechnologien keinem spezifischen wissenschaftlich-theoretischen Korpus.[2] Sie wurden auch nicht, wie die Funktechnik, zusammen mit einer wissenschaftlichen Theorie entwickelt. Natürlich hing ihre Entwicklung von einer Vielzahl mathematischer Theorien ab (der booleschen Algebra, Shannons Informationstheorie, Rekursionstheorie usw.), sie waren jedoch nicht das Ergebnis der Anwendung eines bestimmten Theoriekorpus. So hat der bedeutende Computerhistoriker Michael Mahoney[3] darauf hingewiesen, dass die Informatik

> eher die Form einer Familie lose miteinander verbundener Forschungsprogramme [hat] als die einer kohärenten, durch empirische Resultate bestätigten allgemeinen Theorie. Bislang hat sich noch keines der mathematischen Modelle als der Diversität der Datenverarbeitung adäquat erwiesen, und außerdem hängen die verschiedenen Modelle nicht wirklich zusammen. Welche Mathematik man verwendete, war abhängig von der jeweiligen Fragestellung, und bei manchen Fragen gab es keine Mathematik, die als theoretische Begründung für das in der Praxis Geleistete hätte fungieren können.[4]

«*Der* Computer» wurde genau genommen nicht «erfunden». In der Formulierung von Mahoney heißt es: «Auch wenn von anderen Technologien behauptet werden kann, sie hätten ein jeweils spezifisches Wesen und seien für eine bestimmte Wirkung geplant, so gilt das nicht für den Computer. Oder vielmehr: Die Natur des Computers ist proteisch».[5] Laut Mahoney gab es daher lange Zeit «auf die Frage ‹Was ist ein Computer oder wie soll er sein?› keine eindeutige Antwort». Computer und Datenverarbeitung erhielten erst im Lauf eines jahrzehntelangen Prozesses ohne absehbares Ende «ihre moderne Gestalt».[6] Und es gibt keinen Grund anzunehmen, dass der Begriff des Computing sich gefestigt und stabilisiert hätte: Solange die Formbarkeit «des Computers» in alle denkbaren Richtungen erforscht wird, ist das letzte Wort noch nicht gesprochen. Es wäre falsch, von «dem Computer» als *einer* Technologie zu sprechen. Er ist eine proteische Technologie, deren Entwicklung in einem viel radikaleren Sinn durch die *praktischen Anwendungen* bestimmt wird, als das bei anderen Technologien der Fall ist.

Computertechnologien entstanden aus den Koordinationsproblemen bei der Organisation vernetzter Arbeit in der modernen industriellen Zivilisation. Aus der Entwicklung der Industriegesellschaft ergab sich beispielsweise massiver Rechenbedarf bei der Erstellung astronomischer Tabellen für die Seefahrt. Ungenauigkeiten der Tabellen konnten zum Verlust von Schiffen, Menschen und Ladungen führen. Andere Formen der Massenkalkulation, etwa Logarithmentafeln, erlangten ähnlich große Bedeutung in allen Industriebereichen, im Handel und für die Kriegführung. Nach der Französischen Revolution begannen Techniker und Wirtschaftsleute wie Gaspard de Prony, umfangreiche Rechenarbeiten nach dem Muster der für die frühe industrielle Organisation

2 Der Mythos, die Computertechnologien hätten ihren Ursprung in einem bestimmten mathematischen Theoriekorpus, findet sich verbreitet in Berichten zur Geschichte der Informatik, z. B. bei Martin Davis: *The Universal Computer. The Road from Leibniz to Turing*. New York 2000 und George Dyson: *Turing's Cathedral. The Origins of the Digital Universe*, London 2012.

3 Mittlerweile ist eine gute, von Thomas Haigh herausgegebene Aufsatzsammlung von Mahoney verfügbar: Michael Sean Mahoney: *Histories of Computing*. Cambridge, Mass., London 2011.

4 Michael S. Mahoney: Computers and mathematics: The search for a discipline of computer science, in: J. Echeverria u. a. (Hg.): *The Space of Mathematics. Philosophical, Epistemological, and Historical Explorations*, Berlin, New York 1992, 349–363, hier 361.

5 Michael S. Mahoney: The histories of computing(s), in: *Interdisciplinary Science Reviews*, Vol. 30, Nr. 2, 2005, 119–135, hier 122.

6 Mahoney: Computers and mathematics, 349.

charakteristischen Arbeitsteilung der «Manufakturen» zu organisieren.[7] Mit der Entfaltung der industriellen Zivilisation und ihrer immer weitergehenden sozialen Arbeitsteilung wurden Berechnungen in zunehmend größerem Maß durch kooperative Arbeit bewältigt: Die Buchhaltung und Gehaltsabrechnung in Industriebetrieben, die Massenkalkulationen in Versicherungen und Geldinstituten erforderten nach einigen Jahrzehnten ganze Armeen «menschlicher Computer»:[8] auf routinisierte Rechenarbeit spezialisiertes Verwaltungspersonal nach dem Muster industrieller Arbeitsteilung.

Mitte des 20. Jahrhunderts war die für eine industrielle Kriegsführung erforderliche Rechenarbeit so immens geworden, dass bei der Entwicklung neuer Waffentypen wie Nuklearwaffen die Kosten für die Koordination der kooperativ durch «menschliche Computer» durchgeführten Massenkalkulationsarbeit den Nutzen dieser Organisationsform trotz der Verwendung von Tisch- und Lochkartenrechnern überstiegen. «In den 1940er Jahren […] wurde das Ausmaß mancher von Physikern und Technikern benötigten Berechnungen so groß, dass die Arbeit nicht mehr mit Tischrechenmaschinen zu bewältigen war. Die Entwicklung schneller Großrechner wurde zu einem dringenden Problem.»[9] Das war die vorrangige Motivation für die Entwicklung der ersten elektronischen digitalen Computer. Alan Turing formulierte das in einem Ende 1945 verfassten Bericht als Aufgabe, die «menschliche Bremse» aus dem Prozess der Berechnung zu eliminieren:

> Es darf keine Arbeit mehr für das Herausnehmen von Material in die Maschine und dessen Wiedereinsetzen im richtigen Moment erforderlich sein. Um all das muss sich die Maschine selbst kümmern.[10]

Die frühen elektronischen Computer waren also kaum mehr als große, sehr schnelle Rechenmaschinen, die bei entsprechender manueller oder automatischer Einstellung eine ganze Kalkulationsaufgabe ausführten, die bis dahin für eine große Anzahl hochspezialisierter menschlicher Computer zerlegt worden war. Als in den 1950er Jahren die kommerzielle Nutzung dieser Technologie begann, wurde sie weiterhin zur Stapelverarbeitung eingesetzt. Der einzige Unterschied lag darin, dass Geschäftsanwendungen wie die Gehaltsabrechnung die Verarbeitung riesiger Datenmengen beinhalteten. Wie bei der Computernutzung für automatisierte Massenkalkulationen wurde die speicherprogrammierte Datenverarbeitung auch in der Verwaltung als Substitutionstechnologie eingesetzt: zur Automatisierung der Arbeit von zentralen Datenverarbeitungsabteilungen großer Organisationen, und zum Ersatz ganzer Batterien von Lochkartenrechnern durch einen einzigen Rechner, etwa den IBM 1401. Die meisten im Handel erhältlichen und bei den Behörden während der 1950er und 1960er Jahre installierten Computersysteme waren «einfach die elektronischen Äquivalente der Lochkartenrechenmaschinen, die sie ersetzten».[11] Ihre tatsächliche Planung und Nutzung war also trotz eines Anspruchs auf «Universalität» strikt auf das Eliminieren der «menschlichen Bremse» beschränkt, wie Turing das nannte.

7 Gaspard F. C. M. Riche de Prony: Notice sur les Grandes Tables Logarithmique &c., in: *Recueil des Discourses lus dans la séance publique de L'Academie Royale Des Sciences*, dort datiert Paris 7. Juni 1824, http://sites.google.com/site/babbagedifferenceengine/baron-de-prony's-description-of-the-construct, gesehen am 4.2.2015; Ivor Grattan-Guinness: Work for the hairdressers. The production of de Prony's logarithmic and trigonometric tables, in: *IEEE Annals of the History of Computing*, Vol. 12, Nr. 3, 1990, 177–185; ders.: The computation factory. De Prony's project for making tables in the 1790s, in: Martin Campbell-Kelly u. a. (Hg.): *The History of Mathematical Tables. From Sumer to Spreadsheets*, Oxford 2003, 105–122 [Reprint 2007].

8 David Alan Grier: *When Computers Were Human*, Princeton, Oxford 2005.

9 B. Jack Copeland: Colossus and the rise of the modern computer, in: ders. (Hg.): *Colossus. The Secrets of Bletchley Park's Codebreaking Computers*, Oxford 2006, 101–115, hier 102.

10 Alan M. Turing: Proposed electronic calculator, in: B. J. Copeland (Hg.): *Alan Turing's Automatic Computing Engine. The Master Codebreaker's Struggle to Build the Modern Computer*, Oxford 2005, 369–454, hier 371.

11 Martin Campbell-Kelly, William Aspray: *Computer. A History of the Information Machine*, New York 1996, 157.

Die praktischen Ursprünge des interaktiven Computing

Parallel zur Entwicklung automatischer Berechnungstechnologien, die das Ende der «menschliche[n] Bremse» herbeiführen sollten, entstand eine völlig anders geartete Computertechnologie, deren Zweck nicht in der Automation lag, sondern darin, dass der Rechner Arbeitskräfte bei der Durchführung und Koordination ihrer Arbeit unterstützte. Der Prototyp dieses neuen Computing-Paradigmas wurde unter dem Namen Whirlwind bekannt.[12]

Die Motivation war deutlich anders gelagert als bei der Automatisierung von Rechenarbeit. Im August 1949 zündete die Sowjetunion ihre erste Atombombe, und da dieses Land auch über Kampfflugzeuge zum Abwurf solcher Waffen in den USA verfügte, befanden sich die US-amerikanischen Behörden auf einen Schlag in der Lage eines unzulänglichen Luftverteidigungssystems. Insbesondere die Koordinationskapazität der vorhandenen Systeme war mehr als begrenzt. Judy O'Neill erläutert das Problem wie folgt:

> Im bestehenden System, dem «manuellen System», beobachtete ein Bediener einen Radarbildschirm und schätzte Höhe, Geschwindigkeit und Richtung der durch die Radarantennen erfassten Flugzeuge. Der Bediener überprüfte das erfasste Flugzeug anhand der bekannten Flugrouten und anderer Informationen. Konnte es nicht identifiziert werden, brachte der Bediener Abfangjäger in Angriffsposition. Verließ das Flugzeug den auf dem Radarbildschirm des Bedieners erfassten Bereich, so musste innerhalb weniger Augenblicke die «Übergabe» oder Übertragung der Richtungsdaten des Luftkampfes an den entsprechenden Bediener in einem anderen Sektor erfolgen.[13]

Diese vernetzte Arbeitsorganisation, das «manuelle System», war nicht dafür geschaffen, eine große Anzahl von Flugzeugen mit interkontinentaler Reichweite und möglichen Nuklearwaffen zu bewältigen. Eine radikale Transformation der Organisation der Luftverteidigung war unumgänglich. Schließlich entschied man sich für die Konstruktion eines völlig neuen Luftverteidigungssystems. Das System wurde Semi-Automatic Ground Environment (SAGE) genannt und war auf 23 Zentren in den ganzen USA verteilt. Diesen Zentren wurde die Verantwortung für die Luftraumüberwachung des jeweiligen Sektors und für die potentielle Leitung und Koordination der militärischen Reaktion übertragen. Die Arbeit der einzelnen Zentren sollte durch eine schnelle elektronische digitale Datenverarbeitungsmaschine unterstützt werden, die über ein System von hunderten Fernsprechleitungen Daten von Radarstationen empfangen und verarbeiten konnte, und zwar «in Echtzeit», also entsprechend den gemeldeten Zustandsänderungen. Die Entwicklungsaufgabe wurde 1950 einem Team am Labor für Regelsysteme des MIT unter der Leitung von Jay W. Forrester übertragen, das bereits einen digitalen Flugsimulator entwickelt hatte und damit eine gewisse Erfahrung mit Problemen des Echtzeit-Computing besaß.[14]

Als Grundlage für die Entwicklungsarbeit am Whirlwind-Projekt diente das Paradigma der Speicherprogrammierung, allerdings auf radikal modifizierte

12 Zu Whirlwind siehe z. B. Judy Elizabeth O'Neill: *The Evolution of Interactive Computing through Time-sharing and Networking*, Ph.D. dissertation, University of Minnesota 1992, sowie Campbell-Kelly, Aspray: *Computer*. Eine Darstellung mit Schwerpunkt auf dem organisatorischen Aspekt findet sich in Kent C. Redmond, Thomas M. Smith: *Project Whirlwind. The History of a Pioneer Computer*, Bedford, Mass. 1980 sowie dies.: *From Whirlwind to MITRE. The R&D Story of the SAGE Airdefense Computer*, Cambridge, Mass., London 2000.

13 O'Neill: *The Evolution of Interactive Computing*, 15.

14 Perry Orson Crawford: Application of digital computation involving continuous input and output variables [5.8.1946], in: Martin Campbell-Kelly, M. R. Williams (Hg.): *The Moore School Lectures. Theory and Techniques for Design of Electronic Digital Computers*, Cambridge, Mass. 1985, 374–392 [überarbeitete Ausgabe von *Theory and Techniques for Design of Electronic Computers*, 1947–1948].

Weise. Während der EDVAC und andere speicherprogrammierte Computer für automatische Zahlenverarbeitung im Stapelmodus konzipiert waren und nicht mit der Umgebung interagierten (auch nicht mit dem Bediener), solange nicht der vorher festgelegte Endzustand erreicht war, wurde Whirlwind für die Verarbeitung von Daten während einer fortlaufenden Interaktion mit der Umgebung (Radardaten, Bedieneranweisungen, Zielverfolgung usw.) geplant. Die Geschwindigkeit und Zuverlässigkeit des Computerspeichers wurde dementsprechend zu einer großen technischen Herausforderung. Beträchtliche Energie wurde auf die Entwicklung einer neuen Technologie von Magnetkernspeichern auf der Basis eines Netzes aus winzigen magnetischen Keramikringen verwendet. Durch die erst im Sommer 1953 funktionstüchtige Kernspeichertechnologie wurde Whirlwind der «bei weitem schnellste und zuverlässigste Computer der Welt».[15] Zur Beschleunigung der Betriebsgeschwindigkeit war die Architektur der zugrunde liegenden Schaltkreise für Parallelverarbeitung – «synchrone parallele Logik» – konzipiert.[16]

Whirlwind war zugleich «mit Abstand führend bei den optischen Anzeigemöglichkeiten», die u. a. die «grafische Darstellung der errechneten Resultate auf Luftraumkarten» erleichterten.[17] Als Korrelat gab es eine «Lightgun», mit der die Bediener Objekte auswählen und auf das Display schreiben konnten: «Aufgrund dieser beiden Besonderheiten wurde direkte und simultane Mensch-Maschine-Interaktion möglich […].»[18] Whirlwind und seine Nachfolgemodelle, der Cape Cod- und SAGE-Computer, schufen und definierten ein neues technologisches Paradigma – die sogenannten computerbasierten Echtzeit-Transaktionsverarbeitungssysteme.[19] Diese Technologie erleichterte Arbeitskräften insofern die Zusammenarbeit, als das System ihnen einen gemeinsamen Arbeitsbereich in Form eines Datensatzes oder einer anderen (möglicherweise mit nicht-digitalen Anlagen gekoppelten) digitalen Darstellung bot und sie damit in die Lage versetzte, durch die Änderung des Datenzustandes auf strikt vorgegebene Weise miteinander zu kooperieren:

> Online-Transaktionsverarbeitungssysteme waren so programmiert, dass an Endgeräten Eingaben getätigt werden konnten, der zentrale Prozessor die Berechnung durchführte und die Ausgabe in relativ kurzer Zeit wiederum am Endgerät angezeigt wurde. […] Die Dienste, die angefordert werden konnten, waren auf die bereits im System vorprogrammierten beschränkt.[20]

Die Bediener in den verschiedenen Zentren und auf nationaler Ebene waren insofern in kooperative Arbeit involviert, als ihre individuellen Tätigkeiten im SAGE-System und durch dessen Netzwerk aus Radarstationen, Kommunikationsleitungen, Ortungsgeräten und Übergabeprozeduren mit ihren Kollegen gekoppelt waren. Der SAGE-Computer war zur Unterstützung von Arbeitskräften bei stark routinisierter Verarbeitung (in Echtzeit) der riesigen vom Netzwerk generierten Transaktionsmengen konzipiert. Die Unterstützung kooperativer Arbeit und Koordination blieb daher ziemlich rudimentär.

15 Campbell-Kelly, Aspray: *Computer*, 167.
16 Redmond, Smith: *Project Whirlwind*, 217.
17 Ebd., 216.
18 Ebd.
19 Vgl. O'Neill: *The Evolution of Interactive Computing*, 22.
20 Ebd., 22–23.

Angesichts der untilgbaren Kontingenz der Anwendungen (der grundsätzlichen Ungewissheit der Lage am Himmel, der oft zweifelhaften Zuverlässigkeit von Radardaten und Übertragungsleitungen) und der extrem sicherheitsrelevanten Art der Arbeit (nämlich der Verhinderung eines atomaren Angriffs) war es von Anfang an Teil der Planungsanforderungen des Whirlwind-Projekts, die Improvisationsfähigkeit der Arbeitskräfte zu unterstützen. Das SAGE-System war nicht dafür konzipiert, die «menschliche Bremse» zu eliminieren, sondern dafür, eine Kooperation der Bediener bei der Erfüllung jeder Aufgabe zu ermöglichen, die nicht mehr manuell koordiniert werden konnte. SAGE wurde also von Anfang an als «halbautomatisches» System geplant. Dementsprechend waren der Cape Cod-Prototyp und das endgültige SAGE-Computersystem mit Absicht für interaktives Computing geplant, samt Echtzeit-Computing, grafischen CRT-Displays, Handgeräten für die Auswahl (Lightguns, Joysticks) und direkter Manipulation.

Ich betone diesen Aspekt, weil es dieses Planungsmerkmal war, das letzten Endes die technische und konzeptionelle Grundlage für das moderne interaktive Computing bildete – und weil die Historiker der Datenverarbeitungstechnologie es oft übersehen. Die Historiker Paul Edwards beispielsweise behauptet, dass SAGE für die «Automatisierung» der Luftverteidigung konzipiert war und diese Hoffnung enttäuschte:

> Die von SAGE versprochene *automatische* Kontrolle war und ist bis heute weitgehend eine Illusion. Ein Großteil der gesamten Aufgabe verblieb – unabhängig vom Potenzial der Computer und ihrer Programme – bei den menschlichen Bedienern und ihrer Organisation. Alle Versuche, diesen Teil der Arbeit – in Form formaler, in Handbüchern verschlüsselter Abläufe – zu «programmieren», kamen aufgrund der unbändigen, noch nicht in das System integrierten Komplexitäten ins Stocken.[21]

Laut Edwards zeigte SAGE auf diese Weise die «Unmöglichkeit, jede denkbare Situation im Vorhinein zu berücksichtigen oder auch nur zu artikulieren, und damit die Notwendigkeit, auf das menschliche Urteil zurückzugreifen».[22] Er bringt keinen Beleg für diese Behauptung (abgesehen von einer *späteren* «Kritik der Computertechnologie»[23]). Dabei wurde SAGE zur Unterstützung des «menschlichen Urteils» geschaffen; und es war kaum überraschend, dass «ein Großteil der gesamten Aufgabe» bei den «menschlichen Bedienern und ihrer Organisation» verblieb. So war es im Planungsprozess angelegt. Eine Zusammenfassung der Planungsgrundsätze des Cape Cod-Prototyps lautete:

> [Der geplante operative Ablauf] ist jedoch nicht zur Gänze automatisch. Das Auslösen einer neuen Verfolgung erfolgt automatisch oder manuell, oder es können beide Methoden verwendet werden, jeweils in unterschiedlichen geografischen Bereichen des Systems. Auch die Entscheidung, bei welchen Flugzeugbahnen es sich um Ziele und bei welchen es sich um Abfangjäger handelt, wird durch Menschen gefällt und manuell mittels der Lightgun in das Gerät eingegeben. […] Die von der Maschine durchgeführten Aktionen werden durch manuelle Betätigung eines Auswahlschalters

21 Paul N. Edwards: *The Closed World. Computers and the Politics of Discourse in Cold War America*. Cambridge, Mass., London 1996, 108.
22 Ebd., 109.
23 Ebd., 109, 392 f.

> definiert, die der Computer automatisch erfasst und interpretiert (beispielsweise «Dieses Flugzeug als Abfangjäger behandeln»). Die Menschen treffen Entscheidungen und improvisieren, der Computer übernimmt, von ihnen beaufsichtigt, die Routineaufgaben. Zur Erleichterung der Beaufsichtigung durch Menschen ist ein schnelles, flexibles Anzeigesystem erforderlich.[24]

Die Grenzen dieser Flexibilität waren im Whirlwind-Paradigma jedoch von Anfang an deutlich. Beispielsweise war den Planern klar, dass die Bediener in manchen Situationen auf andere Weise mit ihren Kollegen (vielleicht in einem anderen Zentrum) interagieren können müssten, als dies im System vorprogrammiert war. So wurde im Sommer 1953 ein Telefonsystem installiert, über das Bediener an unterschiedlichen Standorten einfach persönlich miteinander sprechen konnten.[25] Das Rechensystem unterstützte die Bediener also nicht beim Entwickeln von Koordinationspraktiken, und Unvorhergesehenes musste außerhalb des Rechensystems gehandhabt werden – in gewöhnlichen Gesprächen unter den Bedienern eines Zentrums oder in Telefonaten zwischen den Bedienern mehrerer Zentren.

Die Lehrjahre des interaktiven Computing

Bedenkt man die spätere Ubiquität des interaktiven Computing, so scheint es verblüffend, dass es nach dem enormen Aufwand des Whirlwind-Projekts auf spezielle Bereiche der Online-Massentransaktionsverarbeitung, wie Flugsicherung und Flugreservierung, begrenzt blieb. Die Gründe dafür liegen jedoch auf der Hand. Rechenleistung war extrem teuer. Und das blieb so bis in die 1980er Jahre, bis die Entwicklung von integrierten Schaltkreisen und Mikroprozessoren für Digitaluhren und Tischrechner die Massenproduktion von Rechnern wirtschaftlich tragbar gemacht hatte.[26] Interaktives Computing blieb bis dahin auf jene Bereiche beschränkt, in denen die Kosten dieser Art von Computerausstattung gerechtfertigt waren. Anderswo mussten Computersysteme aufgrund ihrer Kosten bei fast voller Kapazitätsauslastung arbeiten. Folglich arbeiteten die meisten der wenigen Computer in den 1960er Jahren im Stapelverarbeitungsmodus einen Auftrag nach dem anderen ab, oder wie J. C. R. Licklider treffend formuliert hat, «der übliche Betriebsmodus in Rechenzentren verlief nach dem Muster der Reinigungsfirma um die Ecke (‹in by ten, out by five›)».[27]

Auf diese Weise wurde das Wachstum des interaktiven Computing so gut wie blockiert. Das Whirlwind-Paradigma starb jedoch nicht aus, sondern hinterließ eine interaktive Computing-«Kultur», die in vagen Begriffen wie «Mensch-Computer-Symbiose»,[28] «Erweiterungssystem» des «menschlichen Intellekts»[29] und «Mensch-Maschine-System»[30] zum Ausdruck kam und auf jene Praxisgemeinschaft zurückgehen, die durch das Paradigma geschaffen worden war.

Erstens gab es eine unmittelbare personelle Kontinuität: «Die Menschen, die an den beiden Projekten beteiligt waren, hatten die Möglichkeiten der Interaktion zwischen Mensch und Computer zu schätzen gelernt. Dieses Verständnis

24 C. Robert Wieser: Cape Cod System and demonstration [13.3.1953], Cambridge, Mass., MIT Lincoln Laboratory, Division 6, 2. – Project Whirlwind Limited Memorandum VI - L-86. http://dome.mit.edu/handle/1721.3/41510, gesehen am 6.2.2015.

25 Vgl. Redmond, Smith: *From Whirlwind to MITRE*, 310 f.

26 Vgl. z. B. Ernest Braun, Stuart Macdonald: *Revolution in Miniature. The History and Impact of Semiconductor Electronics*, Cambridge 1978 [Taschenbuchausgabe 1980, 2. Auflage 1982]; Michael Riordan, Lillian Hoddeson: *Crystal Fire. The Invention of the Transistor and the Birth of the Information Age*. New York, London 1997 [Taschenbuchausgabe 1998]; John Orton: *Semiconductors and the Information Revolution. Magic Crystals that Made IT Happen*, Amsterdam u. a. 2009.

27 Joseph Carl Robnett Licklider, Welden E. Clark: On-line man-computer communication, in: G. A. Barnard (Hg.): *SJCC'62. Proceedings of the Spring Joint Computer Conference, 1–3 May 1962, San Francisco, California*, 1962, Vol. 21, 1962, 113–128, hier 114.

28 Joseph Carl Robnett Licklider: Man-computer symbiosis (IRE Transactions on Human Factors in Electronics, March 1960), in: R. W. Taylor (Hg.): *In Memoriam: J. C. R. Licklider, 1915–1990*, Palo Alto, Cal. 1990, 4–11. Ebenfalls abgedruckt in Adele Goldberg: *A History of Personal Workstations*, Reading, Mass. 1988.

29 Douglas C. Engelbart: Augmenting human intellect. A conceptual framework (Prepared for: Director of Information Sciences, Air Force Office of Scientific Research, Washington 25, D.C., Contract AF49(638)-1024), Menlo Park, Cal. Oktober 1962. – AFOSR-3233. http://www.dougengelbart.org/pubs/augment-3906.html, gesehen am 6.2.2015.

30 Ivan Edward Sutherland: Sketchpad. A man-machine graphical communication system, in: E. C. Johnson (Hg.): *SJCC'63. Proceedings of the Spring Joint Computer Conference, 21–23 May 1963, Detroit, Michigan*, Vol. 23, Santa Monica, Cal. 1963, 329–346.

ermöglichte die Weiterentwicklung des interaktiven Computing, zusammen mit Timesharing und Vernetzung».[31] Darüberhinaus besaßen das Whirlwind-Projekt und die nachfolgenden SAGE-Forschungs- und Entwicklungsprojekte gewaltige Ausmaße. 1959 war die Hälfte aller ausgebildeten Programmierer in den USA mit der Entwicklung von Software für das SAGE-System beschäftigt.[32]

Zweitens wurde der ursprüngliche Whirlwind-Prototyp nach dem Ende des Whirlwind-Projekts im Jahr 1953 nicht verschrottet, sondern blieb auf dem MIT-Campus verfügbar. Außerdem wurden zur Weiterentwicklung von SAGE auf der Whirlwind-Architektur basierende experimentelle transistorisierte Computer konstruiert. Auch diese Computer wurden 1958 dem MIT-Campus überlassen und intensiv von Studierenden höherer Semester und Forschern genutzt: «Die Haltung der Leute, die den Whirlwind-Computer und diese Testcomputer verwendeten, sollte eine ‹Kultur› des interaktiven Computing entstehen lassen, in der Computer als Partner des kreativen Denkens von Menschen betrachtet wurden.»[33] Fernando Corbató z. B., der später eine zentrale Rolle bei der Entwicklung des Timesharing spielte, erinnert sich, dass «viele von uns [am MIT] unsere Erfahrungen am Whirlwind gesammelt hatten. Whirlwind war in mancher Hinsicht ein Gerät wie ein großer Personal Computer, auch wenn die Anlage aus Effizienzgründen nicht auf den Einbau von Stapelverarbeitungen verzichten konnte. Wir hatten Bildschirme. Wir hatten Schreibmaschinen und wussten irgendwie, was es bedeutete, mit einem Computer zu interagieren, und erinnerten uns daran.»[34]

Drittens konnte das eher restriktive ökonomische Regime, das Licklider so eindringlich beschreibt («in by ten, out by five») zwar bei administrativen Abläufen (wie Gehaltsabrechnungen und anderen Kalkulationen) toleriert werden, die seit der Einführung von Lochkartenrechnern nach Grundsätzen der Stapelverarbeitung organisiert waren. Dieses Regime schloss aber alle Anwendungen aus, die hochfrequente oder Echtzeitinteraktionen zwischen dem Benutzer und digitalen Darstellungen erfordern. Das «In by ten, out by five»-Regime machte Programmieren und insbesondere dessen Fehlerbehebung zu einer lähmenden Beschäftigung. Diese Situation motivierte Computertechniker, sich alternative Funktionsweisen für die eigene Arbeit auszudenken. Wie O'Neill beschrieben hat, waren Forscher am MIT, die ihre Erfahrungen mit Whirlwind gesammelt hatten und interaktives Computing aus erster Hand kannten, «nicht mehr willens, die Stapelmethode der Computernutzung für ihre Arbeit zu akzeptieren», und «suchten nach Möglichkeiten einer interaktiven Nutzung von Computern»; damals wäre es jedoch «zu kostspielig gewesen, jeden Benutzer mit einem eigenen Computer auszustatten».[35] So wurde um das Jahr 1960 die Idee verfolgt, mit einem zentralen Computersystem mehrere Benutzer «simultan» zu versorgen. Die Lösung war, mit den Worten eines ihrer Urheber, John McCarthy, ein «Betriebssystem», das «jedem Benutzer ständigen Zugriff auf die Maschine gewährte» und ihm erlaubte, «sich so zu benehmen, als habe er die alleinige Kontrolle über einen Computer».[36] Das erste funktionierende Betriebssystem dieser

31 O'Neill: *The Evolution of Interactive Computing*, 11.

32 Vgl. Martin Campbell-Kelly: *From Airline Reservations to Sonic the Hedgehog. A History of Software Industry*, Cambridge, Mass., London 2003. Siehe auch Claude Baum: *The System Builders. The Story of SDC*. Santa Monica, Cal. 1981.

33 O'Neill: *The Evolution of Interactive Computing*, 24.

34 Fernando J. Corbató: Oral-History-Interview. Interviewt von A. L. Norberg, 18 .4.1989, 14.11.1990, in Cambridge, Mass. Interview-Transkript, Charles Babbage Institute, University of Minnesota, Minneapolis (OH 162), 0–255, 14.

35 O'Neill: *The Evolution of Interactive Computing*, 44.

36 John McCarthy: Reminiscences on the history of time sharing. Stanford University, in: http://www-formal.stanford.edu/jmc/history/timesharing/timesharing.html, dort datiert Winter/Frühling 1983, gesehen am 6.2.2015.

Art war anscheinend das Compatible Time-Sharing System (CTSS). Es wurde durch ein Team unter der Leitung von Fernando Corbató am MIT geschaffen und 1961 erstmals vorgeführt.[37] Die verschiedenen Benutzer waren über Terminals mit dem «Host»-Computer verbunden, und jeder hatte Zugriff auf die Rechenleistung des «Host», als wäre er der alleinige Benutzer.

Der vierte und im Nachhinein – jedenfalls in konzeptueller Hinsicht – wichtigste Punkt war die beharrliche Forschungsarbeit von Douglas Engelbart an der Weiterentwicklung des interaktiven Computing-Paradigmas mit Blick auf die «Erweiterung des menschlichen Intellekts» mithilfe der Computing-Technologie.[38] Eine bedeutende Etappe dieser Arbeit war 1968 erreicht, als Engelbart und seine Kollegen zeigen konnten, dass viele der Technologien, die wir heute als wesentliche Komponenten des interaktiven Computing ansehen (direkte Manipulation, Bildschirme mit Rastergrafik, Maus, Nachrichtenübermittlung usw.), mit einer Architektur, die uneingeschränkten Zugriff auf *mehrere* Anwendungsprogramme bot, auf *integrierte* Weise realisiert werden konnten.[39]

Die langwierige und enorm teure Forschung, die in den Bau des Whirlwind und seines Nachfolgers einging, stellte die grundlegenden Prinzipien und Techniken des interaktiven Computing-Paradigmas bereit. Das Paradigma war jedoch auf wenige Computerlabors und die Erinnerungen und Ambitionen der Whirlwind- und SAGE-Veteranen beschränkt. Die Kontinuität des Paradigmas war daher labil. Tatsächlich brauchte es eine Generation, bis die Technologien des interaktiven Computing für Arbeitskräfte jenseits der kleinen Gruppe von Computerwissenschaftlern und des Personals in bestimmten zeitkritischen Arbeitsbereichen, etwa der Luftverteidigung und Flugreservierung, zur praktischen Realität wurden.

Das interaktive Computing wird praktische Realität

Als Engelbart in der zweiten Hälfte der 1960er Jahre seine experimentellen Arbeiten verfolgte, steckten die integrierten Schaltkreise noch in den Kinderschuhen. Die Computerplattform, die seinem Labor zu Beginn der Experimente zur Verfügung stand, bestand aus ein paar Minirechnern. 1967 jedoch konnte er einen ersten Timesharing-Computer anschaffen, dieser kostete über 500.000 US-Dollar.[40] Erst die Verfügbarkeit immer leistungsstärkerer integrierter Schaltkreise seit Anfang der 1970er Jahre führte zu einer radikalen Veränderung der dem Computer zugrunde liegenden Technologie. Das interaktive Computing profitierte von dieser Erleichterung. Der erste Schritt erfolgte 1972, als Techniker am Xerox Palo Alto Research Center (PARC), die auf der Straßenseite gegenüber Engelbarts Labor am Stanford Research Institute arbeiteten und sich von dessen Tätigkeit inspirieren ließen, etwas zu bauen begannen, was sie «Personal Computer» nannten: den Xerox Alto.[41] Der Alto kam 1974 heraus, nur sieben Jahre, nachdem Engelbart 500.000 US-Dollar für einen Timesharing-fähigen Computer ausgegeben hatte. Die Leistung entsprach der

37 Vgl. Fernando J. Corbató, Marjorie Merwin-Daggett, Robert C. Daley: An experimental time-sharing system, in: Barnard (Hg.): *SJCC'62*, 335–344.

38 Engelbart: *Augmenting human intellect*. Siehe auch Thierry Bardini: *Bootstrapping. Douglas Engelbart, Coevolution, and the Origins of Personal Computing*, Stanford, Cal. 2000.

39 Vgl. Douglas C. Engelbart, William K. English: A research center for augmenting human intellect, in: *FJCC'68. Proceedings of the Fall Joint Computing Conference, 9–11 December 1968*, San Francisco, Cal. Teil 1, Vol. 33, New York 1968, 395–410. Ebenfalls abgedruckt in I. Greif (Hg.): *Computer-Supported Cooperative Work. A Book of Readings*, San Mateo, Cal. 1988, 81–105.

40 Vgl. Bardini: *Bootstrapping*, 123, 251.

41 Vgl. Butler W. Lampson: *Why Alto*, in: XEROX Inter-Office Memorandum, Palo Alto, Cal., dort datiert 19.12.1972, http://research.microsoft.com/en-us/um/people/blampson/38a-whyalto/Acrobat.pdf, gesehen am 6.2.2015; Diana Merry, Ed McCreight, Bob Sproull: *ALTO. A Personal Computer System Hardware Manual*, Palo Alto, Cal. 1976 [Version 23; erste Version 1975]; Charles P. Thacker u. a.: *Alto. A personal computer*, in: Xerox PARC, dort datiert 7.8.1979. – CSL-79-11 [Reprint Februar 1984], http://research.microsoft.com/en-us/um/people/blampson/25-Alto/25-Alto.pdf, gesehen am 10.2.2015.

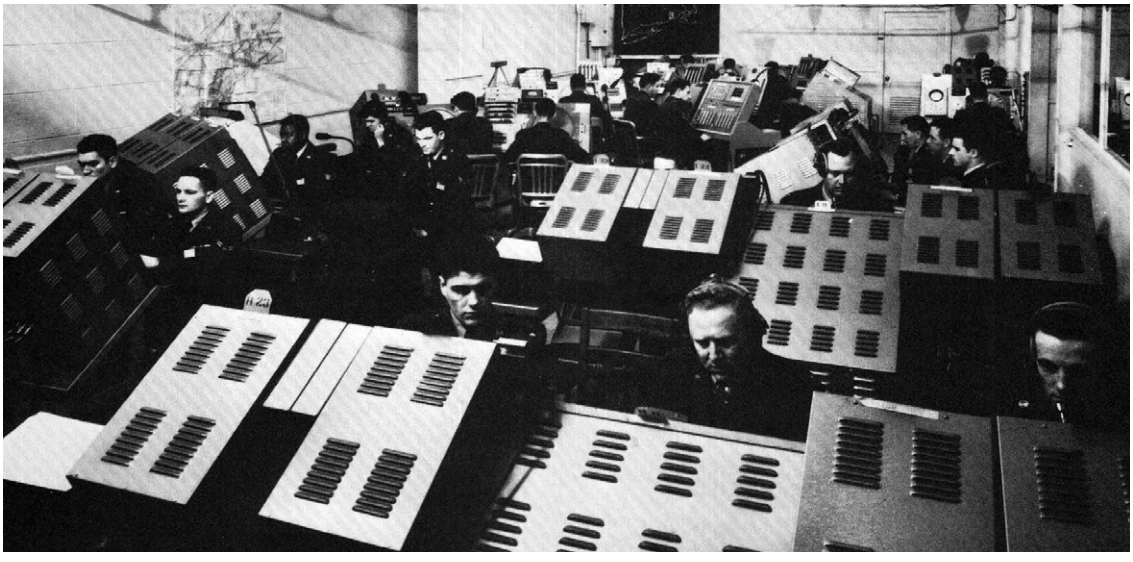

Abb. 1 **Prototyp eines Monitorraums zur Steuerung von Abfang-Operationen, SAGE-Leitungszentrum im Barta Building des Massachusetts Institute of Technology, 1953**

eines Minirechners, die Kosten waren jedoch auf rund 12.000 US-Dollar geschrumpft. Das lag zwar immer noch deutlich über der unmittelbaren kommerziellen Verwertbarkeit, aber der Alto wurde ohnehin in erster Linie als eine experimentelle Plattform entwickelt. Schließlich wurden 1.500 Geräte hergestellt und von Forschern bei Xerox und an einigen Universitäten verwendet.[42]

Die der Alto-Konzeption zugrunde liegende Vision des «Personal Computing» wurde klar formuliert: «Mit ‹Personal Computer› meinen wir ein nicht gemeinsam genutztes System mit ausreichend Verarbeitungsleistung, Speicher und Eingabe-Ausgabe-Potenzial für den Rechenbedarf eines einzelnen Nutzers.»[43] Entsprechend dieser Vision war der Alto in seiner Konzeption primär für «Büroarbeit» gedacht:

> Die wichtigsten für den Alto vorgesehenen Anwendungen waren interaktive Textbearbeitung zur Dokument- und Programmvorbereitung, Unterstützung für den Programmentwicklungsprozess, Experimentieren mit Echtzeit-Animation und Musikgenerierung sowie das Betreiben einer Reihe experimenteller Büro-Informationssysteme. Das Hardware-Design war deutlich durch diese Vorannahmen betreffs der möglichen Anwendungen geprägt. Es tendierte zur Interaktion mit dem Benutzer, weg von umfangreicher numerischer Verarbeitung.[44]

Nach einem langwierigen Prozess des Experimentierens und fortlaufender Planung folgte auf den Alto der Xerox 8010 Star, der 1981 herauskam.[45]

> Der Alto diente als wertvoller Prototyp für den Star. Es wurden letzten Endes über tausend Altos hergestellt, und Alto-Benutzer hatten damit im Laufe von acht Jahren mehrere tausend Arbeitsjahre Erfahrung – der wohl größte Prototypenentwicklungsaufwand der Geschichte. Mitglieder des Xerox Palo Alto Research Center schrieben dutzende experimentelle Programme für den Alto. Ohne die kreativen Ideen der Autoren dieser Systeme wäre Star in seiner vorliegenden Form nicht möglich gewesen.[46]

42 Vgl. Charles P. Thacker: Personal distributed computing. The Alto and Ethernet hardware, in: A. Goldberg (Hg.): *A History of Personal Workstations*, Reading, Mass. 1988, 267–289.

43 Merry u. a.: *ALTO*, § 1.0.

44 Thacker u. a.: *Alto*, 3.

45 Vgl. David Canfield Smith u. a.: Designing the Star User Interface, in: *Byte*, April 1982, 242–282; David Canfield Smith, Charles Irby, Ralph Kimball, Eric Harslem: The Star user interface. An overview, in: R. K. Brown, H. L. Morgan (Hg.): *AFIPS '82. Proceedings of the National Computer Conference, 7–10 June 1982, Houston, Texas*, Arlington, Virginia 1982, 515–528.

46 Smith u. a.: The Star user interface, 527 f.

Nicht zu vergessen: «der wohl größte Prototypenentwicklungsaufwand der Geschichte» mit «mehrere[n] tausend Arbeitsjahre[n] Erfahrung» und über tausend Altos, auf denen «dutzende experimentelle Programme» liefen. Das interaktive Computing fiel zweifellos nicht einfach vom Himmel.

Und weniger als zehn Jahre nach der Veröffentlichung des Alto konnte Apple den ersten Macintosh zu einem Preis von rund 1.200 US-Dollar auf den Markt bringen, d. h. zu nur einem Zehntel der Kosten des Alto ein Jahrzehnt zuvor. Eine wirtschaftlich tragfähige Plattform für interaktives Computing war nun praktische Realität.

Die lange Reifephase der interaktiven Computertechnologien – angefangen von der Whirlwind- bis zur Xerox-Generation des interaktiven Computing-Paradigmas – war keine Folge theoretischer Defizite. Die Technologien des interaktiven Computing wurden niemals primär aus bereits vorhandenem theoretischen Wissen abgeleitet. Es gab keine mathematische Theorie, die eine Grundlage oder Begründung für interaktives Computing hätte abgeben können (und es wurde auch später keine geschaffen). Ebenso wenig kann das interaktive Computing als Folge der Anwendung einer psychologischen Theorie gelten. Der bedeutende HCI-Forscher Jack Carroll hat darauf hingewiesen, dass es «in einigen der zukunftsträchtigsten und folgenschwersten Planungsarbeiten zur Mensch-Maschine-Schnittstelle während der letzten 25 Jahre [etwa Sutherlands Sketchpad und Engelbarts NLS in den 1960er Jahren] überhaupt keine explizite Bezugnahme auf Psychologie gibt» und «die ursprünglichen Benutzeroberflächen zur direkten Manipulation» – etwa des Xerox Alto (1973) und des Xerox Star (1981) – «wenig oder nichts wissenschaftlich herleiteten, da in Wirklichkeit der Einfluss in die umgekehrte Richtung ging».[47]

Mehr noch: Die Prinzipien der «direkten Manipulation» wurden erst ein Jahrzehnt *nach* dem Herauskommen des Alto formuliert und systematisiert (durch Shneiderman sowie Hutchins, Hollan und Norman).[48] Die HCI-Forschung entwickelte sich als Systematisierungsbemühung *post festum*, nämlich mit dem Ziel des Verstehens der interaktiven Computing-Techniken, die bereits im Lauf von dreißig Jahren mit dem Whirlwind, dem Alto, dem Star und dem Macintosh bereits entstanden waren. Es gab bis dahin keine theoretischen Grundlagen in der Informatik. Die interaktiven Computing-Technologien der zweiten Generation wurden vielmehr von Computertechnikern entwickelt, um Anforderungen zu erfüllen, die sie selbst auf Basis der Prinzipien und Konzepte, die sie aus der eigenen täglichen Arbeitspraxis kannten, formuliert hatten. Einerseits bauten die Techniker bei der Entwicklung des Alto und des Star auf Konzepte auf, die jedem Erwachsenen im Westen, der in einer Büroumgebung arbeitete, bestens vertraut waren, und verallgemeinerten sie. In ihren eigenen Worten:

> Die Welt des Star ist in Form von Objekten organisiert, die Eigenschaften haben und an denen Aktionen ausgeführt werden können. Beispiele für solche Objekte sind Textzeichen, Textabsätze, grafische Linien, grafische Illustrationen, mathematische Summierungszeichen, mathematische Formeln und Symbole.[49]

47 John M. Carroll, Wendy A. Kellogg, Mary Beth Rosson: The task-artifact cycle, in: J. M. Carroll (Hg.): *Designing Interaction. Psychology at the Human-Computer Interface*, Cambridge 1991, 74–102, hier 79.

48 Ben Shneiderman: Direct manipulation. A step beyond programming languages, in: IEEE *Computer*, Vol. 16, Nr. 8, August 1983, 57–69; Edwin L. Hutchins, James D. Hollan, Donald A. Norman: Direct manipulation interfaces, in: D. A. Norman, S. W. Draper (Hg.): *User Centered System Design*, Hillsdale, New Jersey 1986, 87–124.

49 Smith u. a.: The Star user interface, 523.

Andererseits griffen die Entwickler bei der Interpretation dieser Alltagskonzepte auf den konzeptuellen Apparat der objektorientierten Programmierung zurück (Objekte, Objektklassen, Eigenschaften, Nachrichten), den Kristen Nygaard und Ole-Johan Dahl als Technik für «quasi-parallele Verarbeitung» entwickelt hatten[50] und der in den 1960er Jahren durch Alan Kay und andere in der Gestalt von Smalltalk weiterentwickelt wurde:

> Jedes Objekt hat Eigenschaften. Zu den Eigenschaften von Textzeichen gehören Schriftstil, Größe, Schriftart und Schriftschnitt (beispielsweise fett oder kursiv). Zu den Eigenschaften von Absätzen gehören Einrückung, Einzug und Ausrichtung. Zu den Eigenschaften grafischer Linien gehören Stärke und Struktur (beispielsweise durchgezogen, gestrichelt, gepunktet). Zu den Eigenschaften von Dokumentsymbolen gehören Name, Größe, Autor und Erstellungsdatum. Die Eigenschaften eines Objekts sind also vom Objekttyp abhängig.[51]

Auf ähnliche Weise konnten Techniker bei der Manipulation von Datenstrukturen auf «grundlegende Konzepte der Computerwissenschaft» aufbauen, um anwendungsunabhängige oder «generische» Befehle zu schaffen, die den Benutzer in die Lage versetzten, mehrere Anwendungen zu meistern und Daten zwischen Anwendungen zu «verschieben»:

> Der Star verfügt über einige Befehle, die im ganzen System eingesetzt werden können: MOVE, COPY, DELETE, SHOW PROPERTIES, COPY PROPERTIES, AGAIN, UNDO und HELP (VERSCHIEBEN, KOPIEREN, LÖSCHEN, EIGENSCHAFTEN ANZEIGEN, EIGENSCHAFTEN KOPIEREN, ERNEUT, RÜCKGÄNGIG MACHEN und HILFE). Jeder dieser Befehle funktioniert immer auf dieselbe Weise, unabhängig vom ausgewählten Objekttyp. Wir können solche Befehle folglich ‹generische Befehle› nennen. […] Befehle sind elementarer als in anderen Computersystemen. Sie streifen unnötige anwendungsspezifische Semantik ab, um an die zugrunde liegenden Prinzipien heranzukommen. Die generischen Befehle sind aus fundamentalen Konzepten der Computerwissenschaft abgeleitet, da sie auch den Operationen in Programmiersprachen zugrunde liegen.[52]

Es bleibt zentral, dass diese Planungskonzepte die eigene praktische Erfahrung der Techniker widerspiegelten, insofern sie die Konzepte ihrer Arbeit (typografische Elemente wie Zeichen, Absätze, Schriftstile usw.) verallgemeinerten und zugleich ähnliche und bereits generalisierte Konzepte der Computerwissenschaft (verschieben, kopieren, löschen, etc.) nutzen konnten. Konzepte wie «Zeichen», «Absatz», «Linie» und «Illustration» oder «kopieren» und «einfügen» waren Teil ihrer eigenen Praxis, ebenso wesentlich wie Stift und Papier. In der Folge konnten die interaktiven Computertechnologien lange Zeit durch Techniker weiterentwickelt und verfeinert werden, die auf der Basis ihrer Alltagspraxis in einem ziemlich intuitiven iterativen Planungs- und Bewertungsprozess Tools für den Eigengebrauch schufen. Die Mitglieder des Macintosh-Entwicklungsteams waren ähnlich motiviert: «Wir waren selbst unsere idealen Kunden und planten etwas, was wir am liebsten selbst haben wollten.»[53] Sie kannten das Konzept des interaktiven Computing aus Engelbarts Arbeit (der

50 Ole-Johan Dahl, Kristen Nygaard: SIMULA. An ALGOL-based simulation language, in: *Communications of the ACM*, Vol. 9, Nr. 9, September 1966, 671–678. Siehe auch Kristen Nygaard, Ole-Johan Dahl: The development of the SIMULA languages, in: *ACM SIGPLAN Notices*, Vol. 13, Nr. 8: History of programming languages conference, August 1978, 245–272.

51 Smith u. a.: The Star user interface, 523.

52 Ebd., 525.

53 Andy Hertzfeld: *Revolution in the Valley*, Sebastopol, Cal. 2005, xvii f.

seinerseits auf der Erfahrung mit Whirlwind aufgebaut hatte). Aus ihrem eigenen Leben wussten sie, welche Anforderungen sie daran stellen würden, und sie verfügten über das technische Können, ihre Ideen umzusetzen.

Die praktischen Ursprünge der interaktiven Anwendungsprogramme

Oft wird vergessen (auch in der Forschung zur Mensch-Computer-Interaktion), dass «Benutzer» (ausgenommen natürlich Computertechniker) nicht mit «Computern» interagieren, nicht einmal im interaktiven Computing-Paradigma. Praktiker verwenden für die Interaktion mit den Objekten und Prozessen in ihrer Arbeit eher Rechenwerkzeuge oder Darstellungen davon, und im interaktiven Computing-Paradigma können sie das flüssig, d. h. «in Echtzeit», synchron mit dem Rhythmus ihrer Arbeitsaktivitäten tun. Damit interaktives Computing einen praktischen Wert erhält und das Rechenwerkzeug in die jeweilige Praxis integriert, muss es den Betroffenen erlauben, ihren Arbeitsbereich in den eigenen Kategorien (durch Objekte und funktionale Elemente) des jeweiligen Gebiets darzustellen und zu verarbeiten.

Im Whirlwind-Paradigma sollte Computing den Bedienern ermöglichen, mit Kategorien wie «Ziel», «Verfolgung», «Flughöhe» usw. zu arbeiten. Im Xerox-Paradigma hingegen sollte Computing das Arbeiten mit elementaren grafischen Objekten schriftlicher Zeichensysteme (Text, Zahlen, geometrischen Objekten) und deren Kombinationen ermöglichen. Im Gegensatz zu Whirlwind wurde der Alto nach dem Muster der durch das Timesharing-Computing eingeführten «Arbeitsteilung» konzipiert, d. h. mit einem spezialisierten Programm – einem «Betriebssystem» –, das den Benutzer in die Lage versetzte, eine Reihe spezialisierter Anwendungsprogramme in beliebigen Kombinationen auszuführen. Entsprechend stellte man sich die Elemente als Ressourcen nicht für den Benutzer, sondern für die Anwendungsprogramme vor. Obwohl Anwendungsprogramme wesentlich für die Validierung des Konzepts «Personal Computing» waren, versuchte Xerox PARC in erster Linie nicht, eine Suite von Anwendungsprogrammen zu schaffen, sondern eine Plattform, auf der Benutzer eine Suite von Anwendungsprogrammen auf interaktive Weise ausführen konnten.

Die paradigmatischen Fälle interaktiver Anwendungsprogramme, etwa Textverarbeitung, Kalkulationstabellen, Desktop-Publishing-Programme und rechnergestütztes Konstruieren (CAD), wurden bisher noch bemerkenswert wenig systematisch untersucht. Es scheint die Annahme zu herrschen, dass diese Programme zumindest in konzeptioneller Hinsicht lediglich Anwendungen der Prinzipien der «direkten Manipulation» sind, die durch Schaffung optisch «vertrauter» Bedienungselemente mittels Planungs-«Metaphern» für Praktiker zugänglich wurden.[54] Die Anwendungsprogramme, die interaktives Computing in der Praxis für Nicht-Fachleute nützlich machten, wurden jedoch – einmal mehr – in Wechselbeziehung zur jeweiligen Arbeitspraxis entwickelt und nahmen folglich deren unterschiedlichen Logiken auf.

54 Vgl. etwa Mary Beth Rosson, John M. Carroll: *Usability Engineering. Scenario-Based Development of Human-Computer Interaction*, San Francisco 2001, 128.

Die paradigmatische Killerapplikation für Tabellenkalkulation, VisiCalc, die 1979 mit dem Apple II herauskam, wurde von einem Wirtschaftsstudenten mit Softwaretechnik-Hintergrund, Dan Bricklin, und einem Computerprogrammierer, Bob Frankston, einem Freund Bricklins, entwickelt. Die Entstehungsgeschichte dieses Programms ist mittlerweile Folklore – mit fehlenden historischen Belegen. Bricklin – ein Softwaretechniker mit Erfahrung bei der Entwicklung von Textverarbeitungsanwendungen für Minirechner – empfand es offenbar als unerträglich mühsam, seine Hausarbeiten in Harvard auf einem Arbeitsblatt aus Papier machen zu müssen. Seine brillante Idee bestand darin, das Tabellenformat des Arbeitsblattes – das Gitter mit vorgegebenen arithmetischen Beziehungen zwischen den Zellen – als Modell zu nehmen. Er erklärt das in einem 2004 durchgeführten Oral History-Interview so:

> Ich machte bei VisiCalc Folgendes: Ich fasste alles sofort bei der Eingabe zusammen, die Berechnungen und die Ausgabe erfolgten gleichzeitig damit. Das war meine Vision. Es ist wie bei der Textverarbeitung, wo man im Grunde genommen an der eigentlichen Ausgabe arbeitet, man drückt sich also im Output aus. […] Ich musste alles benennen. Daher versuchte ich herauszufinden, wie man Zellwerte [zur Verwendung in der Berechnungsdefinition] so angibt, dass es funktioniert. […] Schließlich überlegte ich mir, warum soll ich nicht ein Raster verwenden? […] Das war also die grundlegende Idee: Ich verwendete ein Raster, das ich ähnlich wie eine Karte benennen konnte. Natürlich war ich die Verwendung von Rastern gewöhnt, denn genau die benutzten wir bei dem numerischen Zeug, das wir machten […]. Die Zeilen und Spalten dienten nur dazu, die Benennung zu vereinfachen.[55]

Bricklin und Frankston nahmen das Arbeitsblatt als Modell und schufen so VisiCalc nach dem Muster eines zentralen Werkzeugs der Buchhaltungspraxis, das vor über einem Jahrhundert entwickelt worden und seither in Verwendung war. Dieses Werkzeug ist seinerseits ein Spezialfall des Tabellenformats, das auf die sumerische Buchführung vor ungefähr 4.500 Jahren zurückgeht.[56] Die bemerkenswerte Errungenschaft der beiden jungen Männer im Jahr 1978 sollte das traditionelle Werkzeug zusammen mit den dazugehörigen Formatierungskonventionen in ein interaktives Rechenwerkzeug transformieren. Das führte dazu, dass das elektronische Arbeitsblatt oder die elektronische Kalkulationstabelle sofort in der Praxis der Buchhalter und anderer Rechenberufe Fuß fassen konnte.

Für die elektronische Kalkulationstabelle war das «Arbeitsblatt» mehr als nur eine Metapher. Die optische Verwandtschaft war nicht nur oberflächlich. Das Tabellenformat des Arbeitsblattes bietet nicht nur eine Anordnung von (in arithmetischer Beziehung zueinander stehenden) Zahlen, sondern auch eine allgemeine räumliche Ordnung, die – mit markierten Zeilen- und Spaltenkoordinaten – entsprechend den immanenten Kategorisierungsnormen der Praxis der Buchhaltung, Kostenplanung usw. eingesetzt werden konnte. Aufgrund dieser räumlichen Struktur bietet das Arbeitsblatt eine gemäß den zentralen Anliegen der jeweiligen Praxis geordnete Synopse des Problemraums. Beim herkömmlichen Arbeitsblatt gab es das Problem, dass jede Änderung mühsame manuelle

55 Daniel Singer Bricklin, Robert M. Frankston: Oral-History-Interview. Interviewt von M. Campbell-Kelly und P. E. Ceruzzi, 7.5.2004. Interview-Transkript, Charles Babbage Institute, University of Minnesota, Minneapolis (OH 402), 12 f.

56 Vgl. Jack Goody: *The Domestication of the Savage Mind*, Cambridge 1977; Hans J. Nissen, Peter Damerow, Robert K. Englund: *Frühe Schrift und Techniken der Wirtschaftsverwaltung im alten Vorderen Orient. Informationsspeicherung und -verarbeitung vor 5000 Jahren*, Berlin 1990; Eleanor Robson: Tables and tabular formatting in Sumer, Babylonia, and Assyria, 2500 BCE–50 CE, in: Martin Campbell-Kelly u. a. (Hg.): *The History of Mathematical Tables. From Sumer to Spreadsheets*, Oxford 2003, 19–48 [Reprint 2007].

Neuberechnungen und Aktualisierungen erforderte. Die elektronische Version des Arbeitsblattes beseitigte diese langwierige manuelle Aufgabe und behielt zugleich das über Jahrhunderte entwickelte tabellarische Format bei.

Die zweite paradigmatische Killerapplikation, PageMaker, wurde im Juli 1985 von Aldus herausgebracht. PageMaker wurde durch ein professionelles Team für einen bereits ausgereiften Anwendungssoftware-Markt entwickelt. Aber auch die Planung von PageMaker und damit des Desktop-Publishing (DTP) geht auf einen erfahrenen Zeitungsmann, Paul Brainerd, zurück, der über ein Jahrzehnt als Produktionsleiter in der Zeitungsbranche gearbeitet hatte, wo er für das Seitenlayout und die Vorstufenprozesse des Drucks verantwortlich war. «Die ganze Idee des Seitenlayouts lag mir», wie er es selbst formuliert, «sehr am Herzen, da ich diese Arbeit am eigenen Leib erfahren hatte, mit Schablonenmesser und Rasierklingen und Wachs auf der Rückseite des Fotosatzes [d. h. beim Setzen ohne Metallguss]».[57] Nach jahrelanger Arbeit zuerst als Produktionsleiter und dann als Zuständiger für die Einführung der elektronischen Textverarbeitung in die Zeitungsproduktion ging Brainerd in die Textverarbeitungsbranche und gründete nach wenigen Monaten im Jahr 1984 Aldus, wo er (in Koordination mit Apple und Adobe) die Entwicklung von PageMaker leitete. Die Funktionsbeschreibungen für den späteren PageMaker schrieb Brainerd in Zusammenarbeit mit den Technikern.[58]

Das zentrale Werkzeug der Seitenlayout-Praxis war die Musterseite mit ihrem Rand und dem Satzspiegel und Rasterlinien, Kopf- und Fußzeilen, Abschnitten usw. Sie diente zur Standardisierung des Seitenlayouts für alle Seiten und damit für ganze Publikationen und war daher ein unentbehrliches Werkzeug für die Schaffung übersichtlicher und ansprechender Druckschriften. War die Musterseite vorhanden, so bestand die Layout-Arbeit bei traditionellem Fotosatz vor allem darin, den Text und die Illustrationen in das vorbereitete Muster-Layout zu kleben, und zwar, in den Worten Brainerds, «mit Schablonenmesser und Rasierklingen und Wachs auf der Rückseite des Fotosatzes». Die schwierigste Aufgabe dieser Arbeit blieb es, peinlich genau sicherzustellen, dass alle Elemente exakt angeordnet und ausgerichtet waren. Und auch hier erforderte jede Korrektur eine weitere Runde mühsamer Klebearbeit. PageMaker behielt das zentrale Werkzeug des Layoutens (und die dazugehörigen Werkzeuge) bei, in denen sich die Ordnungs-und Verfahrensprinzipien dieser Arbeit und der begleitenden konzeptuellen Schemata widerspiegelten. Da es in DTP möglich war, die erforderliche Klebearbeit auf einer digitalen Darstellung der Seite zu erledigen, reduzierte das die Schinderei bei Änderungen am Layout deutlich. Auch hier wurden die zentralen Werkzeuge der Arbeitspraxis ebenso wie die dazugehörigen Prinzipien fachgerechten Layouts in ein interaktives Rechenwerkzeug transformiert.

Ebenso, aber insgesamt auf einem anderen Niveau, wurde das rechnergestützte Konstruieren (CAD) weitgehend in enger Wechselbeziehung mit praktischen Anliegen der Luft- und Raumfahrt- sowie der Automobilindustrie

57 Paul Brainerd: Oral-History-Interview. Interviewt von S. Crocker, 16.5.2006. Interview-Transkript, Computer History Museum (X2941.2005). Das Oral History-Interview wurde telefonisch geführt.

58 Vgl. Brainerd: Oral History.

von Ford bis Lockheed, von Dassault bis Renault entwickelt.[59] Noch 1980 kostete ein Techniker-«Platz» mit CAD-Systemen auf spezialisierten Minirechnern 125.000 US-Dollar. Die Rechner mussten in einem speziellen klimatisierten Raum installiert werden, und obendrein dauerte die Bediener-Grundschulung mehrere Wochen lang. Daher hatte sich eine Arbeitsorganisation als Norm durchgesetzt, die der Massenkalkulation durch von Tischrechnern unterstützten «menschlichen Computern» ähnelte, mit der «Bremse Mensch» und dem «In by ten, out by five»-Regime:

> Da diese Systeme relativ teuer waren, arbeiteten sie tendenziell im sogenannten «geschlossenen Betrieb». Für die Bediener war die Arbeit an den Grafikkonsolen normalerweise eine Vollzeitbeschäftigung. Techniker und Planer brachten Arbeit in die «CAD-Abteilung» und kamen Stunden oder Tage später zurück, um die geplottete Ausgabe abzuholen, die sie sorgfältig prüften. Die Zeichnungen wurden mit Änderungsanmerkungen wieder an die CAD-Bediener retourniert, die sie überarbeiteten und wiederum dem Anforderer zurückgaben. Nur selten hatte ein Techniker die Möglichkeit, ein System für interaktive kreative Planungsarbeit zu nutzen oder gemeinsam mit einem Bediener an der Arbeit zu sitzen, der unmittelbar auf seine Anregungen einging. Die hohen Kosten dieser Systeme resultierten oft in einem Zwei- oder gar Dreischichtenbetrieb zur maximalen Auslastung.[60]

Im Zuge der Einrichtung der Personal Computer-Plattform in den 1980er Jahren (zunächst als spezialisierte «Arbeitsplätze für technische Planung», später als CAD-Software auf gewöhnlichen PCs) wiederholte sich bei der Entwicklung von CAD das, was auch in der Entwicklung von Kalkulationstabellen und DTP-Anwendungsprogrammen geschehen war. Bei der herkömmlichen technischen und architektonischen Planung spielte die Verwendung von «Pauspapier» eine Schlüsselrolle. Diese Methode bestand einfach darin, ein Blatt durchscheinendes Papier über die Zeichnung zu legen und zu fixieren und dann den Plan mit einem Kugelschreiber oder Bleistift sorgfältig nachzuzeichnen, Linie für Linie.[61] Obwohl dieser Vorgang äußerst arbeitsintensiv war, blieb die Verwendung von Pauspapier – auch als Blaupausen bereits üblich waren – zentraler Bestandteil technischer Planungspraxis: Noch 1980 gehörte sie zum «aktuellen Stand der Produktion architektonischer Zeichnungen».[62] Die Gründe dafür liegen in der Logik der technischen und architektonischen Planungspraxis. Laut Charles Eastman ist die technische und architektonische Planungspraxis «auf eine Reihe von Darstellungen angewiesen, die jeweils einzelne Elemente einer Konstruktion beschreiben». Darin besteht ein Grund für die Komplexität der jeweiligen Praxis:

> Beim manuellen oder computergestützten Planen geht es darum, Elemente zu definieren und diese mittels verschiedener Darstellungen in den vielfachen Dimensionen ihrer Interaktion – geometrisch, strukturell, elektrisch, akustisch usw. – zusammenzusetzen. Diese verschiedenen Darstellungen werden schrittweise mit der Zeit definiert. Zu den vorhandenen Beschreibungen werden neue Elemente hinzugefügt, die weitere Leistungen abbilden, an denen der Planer interessiert ist. [Bei der Planung] werden die Informationen in einer Darstellung erstellt und dann in andere

59 Vgl. David E. Weisberg: *The Engineering Design Revolution. The People, Companies and Computer Systems that Changed Forever the Practice of Engineering*, Englewood, Col., PDF auf: http://www.cadhistory.net, dort datiert 2008, gesehen am 6.2.2015.

60 Weisberg: *The Engineering Design Revolution*, 2–16.

61 Vgl. Chester W. Edwards: *Overlay Drafting Systems*, New York 1984; Frank Woods, John Powell: *Overlay Drafting. A Primer for the Building Design Team*, London 1987.

62 Mark J. Clayton: Computational design and AutoCAD. Reading software as oral history, in: *SIGraDi 2005. IX Congreso Iberoamericano de Gráfica Digital, 21–23 November 2005, Lima*, Peru 2005, 103–107, hier 105.

> übertragen, bis die Konstruktion diverse Kriterien erfüllt, die in den verschiedenen Darstellungen bewertet werden. […] Die Darstellungen der Konstruktion müssen konsistent sein […] Eine Änderung in irgendeinem Teil der Planung muss sowohl in die höheren als auch in die niedrigeren Detaillierungsniveaus übernommen werden.[63]

Die Layertechnik wurde zu einem zentralen Element der Planungspraxis, da sie einen Umgang mit dieser Komplexität ermöglichte. Erstens versetzte sie Techniker und Architekten in die Lage, (möglicherweise tentative) Änderungen an einer Planung vorzunehmen, ohne die gesamte Zeichnung neu erstellen zu müssen. Zweitens konnten so die einzelnen Planer separat an unterschiedlichen Planungsaspekten komplexer Konstruktionen arbeiten, sowohl konzeptionell als auch praktisch. Drittens bot sie ein Strukturgerüst, in dem mehrere Planer getrennt voneinander und doch in geordneter Form arbeiten konnten: Jeder Einzelne arbeitet an seinen Elementen, die in separaten Layers dargestellt sind, und ist zugleich jederzeit in der Lage, seine Teilarbeit auf die Arbeit der anderen zu beziehen.[64] Und viertens konnte aufgrund der Trennung in Layers durch die unterschiedliche Kombination von Layers auf unkomplizierte Weise eine Vielzahl von Kopien erstellt werden, bot sich also eine recht einfache Methode zur Koordination zwischen den verschiedenen Berufen, die in der Planung und Konstruktion oder im Herstellungsprozess involviert waren.

Wie bei Kalkulationstabellen und beim Desktop-Publishing veränderte CAD keineswegs die Logik der bestehenden Praxis. Es ermöglichte eher die Durchführung von analogen Planungen mit höherer Komplexität. An erster Stelle wurden natürlich die Kosten für Änderungen an Zeichnungen reduziert. Durch CAD wurde die Layertechnik auch bei Weitem besser handhabbar.[65] CAD-Pläne können jetzt problemlos aus über 100 Layers bestehen,[66] was seinerseits neue Standards für den Umgang mit Layerstrukturen erforderte, etwa bei der Benennung, den Zeichnungsnotationen usw.[67] Schließlich bleibt zu erwähnen, dass die Layertechnik von der Architektur und der technischen Planung auch auf andere Arbeitsgebiete verlagert wurde, in denen Zeichnungen eine Rolle spielen (siehe beispielsweise Adobe InDesign und Acrobat), und damit auch von Praktikern in Berufen aufgegriffen wurde, die bis dahin diese Techniken nicht verwendet hatten.

Wichtige Paradigmen für Anwendungen zur interaktiven Datenverarbeitung weisen also, kurz gesagt, in ihrer Entwicklung bemerkenswerte Parallelen auf: Sie gehen auf Praktiker zurück, entstanden als praktische Techniken zum Eigengebrauch oder für die Verwendung durch Kollegen, und wurden später verallgemeinert. Analoges gilt für die Ursprünge des *kollaborativen Computing*, etwa für die Entwicklung des ARPANET, für den durchschlagenden Erfolg der E-Mail, die Entstehung des WWW am CERN und die Entstehung des Internet insgesamt. Die Verallgemeinerung des Eigengebrauchs von Programmierern, Wissenschaftlern und anderen Praktikern scheint das zu sein, was die proteische Natur des Computers am wirksamsten entfaltet.

Aus dem Englischen von Leonhard Schmeiser

63 Charles M. Eastman: Why we are here and where we are going. The evolution of CAD, in: *ACADIA 1989. New Ideas and Directions for the 1990s. ACADIA Conference Proceedings. 27–29 October 1989, Gainsville, Florida*, 9–26, hier 12.

64 Vgl. beispielsweise Edwards: *Overlay Drafting Systems*; Fred A. Stitt: *Systems Graphics. Breakthroughs in Drawing Production and Project Management for Architects, Designers, and Engineers*, New York u. a. 1984.

65 Vgl. Stitt: *Systems Graphics*; Chengzhi Peng: Survey of Collaborative Drawing Tools. Design Perspectives and Prototypes, in: *Computer Supported Cooperative Work (CSCW). An International Journal*, Vol. 1, Nr. 3, 197–228.

66 Vgl. Kjeld Schmidt, Ina Wagner: Ordering systems. Coordinative practices and artifacts in architectural design and planning, in: *Computer Supported Cooperative Work (CSCW). The Journal of Collaborative Computing*, Vol. 13, Nr. 5–6, December 2004, 349–408.

67 Vgl. beispielsweise AIA: *AIA CAD Layer Guidelines. U.S. National CAD Standard Version 3*, Washington, D.C. 2005 [3. Auflage; 1. Auflage 1990].